

**Estudio y Aplicación del Procesado Digital de
la Señal a las Radiocomunicaciones Digitales.**

(Tesis Doctoral)

Autor de la Tesis Doctoral:

Juan J. Palma Guerrero

Directores de la Tesis Doctoral:

Dr. Carlos Mascareñas Pérez-Iñigo
Dr. José Luís Cueto Ancela

AGRADECIMIENTOS.

Me gustaría que estas líneas sirvieran para expresar mi más profundo y sincero agradecimiento a todas aquellas personas que con su ayuda han colaborado en la realización del presente trabajo.

Al Doctor D. Carlos Mascareñas Pérez-Iñigo, director de esta tesis, por la orientación, el seguimiento y la supervisión continua de la misma, pero sobre todo por la motivación y el apoyo recibidos a lo largo de estos años.

Al Doctor D. José Luís Cueto Ancela, codirector de esta tesis, por el interés mostrado en esta investigación y las valiosas sugerencias aportadas.

Al Doctor Pieter-Tjerk de Boer, al Sr. Andrea Montefusco, al Sr. Alex Lee, al Sr. Tony Parks y al Sr. Simon Brown por permitirme participar en sus trabajos y Grupos de Investigación.

Igualmente, al Grupo de Investigación Señales, Sistemas y Comunicaciones Navales de la Universidad de Cádiz, por la financiación del Proyecto Marconi que ha permitido la aplicación práctica de esta investigación.

Quisiera hacer extensiva mi gratitud al jurado calificador por aceptar ser parte de este trabajo.

Un agradecimiento muy especial merece la comprensión, paciencia y el ánimo recibido de mi mujer y de mis hijas, así como el apoyo incondicional de mi hermano Manuel Palma.

A todos ellos, muchas gracias.

Juan J. Palma Guerrero

DEDICATORIA:

A mis padres

A mi hermano Manuel.

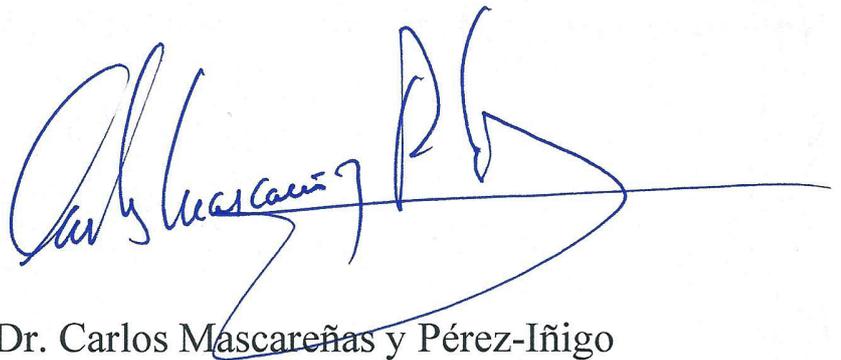
A mi mujer Lourdes y a mis hijas, Cristina y Almudena.

Dr. D. Carlos Mascareñas y Pérez-Iñigo. Profesor Titular del Departamento de Ciencias y Técnicas de la Navegación, con NRP 0079643868 A0504, de la Universidad de Cádiz.

H A C E C O N S T A R :

Que la Tesis Doctoral “**ESTUDIO Y APLICACIÓN DEL PROCESADO DIGITAL DE LA SEÑAL A LAS RADIOCOMUNICACIONES DIGITALES**“, ha sido realizada por D. Juan José Palma Guerrero, bajo mi dirección y la del Prof. Dr. D. José Luis Cueto Ancela y consideramos que cumple todos los requisitos legales vigentes para optar al Grado de Doctor.

Y para que conste, expido el presente documento en Puerto Real a los veintinueve días de marzo de dos mil catorce.



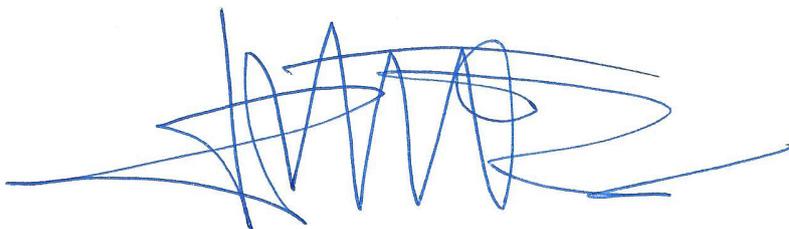
Fdo.- Prof. Dr. Carlos Mascareñas y Pérez-Iñigo

Dr. D. José Luis Cueto Ancela. Profesor Titular del Departamento de Ciencias y Técnicas de la Navegación, con NRP 1603708257 A0504, de la Universidad de Cádiz.

H A C E C O N S T A R :

Que la Tesis Doctoral **“ESTUDIO Y APLICACIÓN DEL PROCESADO DIGITAL DE LA SEÑAL A LAS RADIOCOMUNICACIONES DIGITALES”**, ha sido realizada por D. Juan José Palma Guerrero, bajo mi dirección y la del Prof. Dr. D. Carlos Mascareñas y Pérez-Iñigo y consideramos que cumple todos los requisitos legales vigentes para optar al Grado de Doctor.

Y para que conste, expido el presente documento en Puerto Real el treinta y uno de marzo de dos mil catorce.



Fdo.- Prof. Dr. José Luis Cueto Ancela

ESTUDIO Y APLICACIÓN DEL PROCESADO DIGITAL DE LA SEÑAL A LAS RADIOCOMUNICACIONES DIGITALES.

Índice.

1 Introducción.

1.1. Introducción al Proceso Digital de la Señal.

1.1.1. Telecomunicaciones.

1.1.1.1. Multiplexado.

1.1.1.2. Compresión.

1.1.1.3. El control del eco.

1.1.2. Procesado de audio.

1.1.3. Localización del eco.

1.1.3.1. RADAR.

1.1.3.2. SONAR.

1.1.3.3. Reflexión sismológica.

1.1.4. Procesamiento de imágenes.

1.1.5. Medicina.

1.1.6. Espacio.

1.1.7. Imágenes en productos comerciales.

1.2. Introducción a los Modos de Radiocomunicaciones Digitales.

1.2.1. Introducción.

1.2.2. Radioteletipo (RTTY).

1.2.2.1. BAUDOT (o Murray).

1.2.2.2. ASCII.

1.2.2.3. SITOR.

1.2.2.4. FAX.

1.2.2.5. SSTV (Slow Scan TV-Televisión de barrido lento).

1.2.2.6. ARQ (Automatic Repeat reQuest).

1.2.2.7. FEC (Forward Error Correction).

1.2.2.8. Radio Paquete (Packet-Radio).

1.2.2.9. PACTOR y PACTOR-II.

1.2.2.10. CLOVER y CLOVER-II.

- 1.2.2.11. G-TOR.
 - 1.2.2.12. PSK31.
 - 1.3. Procesado Digital de la Señal y SDR.
 - 1.4. Composición resumida del Demostrador de Conceptos Marconi.
 - 1.5. Aplicación del Demostrador de Conceptos Marconi.
 - 2. Estado de la cuestión.
 - 2.1 Los Equipos de Radio Definidos por Software.
 - 2.1.1 Introducción
 - 2.1.2 Antecedentes.
 - 2.1.3 El foro SDR.
 - 2.1.4 Análisis de los Equipos a utilizar en el Demostrador de Conceptos
 - 2.1.4.1 Introducción histórica
 - 2.1.4.2 Diferencia entre equipos de Radio definidos por Software y Equipos de Radio controlados por Software.
 - 2.2 Descripción tecnológica de los Equipos de Radio definidos por Software.
 - 2.2.1 Antecedentes, el DSP TMS320C30
 - 2.2.1.1 Introducción
 - 2.2.1.2 Buses.
 - 2.2.1.3 Unidad Central de Proceso.
 - 2.2.1.4 Organización de memoria.
 - 2.2.1.5 Control.
 - 2.2.1.6 Periféricos.
 - 2.2.2 Introducción técnica al SDR.
 - 2.2.3 Conversión de frecuencias intermedias en señales IQ.
 - 2.2.4 La Demodulación.
 - 2.3 Sistemas de Radiocomunicaciones basados en Software.
 - 2.3.1 Introducción.
 - 2.3.2 Bloques funcionales de SDR.
 - 2.3.3 Arquitectura SDR.
 - 2.3.4 Ventajas y desventajas de los Sistemas de Comunicaciones basados en SDR.
 - 2.3.5 Plataformas Hardware para su implementación.
 - 2.3.6 DSP.

- 2.3.7 Arquitectura GNU Radio.
 - 2.3.7.1 Descripción general.
 - 2.3.7.1.1 Qué hace GNU Radio.
 - 2.3.7.1.2 Formas de instalar GNU Radio.
 - 2.3.7.1.3 GNU.
 - 2.3.7.1.4 GNU/LINUX.
 - 2.3.7.1.5 Distribuciones Linux.
 - 2.3.7.1.6 GNU Radio.
 - 2.3.7.1.7 Definiciones generales.
 - 2.3.7.2 La Tarjeta USRP.
- 2.3.8 Arquitectura open HPSDR.
 - 2.3.8.1 Introducción.
 - 2.3.8.2 Módulos.
 - 2.3.8.2.1 Módulo ATLAS.
 - 2.3.8.2.2 Módulo OZYMANDIAS (OZY).
 - 2.3.8.2.3 Módulo MERCURY.
 - 2.3.8.2.4 Módulo SASQUATCH.
 - 2.3.8.2.5 Módulo PENELOPE.
 - 2.3.8.2.6 Módulo JANUS.
 - 2.3.8.2.7 Módulo PINOCCHIO.
 - 2.3.8.2.8 Módulo EPIMETHEUS (EPI).
 - 2.3.8.2.9 Módulo ALEXIARES (ALEX).
 - 2.3.8.2.10 Módulo PANDORA.
 - 2.3.8.2.11 Módulo DEMETER.
 - 2.3.8.2.12 Módulo PROTEUS.
 - 2.3.8.2.13 Módulo THOR.
 - 2.3.8.2.14 Módulo GIBRALTAR.
 - 2.3.8.2.15 Módulo PHOENIX.
 - 2.3.8.2.16 Módulo CYCLOPS.
 - 2.3.8.3 Desarrollo del proyecto HPSDR.
- 2.3.9 Arquitectura WebSDR.
 - 2.3.9.1 Descripción general.
 - 2.3.9.2 Instalación WebSDR.
 - 2.3.9.3 Configuración WebSDR.

- 2.3.9.4 Consideraciones generales sobre el cliente WebSDR.
- 2.3.9.5 Consideraciones generales para el servidor WebSDR.
 - 2.3.9.5.1 El receptor Softrock Lite II.
 - 2.3.9.5.1.1 Fuente de alimentación
 - 2.3.9.5.1.2 Oscilador local.
 - 2.3.9.5.1.3 Divisor (40m).
 - 2.3.9.5.1.4 Amplificadores operacionales (40m).
 - 2.3.9.5.1.5 Filtro paso banda (40m).
 - 2.3.9.5.1.6 Mezclador (40m).
 - 2.3.9.5.1.7 Conexiones externas.
 - 2.3.9.5.2 Distribuciones Linux del servidor y tarjetas de sonido.
- 2.3.9.6 Configuración avanzada de WebSDR..
- 2.3.10 Arquitectura QtRadio
 - 2.3.10.1 Objetivos del proyecto.
 - 2.3.10.2 Descripción general.
 - 2.3.10.3 Instalación de QtRadio para Linux-Ubuntu 12.04.
 - 2.3.10.4 Ejecución del programa cliente QtRadio.
 - 2.3.10.5 Ejecución del programa servidor QtRadio.
 - 2.3.10.6 Configuración del servidor QtRadio.
- 2.3.11 Arquitectura SDR-Radio.
 - 2.3.11.1 Introducción.
 - 2.3.11.2 SDR-Radio, conexión cliente / servidor.
 - 2.3.11.3 Servidor de SDR-Radio.
 - 2.3.11.4 Cliente de SDR-Radio.
- 2.4 Voz sobre IP.
 - 2.4.1 Introducción.
 - 2.4.2 Descripción general.
 - 2.4.3 El protocolo SIP.
 - 2.4.4 Calidad del servicio
- 2.5 Telefonía sobre IP.
 - 2.5.1 Introducción.
 - 2.5.2 Descripción general.
 - 2.5.3 Primeros pasos con Asterisk.
 - 2.5.3.1 Ficheros de configuración.

- 2.5.3.2 Configuración de una interfaz con la red pública o una PBX.
- 2.5.3.3 Configuración de los teléfonos-IP SIP.
- 2.5.3.4 Plan de llamadas.
- 2.5.3.5 Conceptos generales sobre canales analógicos y digitales.
- 2.5.3.6 Configuración de un canal de telefonía en Asterisk.
- 2.5.4 Características de la centralita Trixbox.
 - 2.5.4.1 Conceptos generales.
 - 2.5.4.2 Descripción de Trixbox.
 - 2.5.4.3 Consideraciones generales para implementar una centralita Trixbox.
 - 2.5.4.4 Instalación de Trixbox.
 - 2.5.4.5 Configuración de la centralita Trixbox.
 - 2.5.4.5.1 Pantallas de configuración de la centralita Trixbox.
 - 2.5.4.5.2 Pantalla de administración.
 - 2.5.4.5.3 Creación de extensiones.
 - 2.5.4.5.4 Creación de troncales.
 - 2.5.4.5.5 Creación de rutas.
 - 2.5.4.5.6 Planes de marcado.
 - 2.5.4.5.7 Grupos de llamada.
 - 2.5.4.5.8 Colas de llamadas.
 - 2.5.4.5.9 Configuraciones generales.
 - 2.5.4.5.10 Panel rápido del operador (Flash Operator Panel).
 - 2.5.4.6 Configuración avanzada de la centralita Trixbox.
- 2.6 Equipos y Sistemas de comunicaciones IP.
 - 2.6.1 Introducción.
 - 2.6.2 Equipos Softphones (Teléfonos software).
 - 2.6.3 Equipos IP-Phones (Teléfonos electrónicos).
 - 2.6.4 Cámaras IP.
 - 2.6.5 Descripción de Teamspeak.
 - 2.6.6 Programación de Teamspeak3 MARCONI.
 - 2.6.7 Streaming de voz en Linux.
 - 2.6.7.1 Introducción.
 - 2.6.7.2 Cómo calcular el ancho de banda necesario.
 - 2.6.7.3 Límite de Tráfico.

- 3 Objetivos de la Tesis Doctoral.

- 4 Metodología propuesta para la realización de esta Tesis Doctoral.
 - 4.1 Enfoques metodológicos.
 - 4.2 Desarrollo metodológico.
 - 4.3 La Radio definida por Software en la práctica.
 - 4.4 Tipos de SDR.
 - 4.5 Software para SDR.
 - 4.6 Aplicaciones SDR basadas en servidor.
 - 4.7 Aplicaciones VoIP y Telefonía sobre IP.
 - 4.8 Sistema de ayuda al uso y mejora del demostrador Marconi.

- 5 Resultados – Aplicaciones prácticas realizadas en el Laboratorio del Grupo S2CN.
 - 5.1 Introducción.
 - 5.2 Caminos explorados dentro de la Tesis Doctoral.
 - 5.3 Arquitectura de red Marconi.
 - 5.4 Servidor Teamspeak.
 - 5.5 Servidor Trixbox.
 - 5.6 Servidores de Receptores de Radio.
 - 5.7 Página Web Marconi.uca.es.
 - 5.8 Página Foro Marconi.uca.es.
 - 5.9 Servidor de GNU Radio.
 - 5.10 Propuestas de Patentes.
 - 5.10.1 Demostrador de Conceptos de Radiofrecuencia diseñado para su aplicación a la Enseñanza Superior y con aplicaciones, de forma local o remota, al Control y Análisis de Radiocomunicaciones en cualquier banda de Radio.
 - 5.10.2 Demostrador de Conceptos de Radiofrecuencia con aplicaciones múltiples en el campo de la Enseñanza Superior.
 - 5.10.3 Demostrador de Conceptos de Radiocomunicaciones vía satélites ecuatoriales con aplicaciones múltiples en el campo de la Enseñanza Superior.
 - 5.11 Artículos derivados de esta Tesis Doctoral.

- 5.11.1 Artículo publicado en revista Radioaficionados (URE).
- 5.11.2 Artículo publicado en CISTI'2014 - 9ª Conferencia Ibérica de Sistemas y Tecnologías de Información.
- 5.12 Ejemplo de práctica de Radiotecnía ejecutada por alumno del Grado de Ingeniería Radioelectrónica.
- 5.13 Colaboración con la WebSDR de Chile.

6 Conclusiones.

7 Bibliografía.

8 Apéndices.

Apéndice 1. Instalación de las fuentes de GNU Radio.

Apéndice 2. Configuración de Asterisk

Apéndice 3. Estimación del tráfico

Apéndice 4. Configurar un cliente SIP

Apéndice 5. SIP NAT Transversal

Apéndice 6. Instalación de QtRadio (versión SSL)

Apéndice 7. Detalle de instalación del Servidor Teamspeak 3

Apéndice 8. Descripción detallada de comandos de conexión SDR_Radio.

Apéndice 9. Instalación de GNU Radio Marconi.

Índice de figuras.

- Figura 1.1.1 Aplicaciones del procesado digital de la señal.
- Figura 1.1.2 Áreas de aplicación del procesado digital de la señal.
- Figura 1.3.1 Diagrama de bloques de Radio definida por software.
- Figura 1.3.2 Diagrama de bloques de “digital down converter”.
- Figura 1.3.3 Muestras de ADC en el dominio del tiempo.
- Figura 1.3.4 Transformada Rápida de Fourier de la banda de FM con nueve estaciones.
- Figura 1.3.5 Señal modulada en frecuencia.
- Figura 1.3.6 Diagrama de bloques del receptor FM.
- Figura 1.3.7 FFT a la salida del convertidor digital de bajada.
- Figura 1.3.8 FFT de la señal FM demodulada.
- Figura 1.4.1 Red de Estaciones Costeras Españolas de MF y HF del SMSSM.
- Figura 1.4.2 Espectros de una señal A1A a la izquierda y una señal A3E a la derecha.
- Figura 1.4.3 Espectrograma simultáneo de los dos receptores del Websdr de UCA.
- Figura 1.4.4 Diagrama en cascada simultáneo de los dos receptores del Websdr de UCA.
- Figura 1.4.5 Pantalla de comentarios escritos del Websdr de la UCA.
- Figura 1.4.6 Espectrograma de la señal de la Estación Costera Cádiz Radio con definición de los parámetros del receptor Funcube.
- Figura 1.5.1 Diagrama de bloques del programa que controla al receptor Funcube.
- Figuras 2.1.1 y 2.1.2 Receptores controlados por ordenador de la firma ICOM del Grupo S2CN.
- Figura 2.2.1 Tarjeta de aplicación programable Texas Instrument TMS320C30
- Figura 2.2.2 Diagrama de bloques del TMS320C3x.
- Figura 2.2.3 Buses internos del TMS320C3x (SMJ320C30DSP. Texas Instruments).
- Figura 2.2.4 Diagrama de bloques de la CPU (SMJ320C30DSP. Texas Instruments).
- Figura 2.2.5 Organización de memoria (SMJ320C30DSP. Texas Instruments).
- Figura 2.2.6 Mapas de memoria (SMJ320C30DSP. Texas Instruments).
- Figura 2.2.7 Periféricos del TMS320C3x.
- Figura 2.2.8 Diagrama de bloques del equipo SDR-1000.
- Figura 2.2.9 Principio del Método de fase para generar señales SSB.
- Figura 2.2.10 Principio del mezclador en cuadratura con muestreo.
- Figura 2.2.11 Componentes I y Q.
- Figura 2.3.1 Transceptor de MF/HF del Servicio Móvil Marítimo.

Figura 2.3.2 Selección de una frecuencia entre 7,00 MHz y 7,20 MHz.

Figura 2.3.3 Pantalla de conjunto de receptores SDR (Universidad de Twente - 2010).

Figura 2.3.4 Pantalla del receptor Perseus.

Figura 2.3.5 Diagrama de bloques funcionales de SDR.

Figura 2.3.6 Arquitectura resumida de un SDR.

Figura 2.3.7 Bloques principales en este sistema de comunicaciones de Software Radio.

Figura 2.3.8 Universal Software Radio Peripheral.

Figura 2.3.9 Diagrama de bloques del Universal Software Radio Peripheral.

Figura 2.3.10 Multiplexor en el USRP, donde I es la señal en fase y Q la señal en cuadratura.

Figura 2.3.11 Imagen de las tarjetas secundarias Basic RX y Basic TX.

Figura 2.3.12 Logotipo de los equipos compatibles con HPSDR.

Figura 2.3.13 Placa Atlas, versión 1^a.

Figura 2.3.14 Placa Janus. Prototipo versión alfa 2.

Figura 2.3.15 Conjunto de placas Atlas, Janus (en primer plano) y Pinocchio (al fondo).

Figura 2.3.16 Caja Pandora modelo BK959. Aloja las placas Atlas, Ozy, y Janus.

Figura 2.3.17 Transceptor HPSDR de Phil Harman, VK6APH, año 2007.

Figura 2.3.18 Descripción de WebSDR y detalle de localización de servidores.

Figura 2.3.19 Relación de servidores donde se observa su localización y posición en el mapa.

Figura 2.3.20 Componentes de la página WebSDR.

Figura 2.3.21 WebSDR. Chat para establecer comunicación entre usuarios.

Figura 2.3.22 WebSDR. Información de datos recibidos/enviados.

Figura 2.3.23 SoftRock Lite II, cara inferior (Bottom).

Figura 2.3.24 SoftRock Lite II, cara superior (Top).

Figura 2.3.25 Diagrama receptor Softrock Lite II (FiveDashInc).

Figura 2.3.26 Fuente de alimentación Softrock lite II (FiveDashInc).

Figura 2.3.27 Detalle de componentes de fuente de alimentación (FiveDashInc).

Figura 2.3.28 Oscilador local Softrock lite II (FiveDashInc).

Figura 2.3.29 Detalle de componentes del oscilador local. Softrock Lite II (FiveDashInc).

Figura 2.3.30 Divisor Softrock Lite II (FiveDashInc).

Figura 2.3.31 Detalle de componentes del Divisor Softrock Lite II (FiveDashInc).

Figura 2.3.32 Amplificadores operacionales. Softrock Lite II (FiveDashInc).

Figura 2.3.33 Amplificadores operacionales. Softrock Lite II (FiveDashInc).

Figura 2.3.34 Filtro paso-banda. Softrock Lite II (FiveDashInc).

Figura 2.3.35 Filtro paso-banda. Softrock Lite II (FiveDashInc).

Figura 2.3.36 Mezclador. Softrock Lite II (FiveDashInc).

Figura 2.3.37 Mezclador Softrock Lite II (FiveDashInc).

Figura 2.3.38 Conexiones externas. Softrock Lite II (FiveDashInc).

Figura 2.3.39 Diagrama de bloques Softrock Lite II (FiveDashInc).

Figura 2.3.40 Página principal de Web QtRadio.

Figura 2.3.41 Página de descarga del cliente QtRadio para Windows.

Figura 2.3.42 Descarga del último zip de QtRadio.

Figura 2.3.43 Directorio QtRadio.

Figura 2.3.44 Programa QtRadio.

Figura 2.3.45 Pantalla del QtRadio conectado al servidor marconi.uca.es

Figura 2.3.46 Ventana de selección de servidores de QtRadio.

Figura 2.3.47 QtRadio conectado a servidor a través de Internet.

Figura 2.3.48 Detalle de lanzadores de Ubuntu.

Figura 2.3.49 Pantalla principal servidor QtRadio.

Figura 2.3.50 Detalle del lanzamiento de la interfaz para Perseus.

Figura 2.3.51 Servidor DSP de la aplicación servidor de QtRadio.

Figura 2.3.52 Lanzamiento de la interfaz gráfica del Perseus.

Figura 2.3.53 Interfaz gráfica QtRadio.

Figura 2.3.54 Consola SDR-Radio.

Figura 2.3.55 Servidor de SDR-Radio con Funcube conectado.

Figura 2.3.56 Consola conectada a servidor de imagen anterior.

Figura 2.3.57 Iconos de lanzamiento de las aplicaciones SDR-Radio.

Figura 2.3.58 Pantalla principal del servidor SDR-Radio.

Figura 2.3.59 Pantalla connection con un cliente conectado (Funcube).

Figura 2.3.60 Configuración de usuarios. Pestaña user accounts.

Figura 2.3.61 Creación de un nuevo usuario.

Figura 2.3.62 Modificación de la cuenta de usuario.

Figura 2.3.63 Pestaña options. Configuración del servidor SDR.

Figura 2.3.64 Pantalla de configuración “options”.

Figura 2.3.65 Pantalla de información que muestra el servidor.

Figura 2.3.66 Pantalla de selección de equipos y tarjetas de sonido.

Figura 2.3.67 Opciones de publicación del servidor en la Internet.

Figura 2.3.68 Situación global de los servidores de SDR-Radio

Figura 2.3.69 Listado de estaciones SDR-Radio.

Figura 2.3.70 Consola principal de SDR-Radio.

Figura 2.3.71 SDR-Radio conectados localmente a Funcube.

Figura 2.3.72 Pantalla de selección de servidor remoto.

Figura 2.3.73 Relación de servidores remoto SDR-Radio.

Figura 2.3.74 Consola SDR-Radio conectado a servidor DJ9NZ (Karlsruhe).

Figura 2.3.75 Conexión a servidor en red local.

Figura 2.3.76 Conexión a servidor local de SDR-Radio.

Figura 2.3.77 Servidor SDR-Radio.

Figura 2.4.1 Registro de un teléfono SIP.

Figura 2.4.2 Trixbox (Asterisk) conectado a servidor VoIP.

Figura 2.5.1 Instalación Trixbox.

Figura 2.5.2 Comienzo comprobación CD.

Figura 2.5.3 Comprobación del disco.

Figura 2.5.4 Confirmación de chequeo de CD.

Figura 2.5.5 Comienzo de chequeo.

Figura 2.5.6 Chequeo completado.

Figura 2.5.7 Selección de teclado.

Figura 2.5.8 Selección de clave (password).

Figura 2.5.9 Selección de zona horaria.

Figura 2.5.10 Pantalla de configuración.

Figura 2.5.11 Pantalla de configuración de red.

Figura 2.5.12 Interfaz web Trixbox.

Figura 2.5.13 Interfaz web pantalla flash operator panel.

Figura 2.5.14 Pantalla Trixbox Admin.

Figura 2.5.15 Pantalla Trixbox edición de ficheros.

Figura 2.5.16 Pantalla Trixbox información.

Figura 2.5.17 Gestor de teléfonos.

Figura 2.5.18 Información del sistema.

Figura 2.5.19 Mantenimiento del sistema.

Figura 2.5.20 Terminal de acceso al servidor.

Figura 2.5.21 Trixbox pantalla de administración.

Figura 2.5.22 Trixbox. Añadir extensión.

Figura 2.5.23 Trixbox. Definición de extensión.

Figura 2.5.24 Trixbox. Edición de extensión.

Figura 2.5.25 Trixbox. Añadir troncal.

Figura 2.5.26 Trixbox. Añadir troncal Zaptel.

Figura 2.5.27 Trixbox. Adición de la ruta de entrada.

Figura 2.5.28 Trixbox. Edición de la ruta de salida.

Figura 2.5.29 Trixbox. Grupo de llamada.

Figura 2.5.30 Trixbox. Adición de colas.

Figura 2.5.31 Trixbox. Opciones de marcado.

Figura 2.5.32 Trixbox. Buzón de voz.

Figura 2.5.33 Trixbox. Directorio.

Figura 2.5.34 Trixbox. Configuración de fax.

Figura 2.5.35 Trixbox. Configuración de seguridad.

Figura 2.5.36 Trixbox. Flash operator panel.

Figura 2.5.37 Trixbox. Pantalla principal.

Figura 2.5.38 Trixbox. Configuración de ficheros.

Figura 2.5.39 Trixbox. Fichero sip_nat.conf.

Figura 2.5.40 Trixbox. Fichero extensions_customs.conf.

Figura 2.6.1 Vídeo Teléfono Grandstream.

Figura 2.6.2 Detalle de teléfono Grandstream GXV3140.

Figura 2.6.3 Características de teléfono Grandstream GXV3140.

Figura 2.6.4 Display de teléfono Grandstream GXV3140.

Figura 2.6.5 Teléfono Grandstream GXV314 conectado a centralita remota y centralita local.

Figura 2.6.6 Teléfono Grandstream GXV314 Noticias/Stock/Tiempo.

Figura 2.6.7 Teléfono Grandstream GXV314 Marcación y listado de llamadas.

Figura 2.6.8 Teléfono Grandstream GXV314 Directorio y Mensajería.

Figura 2.6.9 Teléfono Grandstream GXV314 Marcación y Llamada.

Figura 2.6.10 Teléfono Grandstream GXV314 Opciones y Redes Sociales.

Figura 2.6.11 Teléfono Grandstream GXV314 Skype.

Figura 2.6.12 Teléfono Grandstream GXV314 Skype. Llamada.

Figura 2.6.13 Teléfono Grandstream GXV314 Twitter y Navegación.

Figura 2.6.14 Teléfono Grandstream GXV314 Radio, Música y Vídeo.

Figura 2.6.15 Teléfono Grandstream GXV314 Configuración.

Figura 2.6.16 Teléfono Grandstream GXV314. Pantalla de información general.

Figura 2.6.17 Teléfono Grandstream GXV314 Pantalla de información de la red.

Figura 2.6.18 Teléfono Grandstream GXV314. Configuración de cuenta. Parámetros generales.

Figura 2.6.19 Teléfono Grandstream GXV314 Configuración de cuenta. Parámetros de red.

Figura 2.6.20 Teléfono Grandstream GXV314 Configuración de cuenta. Parámetros SIP.

Figura 2.6.21 Teléfono Grandstream GXV314 Configuración de cuenta. CÓDECS.

Figura 2.6.22 Teléfono Grandstream GXV314 Configuración de cuenta. Parámetros de llamada.

Figura 2.6.23 Opciones de videocámaras.

Figura 2.6.24 Arquitectura Vídeo IP-SIP (1).

Figura 2.6.25 Arquitectura Vídeo IP-SIP (2).

Figura 2.6.26 Configuración de teléfono Grandstream. Detección de humos.

Figura 2.6.27 Configuración de teléfono Grandstream. Análisis de figura.

Figura 2.6.28 Configuración de teléfono Grandstream. Control de cerradura eléctrica.

Figura 2.6.29 Videocámara Wifi del sistema Marconi.

Figura 2.6.30 Cámara GXV3615W. Interfaz de administración.

Figura 2.6.31 Cámara GXV3615W. Configuración básica: system.

Figura 2.6.32. Cámara GXV3615W. Configuración básica vídeo y audio.

Figura 2.6.33 Cámara GXV3615W. Configuración básica de red.

Figura 2.6.34 Cámara GXV3615W. Configuración básica wifi.

Figura 2.6.35 Cámara GXV3615W. Configuración básica DDNS.

Figura 2.6.36 Cámara GXV3615W. Configuración básica SIP.

Figura 2.6.37 Cámara GXV3615W. Información de estado.

Figura 2.6.38 Página principal de Teamspeak <http://www.teamspeak.com/>.

Figura 2.6.39 Página de descarga de Teamspeak.

Figura 2.6.40 Servidor Teamspeak Marconi.

Figura 2.6.41 Servidor Teamspeak Marconi. Grupos del servidor.

Figura 2.6.42 Servidor Teamspeak Marconi. Permisos asociados a grupos.

Figura 2.6.43 Servidor Teamspeak Marconi. Programar canal, solapa Estándar.

Figura 2.6.44 Servidor Teamspeak Marconi. Programar canal, solapa Audio.

Figura 2.6.45 Servidor Teamspeak Marconi. Programar canal, Solapa Permisos.

Figura 2.6.46 Servidor Teamspeak Marconi. Grupos definidos en los canales.

Figura 2.6.47 Servidor Teamspeak Marconi. Grupos del canal.

Figura 2.6.48 Servidor Teamspeak Marconi. Grupos de canal del usuario.

Figura 4.1.1 Descripción de la metodología a utilizar en el desarrollo de prototipos.

Figura 5.3.1 Arquitectura de red MARCONI.

Figura 5.3.2 Servidores Teamspeak, Trixbox, IPCop, Fldigi/GNU-Radio y Cámara/Videoteléfono.

Figura 5.3.3 Servidores XP, Ubuntu equipos de recepción y Cámara/Videoteléfono.

Figura 5.3.4 Sistema de Control y Distribución de señales.

Figura 5.3.5 Cabecera de radiofrecuencia.

Figura 5.4.1 Firewall IPCop Marconi.

Figura 5.4.2 Pantalla principal del servidor Marconi.

Figura 5.5.1 Panel principal TRIXBOX.

Figura 5.5.2 Videoteléfono conectado a Trixbox (extensión 800).

Figura 5.5.3 Teléfono software ExpressTalk conectado a Trixbox (extensión 100).

Figura 5.6.1 Desktop servidor de receptores XP.

Figura 5.6.2 HDSDR y receptor Funcube Dongle.

Figura 5.6.3 HDSDR y receptor RTL-SDR.

Figura 5.6.4 HDSDR y receptor SDR-IQ.

Figura 5.6.5 HDSDR y receptor Softrock HF.

Figura 5.6.6 Qthid Quisk y receptor Funcube Dongle.

Figura 5.6.7 SdrSharp y receptor Funcube Dongle.

Figura 5.6.8 SdrSharp y receptor RTL-SDR.

Figura 5.6.9 SdrSharp y receptor SDR-IQ.

Figura 5.6.10 Winrad y receptor SDR-IQ.

Figura 5.6.11 Spectravue y receptor SDR-IQ.

Figura 5.6.12 Sigmira con receptor local SDR-IQ (sobre pantalla WebSDR).

Figura 5.6.13 Sigmira pantalla de servidores.

Figura 5.6.14 Sigmira conectado a servidor SDR-Radio con receptor SDR-IQ.

Figura 5.6.15 Cliente SDR-Radio conectado a servidor y usando receptor Funcube.

Figura 5.6.16 SDR-Radio Server pantalla de selección de receptor.

Figura 5.6.17 Cliente SDR-Radio usando receptor RTL-SDR a través de servidor.

Figura 5.6.18 Cliente SDR-Radio usando receptor SDR-IQ a través de servidor.

Figura 5.6.19 Cliente SDR-Radio usando receptor Softrock a través de servidor (detalle de WebSDR con Softrock).

Figura 5.6.20 Cliente CuteSDR usando SDR-IQ a través de SDRxxServer.

Figura 5.6.21 WebSDR con cascadas 40 metros y FM.

Figura 5.6.22 WebSDR. Detalle de diálogo de usuarios.

Figura 5.6.23 Servidor XP con programa WebSDR (Cliente) y Servidor Linux con programa QtRadio.

Figura 5.6.24 Antenas usadas dentro del sistema Marconi.

Figura 5.6.25 Detalle de antenas del sistema Marconi para Flex 1500, Funcube, SDR-IQ.

Figura 5.6.26 Detalle de antenas para Softrock, Perseus, Flex 5000 e ICOM ICM-411.

Figura 5.6.27 Detalle de antenas para receptores integrados en Teamspeak.

Figura 5.6.28 Detalle de antenas para Softrock y Perseus.

Figura 5.6.29 Detalle de antenas para Terratec (RTL-SDR) y SDR-IQ.

Figura 5.6.30 Detalle de distribución de antenas en el CASEM.

Figura 5.6.31 Detalle de distribución de antenas y equipos en el castillete del CASEM.

Figura 5.7.1 Página web del servidor Marconi del grupo S2CN.

Figura 5.7.2 Detalle página web del servidor Marconi del grupo S2CN (1).

Figura 5.7.3 Detalle página web del servidor Marconi del grupo S2CN (2).

Figura 5.8.1 Detalle página web del foro S2CN. Información sobre la aplicación WebSDR.

Figura 5.8.2 Detalle página Web del foro S2CN. Página principal del foro.

Figura 5.9.1 Pantalla principal de GNU Radio.

Figura 5.9.2 Ejemplo de radio FM usando GNU Radio.

Figura 5.10.1 Cabecera de Radiofrecuencia.

Figura 5.10.2 Sistema de Control y Distribución de Señales.

Figura 5.10.3 Diagrama de Bloques Simplificado de la Invención.

Figura 5.10.4 Diagrama de Bloques Simplificado de la Invención.

Índice de tablas.

Tabla 1.1 Tabla con los conceptos que se pueden demostrar con Demostrador Marconi según la asignatura del Grado en Ingeniería Radioelectrónica.

Tabla 2.1 Características de los códecs.

Tabla 5.1 Costes del Demostrador de Conceptos Marconi.

Resumen.

Esta Tesis Doctoral se estructura en seis capítulos más un capítulo de bibliografía y un capítulo de apéndices:

Capitulo 1: Introducción.

Describe qué es el Procesado Digital de la Señal, los Modos de Radiocomunicaciones Digitales e introduce al SDR. Hace una descripción resumida del Demostrador de Conceptos Marconi y de su aplicación.

Capitulo 2: Estado de la cuestión.

Describe la tecnología SDR así como los principales desarrollos SDR: GNU-Radio, HPSDR, WebSDR, QtRadio y SDR-Radio.

Describe la telefonía y la voz sobre IP, así como los diferentes equipos y programas usados.

Capitulo 3: Objetivos de la Tesis Doctoral.

Los objetivos de esta Tesis Doctoral son: crear un Demostrador de Conceptos que ayude a asentar las bases técnicas, el acceso completo a dicho sistema y equipos desde cualquier parte del Mundo donde haya una conexión a Internet y la aplicación a las Radiocomunicaciones tanto de las nuevas tecnologías de la Información y las Telecomunicaciones, así como los avances en ROIP y SDR

Capitulo 4: Metodología propuesta para la realización de esta Tesis Doctoral.

Describe la metodología empleada para el desarrollo de esta Tesis Doctoral, cómo se han realizado y validado las aplicaciones, y cómo se han desarrollado los distintos subsistemas de la aplicación Marconi.

Capitulo 5: Resultados – Aplicaciones prácticas realizadas en el Laboratorio del Grupo S2CN.

Describe las aplicaciones probadas dentro del sistema Marconi, así como los desarrollos realizados. Se recoge información tanto de las patentes como de los artículos publicados en el desarrollo de esta Tesis.

Capitulo 6: Conclusiones.

Se relatan las conclusiones obtenidas dentro de esta Tesis Doctoral: la aplicación del sistema Marconi a la Investigación y la evolución de la ROIP y SDR.

Capitulo 7 y 8 Bibliografía y Apéndices.

Incluye la bibliografía consultada para el desarrollo de esta Tesis así como la relación de apéndices que recogen una información más detallada de algunos de los puntos desarrollados en esta Tesis Doctoral.

1. Introducción.

1.1 Introducción al Procesado Digital de la Señal.

El procesado digital de la señal (Digital Signal Processing. DSP) es una de las tecnologías más ampliamente desarrolladas, aplicándose tanto al área de las comunicaciones, como a imágenes, RADAR, SONAR, música de alta fidelidad y otras. Cada área ha desarrollado sus propios algoritmos, matemáticas y técnicas especializadas.

El procesado digital se caracteriza porque los datos utilizados son señales que provienen del mundo real y pueden ser tanto ondas acústicas, ondas de radio, imágenes, etc. Refiriéndose como DSP al conjunto de algoritmos y técnicas utilizados para manejar estas señales, una vez que éstas se han convertido en datos en un dispositivo electrónico digital.

El objetivo del DSP es muy variado y puede incluir tanto la mejora de las imágenes, como el reconocimiento y generación de la voz, la compresión, almacenamiento y transmisión de datos, etc.

Las raíces de la DSP comienzan hacia las décadas de los 60, 70, cuando aparecieron los primeros ordenadores, se digitalizaron las señales y se empezó a afrontar la pregunta de cómo manejarlas.

Los primeros ordenadores eran muy caros y se dirigió esta tecnología a puntos clave, tales como la seguridad nacional (RADAR, SONAR), la salud, así como a aquellas áreas donde los datos son difíciles de obtener y reemplazar así como donde existe una alta posibilidad de negocio.

La revolución de los ordenadores en las décadas de los años 80 y 90 hizo que el DSP se abriera paso hacia el mercado comercial, apareciendo en equipos móviles, teléfonos, reproductores de DVD, correos de voz, etc.

Se puede ver en figura 1.1.1 los campos de aplicación de la DSP.

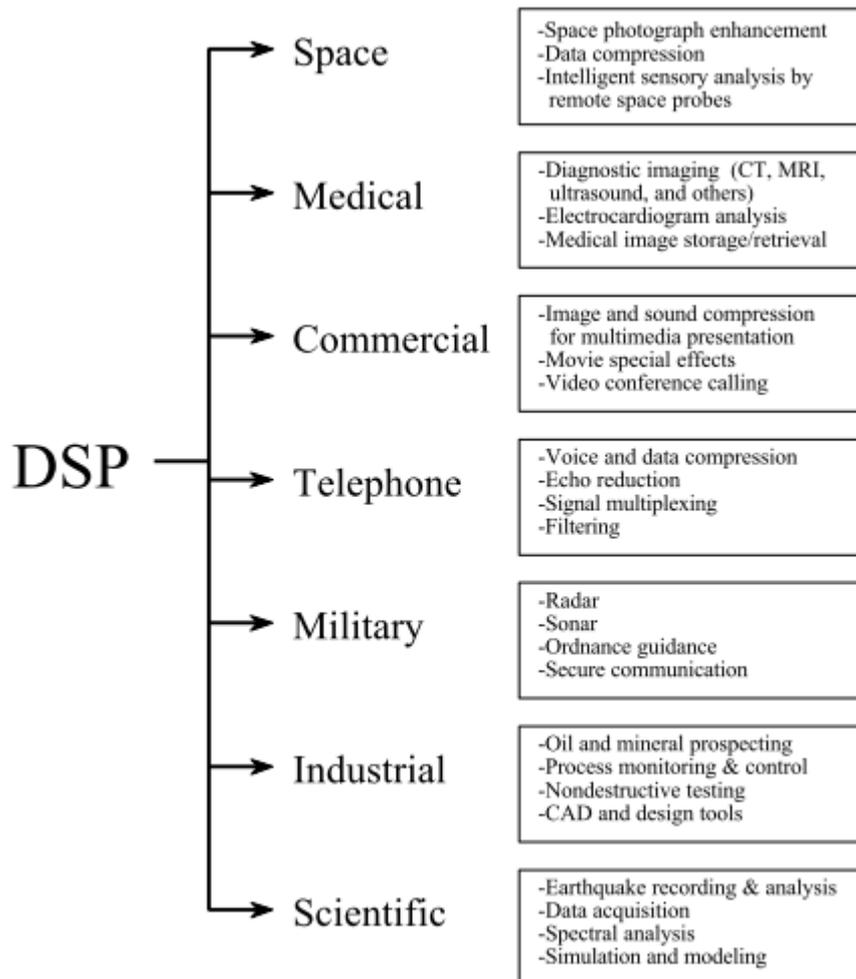


Figura 1.1.1 Aplicaciones del procesado digital de la señal.
Figura cortesía de [SMIT97]

Como vemos en figura 1.1.2 la frontera entre DSP y otras técnicas es más bien difusa, aunque para comprender el ámbito de aplicación de esta tecnología, hay que conocer un poco de estas materias.

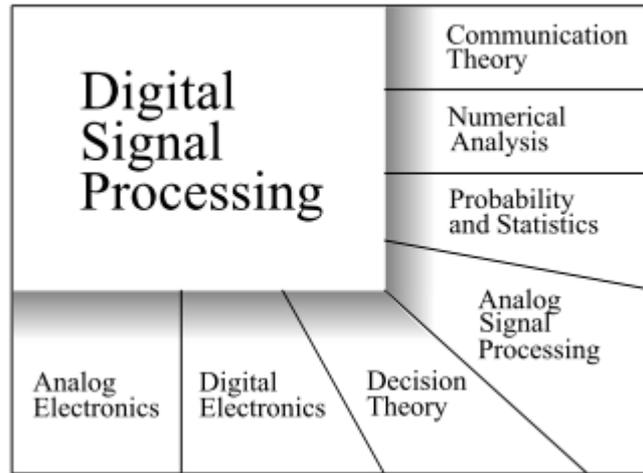


Figura 1.1.2 Áreas de aplicación del procesado digital de la señal.
 Figura cortesía de [SMIT97]

1.1.1 Telecomunicaciones.

Las Telecomunicaciones tratan la transferencia de información de un lugar a otro. Esto incluye muchas formas de información: conversaciones telefónicas, señales de televisión, archivos informáticos, y otros tipos de datos. Para transferir la información, se necesita un canal entre los dos lugares. Este puede ser un par de hilos, la señal de radio, fibra óptica, etc. Las empresas de telecomunicaciones reciben el pago por la transferencia de información de sus clientes, mientras que deben pagar para establecer y mantener el canal. El balance es simple: cuanta más información se pueda pasar a través de un solo canal, más dinero se gana. El DSP ha revolucionado los sectores de las telecomunicaciones en muchas áreas: como la generación de tonos de señalización y detección, filtrado para eliminar el ruido de las líneas, etc.

Tres áreas específicas de las telecomunicaciones son la multiplexación, la compresión y el control de eco.

1.1.1.1 Multiplexado.

El DSP convierte las señales de audio en un flujo de datos digitales en serie, y muchas conversaciones telefónicas se pueden transmitir en un solo canal. La ventaja económica de la transmisión digital es importante.

1.1.1.2 Compresión.

Cuando una señal de voz se digitaliza, gran parte de la información digital es redundante. Es decir, la información transportada por cualquier muestra se duplica en gran medida por las muestras adyacentes. Se han desarrollado gran cantidad de algoritmos para convertir las señales de voz digitalizadas en datos corrientes que requieren menos bits/seg. Éstos se llaman algoritmos de compresión de datos. A su vez se usan los algoritmos de descompresión para restaurar la señal a su forma original.

La compresión más alta alcanzable es de alrededor de 2 kilobits/seg., lo que produce un sonido que está muy distorsionado, pero útil para algunas aplicaciones militares y las comunicaciones submarinas.

1.1.1.3 El control del eco.

El eco es un problema grave en las conexiones telefónicas de larga distancia. Cuando se habla por teléfono, la señal viaja hacia el receptor y una parte de la señal se devuelve como eco. Si la conexión se encuentra a unos cientos de kilómetros, el tiempo transcurrido para la recepción del eco es sólo unos pocos milisegundos. El oído humano está acostumbrado a oír el eco con pequeños retardos de tiempo, y la conexión suena bastante normal. A medida que la distancia se hace más grande, el eco se vuelve cada vez más notable e irritante. El retardo puede ser de varios cientos de milisegundos para comunicaciones a larga distancia, resultando bastante molesto.

El DSP ataca este tipo de problema mediante la medición de la señal de retorno y la generación de una señal adecuada para anular este eco. Esta misma técnica se puede utilizar también para reducir el ruido ambiental.

1.1.2 Procesamiento de audio.

Los dos principales sentidos del ser humano son la visión y la audición y gran parte de la DSP está relacionado con la imagen y el sonido. El DSP ha introducido cambios importantes en el procesado de la música y la palabra. La utilización de datos digitales es importante para evitar la degradación de la señal comúnmente asociada con el almacenamiento y la manipulación analógica.

La generación y el reconocimiento de voz se utilizan para la comunicación entre los seres humanos y máquinas. En grabaciones digitales la voz se digitaliza y almacena, por lo general en una forma comprimida. Durante la reproducción, los datos almacenados se

descomprimen y se convierten de nuevo en una señal analógica. Toda una hora de discurso grabado requiere sólo de unos tres Megas de almacenamiento.

1.1.3 Localización del eco.

Un método común de obtener información sobre un objeto remoto es hacer que una onda se refleje sobre él. Por ejemplo, el RADAR funciona mediante la transmisión de pulsos de ondas de radio, y el examen de la señal recibida. En el SONAR, las ondas de sonido se transmiten a través del agua para detectar submarinos y otros objetos sumergidos.

1.1.3.1 RADAR.

En el sistema de RADAR más simple, un transmisor de radio produce un pulso de energía en una frecuencia de radio de unos pocos microsegundos de longitud, este pulso se envía a una antena altamente direccional, donde la onda de radio resultante se propaga a la velocidad de la luz. Los aviones en el camino de esta onda reflejan una pequeña porción de la energía de vuelta hacia una antena de recepción, situada cerca de la torre de transmisión, en el caso de no ser la misma antena.

La distancia al objeto se calcula a partir del tiempo transcurrido entre el pulso transmitido y el eco recibido. La dirección del objeto se encuentra de forma simple, se sabe dónde se apuntó la antena direccional y cuándo se recibió el eco.

El DSP ha revolucionado el radar. En primer lugar, DSP puede comprimir el pulso después de que se recibe, proporcionando mejor determinación de la distancia, sin reducir la gama de funcionamiento. En segundo lugar, DSP puede filtrar la señal recibida para disminuir el ruido. En tercer lugar, DSP permite la selección rápida y la generación de diferentes formas y longitudes de pulso, entre otras cosas, esto permite que el pulso pueda ser optimizado para una detección en particular.

1.1.3.2 SONAR.

Se divide en dos categorías, activo y pasivo. En el SONAR activo, los pulsos de sonido entre 2 kHz y 40 kHz se transmiten en el agua, y los ecos resultantes se detectan y analizan. El uso de los sonares activos incluye: la detección y localización de los cuerpos de submarinos, la navegación, la comunicación y la representación del fondo del mar. Es típico un alcance de operación máximo de 10 a 100 km. En comparación, el sonar pasivo simplemente escucha sonidos bajo el agua,

que incluye: la turbulencia natural, la vida marina y los sonidos mecánicos de submarinos y buques de superficie. El SONAR pasivo no emite energía, es ideal para las operaciones encubiertas. La aplicación más importante de sonar pasivo es en sistemas de vigilancia militar que detecta y rastrea submarinos. El SONAR pasivo normalmente utiliza frecuencias inferiores al SONAR activo, al propagarse a través del agua con menor absorción, el alcance de detección pueden ser miles de kilómetros.

El DSP ha revolucionado el SONAR en muchas de las mismas áreas que el RADAR: generación de pulso, compresión de impulsos y el filtrado de señales detectadas.

El SONAR es más simple que el RADAR debido a las frecuencias más bajas involucradas.

Bajo otro punto de vista, el SONAR es más complejo que el RADAR porque el entorno es mucho menos uniforme y estable

1.1.3.3 Reflexión sísmológica.

Los geofísicos descubrieron que la estructura de la corteza de la tierra podría ser determinada con sonido. Se podía provocar una explosión y registrar los ecos de las capas de la tierra hasta 10 kilómetros por debajo de la superficie. El método sísmico de reflexión se convirtió rápidamente en el método principal para la localización de yacimientos de petróleo y minerales.

En el caso ideal, un pulso de sonido enviado al suelo produce un solo eco para cada capa límite a través de la que pasa el pulso. Cada eco que vuelve a la superficie debe pasar por todas las otras capas límite por encima de donde es originario. Esto puede resultar en más eco rebotando entre las capas, dando lugar a ecos de ecos que se detectan en la superficie. Estos ecos secundarios pueden hacer que la señal detectada sea muy complicada y difícil de interpretar.

El Procesamiento Digital de Señales ha sido ampliamente utilizado para aislar el eco principal de los ecos secundarios en los sismogramas de reflexión.

1.1.4 Procesamiento de imágenes.

Las imágenes son señales con características especiales. En primer lugar, son una medida de un parámetro en el espacio (distancia), mientras que la mayoría de las señales son una medida de un parámetro en el tiempo. En segundo lugar, contienen una gran cantidad de información. En tercer lugar, el juicio final de la calidad es a menudo

subjetivo. Estas características especiales han hecho el procesamiento de la imagen un subgrupo distinto dentro del DSP.

1.1.5 Medicina.

Los rayos X estaban limitados por varios problemas hasta que aparecieron las técnicas DSP. En primer lugar, los órganos superpuestos en el cuerpo pueden esconderse entre sí. En segundo lugar, no siempre es posible distinguir entre tejidos similares. En tercer lugar, las imágenes de rayos X muestran la anatomía, la estructura del cuerpo, y no la fisiología, el funcionamiento del cuerpo. En cuarto lugar, la exposición a los rayos X puede causar cáncer, haciendo que sea utilizado con moderación y sólo con justificación adecuada.

El problema de los órganos superpuestos se resolvió con la introducción del primer escáner de tomografía computarizada. La tomografía computarizada es un clásico ejemplo de procesamiento digital de la señal digital. Se pasan los rayos X desde muchas direcciones a través de la sección del cuerpo del paciente que está siendo examinado. Las señales se convierten en datos digitales y se almacenan en un ordenador. La información se utiliza entonces para calcular imágenes que muestran secciones del cuerpo. Estas imágenes muestran mucho mayor detalle que las técnicas convencionales, lo que permite significativamente mejor diagnóstico y tratamiento

Los problemas de los rayos X se han resuelto mediante el uso de la energía de las ondas de radio y sonido. El DSP juega un papel clave en todas estas técnicas. Por ejemplo, la resonancia magnética usa campos magnéticos en conjunción con ondas de radio para sondear el interior del cuerpo humano. Ajustando la fuerza y frecuencia de los campos se consigue que los núcleos atómicos en una región localizada del cuerpo entren en resonancia. Esta resonancia da lugar a la emisión de una onda de radio secundaria, detectando con una antena colocada cerca del cuerpo, la fuerza y las otras características de esta señal y proporcionando información localizada en la región en resonancia.

1.1.6 Espacio.

A veces, sólo hay que sacar el máximo provecho de una mala imagen, es el caso de las imágenes tomadas por los satélites no tripulados y vehículos de exploración en el espacio. El DSP puede mejorar la calidad de las imágenes tomadas bajo condiciones extremadamente desfavorables en varios modos: brillo, ajuste de contraste, detección de bordes, reducción de ruido, ajuste de enfoque,

reducción de desenfoque en el movimiento, etc. Las imágenes que tienen distorsión espacial, como las que aparecen cuando se toma una imagen plana de un planeta esférico, también se pueden transformar a una representación correcta. Muchas de las imágenes individuales también se pueden combinar en una sola base de datos, permitiendo que la información se muestre de una forma única.

1.1.7 Imágenes en productos comerciales.

Los sistemas comerciales deben ser baratos, y esto no encaja bien con las grandes velocidades de transferencia de datos. Al igual que con las señales de voz, las imágenes contienen una enorme cantidad de información redundante y se puede reducir a través de algoritmos que disminuyan el número de bits necesarios para su representación. La televisión y otras imágenes en movimiento son especialmente adecuadas para la compresión, ya que la mayoría de las imágenes siguen siendo las mismas de cuadro a cuadro.

Los productos que usan la tecnología DSP son: videoteléfonos, programas de ordenador que muestran imágenes en movimiento y la televisión digital .

1.2 Introducción a los modos de Radiocomunicaciones Digitales.

1.2.1 Introducción.

Se conocen genéricamente por Comunicaciones Digitales (CC.DD.) aquellas que tienen lugar entre equipos que utilizan códigos basados en dos o más estados de los circuitos: conducción/no conducción, designados como "ON/OFF" ó "1/0". Las señales generadas por la voz, son de tipo analógico, convirtiéndose a digital para aumentar su capacidad de transmisión y manipulación.

La evolución de los ordenadores ha potenciado decisivamente las comunicaciones digitales, haciéndolas asequibles tanto a la industria como a los particulares, y ha permitido experimentar una serie de modalidades de comunicaciones digitales algunas de las cuales se han popularizado y extendido notablemente durante los últimos años.

Se trata básicamente de teclear la información en un ordenador, o disponer de la que tenga almacenada, para entregarla al módem (modulador/demodulador) que la procesa (codifica) y la entrega a su vez al equipo de radio para su retransmisión. O bien, una vez

recibida del equipo receptor, a través de la demodulación y decodificación la facilita al ordenador para su proceso, lectura o almacenamiento.

Debe intervenir además un segundo equipo que controla el protocolo que puede ser relativamente simple o complejo, según la modalidad. Este equipo puede funcionar como un apéndice del ordenador (generalmente adosado al módem) o bien como un programa especializado del propio ordenador. Tiene por misión encargarse de la correcta interpretación de los caracteres y, en según que casos, de la integridad de la información y del gobierno de la operativa del módem, que a su vez controla al equipo. A este equipo se le denomina TNC o controlador del Terminal de nodo.

Existen diversas formas de modulación, pero las básicas para comprender el funcionamiento de las comunicaciones digitales son las que se emplean en la operación en Radioteletipo (RTTY) y sus derivados y frecuentemente (aunque no exclusivamente) en Radiopaquete (*Packet Radio*). Ambas conservan en su esencia una gran similitud con la operación en Radiotelegrafía (CW). En esta última se trata de transmitir y/o cortar una portadora que está (SI/ON) o no está (NO/OFF), mientras que en radioteletipo la portadora permanece constantemente durante la transmisión y lo que varía constantemente es su frecuencia que se mueve entre dos valores conocidos como MARCA y ESPACIO. La MARCA o estado SI/ON y al ESPACIO le corresponde el estado NO/OFF. A este sistema de modulación se le conoce como Manipulación por Desplazamiento de Frecuencia, FSK (*Frequency Shift Keying*).

Cuando lo que varía son los tonos que se utilizan para modular la portadora, empleando uno más agudo como MARCA, frente a otro más grave, ESPACIO, a este sistema se le conoce como Manipulación por Desplazamiento de Audio-Frecuencia AFSK (*Audio Frequency Shift Keying*).

Otra forma de modulación es la Manipulación por Desplazamiento de Fase: PSK (*Phase Shift Keying*). Aquí la marca y el espacio se logran invirtiendo o desplazando la fase. Tiene diversas derivadas como el Desplazamiento Binario de Fase, BPSK (*Binary Phase Shift Keying*), QPSK (*Quadrature Phase Shift Keying*), etc. Se emplean tanto en la operación en banda lateral única (SSB) como en modulación en frecuencia (FM).

Existen otras formas sensiblemente diferentes de modulación y modalidades que no encajan exactamente con las descritas, pero al tener el común denominador de utilizar microprocesadores y ordenadores para su ejecución, se engloban todas dentro del colectivo genérico conocido como Comunicaciones Digitales.

1.2.2 Radioteletipo (RTTY).

El Radioteletipo es la modalidad más antigua de comunicación digital si exceptuamos el Morse. Se emplea básicamente en HF. La operación en VHF y superiores es rara y ha sido desplazada últimamente por el radiopaquete.

Se distinguen varios tipos: el Baudot (o Murray), el ASCII y el SITOR, y dentro de esta última dos operativas diferentes: ARQ y FEC.

Los programas más utilizados por los aficionados son: HAMCOMM, JVComm32 y FLDigi.

1.2.2.1 BAUDOT (o Murray).

Se conoce también como Alfabeto Telegráfico Internacional Número 2 (ITA2). Representa cada carácter mediante un grupo de cinco bits, de manera que cada carácter tiene una combinación diferente de bits, por lo que solo existen 32 combinaciones posibles. Todos los textos van en mayúsculas y en las cifras y los signos de puntuación son obligados pasar del juego de letras (LTRS) a cifras (FIGS). La velocidad de transmisión estándar en HF es de 45 baudios (60 palabras por minuto) con modulación FSK o AFSK, pero también se utilizan 50, 56, 75, 100, y 300 o más baudios.

Murray fue un Ingeniero Radio americano que modificó el código someramente.

Los programas más utilizados por los aficionados son: HAMCOMM, JVComm32 y FLDigi.

1.2.2.2 ASCII.

ASCII significa *American National Standard Code For Information Interchange*. Corresponde al Alfabeto Telegráfico Internacional Número 5 (IT5). La diferencia con el Baudot es que cada carácter se forma con 7 bits con lo que aumenta el juego de caracteres y se pueden representar mayúsculas y minúsculas y caracteres de control. Este alfabeto fue ideado para el uso con ordenadores y transmisión de datos y es el básico en estos sistemas. La velocidad de transmisión por radio generalmente es de 110 o 300 baudios y modulación FSK, en HF.

Ambos, Baudot y ASCII no disponen de sistema de detección de errores por lo que si una emisión es interferida o la recepción resulta dificultosa, puede perderse fácilmente información por lo que tiene muy limitada su operación en modo desatendido o automático. Nunca existe la certeza de que el mensaje o el texto ha llegado íntegro al receptor.

Los programas más utilizados por los aficionados son: HAMCOMM, JVComm32 y FLDigi.

1.2.2.3 SITOR.

Simplex Telex Over Radio se empezó a utilizar a principios de los 80. Al igual que el Baudot, representa cada carácter mediante un grupo de cinco bits.

Está pensado para la operación en condiciones adversas en HF (SITOR estaba dedicado al uso marítimo). Es básicamente RTTY pero añadiéndole detección de errores.

La velocidad de transmisión suele ser de 100 baudios, modulación FSK.

Las estaciones que operan SITOR se clasifican en *Information Sending Station* (ISS) y *Information Receiving Station* (IRS), respectivamente transmisores y receptoras de información. La estación que origina la comunicación se denomina "master" (OISS) y la correspondiente "slave" (OIRS).

En la versión de aficionado o AMTOR está diseñado de forma que para su identificación las estaciones utilizan un código de cuatro letras. Esto es incompatible con los indicativos otorgados, por lo que la mayoría de estaciones optan por usar la primera y las tres últimas letras.

Los programas más utilizados por los aficionados son: HAMCOMM, JVComm32 y FLDigi.

1.2.2.4 FAX.

Es una modalidad pensada para la transmisión y recepción de imágenes. Aunque existen módems y programas aptos para transmitir en enlaces terrestres, bien sea por HF como por VHF, y existen frecuencias reservadas, según los planes de banda de la IARU, los radioaficionados la utilizan mayoritariamente para la recepción de mapas y cartas meteorológicas WEFAX (*Weather Fax*), bien sea procedentes de estaciones comerciales terrestres en HF, como de satélites polares (NOAA, Meteor, Okean, etc.) en 136 MHz o de geoestacionarios (Meteosat, GOES, GMS) en 1.6 GHz.

Existe un formato asequible a los usuarios de baja resolución, llamado ATP (*Automatic Picture Transmission*), y otros más complejos como el llamado HRPT (*High Resolution Picture Transmission*).

Los programas más utilizados son JVFAX, HFFAX, WEFAX.

1.2.2.5 SSTV (*Slow Scan TV-Televisión de barrido lento*).

Es una modalidad que permite el envío y la recepción de imágenes de calidad a través de enlaces de radio. Se usa sobretodo en HF pero también se registra actividad en VHF y superiores.

Hay diversos procedimientos para codificar imágenes SSTV con el fin de transmitir las pero la base de todas ellas consiste en digitalizar la imagen dibujando una serie de líneas, de izquierda a derecha, de arriba abajo. Se transmiten cada una de estas líneas mediante variaciones de tono que corresponden a variaciones de brillo. Cuanto más alto el tono, más brillo.

Para la detección del final de línea existen dos procedimientos: en formato asíncrono se transmite un tono de baja frecuencia de 1.200 Hz. Los otros se basan en patrones de tiempo exacto, predeterminado.

Para la retransmisión de imágenes en color, según la modalidad se utilizan partes de cada línea para la información de brillo y color, respectivamente. Otras modalidades utilizan líneas distintas para información de brillo y color.

Según la modalidad aumenta el número de líneas, permitiendo mayor o menor definición. Varía entre 120 (blanco y negro) a 256 (color) en modos asíncronos y entre 240 y 400 en modos no asíncronos.

Los programas más utilizados son JVcomm32, MMSSTV y Multimode.

1.2.2.6 ARQ (*Automatic Repeat reQuest*).

Es la modalidad A de SITOR. La estación transmisora (ISS) emite la información en bloques de tres caracteres a la receptora (IRS). Una vez transmitido un bloque, la ISS pasa a la escucha de la IRS para recibir su "acuse de recibo". Si la receptora se lo da, transmite tres caracteres más, caso contrario, los repite hasta que no reciba la conformidad de la receptora. Los tiempos de transmisión y escucha son inferiores al cuarto de segundo por lo que esta modalidad exige un gran esfuerzo a los conmutadores de los equipos que deben ser capaces de aguantar este ritmo. Esta modalidad se utiliza

para contactos entre dos estaciones. Es fácilmente identificable por su sonido característico conocido como "chirp".

El modo L (*listen*) se utiliza para escucha y control de la operativa en modo A.

Los programas más utilizados son el JVcomm32 y TrueTTY.

1.2.2.7 FEC (Forward Error Correction).

Modo B de SITOR es un sistema que se puede denominar de "radiodifusión" (*broadcast*) entendiéndose por ello que solo existe una transmisora denominada "estación transmisora B-Colectiva" (CBSS) y un número indeterminado de "estaciones receptoras B-Colectivas" (CBRS). La CBSS envía cada carácter dos veces. La transmisión primera de un carácter llamada DX, va seguida de otros cuatro. Luego envía la repetición llamada RX.

Modo B-selectivo: Este modo está previsto para la transmisión hacia una sola estación o grupo de ellas.

Los programas más utilizados por los aficionados son: HAMCOMM, JVComm32 y FLDigi.

1.2.2.8 Radio paquete (Packet-Radio).

Esta es la modalidad principal de las CC.DD, desde su aparición ha desplazado a las modalidades anteriores debido a sus mejores prestaciones. Es un sistema de transmisión de datos, libre de errores y, respecto a los anteriores de "alta" velocidad, muy apto para transferir grandes transferencias de datos.

Prevé las colisiones y pérdidas por cambios de propagación o interferencias.

Permite la racionalización del espectro compartiendo canal o frecuencia varias estaciones al mismo tiempo. Con la ventaja sobre el AMTOR que permite coexistir diversas transmisoras y diversas receptoras a la vez.

Contempla el uso en red por lo que, estaciones que no dispongan de enlace entre sí, pueden contactar utilizando terceras como repetidoras.

Permite la operación en "semi" y "full-duplex" y conexiones simples o múltiples, así como con la propia estación.

La información transmitida va seccionada en "paquetes" de ahí su nombre. Cada paquete, además de la propia información añade datos sobre dirección, comprobación de errores y control. En la información sobre la dirección se incluyen los indicativos de

la estación emisora y de la receptora, así como los de aquéllas que, de haberlas, estén siendo utilizadas como repetidoras.

Los programas más utilizados son: TrueTTY, SeaTTY y SkySweeper.

1.2.2.9 PACTOR y PACTOR-II.

El sistema Pactor corrige las deficiencias en calidad del Amtor y el Radiopaquete en enlaces HF. Pactor es un desarrollo de la firma SCS. Se usa en modulación FSK y permite no solo el juego de caracteres ASCII completo sino también transferencia binaria y compresión Huffman. Pactor opera a 200 baudios y Pactor II a 1.200 siendo compatibles entre sí.

Su robustez en condiciones precarias de enlace hace que Pactor-II vaya desplazando al sistema Radiopaquete en HF, convirtiéndose en el estándar de estas bandas para CC.DD.

El programa más utilizado es Multimode.

1.2.2.10 CLOVER y CLOVER-II:

El sistema Clover fue desarrollado por Ray Petit, W7GHM y lo dió a conocer el año 1.990 con el nombre de "Cloverleaf". Posteriormente otros radioaficionados y la casa HAL aportaron mejoras, hasta la aparición de Clover-II. Utiliza modulación PSK mediante la cual facilita simulación "full-duplex" y alcanza los 750 baudios. Es apto para la operación en HF, especialmente cuando las condiciones son correctas. Necesita transceptores muy estables en frecuencia. Los datos se transfieren automáticamente entre dos estaciones enlazadas. Una característica interesante de esta modalidad es la habilidad para el ajuste automático de la potencia de transmisión requerida en los enlaces.

Los programas más utilizados son Clovermail, DSPWin y DSRTTY.

1.2.2.11 G-TOR.

El nombre proviene de la contracción de Golay-Tor, es una innovación de la casa KANTRONICS. Se trata de una mejora del sistema Pactor. Incorpora como este último el protocolo de compresión Huffman y permite la transferencia binaria con la adición de un protocolo diseñado por M. Golay y empleado en la retransmisión de fotografías de Saturno y Júpiter desde el Voyager. Se opera en modo FSK y tiene la habilidad de

conmutar automáticamente la velocidad de transmisión entre 100 y 300 baudios si varían las condiciones.

El programa más usado es GTOR.

1.2.2.12 PSK31.

Este sistema ha sido diseñado por Peter Martínez G3PLX, autor de Amtor, desarrollando una idea de Pawel Jalocho SP9VRC. Se trata de una forma de realizar contactos "teclado a pantalla" similar al RTTY. No dispone de un protocolo a nivel de enlace. El emisor y los receptores se sincronizan solos e incorpora corrección de errores. Se basa en el empleo de un alfabeto de longitud variable (varicode) de 255 caracteres que incorporan al completo el alfabeto ANSI. La velocidad de transmisión es de 31,25 bps y se consigue una velocidad real de 50 p.p.m. Utiliza dos tipos de modulación: BPSK y QPSK. Su otra gran ventaja es que ocupa un ancho de banda de 40Hz a -3Db (el de los otros modos varía entre 300 y 600) por lo que resulta apropiado para condiciones difíciles y la utilización de filtros estrechos.

El programa más usado en Multipsk.

1.3 Procesado Digital de la Señal y SDR.

Radios Definidas por Software (SDR) es la técnica de conseguir acercar el código a la antena, convirtiendo problemas "hardware" de radio en problemas "software" de radio.

La característica fundamental de la Radio Definida por Software es que el software define las formas de onda transmitidas, y el software demodula las formas de onda recibidas.

Radio Definidas por Software es una revolución en el diseño de la radio debido a su capacidad para crear radios que se puedan modificar sobre la marcha, creando nuevas opciones para los usuarios. En líneas generales, las radios por software pueden hacer casi cualquier cosa que una radio tradicional pueda hacer. La parte emocionante es la flexibilidad que el software le proporciona. En lugar de tener equipos de función fija, en los próximos años vamos a ver un movimiento a los equipos de comunicación universales.

Lo más importante de todo es la posibilidad de construir sistemas de comunicación descentralizados. Si nos fijamos en los sistemas de hoy en día, la gran mayoría están basados en la infraestructura.

Las Emisoras de Radio y TV ofrecen una vía de un solo sentido, están fuertemente regulados y el contenido está controlado por una serie de organizaciones oficiales. Un sistema centralizado limita la velocidad de la innovación.

En lugar de usar teléfonos móviles con capacidad limitada por el operador, que se puedan utilizar sólo si existe infraestructura, se podrían construir dispositivos más inteligentes. Estos dispositivos propiedad de los usuarios generarían la red.

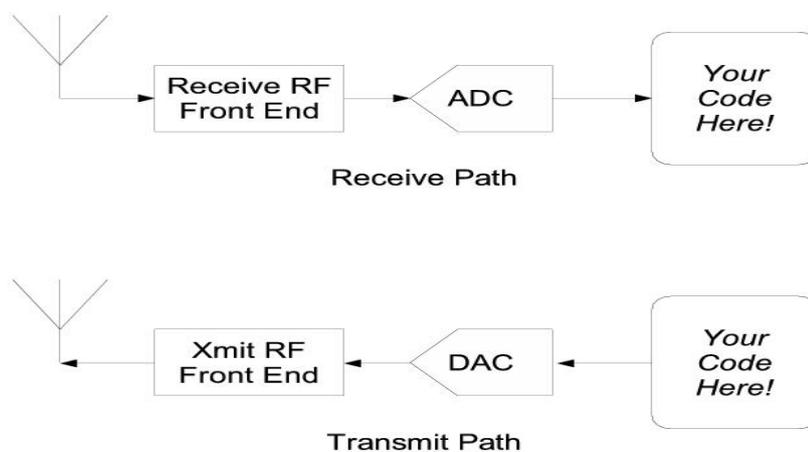


Figura 1.3.1 Diagrama de bloques de Radio definida por software.
Figura cortesía de [BLOS04]

La figura 1.3.1 muestra un diagrama de bloques típico de una radio definida por software. Para entender la parte del software de la radio, primero tenemos que entender un poco sobre el hardware asociado. En la Figura 1.3.1 vemos una antena, un bloque frontal de RF, un convertidor de analógico a digital (ADC) y una cantidad de líneas de código. El convertidor de analógico a digital es el puente entre el mundo físico de señales analógicas continuas y el mundo de muestras digitales discretos manipulados por el software.

Los ADCs tienen dos características principales: la frecuencia y el rango dinámico de muestreo. La frecuencia de muestreo es el número de veces por segundo que el ADC mide la señal analógica. El rango dinámico se refiere a la diferencia entre la señal más pequeña y más grande que se puede distinguir: es una función del número de bits en la salida digital del ADC y el diseño del convertidor. Por ejemplo, un convertidor de 8 bits

como máximo puede representar 256 (2 elevado a 8) niveles de señal, mientras que un convertidor de 16 bits representa hasta 65.536 niveles.

Antes de ir al software, tenemos que hablar un poco de la historia. En 1927, un físico de origen sueco e ingeniero eléctrico llamado Harry Nyquist determinó que para evitar la pérdida de la forma de la señal o aliasing, cuando una señal se convierte de analógica a digital, la frecuencia de muestreo del ADC debe ser al menos el doble de la frecuencia más alta contenida en el ancho de banda de la señal de interés.

Suponiendo que estamos tratando con señales donde el ancho de banda de interés va de 0 a f_{max} , el criterio de Nyquist establece que nuestra frecuencia de muestreo debe ser de al menos $2 * f_{max}$. Pero si nuestro ADC funciona a 20MHz, ¿cómo podemos escuchar una radio FM de 100 MHz? La respuesta es la etapa inicial de RF (o front-end). La etapa inicial de RF (o front-end) desplaza el rango de frecuencias que aparecen en su entrada a un rango más bajo a su salida. Por ejemplo, podríamos imaginar un front-end de RF que desplaza las señales que ocurren en el rango de 90 a 100 MHz hasta el rango 0 a 10MHz.

Podemos tratar el front-end de RF como un bloque mezclador con un solo cometido, situar el centro de la banda de entrada que hay que convertir en la frecuencia intermedia. La potencia de los ordenadores actuales, por encima de 2 GHz, permiten procesar 2000 millones de muestras en coma flotante en un filtro FIR, esto permite que hoy sea posible hacer cualquier sistema de comunicaciones, impensable hace unos años.

Con un ordenador de 2 GHz y 256 MB de RAM es suficiente para procesar la señal, debiendo estar dotando de una tarjeta de sonido de al menos 96 KHz, obviamente hace falta un front-end de RF que pueda desplazar las frecuencias de radio a las bandas de audio.

Una vez muestreadas las señales, se bajan a banda base y se realiza el proceso de DIEZMADO. Este proceso se realiza por el módulo “digital down converter”: ver figura 1.3.2 .

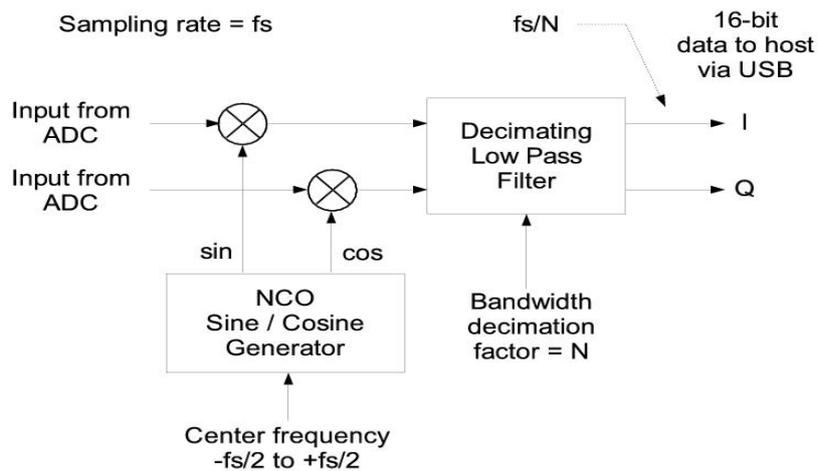


Figura 1.3.2 Diagrama de bloques de “digital down converter”.
Figura cortesía de [BLOS04]

Si tenemos un módulo “Front-End” de RF que seleccione una banda de 6 MHz en la banda de 50 MHz - 800 MHz y lo transforme a una banda de 6 MHz centrada en 5.75MHz. Al muestrear en el tiempo tenemos una representación como se ve en figura 1.3.3.

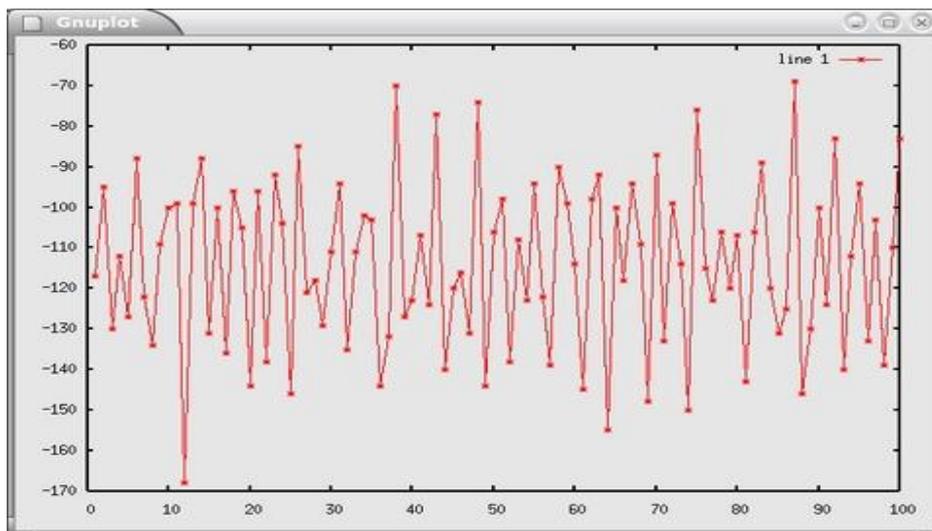


Figura 1.3.3 Muestras de ADC en el dominio del tiempo.

Figura cortesía de [BLOS04]

El dominio de la frecuencia proporciona información adicional. En este caso, si

tomamos las muestras de la señal en el tiempo y calculamos la transformada discreta de Fourier usando la transformada rápida de Fourier (FFT). Esto nos da una representación de las frecuencias contenidas en la señal de entrada. La Figura 1.3.4 muestra el espectro resultante. El eje x es la frecuencia, y el eje y es la potencia en decibelios ($10 * \log_{10}$ de la potencia). El límite inferior es cero Hz, y el extremo superior está a 10 MHz.

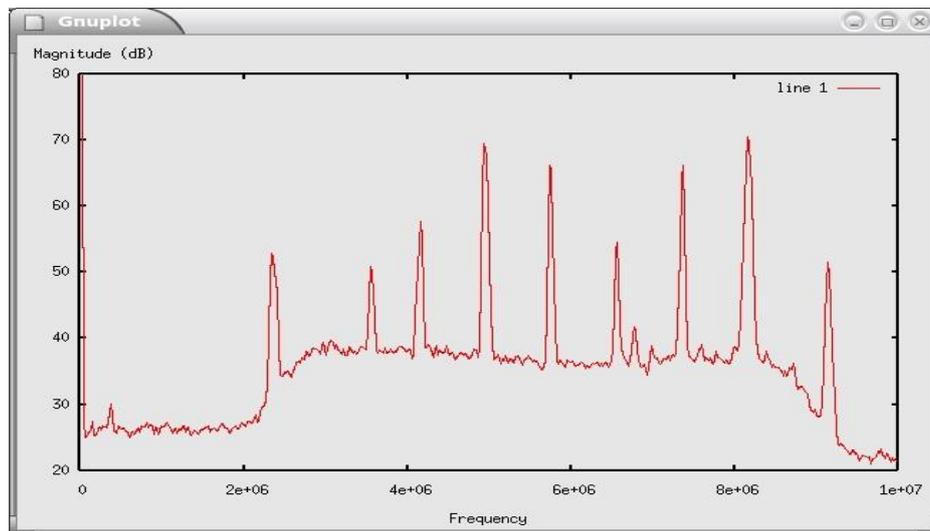


Figura 1.3.4 Transformada Rápida de Fourier de la banda de FM con nueve estaciones.

Figura cortesía de [BLOS04]

Cada uno de los picos en la Figura 1.3.4 es una estación de radio. Nuestro software” ve” todos a la vez. Para escuchar una emisora, necesitamos una manera de separarla de todas las demás, traducirlo a la banda base (DC, 0 Hz) y revertir el efecto de la modulación de frecuencia (suponemos por simplicidad nuestra señal modulada en frecuencia).

En el caso de la frecuencia modulada la señal viene dada por la siguiente fórmula:

$$f(t) = K * m(t) + k$$

$m(t)$ es la señal de entrada, k es una constante y la frecuencia f_c es la frecuencia de la portadora (por ejemplo, 100 MHz).

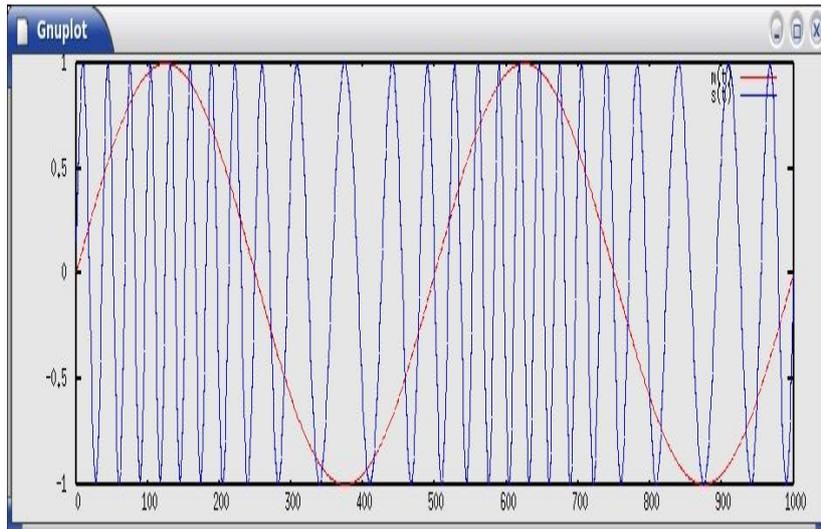


Figura 1.3.5. Señal modulada en frecuencia.
Figura cortesía de [BLOS04]

La figura 1.3.6 muestra la estrategia para escuchar una estación de FM. Eliminamos la portadora y nos quedamos con una señal en banda base cuya frecuencia instantánea es proporcional al mensaje original $m(t)$.

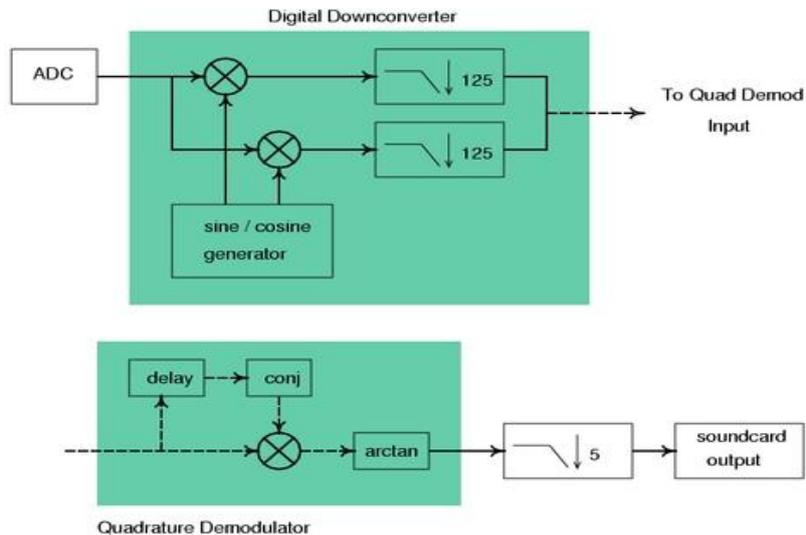


Figura 1.3.6 Diagrama de bloques del receptor FM
Figura cortesía de [BLOS04]

La primera parte es fácil, nos deshacemos de la portadora mediante el uso de nuestro software “Digital Down Converter” (DDC). Este bloque está compuesto conceptualmente de un oscilador de valores numéricos de seno y coseno oscilando a la frecuencia que queremos llevar a cero, un mezclador y un decimador y filtrado FIR.

El convertidor digital hace su trabajo mediante el aprovechamiento de una identidad trigonométrica que dice que cuando se multiplican dos sinusoides de frecuencia f_1 y f_2 , el resultado se compone de dos nuevas sinusoides, una en $f_1 + f_2$ y la otra en $f_1 - f_2$. En nuestro caso, se multiplica la señal entrante por una señal a la frecuencia de su portadora. La salida consiste en dos componentes, uno a doble de la frecuencia de la portadora y otro a cero. Nos deshacemos del componente doble con un filtro de paso bajo, que nos deja la señal de banda base.

El siguiente paso (demodulador en cuadratura) es calcular la frecuencia instantánea de la señal en banda base. Aproximamos la diferenciación de la fase, determinando el ángulo entre dos muestras adyacentes. Podemos determinar el ángulo entre dos muestras consecutivas multiplicando uno por el complejo conjugado del otro y tomando el arco tangente del producto.

La Figura 1.3.8 muestra la salida del demodulador en cuadratura: en esta figura se pueden ver todos los componentes de la forma de onda de FM.

De 0 a aproximadamente 16 KHz es la señal de audio izquierda más derecha ($L + R$). El pico a 19 KHz. es el tono piloto estéreo. La información estéreo izquierda menos derecha ($L-R$) está centrada a la frecuencia doble de la del piloto (38 kHz). Subportadoras adicionales se encuentran en la región de 57KHz-96KHz.

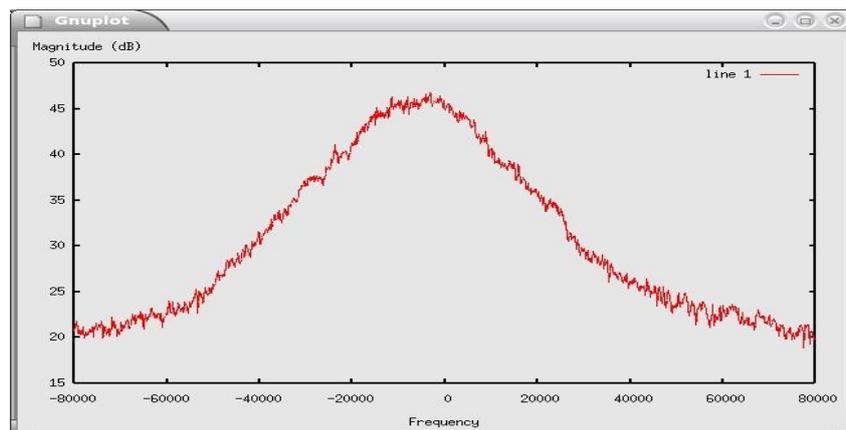


Figura 1.3.7. FFT a la salida del convertidor digital de bajada.
Figura cortesía de [BLOS04]

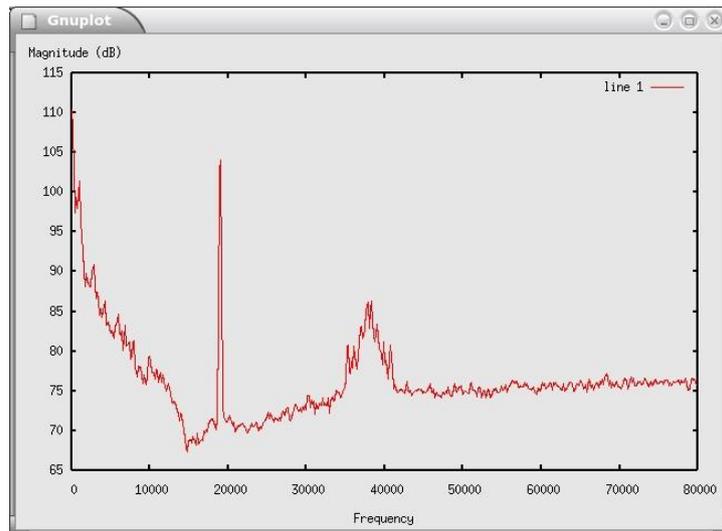


Figura 1.3.8. FFT de la señal FM demodulada.
Figura cortesía de [BLOS04]

Para simplificar la salida del demodulador en cuadratura la pasamos por un filtro pasabajo con una frecuencia de corte de 16 kHz. Esto nos da una salida monoaural que conectamos a la salida de la tarjeta de sonido.

1.4 Composición resumida del Demostrador de Conceptos.

El objetivo de esta tesis doctoral es el estudio y aplicación de los desarrollos actuales del Procesado Digital de la Señal, de forma que esto sirva de base para la posterior investigación y estudio.

Es por ello que se ha realizado de forma práctica un demostrador donde se aplica e integra tanto la parte de telefonía, como vídeo, audio y SDR. Incidiendo para su desarrollo en iniciativas que están evolucionando en este momento y exploran nuevas formas de usar la tecnología DSP / SDR. Se ha participado para ello en los grupos de investigación y se han aplicado al demostrador de conceptos.

Las aplicaciones implementadas en el demostrador de conceptos, son útiles para los objetivos pretendidos dentro de esta Tesis: análisis, aplicación y ayuda a la investigación en estas áreas, dentro del grupo de investigación (S2CN) o en el entorno Universitario en el que se desarrollan. Ninguna, en este momento, tienen una base comercial (salvo alguna excepción), y algunas de ellas, mientras se desarrollaba esta

El programa TS3 Marconi ha demostrado su valía tanto en docencia con el Curso de Navegación Oceánica para personal del Servicio Marítimo de la Guardia Civil, así como en el simulacro de Tsunami celebrado en la provincia de Cádiz el 1 de diciembre de 2012 por la Red Radio de Emergencia gaditana [REME13], adscrita a la Subdelegación del Gobierno Central en Cádiz y que fue seguido en directo por los colaboradores de la Dirección General de Protección Civil y Emergencias de 23 provincias españolas tanto por Onda Corta como por el TS3 Marconi. También ha sido utilizado en el ejercicio mundial de comunicaciones de emergencia GlobalSet 2013 celebrado en Mayo de 2013 y en el ejercicio anual de la Red Radio de Emergencia de la Subdelegación Provincial del Gobierno en Cádiz denominado “Apagón 2013”, celebrado el 14 de diciembre del mismo año.

Además de los canales del Servicio Móvil Marítimo se han simulado canales virtuales de comunicaciones para Instructores de Náutica, Inglés, Servicio Marítimo de la Guardia Civil, canales de Pruebas, tres canales de receptores de radio reales en VHF marítimo, 2182,0 kHz y Canal 19 de la Banda Ciudadana y cinco canales para la Red Radio de Emergencia que son los más utilizados, tres para la REMER gaditana, uno para la REMER Nacional y un último como canal de Socorro o Silencio, todos ellos con acceso por contraseña.

Con la utilización del TS3 Marconi se permiten las prácticas de radiocomunicaciones a nivel mundial, en cualquier idioma y compartiendo recursos informáticos de forma gratuita sin tener que conocer el funcionamiento de los equipos, con lo que se ahorra el tiempo de enseñanza del manejo del simulador y se va directamente a practicar los Procedimientos Radioeléctricos de rutina, llamadas comerciales, socorro, urgencia o seguridad, siendo posible la transmisión de datos en forma de audio de banda base y la decodificación simultánea y remota mediante los programas de ordenador apropiados, ya sea para Llamada Selectiva Digital, Morse, Radioteletipo ARQ o FEC, facsímil meteorológico o cualquier otro de uso común en la Marina Mercante, por ejemplo el Navtex.

Una vez puesto en marcha el TS3 Marconi se procedió a desarrollar una centralita telefónica digital basada en un ordenador compatible PC que fuera capaz de integrar las comunicaciones analógicas convencionales telefónicas conmutadas con las nuevas redes

IP y que incluso permitiera la transmisión de imágenes desde cámaras IP dotadas de micrófono tanto por cable como por wifi y que a su vez, éstas fueran clasificadas como terminales accesibles desde un teléfono IP dotado de pantalla o bien desde la pantalla de un ordenador convencional que se encontrase integrado con la subred Marconi o incluso una Tableta o un smartphone por Wifi.

Creemos que no es necesario explicar que las comunicaciones interiores de los buques de Guerra, Mercantes, Pesca y Recreo se dirigen a la integración total de redes IP para todos los usos, ya sea Sistema de Combate, Control total del buque, Sistema Integrado de Navegación o Comunicaciones de Voz, Datos e Imagen entre lugares remotos del buque. Por lo tanto, la programación sabiamente estructurada de una centralita telefónica digital es imprescindible y es necesario disponer de un demostrador de conceptos que ayude a comprender todas las tareas y parámetros que son necesarios para alcanzar la óptima programación de dicha red.

Ya desde el punto de vista de las asignaturas de Radiotecnia I y Radiotecnia II del Grado en Ingeniería Radioelectrónica es imprescindible que se afiance el conocimiento sobre los Modos de Emisión que define el Reglamento de Radiocomunicaciones de la Unión Internacional de Telecomunicaciones (fig. 1.4.2) y asocie a estos Modos de Emisión los conceptos de Anchura de Banda Necesaria, Anchura de Banda Ocupada, Densidad Espectral, Portadora y Bandas Laterales, Bandas de Frecuencias, Canales, Servicios, Velocidad de Transmisión, alcance o cobertura, etc.

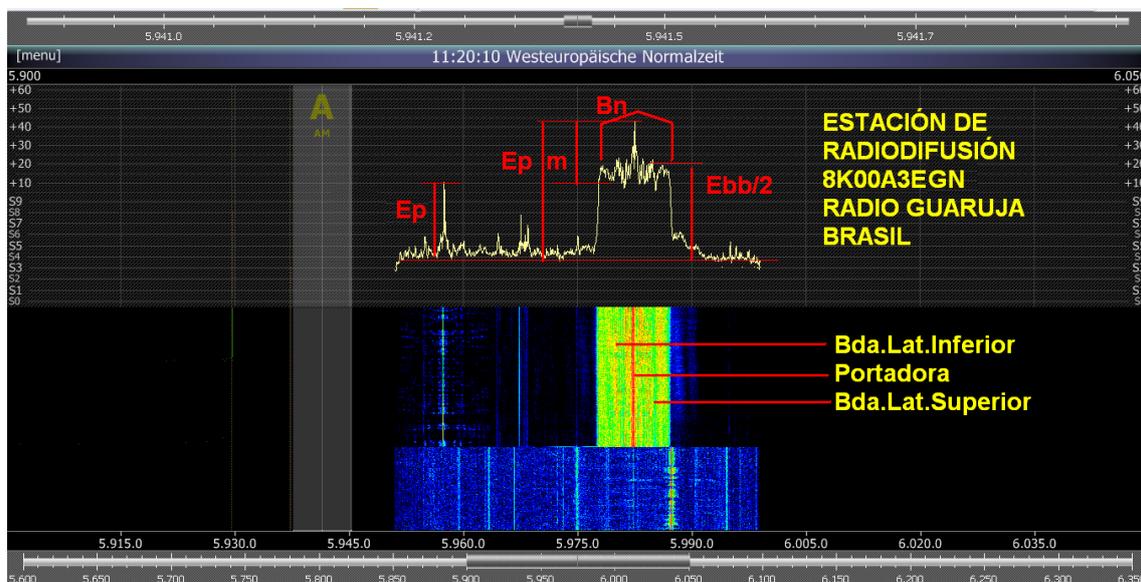


Figura 1.4.2 Espectros de una señal A1A a la izquierda y una señal A3E a la derecha.

A estos conceptos llegaban los Oficiales Radioelectrónicos a base de FE y práctica profesional, prueba y error y sobre todo el estricto cumplimiento del Reglamento de Radiocomunicaciones, pero no es hasta la primera década del Siglo XXI que aparecen los equipos de radio definidos por software (SDR), en los cuales se minimizan los componentes electrónicos y se maximizan la velocidad de muestreo de señales y procesamiento digital de datos, permitiendo el filtrado o sintonización matemática, la transmisión de dichos datos por la red Internet y la decodificación de dichos datos en modo local pero a cientos o miles de kilómetros de donde se recibieron.

Se está logrando acceder remotamente a uno o varios receptores de radio sintonizados en distintas bandas de radio en distintos lugares de la Tierra y que puedan analizar en el tiempo y en frecuencia las distintas comunicaciones que se están captando por estaciones voluntarias (normalmente universitarias) integradas en el Proyecto Websdr.org liderado por el Dr. Pieter de Boer de la Universidad de Twente, Holanda. [BOER13]

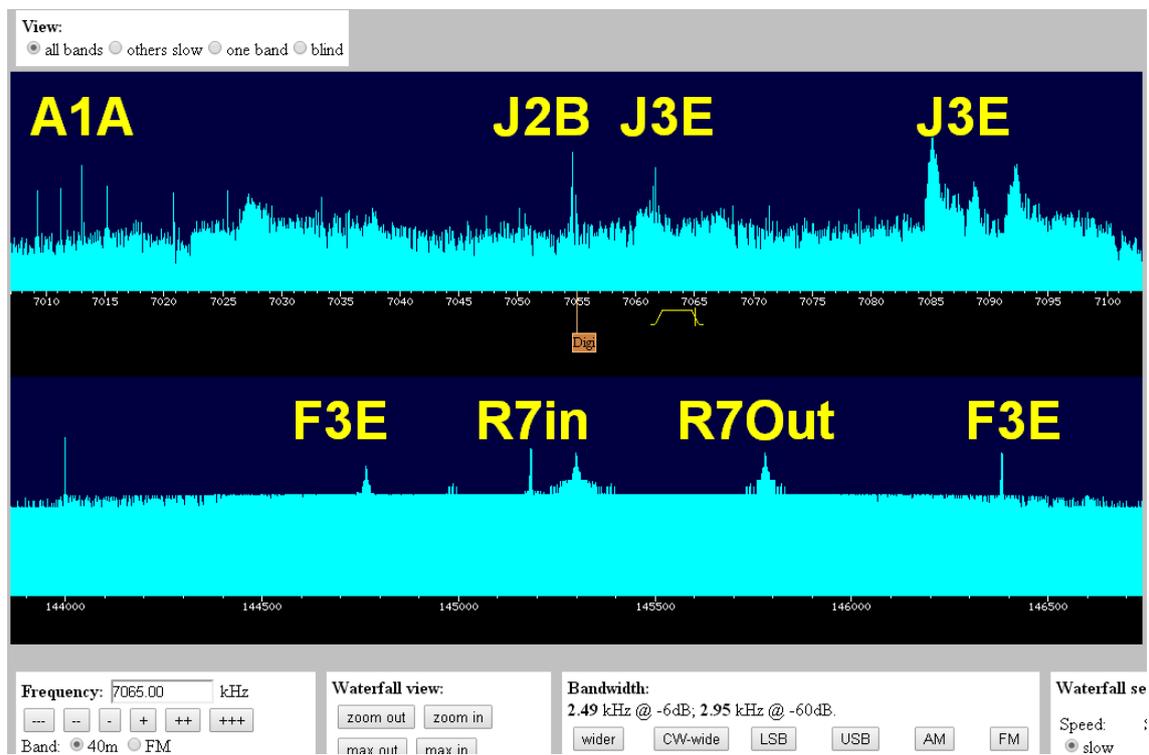


Figura 1.4.3. Espectrograma simultáneo de los dos receptores del Websdr de UCA.

El Grupo S2CN de la UCA ha aportado dos receptores de radio, uno en la banda de aficionado de 7 MHz y otro en la de 144-146 MHz para uso y disfrute tanto de la

comunidad Universitaria como de la de los radioescuchas, siendo múltiples sus aplicaciones (figs 1.4.3, 1.4.4 y 1.4.5). El servidor, receptores y antenas de la UCA es la estación que más al Sur se encuentra de toda Europa, captándose señales portuguesas, francesas e italianas en horas de buena propagación a parte de las españolas y de vez en cuando alguna estadounidense, siendo visitada su página web (marconi.uca.es:8901) por radioaficionados de todo el Mundo, tanto por curiosidad como para comprobar sus propias emisiones.

Los receptores basados en Web pueden ser utilizados por más de 100 usuarios simultáneamente, estando limitado su número a la velocidad de transmisión del servidor del receptor utilizado.

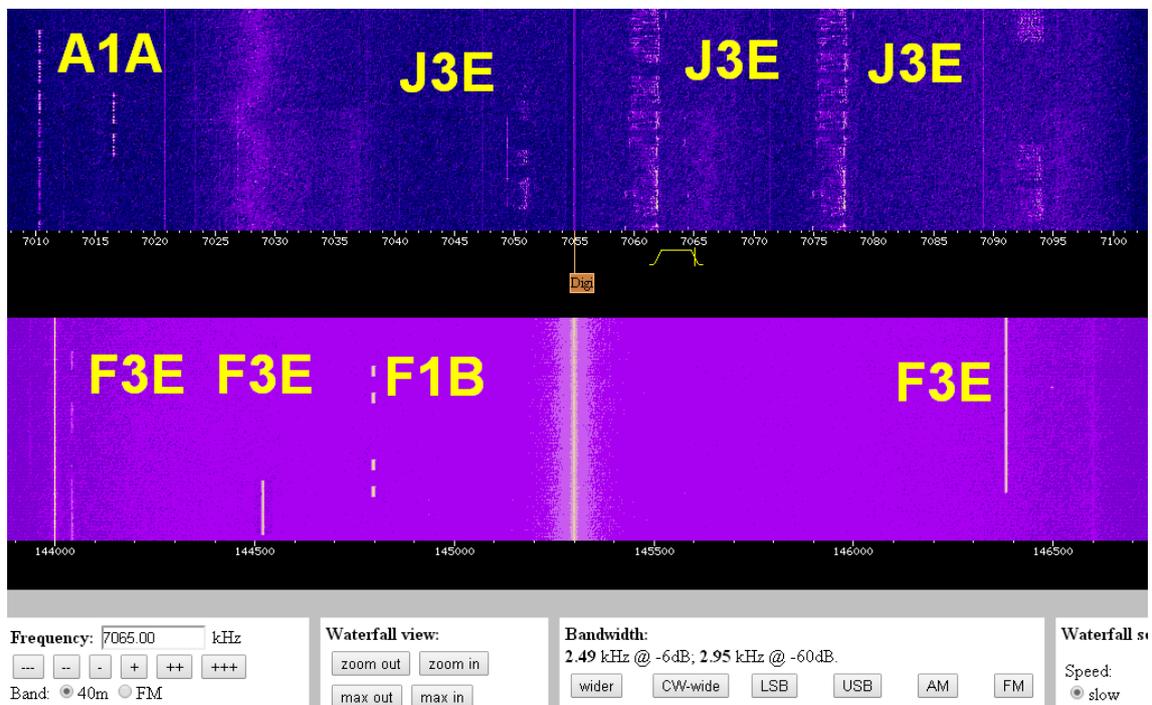


Figura 1.4.4 Diagrama en cascada simultáneo de los dos receptores del Websdr de UCA.

A parte de los receptores Websdr se deseaba poder telecontrolar desde cualquier parte del Mundo y vía internet, un receptor de radio de LF-HF, lo cual se realizó mediante un receptor SDR Perseus, con el único inconveniente de que si bien es posible la escucha multiusuario no es posible su control de esta manera, por lo que se cumple la norma de que el que lo encuentra “libre” lo controla y los que llegan más tarde sólo escuchan. Este sistema también tiene su gran aplicación, ya que permite compartir recursos y

además inyectar la señal recibida en el servidor TS3 Marconi en un canal abierto al efecto.

Para terminar con la parte técnica, se han compilado, adaptado y configurado programas de control y decodificación de señales de equipos SDR en configuración Cliente-Servidor de manera que un único ordenador PC ha administrado cuatro receptores SDR a los cuales se les ha integrado con cuatro usuarios remotos distintos de manera simultánea.

Desde el punto de vista metodológico, para la realización de esta Tesis, se ha querido obtener la realimentación de la opinión y crítica constructiva de los usuarios, lo cual se ha podido realizar tanto por escrito como de palabra a través del servidor Websdr de la UCA como a través del TS3 Marconi, el correo electrónico directo o el foro del Grupo S2CN incluido en la página web del Grupo: marconi.uca.es.

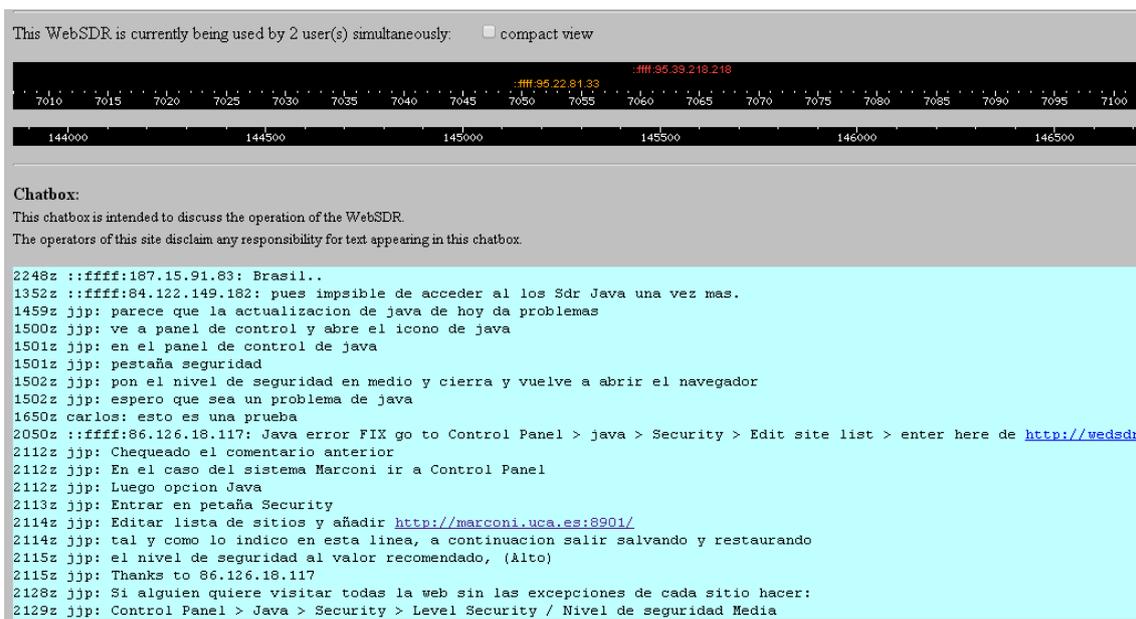


Figura 1.4.5 Pantalla de comentarios escritos del Websdr de la UCA.

Queremos resaltar que ninguna de las soluciones que se han utilizado durante la realización de esta Tesis Doctoral es una “solución comercial” o solución *ad hoc* ya que los programas de control de los receptores de radiocomunicaciones SDR no permitían la configuración cliente-servidor y ha debido buscarse, seleccionarse e incluso adaptarse las soluciones informáticas, muchas veces versiones Beta, de los grupos de desarrollo

en nuestros equipos y para ello hemos tenido que integrarnos con ellos e interrelacionarnos para obtener nuestra “propia solución” al problema planteado y para ello hemos tenido que alcanzar el nivel adecuado de conocimientos con los que pudiéramos “hablar de tú a tú” con los desarrolladores de cada uno de las aplicaciones de software remoto.

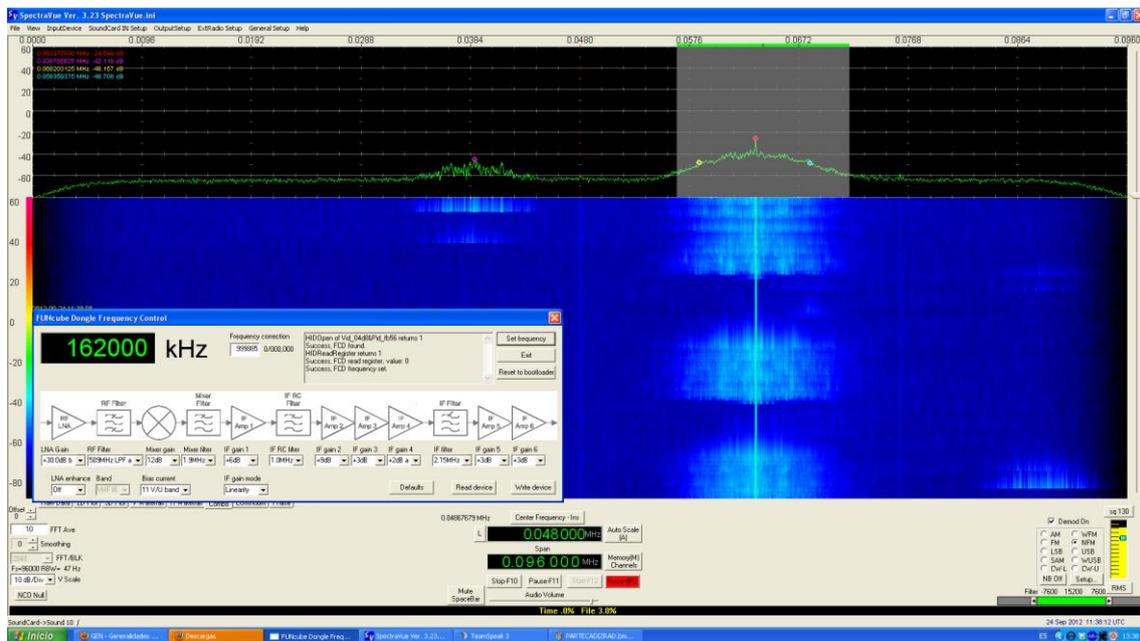


Figura 1.4.6 Espectrograma de la señal de la Estación Costera Cádiz Radio con definición de los parámetros del receptor Funcube.

1.5 Aplicación del Demostrador de Conceptos Marconi.

A continuación se presenta una Tabla con los conceptos que se pueden demostrar con él según la asignatura del Grado en Ingeniería Radioelectrónica, si bien debe ser el profesor de cada asignatura el que prepare la práctica más adecuada para el Concepto que quiera demostrar.

Para ello se ha tenido que recurrir a los descriptores de las asignaturas relacionadas en la Memoria del Grado en Ingeniería Radioelectrónica de la Universidad de Cádiz y no se ha pormenorizado el concepto a demostrar porque no se ha encontrado una referencia escrita, dejándose el amplio descriptor.

Asignatura	Concepto Demostrado	Equipo a utilizar y software.
Física II. Electromagnetismo y Ondas.	Frecuencia y sintonía. Polarización de Antenas Antenas	Antenas y receptores del Demostrador.
Informática	Elementos de un ordenador y sus funciones. Redes de ordenadores. Sistemas Operativos. Diseño y consulta de base datos. Programación.	Ordenadores del Demostrador y Red de Área Local Utilizada. Equipos SDR.
Expresión Gráfica	Análisis e interpretación de planos y esquemas	Diagramas de bloque del Demostrador y Manuales de equipos.
Inglés Marítimo	Maritime safety. Ship's handling. Marine engines. Standard Marine Communication Phrases.	Simulador TS3-Marconi, transceptor VHF, receptor Perseus.
Dispositivos Electrónicos e Instrumentación	Representación de Señales en los dominios del Tiempo y la Frecuencia. Respuesta en frecuencia de los dispositivos electrónicos.	Receptores SDR y Programas de control de los receptores SDR.
Circuitos Analógicos.	Filtros. Amplificadores. Osciladores. Fuentes de alimentación. Amplificadores operacionales. Convertidores Analógico-digitales	Programas de control de los receptores SDR. GNU radio.
Radiotecnía I	Generación de los campos electromagnéticos. Tipos de Radiopropagación en función de la banda de frecuencias. Determinación del enlace idóneo. Antenas embarcadas. Tipos,	Antenas del Demostrador. Cables coaxiales y conectores del Demostrador. Receptores SDR, Manuales y Programas de Control de: Funcube, RTL-SDR, Perseus,

	<p>aplicación, parámetros y diseño. Modulaciones Analógicas y Digitales aplicadas al buque. Comprensión técnica y aplicación de la normativa de la UIT. Cálculo del ancho de banda ocupado. Utilización idónea del canal. Principios básicos de los circuitos electrónicos aplicados a los equipos del SMSSM. Lectura de diagramas de instalación y esquemas radiotécnicos. Fundamentos de Compatibilidad Electromagnética.</p>	<p>SDR IQ y Softrock. Página Web: marconi.uca.es Programas de Análisis de Banda Base: Audacity, Spectravue, Spectran y Spectralab. GNU radio.</p>
Radiotecnía II	<p>Equipos Transmisores de Radio del SMSSM. Equipos Receptores de Radio del SMSSM. Equipos Transceptores de Radio del SMSSM Radioteletipo. Llamada Selectiva Digital. Comunicaciones Vía Satélite. Transceptores del Sistema Automático de Identificación de buques. Transceptores de VHF-AM aéreo del SMSSM. Repetidores.</p>	<p>Antenas del Demostrador. Cables coaxiales y conectores del Demostrador. Receptores SDR, Manuales y Programas de Control de los receptores: Funcube, RTL-SDR, Perseus, SDR IQ y Softrock y de los Transceptores Flex 1500 y 3000. Programas de Decodificación de Comunicaciones: JVComm32, HFFAX, WXSAT, Sigmira, Hamradio Deluxe, HDSDR, FLDigi, FLARQ, QTRadio, SDR-Sharp, Power SDR, ShipPlotter, FCHID, Programas de Seguimiento de Satélites: Orbitron. Programas de Análisis de Banda Base: Audacity, Spectravue, Spectran y Spectralab. GNU radio.</p>
Procedimientos Radioelectrónicos	<p>Reglamentación de las radiocomunicaciones Apéndices al reglamento de radiocomunicaciones. Servicios especiales de radiocomunicaciones. Procedimientos general y de socorro.</p>	<p>Demostrador SDR Marconi controlado remotamente. Aplicación TS3 Marconi.</p>
Sistemas de Radionavegación y Prácticas de	<p>Señales Radiogoniométricas. Señales del Sistema VOR. Señales del Sistema AIS.</p>	<p>Los dos últimos con los receptores RTL-SDR dotados de convertidores de</p>

Radionavegación	Señales GPS. Señales RADAR	frecuencia de bajo ruido (LNB).
Comunicaciones Interiores y Prácticas de Comunicaciones Interiores	Grabación y reproducción del sonido. Grabación y reproducción de video. Sistemas de Telefonía conmutada. Sistemas de Numeración. Sistemas de voz sobre protocolo Internet. Sistemas de video sobre Protocolo Internet. Centralitas Telefónicas Digitales. Teléfonos y Cámaras IP. Hardphone y Softphone.	Programas de Análisis de Banda Base: Audacity, Spectravue, Spectran y Spectralab. Aplicación TS3-Marconi. Aplicación Asterisk. Centralita Trixbox Switches, enrutadores, red WIFI, red de área local, Videoteléfonos IP y Cámaras IP.
Equipos del SMSSM y Prácticas de Radiocomunicaciones	Operación de equipos y subsistemas de comunicaciones en el SMSSM: Procedimientos operativos en comunicaciones generales y de emergencia. Principios básicos del Servicio Móvil Marítimo y Servicio Móvil Marítimo por Satélite. Principios generales de los sistemas de comunicaciones NBDP y TOR. Operaciones de búsqueda y salvamento (SAR)	Aplicación TS3-Marconi. Demostrador SDR Marconi. Página marconi.uca.es Programas de Decodificación de Comunicaciones: JVComm32, HFFAX, WXSAT, Sigmira, Hamradio Deluxe, HDSDR, FLDigi, FLARQ, QTRadio, SDR-Sharp, Power SDR, ShipPlotter, FCHID, Programas de Seguimiento de Satélites: Orbitron.
Inglés Radioelectrónico	Global Maritime Distress and Safety System. Standard Marine Communication Phrases.	Aplicación TS3-Marconi.
Telemática Marítima	Redes de área Local. . Topologías de Redes de Área Local Embarcada. Topología Física y Topología Lógica. Redundancia. Instalación de Cableado de Redes LAN. Normativa Marítima Nacional e Internacional. Niveles ISO de la OSI. Aplicaciones Navales. Elementos de una Red de Área Local embarcada: Hubs. Routers. Switches. Bridges. Mass Memory. Convertidores de protocolos. Repetidores.	Demostrador de Conceptos completo, incluyendo los receptores y transceptores SDR, las tarjetas de sonido internas y externas, los sistemas operativos y recursos de programación, los ordenadores, centralitas telefónicas IP, videoteléfonos y cámaras IP por cable y WIFI, red WIFI, tabletas, ordenadores portátiles y smartphones del alumnado. Programas: Aplicación TS3-Marconi,

	<p>Configuración básica de Routers.</p> <p>Protocolos digitales para comunicaciones: Punto a Punto. DCE – DTE. DTE – DTE.</p> <p>Protocolos TCP-IP, NMEA 0180, NMEA 0182, NMEA 0183 y NMEA 2000.</p> <p>Normalización Modem nulo, RS232, RS422, USB, modems telefónicos.</p> <p>Redes Inalámbricas de Área Local. IEEE 802.11x.</p> <p>Análisis de las señales Digitales de una LAN: En frecuencia y en el tiempo.</p> <p>Localización y reparación de averías.</p> <p>Configuración de Redes LAN embarcadas.</p> <p>Aplicaciones Navales Telemáticas.</p> <p>Sistema Universal de Identificación de Datos (UAIS).</p> <p>Sistemas Integrados de Navegación (Voyager, Cannet, Navnet, y otros).</p> <p>Sistemas de Radiocomunicaciones por Vo-IP.</p> <p>Sistemas de seguimiento del tráfico Marítimo (Vessel Traffic Systems).</p>	<p>ShipPloter, PlanePlotter, Web RTC, VNC, Streaming de Voz y Video. Ice e ICecast.</p>
Programación	<p>Sistemas Operativos: Linux (Debian, Ubuntu, Linaro, Centos), Windows.</p> <p>Lenguajes C++, Java, Android, HTML, CSS, Eclipse, QT.</p>	<p>Ordenadores de control de los Equipos SDR. GNU radio.</p>
Proyecto de Fin de Grado	<p>Manejo de manuales y normativa.</p> <p>Operación de los sistemas de radiocomunicaciones y de radionavegación.</p> <p>Interpretación de circuitos, sistemas y esquemas de los equipos propios del Grado.</p>	<p>Simulación de Formas de Onda Modulada mediante Matlab, QT Radio o GNU radio.</p> <p>Diseño y/o Programación de Equipos SDR a partir de los conocimientos adquiridos.</p> <p>Telecontrol de equipos</p>

Tabla 1.1 Tabla con los conceptos que se pueden demostrar con demostrador Marconi según la asignatura del Grado en Ingeniería Radioelectrónica.

En el momento de redacción de esta Tesis Doctoral se debate en la Escuela de Ingenierías Marina, Náutica y Radioelectrónica de la Universidad de Cádiz la estructura, las materias del Máster en Radioelectrónica (nombre aún no definido) y sus contenidos.

Creemos que no es necesario explicar que el Demostrador de Conceptos será de gran aplicación a todas las asignaturas relacionadas con la Teoría de la Señal y la Teoría de la Comunicación, ya que se pueden alterar las funciones de transferencia de cada bloque electrónico virtual que forma parte del software que emula el equipo real, y por lo tanto, variar su respuesta de forma permanente o temporal.

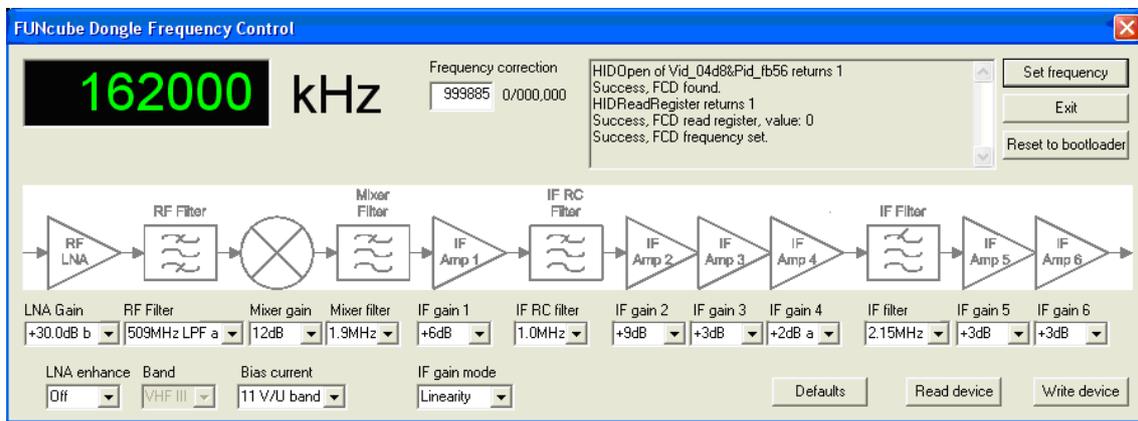


Figura 1.5.1 Diagrama de bloques del programa que controla al receptor Funcube.

2. Estado de la cuestión.

2.1 Los equipos de Radio Definidos por Software.

2.1.1 Introducción.

El término “Software Radio” fue acuñado por Joe Mitola en 1.991 para referirse a un tipo de radios reprogramables o reconfigurables [HARA et al. 02].

En otras palabras, una misma pieza de hardware puede realizar diferentes funciones en distintos instantes de tiempo con la introducción de cambios en su configuración mediante software.

Se define SDR1 como un receptor de radio sustancialmente definido en software y cuyo comportamiento en capa física puede ser significativamente alterado a través de cambios en dicho software, lo que claramente constituye una ventaja significativa y un motivo de peso para que se piense en la realización de esfuerzos a nivel de investigación y desarrollo orientados a hacer de SDR una tecnología comercialmente viable para soportar otros adelantos como la interoperabilidad, la coexistencia, el MIH y la convergencia desde capa física.

Particularmente MIH (normalizado por el Grupo de Trabajo 802.21 del IEEE3) requiere que en capa física haya una tecnología capaz de soportar los cambios dinámicos impuestos por las capas superiores, y la que mejor satisface estos requisitos es SDR, pero aunque tiene ya más de 10 años, las implementaciones comerciales de esta tecnología apenas empiezan a ser consideradas debido a los costos de los componentes y a la capacidad de éstos.

A pesar de que internacionalmente se está trabajando de manera intensiva en SDR y en su evolución, lastimosamente es una tecnología que ha sido abordada por muy pocos investigadores civiles a nivel nacional, lo que claramente debe ser remediado considerando las implicaciones y el impacto que a nivel técnico y comercial tendrá la implementación de este tipo de tecnologías en dispositivos que se ofrecen al usuario final.

Es necesario entonces iniciar un proceso que permita conocer la tecnología SDR y mostrarla a la comunidad académica/científica nacional con el objetivo de motivar al planteamiento de nuevas aproximaciones y lograr un aporte significativo a la evolución de sistemas y redes inalámbricas en el país.

En este orden de ideas, cabe preguntarse: ¿Qué tecnología de capa física es la principal candidata para soportar interoperabilidad y coexistencia inalámbrica?, ¿Qué es SDR?, ¿Qué limitaciones y ventajas tiene su implementación comercial?, ¿Qué orientaciones han sido propuestas por los grupos de trabajo?, ¿Qué soluciones se encuentran disponibles, cómo funcionan, y cómo podrían mejorarse?, y finalmente ¿Cómo motivar a la comunidad académica/científica nacional para la investigación en el tema?

Considerando entonces lo anterior, se puede desarrollar un proyecto de investigación en radiocomunicaciones marinas mediante el cual se haga una primera aproximación al concepto SDR realizando una descripción de sus aspectos técnicos funcionales como tecnología de capa física y desarrollando –a manera de demostración– una implementación software/hardware.

Esta demo y la base de conocimiento que dejará el proyecto permitirán responder en gran medida los interrogantes que se han planteado y posibilitarán la realización de propuestas para futuros trabajos que profundicen en la temática y que se espera que se conviertan en la Red de Receptores para Emergencia Marítima en un futuro cercano.

2.1.2 Antecedentes.

Durante el proceso de evolución de las telecomunicaciones inalámbricas se han generado gran cantidad de tecnologías que tienen como objetivo facilitar y optimizar el intercambio de información entre entidades geográficamente dispersas, pero las mismas diferencias técnicas crean incompatibilidades entre los estándares desarrollados y los equipos relacionados, complicando aspectos que van desde la prestación de servicios como el roaming, hasta la actualización y reemplazo de tecnologías a bajo costo.

El concepto SDR surge para solucionar parcialmente estos inconvenientes de compatibilidad e interoperabilidad. La primera implementación conocida del concepto SDR fue en el proyecto militar estadounidense SpeakEasy [LACK et al. 95], cuyo

objetivo principal era implementar más de 10 tipos de tecnologías de comunicaciones inalámbricas – que en el momento eran las más usadas por las Fuerzas Armadas de Estados Unidos – en un equipo programable, el cual operaría en una banda de frecuencias desde los 2MHz hasta los 200MHz. Un objetivo adicional del proyecto era que el prototipo debía tener la posibilidad de actualizar su código para que así se pudieran tener en cuenta los estándares futuros.

Dicho proyecto empezó en 1991 y solo en 1995 fue posible lograr todos los objetivos planteados. Sin embargo hubo cierto descontento debido a que sólo se podía mantener una comunicación a la vez, por lo cual se modificaron sus alcances llevándolo así a una segunda fase en la cual se trabajarían aspectos como disminución de peso y costo, incremento en su capacidad de procesamiento, simultaneidad de comunicaciones, diseño basado en arquitecturas de software libre, entre otros. La nueva fase del proyecto sólo necesitó 15 meses para lograr sus objetivos, obteniendo así importantes resultados que llevaron a la producción del dispositivo diseñado, el cual trabajó en el rango de 4 MHz a 400 MHz.

SpeakEasy es el primer proyecto que se conoce que haya trabajado con FPGA4 para procesamiento de datos digitales radiados, pero en la actualidad SDR es una tecnología que ha tomando gran auge en el mundo y sobre la cual se pueden encontrar diferentes investigaciones y desarrollos, como por ejemplo implementaciones de SDR usando DSP5 descritos en la tesis “Demonstration of the Software Radio Concept” de la Universidad de Twente [SCHI00], la implementación realizada por Gerald Youngblood de SDR para radioaficionados usando la tarjeta de sonido de un computador personal [YOUN03], kits de desarrollo de SDR como el SoftRock-40 que trabaja con el software libre PowerSDR de la empresa FlexRadio y proyectos como el del VirginiaTech [TRAN et al. 98], entre muchos otros.

En México y Brasil también se conocen algunos desarrollos realizados y primeras aproximaciones

2.1.3 El foro SDR.

El SDR Forum es un foro para la investigación y desarrollo mundial en SDR compuesto por más de 100 empresas, instituciones y organizaciones (Altera, Xilinx, NASA, VirginiaTech, Toshiba, Samsung, Lockheed, Motorola, QUALCOMM, Hitachi, Ohio Aerospace Institute, entre otras), cuyo objetivo es promocionar el uso de SDR desarrollando estándares y especificaciones del mismo, logrando así la divulgación de dicha tecnología para soportar necesidades tanto militares y civiles como comerciales.

Para lograr dicho propósito, el SDR Forum prepara una serie de eventos periódicos en los que se presentan productos, avances y estándares que proponen las empresas inscritas en él con respecto al tema mencionado. Adicionalmente, el SDR Forum tiene un sitio Web en el cual publica noticias y muchos de los aportes que han hecho las empresas asociadas a éste, además de gran cantidad de documentos relacionados que pueden ser descargados por cualquier usuario y otros sólo están disponibles para miembros de la organización.

2.1.4 Análisis de los equipos a utilizar en el demostrador de conceptos.

2.1.4.1 Introducción histórica.

Tradicionalmente los equipos receptores y transceptores de radiocomunicaciones son equipos constituidos por multitud de componentes electrónicos, los cuales forman circuitos sintonizadores, etapas de frecuencia intermedia, detectores, amplificadores de baja frecuencia, etc., es decir, están constituidos por "hardware". Posteriormente, en los años 1980's y 1990's se introdujeron microprocesadores en estos equipos para el control de funciones internas (controles desde teclados y pulsadores) y para añadir nuevas prestaciones (relojes, pantallas informativas, programadores, etc.), y también se introdujo la posibilidad de controlar los equipos de radio desde un ordenador, añadiendo al equipo de radio puertos de comunicación o interfaces para la conexión al ordenador.

En estos casos, y usando el software adecuado, es posible controlar desde el ordenador numerosas funciones del equipo de radio, igual o mejor que desde los controles del propio equipo. También en la década de los 1990's comenzó la introducción en los

modernos equipos de radio de los chips DSP o "Procesadores Digitales de Señal", los cuales permiten mediante técnicas digitales realizar filtros de paso de banda y de supresión de ruidos, entre otras posibilidades, muy eficaces, mejor que los realizados tradicionalmente con circuitos analógicos.

Un chip DSP es básicamente una especie de CPU en miniatura, un chip microprocesador con un juego de instrucciones pequeño, pero capaz de ejecutarlas a una velocidad muy superior a la velocidad de una CPU corriente, gracias a una estructura simplificada y al reducido juego de instrucciones.

En cualquier caso, siempre se trata de equipos de radio realizados enteramente con componentes electrónicos, o sea, en términos informáticos se definirían como "radios hardware". Pero desde principios de la década del 2000 radioaficionados como Gerald Youngblood (AC5OG), comenzaron a investigar y desarrollar un nuevo concepto de equipos de radiocomunicaciones, los equipos de radio desarrollados por programa o "radios software", en siglas SDR (Software Defined Radio), en los que la parte hardware (circuitería) es mínima, y la mayor parte de las funciones que definen un equipo de radio se definen por software (programas) en un ordenador PC o de otro tipo, dotado de tarjeta de sonido (requisito necesario).

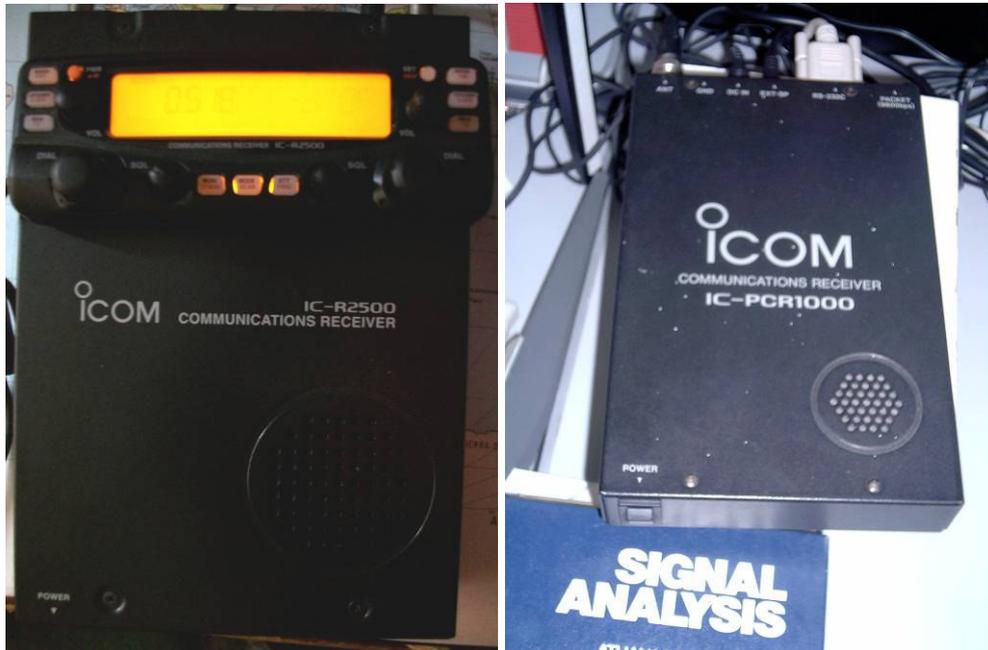
La SDR supone realizar la mayor parte de las funciones de un equipo de radio, incluso las más importantes, mediante el software implementado en un ordenador. Por tanto, y antes de profundizar más en lo que es una SDR, se puede explicar de qué no se trata.

2.1.4.2 Diferencia entre equipos de radio definidos por software y equipos de radio controlados por software.

En primer lugar hay que decir que hay una enorme diferencia entre una radio definida por programas (SDR) y otra controlada por programas.

Casi todos los equipos de radio modernos dotados de interfaces informáticos son equipos que tienen la opción de ser controlados por ordenador, permitiendo que desde este último se gobiernen y visualicen parte o todas las funciones y parámetros que normalmente están presentes en el frontal del equipo: frecuencia, elección de modo de operación (AM, FM, CW, SSB...), control automático de ganancia (CAG), etcétera.

Incluso hay equipos de radiocomunicaciones que ni siquiera tienen un panel frontal de mandos e indicadores, siendo controlados totalmente desde el ordenador, el cual realiza todas las funciones del panel de mando del equipo, mostrando además éste en la pantalla del ordenador.



Figs. 2.1.1 y 2.1.2 Receptores controlados por ordenador de la firma ICOM del Grupo S2CN.

Tampoco las SDR se refieren al uso de las sofisticadas técnicas de procesado digital de señal, mediante la introducción de chips DSP, implementadas a nivel de las etapas de audio, para mejorar la inteligibilidad y calidad de las señales. Las técnicas DSP permiten realizar filtros de audio mediante técnicas digitales que son mucho más efectivas que las tradicionalmente realizadas con componentes electrónicos analógicos, e incluso se pueden programar sus características por software. Y aunque las técnicas DSP se usan principalmente a nivel de las etapas de audio de los equipos de radiocomunicaciones, procesando digitalmente las señales de audio, también hay equipos de radio que introducen el procesado digital de señales DSP a nivel de frecuencia intermedia, aunque en estos casos siempre se hace sobre una última frecuencia intermedia de valor muy bajo, sobre los 40 kHz, lo que permite manejarla casi como una señal de audio, y por tanto al alcance de los chips DSP más comunes.

Las técnicas DSP comenzaron a ser introducidas en la década de los 90's para funciones eficaces de filtrado y de reducción de ruido en las etapas de audio, y actualmente se usan también para las etapas de FI de los receptores de radio más modernos, con mejores prestaciones que las clásicas etapas de FI de filtros resonantes sintonizados a la FI. Pero en cualquier caso, tanto unos como otros no dejan de ser al fin y al cabo equipos de radio convencionales, a pesar de los "añadidos avanzados" que estas técnicas puedan introducir.

Una radio software (SDR), en cambio, tiene casi todos sus "componentes" definidos y funcionando en forma de programas en un ordenador, a excepción de un mínimo de componentes físicos necesarios, externos al ordenador, que no pueden ser definidos por software en el ordenador. Y mientras no sea activado ese software o conjunto de programas, el equipo de radio no será tal, sino que será un simple conjunto de unas cuantas placas electrónicas externas, incapaces de hacer nada práctico.

Es el software SDR que se haga funcionar en el ordenador el que define el esquema de modulación a emplear (AM, FM, SSB...), el tipo de silenciador (squelch), cómo actúa el CAG, y, en fin, todo el equipo de radio. Y aunque las radios SDR funcionan con el software SDR que se hace funcionar en el ordenador, la mayor parte del trabajo no lo hace en sí el ordenador, sino el dispositivo SDR empleado en éste (que, como veremos, habitualmente es la tarjeta de sonido).

Además una radio SDR es muy flexible, ya que modificando o reemplazando sus programas de software, o añadiendo nuevos programas, se consiguen modificar sus funcionalidades: como es añadir nuevos modos o mejorar sus prestaciones. Ello permite también acomodar el SDR a las necesidades de cada tipo de usuario (radioaficionados, servicios de emergencias, etc.).

La SDR es algo novedoso y representa un concepto que puede no ser fácil de asimilar. Para entenderlo se explicará a continuación, aunque describiendo el proceso de recepción, ya que el de transmisión es muy similar aunque a la inversa.

2.2 Descripción tecnológica de los Equipos de Radio definidos por Software.

2.2.1 Antecedentes, el DSP TMS320C30.

2.2.1.1 Introducción.

Antes de empezar a hablar de SDR hemos de hacer una introducción a la electrónica clásica que posibilitó el Procesado Digital de la Señal en la década de los 90.

En la figura 2.2.1 se puede apreciar la fotografía de una tarjeta de aplicación programable Texas Instrument TMS320C30 (Procesador digital de coma flotante), la cual se programa en Código Máquina.

Los Procesadores Digitales de Señal (DSP) están diseñados para realizar las operaciones de multiplicación y suma requeridas por el Procesado Digital de Señal en muy pocos ciclos de reloj.



Figura 2.2.1 Tarjeta de aplicación programable Texas Instrument TMS320C30

Los DSP de coma flotante se suelen utilizar en los campos de aplicación del procesado digital con mayor carga computacional. Algunas de las aplicaciones más usuales de este tipo de procesadores son: procesado de vídeo e imagen, gráficos 3D, RADAR, SONAR, instrumentación de precisión, procesado vectorial, comunicaciones de alta velocidad, aplicaciones multiproceso, etc.

Los TMS320C3x constituyen la tercera generación de DSP de Texas Instruments. Son procesadores de coma flotante, con 32 bits de bus de datos y capaces de alcanzar hasta 40 MFLOPS y 20 MIPS.

Como características más relevantes, podemos destacar una ALU y multiplicador de coma flotante/coma fija, dos generadores de direcciones de memoria, dos temporizadores de 32 bits y dos puertos serie para transferencias de 8/16/24/32 bits. El TMS320C30 incluye un controlador DMA y dispone de soporte para multiproceso. La memoria interna incluye 4k palabras de ROM, 2k palabras de RAM y caché de programa. Su estructura interna se muestra en la Figura 2.2.2.

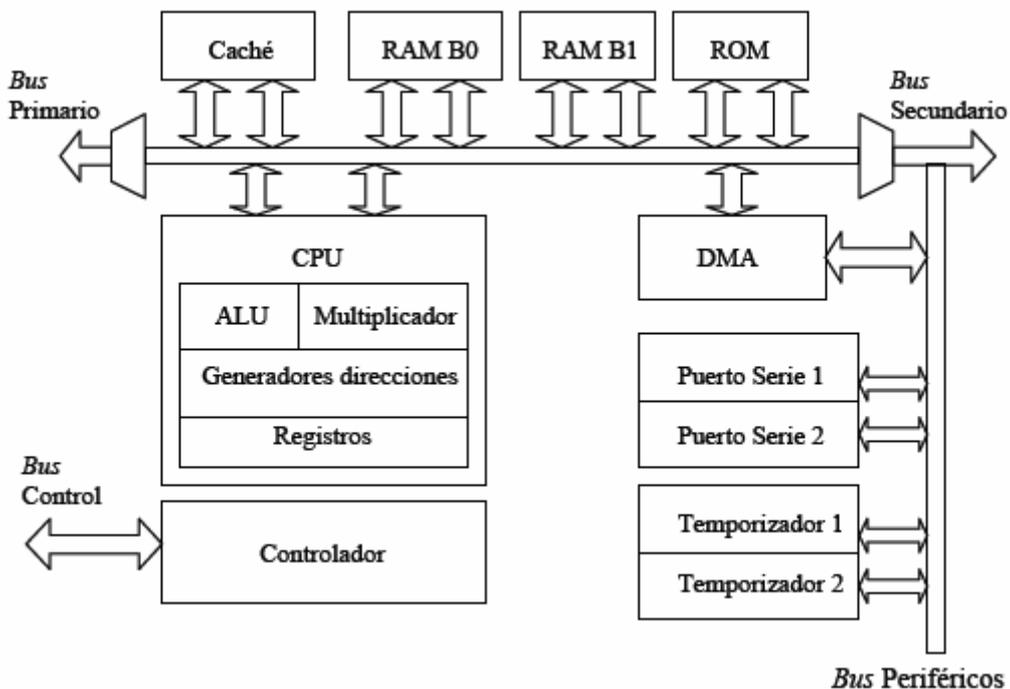


Figura 2.2.2. Diagrama de bloques del TMS320C3x. Figura de [GUER et al. 11]

2.2.1.2 Buses.

El TMS320C3x dispone de una estructura de buses internos y dos buses externos. Los buses de direcciones permiten direccionar hasta un máximo de 16M palabras. En la Figura 2.2.3 puede verse la estructura de buses internos.

Buses internos

Hay tres grupos de buses internos, correspondientes a la memoria de programa, la de datos y del controlador DMA, lo que permite buscar instrucciones, acceder a datos y efectuar operaciones DMA de forma paralela.

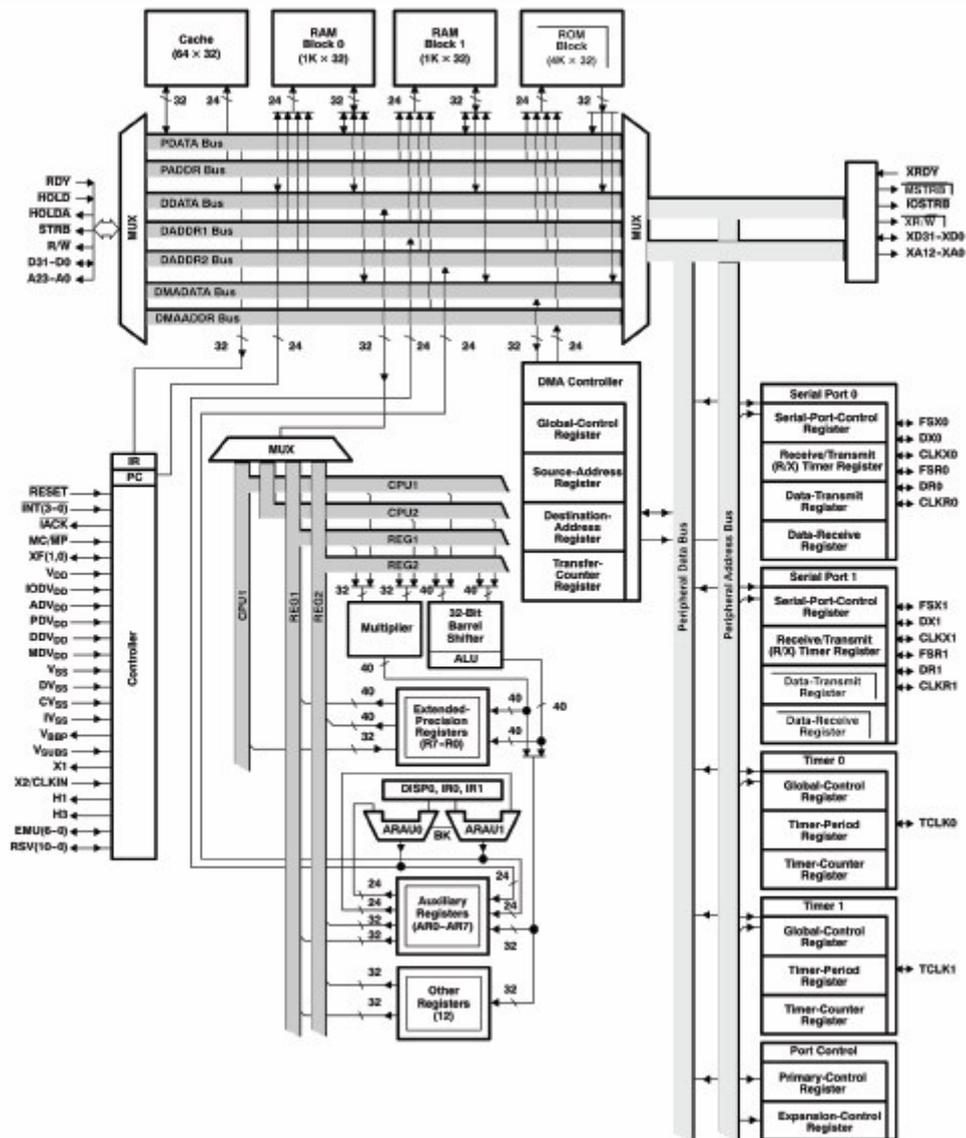


Figura 2.2.3 Buses internos del TMS320C3x (SMJ320C30DSP. Texas Instruments).
Figura de [GUER et al. 11]

Programa

- Direcciones: PADDR (24 bits). Conectado al PC. Dirige RAM, ROM y caché.
- Datos: PDATA (32 bits). Proporciona instrucción a IR.

Datos

- Direcciones: DADDR1/DADDR2 (24 bits). Direcciona ROM y RAM, con 2 direcciones simultáneamente. Conecta con ARAUs y ARs.
- Datos: DDATA (32 bits). Obtiene datos de ROM y RAM. Conecta con CPU1/CPU2.
- CPU1/CPU2: buses de datos (32 bits) de la CPU.
- REG1/REG2: buses de datos (40 bits) para conexión con los registros de la CPU.

DMA

Buses de conexión entre periféricos y memoria para acceso DMA independiente.

- Direcciones: DMAADDR (24 bits).
- Datos: DMADATA (32 bits).

Buses externos

Bus primario: Direcciones (24 bits) y datos (32 bits).

Bus expansión: Direcciones (13 bits) y datos (32 bits). Principalmente para conexión con periféricos.

2.2.1.3 Unidad Central de Proceso.

La CPU del TMS320C3x engloba multiplicador, ALU, desplazadores, generadores de direcciones, los ficheros de registros y una estructura de buses internos.

Multiplicador

El multiplicador realiza multiplicaciones enteras (24 bits) y de coma flotante (32 bits) en un solo ciclo de reloj. Su funcionamiento puede realizarse en paralelo con operaciones de la ALU en el mismo ciclo. Las multiplicaciones de coma flotante utilizan entradas de 32 bits y producen un resultado de 40, mientras que en coma fija los datos de entrada son de 24 bits y producen resultados de 32 bits.

Unidad Aritmético Lógica (ALU)

La ALU ejecuta operaciones en un solo ciclo sobre enteros de 32 bits, operaciones lógicas de 32 bits y operaciones sobre datos en coma flotante de 40 bits, incluyendo

conversiones entero/coma flotante. El desplazador se utiliza para desplazar a derecha o izquierda hasta 32 bits de la palabra en un solo ciclo.

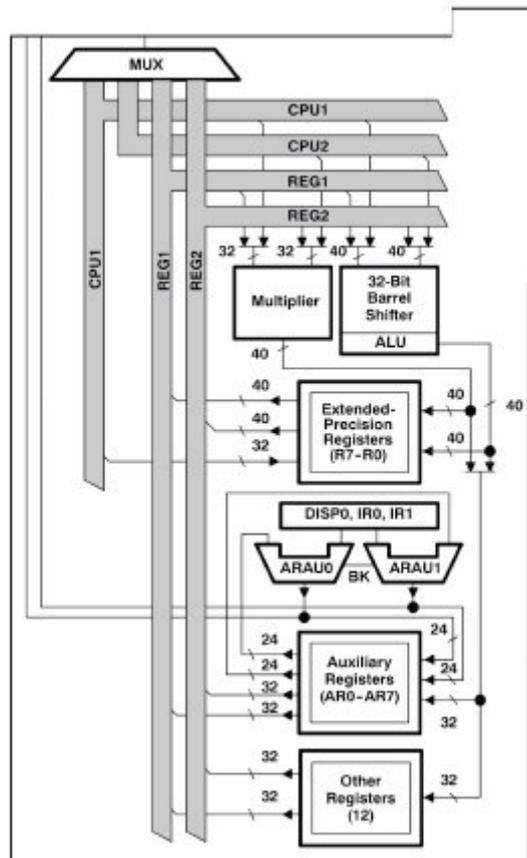


Figura 2.2.4 Diagrama de bloques de la CPU (SMJ320C30DSP. Texas Instruments).
Figura de [GUER et al. 11]

Los buses internos permiten acceder a dos operandos de memoria o del fichero de registros para implementar multiplicaciones y sumas/restas en paralelo, o cuatro operandos enteros en coma fija en un solo ciclo.

Unidades Aritméticas de los Registros Auxiliares (ARAUs).

ARAU0 y ARAU1 permiten generar dos direcciones en el mismo ciclo. Las ARAU operan en paralelo con el multiplicador y la ALU. Soportan direccionamiento con desplazamientos, registros índices, gestión de buffer circular y bit revertido.

Fichero de Registros de la CPU

El TMS320C30 contiene 28 registros contenidos en un fichero de registros multipuerto. Todos ellos pueden operar con el multiplicador y la ALU, y pueden usarse como registros de propósito general, aunque tienen algunas funciones especiales.

Podemos agruparlos en las siguientes clases:

- Registros de precisión extendida (R7-R0): almacenan y soportan operaciones con números enteros de 32 bits y de coma flotante de 40 bits.
- Registros auxiliares (AR7-AR0): de 32 bits, pueden ser modificados por las ARAU. Su función primaria es la generación de direcciones de 24 bits. Pueden utilizarse también como contadores de bucle o como registros de propósito general de 32 bits, que pueden ser utilizados por el multiplicador o la ALU.
- Registros de direccionamiento: además de los auxiliares, el procesador dispone de otros registros para implementar distintos tipos de direccionamiento:
 - El puntero de página de datos (DP) tiene 32 bits, de los cuales los 8 LSB se utilizan en el modo de direccionamiento directo como puntero de página. Las páginas de datos son de 64k palabras, con un total de 256 páginas.
 - Los registros índice (IR0, IR1) de 32 bits contienen el valor usado por el ARAU para direccionamiento indexado.
 - El registro de tamaño de bloque (BK) de 32 bits lo utiliza el ARAU para direccionamiento circular, indicando el tamaño del bloque de datos.
- Registros de control: que incluyen además de la gestión de la pila, las interrupciones, la E/S, etc., el control de bucles, mediante el contador de repetición (RC), de 32 bits, que especifica el número de repeticiones de un bucle de programa, y RS y RE, que definen el comienzo y fin, permitiendo así bucles multiinstrucción.

2.2.1.4 Organización de memoria

Los espacios de programa, datos y E/S están contenidos en el espacio total direccionable de 16M palabras del TMS320C30.

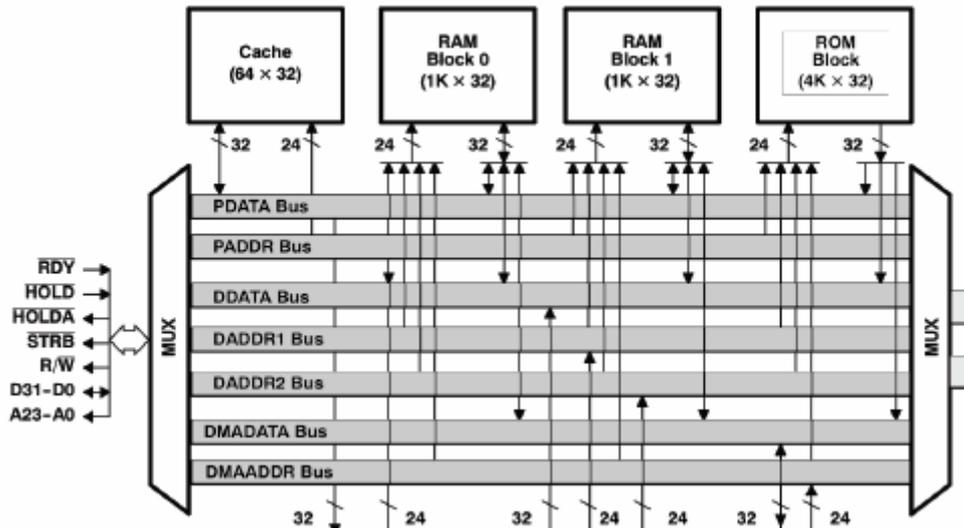


Figura 2.2.5 Organización de memoria (SMJ320C30DSP. Texas Instruments). Figura de [GUER et al. 11]

RAM, ROM y caché

La figura 2.2.5 muestra la organización de la memoria interna del procesador.

Tanto la RAM (bloques 0 y 1, cada uno de 1k palabras), como la ROM (de 4k palabras) son bipuertos que soportan dos accesos de la CPU en el mismo ciclo. Los buses de programa, datos y DMA separados permiten simultáneamente búsquedas de instrucción, lectura/escritura de datos y operaciones DMA.

Existe una caché de instrucciones de 64 palabras para almacenar temporalmente las secciones de código más frecuentemente utilizadas, reduciendo los accesos a memoria principal y permitiendo así la utilización de tecnologías más lentas. La caché puede operar de forma automática completamente transparente al usuario. Está dividida en dos segmentos de 32 palabras cada uno, y utiliza un algoritmo LRU para reemplazamiento de los segmentos en caso de no hallarse presente el código en un acceso.

Mapa de memoria

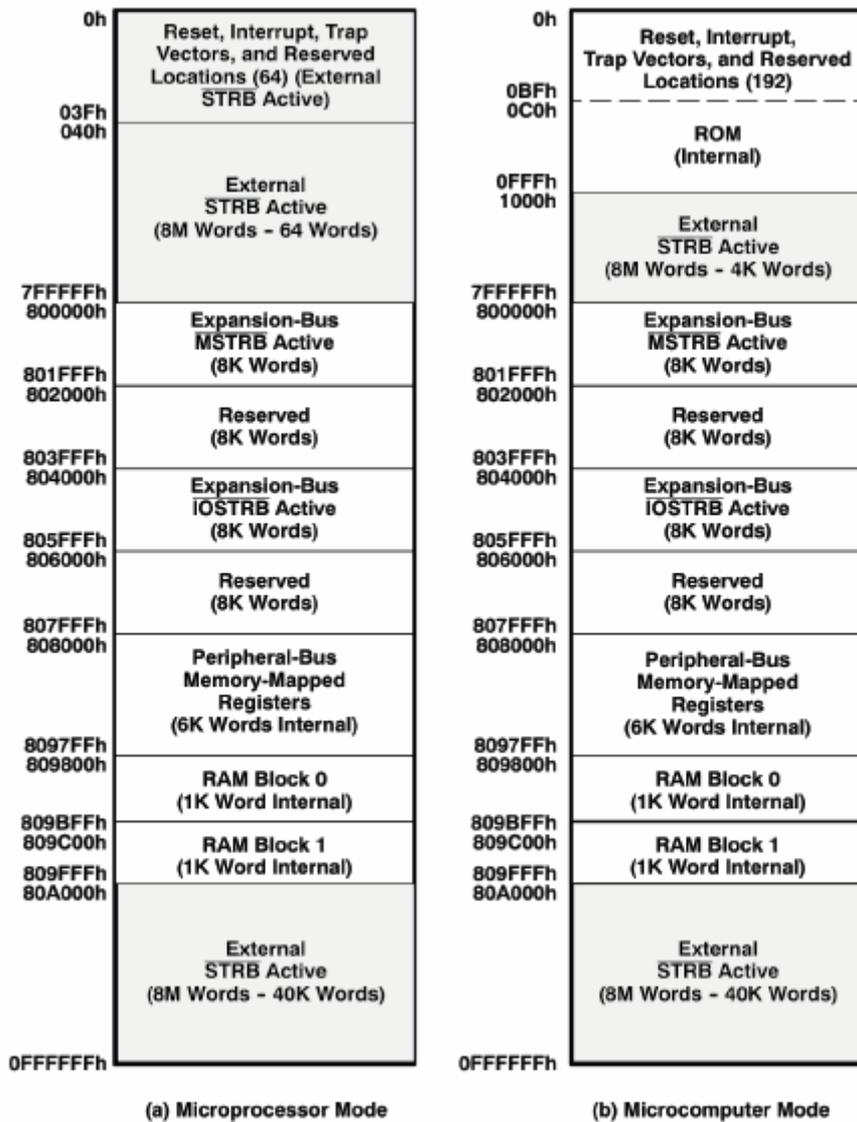


Figura 2.2.6 Mapas de memoria (SMJ320C30DSP. Texas Instruments). Figura de [GUER et al. 11]

El mapa de memoria depende del modo de funcionamiento del procesador. Las direcciones 800000h-801FFFh corresponden a áreas de memoria del bus de expansión, mientras que las 804000h-805FFFh corresponden a áreas de E/S del mismo bus.

Las posiciones 808000h-8097FFh corresponden a los registros mapeados que controlan los periféricos internos. Las posiciones 000h-0BFh están reservadas para los vectores de interrupción.

2.2.1.5 Control.

El controlador del TMS320C3x gestiona las siguientes funciones:

- Control pipeline: la pipeline es de 4 etapas: 1) Búsqueda instrucción; 2) Decodificación; 3) Búsqueda de operandos; 4) Ejecución. Prioriza las etapas y resuelve conflictos posibles con DMA (menor prioridad). Las operaciones de salto pueden ser estándar o retardada.
- Control de bucles multiinstrucción.
- Control de interrupciones: soporta 4 interrupciones externas, diversas interrupciones internas y una señal externa no enmascarable RESET.
- Control de modo de funcionamiento multiprocesador.

2.2.1.6 Periféricos

El TMS320C30 integra dos contadores, dos puertos serie y un controlador de DMA. Todos ellos están controlados por registros mapeados, y disponen de un bus específico que permite comunicación independiente.

Temporizadores

Los temporizadores son contadores/temporizadores de 32 bits de propósito general. Pueden utilizar reloj interno o externo. El procesador los utiliza para generar la frecuencia de muestreo de un conversor A/D, o para producir una interrupción al controlador de DMA que comience una transferencia de datos. Con reloj externo, los temporizadores se pueden utilizar como contadores de eventos, que produzcan una interrupción cuando se alcance determinada cuenta programada.

Puertos serie

El TMS320C30 dispone de dos puertos serie bidireccionales totalmente independientes. Cada puerto puede configurarse para transferencias de 8, 16, 24 o 32 bits por palabra en ambas direcciones. El reloj puede configurarse internamente utilizando alguno de los temporizadores, o externamente. Pueden trabajar en modo de transferencia continua, lo que permite transmitir y recibir cualquier número de palabras sin nuevos pulsos de sincronización.

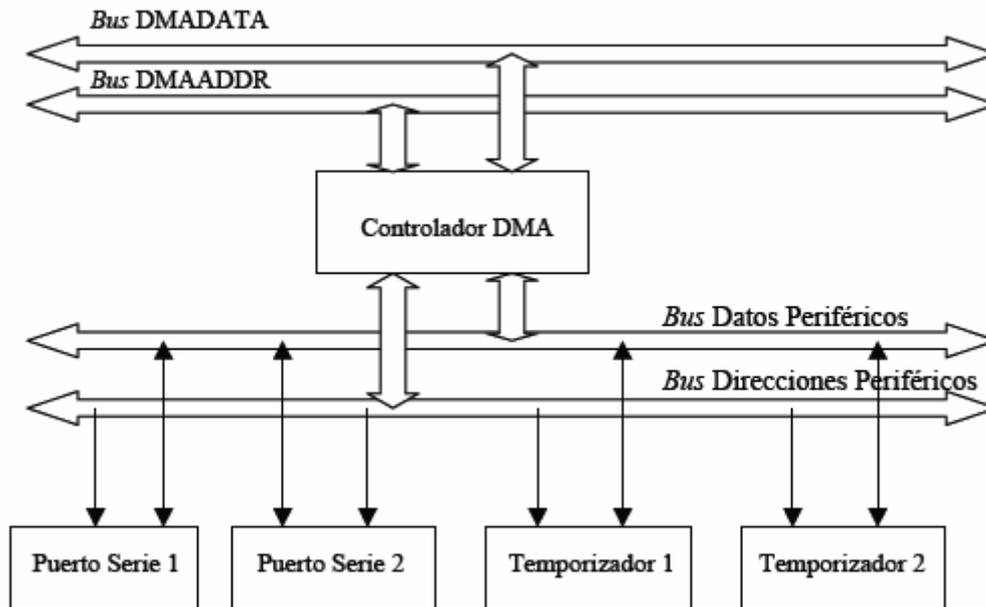


Figura 2.2.7 Periféricos del TMS320C3x. Figura de [GUER et al. 11]

Controlador DMA

El controlador interno de DMA permite realizar accesos E/S sin interferir con la CPU, permitiendo así transferencias entre memorias y periféricos (ADC, puertos serie, etc.) lentos sin reducir la capacidad computacional de la CPU.

El controlador contiene sus propios generadores de direcciones, registros fuente y destino, y contador de transferencias. El controlador puede acceder a cualquier posición del mapa de memoria, incluyendo los periféricos mapeados, a través de buses dedicados que minimizan los conflictos de acceso entre controlador y CPU. Las operaciones DMA son transferencia de bloques o palabras de datos a o desde memoria.

2.2.2 Introducción técnica al SDR.

YOUNGBLOOD [YOUN03], diseñó una etapa frontal de radiofrecuencia (RF) cuya función es convertir la señal de radio recibida, trasladándola a frecuencias muy inferiores, en la banda de audio. Esta etapa es un detector un tanto especial de conversión directa (es decir, de frecuencia intermedia cero o muy baja), a la cual se le han añadido unos filtros de banda seleccionables en el paso de antena.

Con esta etapa frontal, la señal de radiofrecuencia es bajada al rango de las frecuencias de audio, pero sigue siendo una señal sin demodular. El siguiente paso es demodular la

señal, y para ello se hace uso de un ordenador con tarjeta de sonido y el programa adecuado. Esto es lo que se muestra en el diagrama de la figura 2.2.8.

No debe confundirse esta conversión directa con el proceso de demodulación que permite extraer la señal de voz (moduladora) de la señal de RF, pues ello es lo que se hace en el ordenador con el programa adecuado.

La etapa de conversión directa lo que hace es trasladar las señales de RF a otras frecuencias muy inferiores, pero aunque las señales obtenidas en la conversión son señales en el rango de las frecuencias de audio, no significa que sean señales ya desmoduladas (aunque para varios tipos de modulación, como la telegrafía o la banda lateral única, sí lo serán).

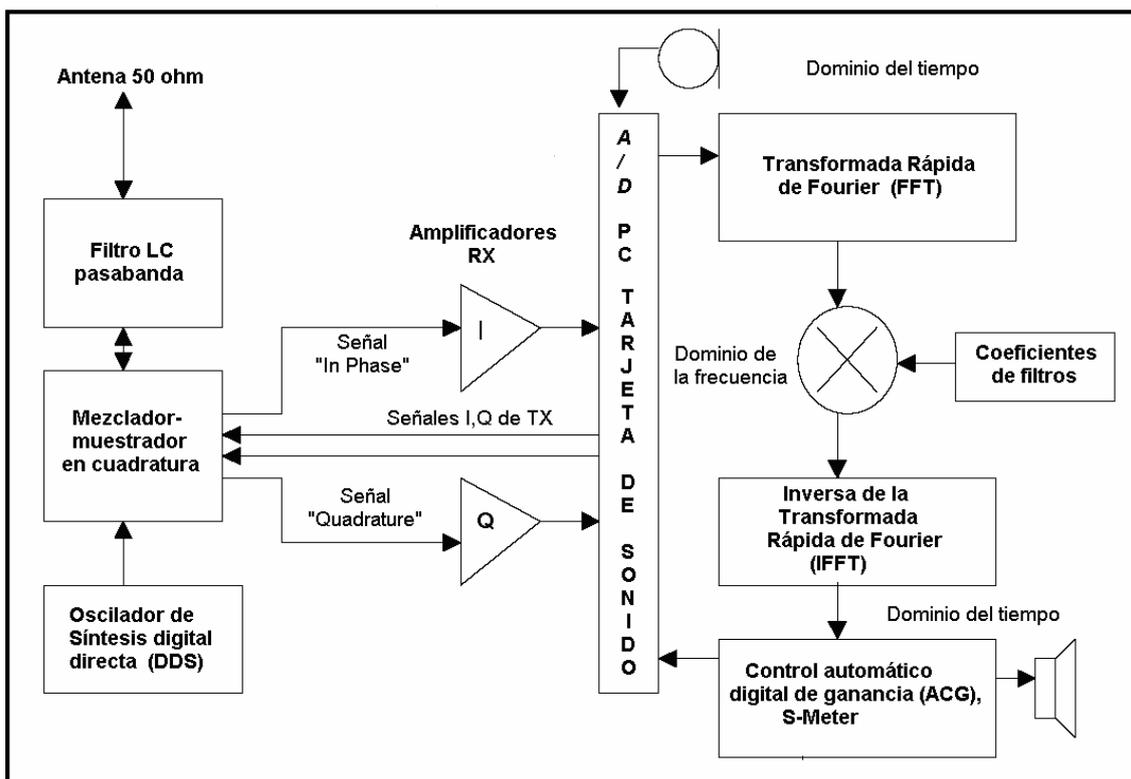


Figura 2.2.8 Diagrama de bloques del equipo SDR-1000

Figura cedida por cortesía de [YOUN03]

Dicha señal ya desplazada en frecuencia a la banda de audio (pero todavía sin demodular) ocupa un margen de frecuencias que entra dentro del margen que acepta en

sus entradas analógicas la tarjeta de sonido. Dependiendo de la tarjeta empleada, una tarjeta de sonido puede aceptar señales de hasta 96 KHz, y más altas.

La tarjeta de sonido incluye unos conversores analógico-digital (A/D) que digitalizan las señales presentes en las entradas de la tarjeta, generando un flujo continuo de bits que representan digitalmente las señales analógicas de entrada. A este respecto, se puede decir que el conversor A/D está conectado casi directamente a la antena. La tarjeta de sonido además proporciona todas las funcionalidades DSP (Procesamiento Digital de Señales) que van a ser necesarias para manejar la señal entrante ya digitalizada.

En la digitalización de las señales de entrada, se transforman las señales analógicas de entrada, que son "continuas en el tiempo" (y con un número de valores de amplitud teóricamente infinitos), en una secuencia de señales "discretas en el tiempo", esto es, que se presentan a intervalos de tiempo determinados, y que se denominan "muestras". Una vez digitalizadas estas muestras de la señal analógica (transformando cada muestra en un número fijo de bits), las señales eléctricas que las componen (los bits) tendrán un número de valores posibles de amplitud fijo y determinado, dos en el caso de las señales digitales binarias (denominados "0" y "1" lógicos).

El proceso por el que las señales analógicas de entrada se transforman en señales discretas en el tiempo se denomina "muestreo". A este respecto, en 1933 Harry Nyquist [NYQU28] estableció que cuando se digitalizan señales analógicas, para que posteriormente se pueda recuperar la señal analógica original mediante el proceso inverso (Conversión Digital-Analógica), la señal analógica debe ser muestreada a una velocidad como mínimo igual al doble de la frecuencia más alta presente en la señal analógica. La frecuencia máxima de la señal analógica de entrada es recomendable que sea limitada mediante el empleo de un filtro de paso banda, denominado "filtro antialiasing". Ello evita el efecto de "aliasing", por el cual, al recuperar por el proceso inverso la señal analógica original, ésta puede aparecer distorsionada. El aliasing aparece cuando en el proceso de digitalización, la señal analógica de entrada tiene componentes cuyas frecuencias son superiores a la mitad del valor de la frecuencia de muestreo, y no son suprimidas antes de realizar el muestreo.

Así, por ejemplo, el oído humano puede llegar a percibir frecuencias en el rango de 20 Hz a 20 KHz (en el mejor de los casos), y por ello la digitalización de las señales de audio de alta fidelidad requiere que sean muestreadas al menos a 40 KHz (40.000 muestras por segundo). De hecho, en el caso de los CD's de audio el muestreo empleado es de 44,100 KHz (lo cual previene también el aliasing). Esta velocidad de muestreo la soporta actualmente también cualquier tarjeta de sonido para sus conversores A/D. Además, muchas tarjetas de sonido incluyen un filtro antialiasing interno con una frecuencia de corte próxima a los 20 KHz.

Una vez la señal ha sido muestreada y digitalizada por la tarjeta de sonido, podemos procesarla como queramos, por ejemplo demodulándola: en modulación de amplitud (AM) se haría detectando la amplitud de la envolvente de la señal, en modulación de frecuencia (FM) habría que seguir las variaciones de frecuencia de la señal. Todos estos procesos se realizan mediante cálculos matemáticos adecuados por software, pues al fin y al cabo las señales digitales son señales discretas que representan valores numéricos (en formato binario) y por tanto (éstos) se pueden tratar matemáticamente con el software adecuado para realizar algo. Los resultados numéricos de estos tratamientos matemáticos son también representados por señales digitales, las cuales se llevan internamente a unos conversores Digital-Analógicos (D/A) implementados en la propia tarjeta de sonido, para convertirlas a señales de audio equivalentes, que son amplificadas y conducidas a las salidas de altavoz de la tarjeta de sonido (caso de las modulaciones de fonía), o que son procesadas para mostrar la información en la pantalla del ordenador (caso de los modos digitales).

Uno de los posibles procesos de la señal de audio digitalizada es su demodulación, y éste es un proceso relativamente simple que incluye procesos de filtrado, desplazamiento de nivel y algunas otras operaciones que se pueden realizar sin muchos problemas por software. Cualquier proceso al que queramos someter la señal de audio digitalizada se puede hacer por software, usando el programa adecuado en el ordenador, y usando la tarjeta de sonido como digitalizador de la señal analógica de audio.

Las tarjetas de sonido son capaces de operar con señales de audio analógico de hasta unos 20 KHz como mínimo (hasta 48 KHz y más, si la tarjeta es de calidad), y la queremos emplear para procesar señales de RF de varios MHz de frecuencia, recibidas en la antena. Dado que la tarjeta de sonido no puede manejar frecuencias tan elevadas,

es necesario convertir las señales de RF a señales de frecuencias muy inferiores, en el rango de frecuencias que puede aceptar la tarjeta de sonido (es decir, en el rango de las bajas frecuencias), manteniendo éstas el mismo esquema de modulación de las señales de RF recibidas.

El sistema más habitual para convertir una señal de una banda de frecuencias a otra es mediante el uso de un mezclador, el cual combina dos señales, la que recibimos en antena, con la procedente de un oscilador local, la cual será una señal pura sin modular, y muy estable en frecuencia. Esto es típico en los receptores superheterodinos para convertir la señal recibida a la comúnmente denominada FI o frecuencia intermedia (en Inglés: Intermediate Frequency o IF).

2.2.3 Conversión de frecuencias intermedias en señales I/Q.

Conocido como "Método de fase o I/Q", consiste en usar un mezclador doblemente balanceado al cual se aplican dos versiones de la frecuencia del oscilador local, una en fase y la otra desfasada 90 grados, y dos versiones de la señal recibida en antena, una en fase y otra desfasada 90 grados. Si esta última estaba desfasada exactamente +90 grados respecto a la otra, a la salida del mezclador se obtenía la banda lateral superior, cancelando la banda lateral inferior, mientras que si está desfasada -90 grados, la que se cancelaba era la banda lateral superior, obteniéndose la banda lateral inferior.

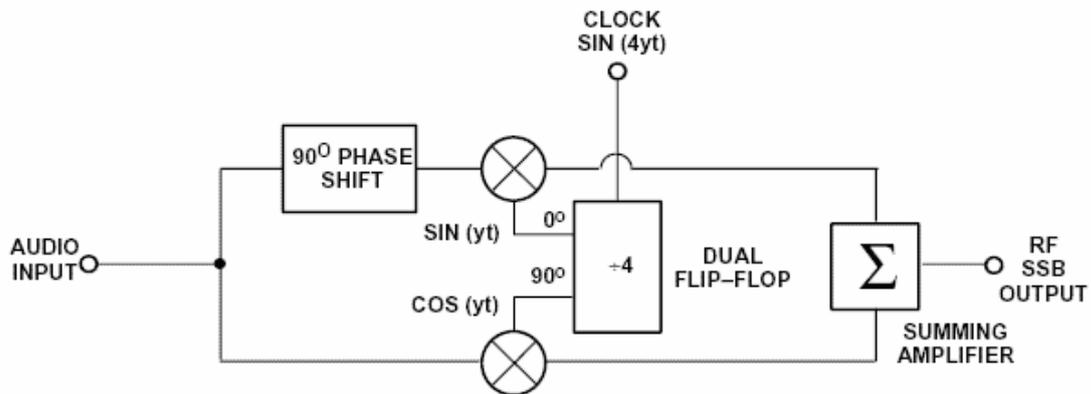


Figura 2.2.9 Principio del Método de fase para generar señales SSB

Este procedimiento, empleado tanto para la modulación como para la demodulación, se denomina Mezcla en cuadratura, Detección de SSB por giro de fase, o Mezcla con rechazo de imagen. El doble mezclador empleado se denomina Mezclador de cuadratura

o Mezclador IQ. Esta última denominación se debe porque a la señal convertida correspondiente a la original no desfasada se denomina Señal en Fase o Señal I ("In phase signal"), mientras que la señal convertida correspondiente a la señal original desfasada +90 (o -90) grados se denomina Señal en Cuadratura o Señal Q ("Quadrature signal").

A la salida de los dos mezcladores se obtienen las señales I y Q, que están en Cuadratura de Fase. Ambas señales son exactamente iguales, pero desfasadas 90 grados (en el caso ideal). Llevando estas señales a un circuito sumador adecuado, se obtendría a la salida de éste la señal deseada, libre de la frecuencia imagen. Es en el circuito sumador donde, dependiendo de la fase de la señal Q respecto a la señal I, se suprime como frecuencia imagen las frecuencias recibidas que estén por encima o por debajo de la frecuencia del oscilador local, reforzándose la otra.

Como muestra la figura 2.2.10, el oscilador local genera la frecuencia de mezcla, f_{lo} , que es inyectada directamente al mezclador inferior (señal "seno") para obtener la señal I a la salida del mezclador. Una parte de la señal f_{lo} pasa por un desfasador de 90 grados para obtener una señal del oscilador local desfasada 90 grados (señal "coseno") que es llevada al mezclador superior, obteniéndose a su salida la señal en cuadratura Q.

Las señales I y Q pasan por sendos filtros de paso bajo (LPF) para eliminar las frecuencias no deseadas que aparecen en los procesos de mezclado de las señales, filtrando la señal deseada, y a continuación, son muestreadas (a la frecuencia de muestreo f_s) y digitalizadas individualmente en sendos conversores analógico-digitales (A/D) para obtener las señales I y Q digitalizadas ($I(t)$, $Q(t)$, discontinuas en el tiempo).

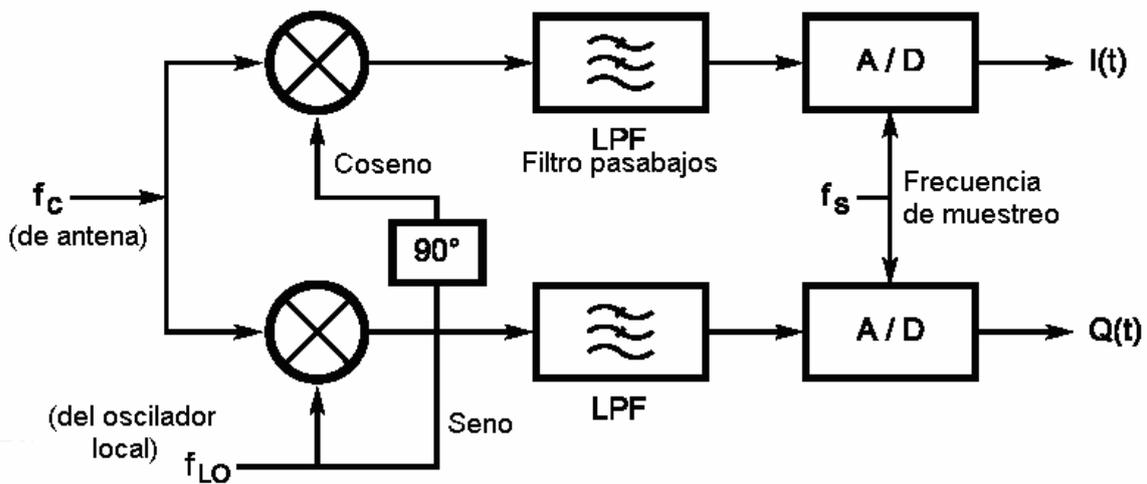


Figura 2.2.10 Principio del mezclador en cuadratura con muestreo.

Ilustración cedida por cortesía de [YOUN03]

Una vez realizado esto, el software SDR cargado en el ordenador se encargará de todo lo demás, obteniéndose el resultado del procesado de las señales en los altavoces conectados a la salida de la tarjeta de sonido.

2.2.4 La demodulación.

Hasta ahora hemos descrito cómo se traslada la señal de RF a frecuencias de audio (es decir, a Banda base), y la generación de las componentes I y Q; falta todavía filtrar y demodular las señales, además de otros procesados en audio que se deseen. Ello se realiza digitalmente, por lo que hay que digitalizar las señales I y Q y luego procesarlas digitalmente. Por ello el diseño de [YOUN03] emplea una tarjeta de sonido común para digitalizar las señales I y Q y para proporcionar todas las funciones del receptor, sacando partido a la potencia del DSP implementado en la tarjeta de sonido y usando el software adecuado.

Sin emplear más que los programas adecuados, casi todas las tarjetas de sonido pueden ser programadas para actuar como un CAG, demodular una señal, eliminar señales no deseadas (como en los equipos más caros con DSP), reducir el ruido (NB), silenciar (squelch), absolutamente todo lo que puedan hacer los equipos de radio, más algunas cosas de las que éstos son incapaces; y todo ello en el ordenador. Los mismos principios

rigen en el otro sentido en transmisión, desde el micrófono que capta la voz hasta la señal enviada hacia la antena.

Sólo como ejemplo, se expondrán los principios de la demodulación en un receptor de este tipo, uno de los procesos básicos de cualquier receptor.

Cuando tenemos una señal modulada en amplitud (AM), lo único importante de la señal es la amplitud de la envolvente de la señal de RF (no la amplitud instantánea de la onda, sólo la de pico o cresta de cada semiciclo de la onda). Un detector de AM a diodo simplemente responde a la amplitud de la envolvente, que es la señal moduladora de BF que interesa obtener.

En un receptor que maneje señales I y Q, al estar ambas desfasadas 90 grados entre sí, si se representan vectorialmente sus amplitudes y fases en un gráfico de ejes de abscisas y ordenadas (x,y), la representación será del siguiente tipo (diagrama de fases en el plano complejo):

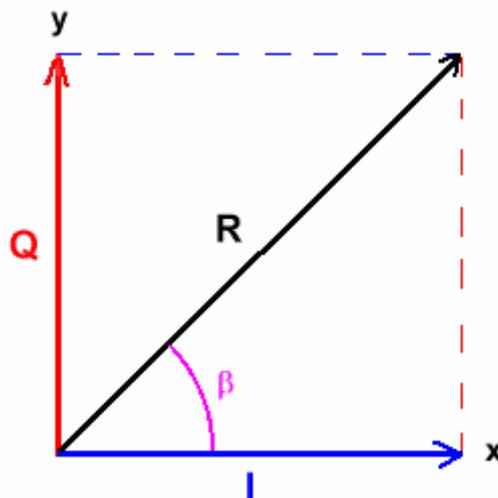


Figura 2.2.11 Componentes I y Q.

donde la longitud de los vectores I y Q representan las amplitudes de dichas señales. En este tipo de diagramas vectoriales, los ángulos de fase se giran en sentido antihorario.

Con este tipo de diagramas, el conjunto de las dos componentes I y Q se puede representar por un único vector resultante R, con un ángulo de fase β . Dado que las componentes I y Q de este diagrama son "ortonormales", esto es, perpendiculares entre sí (desfasadas 90 grados entre sí), el valor de la amplitud instantánea del vector R se

calcula fácilmente por el teorema de Pitágoras, ya que R es la hipotenusa del triángulo recto formado por I y Q como lados catetos:

$$R = \sqrt{I^2 + Q^2} \quad (1.1)$$

Las señales I y Q representan a la envolvente de la señal de antena, por lo que el conocimiento del valor de la amplitud de cualquiera de ellas a lo largo del tiempo indica cómo es la forma de la envolvente y por tanto de la señal moduladora, y por tanto establece la demodulación de amplitud. Pero como la señal Q sigue las mismas variaciones de amplitud que la señal I, basta conocer cómo varía el valor de amplitud del vector R a lo largo del tiempo para conocer la forma de la envolvente de la señal de RF de antena, y por tanto demodular en amplitud ésta, y además demodulándola con ganancia, ya que el valor de amplitud del vector R será siempre igual o mayor al del vector I.

Sin embargo, en las modulaciones de fase, como son la FM (modulación de frecuencia) o la PM (modulación de fase), en la modulación el ángulo de fase de la portadora varía con la amplitud de la señal moduladora (y en el caso de la FM ello conduce a la apariencia que la frecuencia de la portadora varía con la amplitud de la señal moduladora), y por ello la demodulación depende de las variaciones de la fase instantánea de la señal de RF.

En FM y PM a nivel de las señales I y Q implica que las amplitudes de estas dos componentes no varían de la misma forma en cada instante, y por tanto el ángulo de fase β del vector R variará con el tiempo. De cómo varía este ángulo de fase dependerá la forma de la señal moduladora, y por tanto, el conocimiento del valor de este ángulo de fase con el tiempo permite la demodulación en fase o en frecuencia. Si conocemos las amplitudes instantáneas de las componentes I y Q, el ángulo de fase β se puede deducir mediante la siguiente expresión:

$$\beta = \text{Arctang}(Q / I) \quad (1.2)$$

Donde la función arcotangente es la función trigonométrica inversa a la función

tangente de un ángulo.

Al digitalizar las señales I y Q, se obtienen una sucesión de conjuntos de bits discretos en el tiempo, que codifican numéricamente el valor de amplitud de ambas señales, es decir, la digitalización de las señales I y Q da lugar a una sucesión de números binarios que indican las amplitudes instantáneas de ambas señales a intervalos regulares en el tiempo. Y como números que son, ya pueden ser manejados matemáticamente, por lo que ya se pueden aplicar las fórmulas anteriores para conocer los valores del vector R y de su ángulo de fase β , que permitirán conocer los valores de amplitud de la envolvente, en el caso de las modulaciones de AM, y de los desplazamientos de fase y frecuencia de la señal recibida, en el caso de las modulaciones FM y PM. Es decir, el cálculo numérico permite la demodulación de las señales de RF moduladas en AM, FM y PM. Los resultados de estos procesos matemáticos son también valores binarios, que al ser aplicados a un conversor digital-analógico (incluido en la tarjeta de sonido), permite obtener la señal moduladora ya a nivel analógico, la cual una vez amplificada es llevada al altavoz conectado a la tarjeta de sonido.

Para el caso de las modulaciones de banda lateral única (SSB), el proceso es algo más complicado, ya que el cálculo numérico sobre los valores digitales de las componentes I y Q requiere de más pasos de cálculo que para la AM y la FM. Pero conociendo las expresiones matemáticas que se han de aplicar para determinar la forma de la señal moduladora a partir de los valores de amplitud de R y del ángulo de fase β , tampoco es mucho problema realizar la demodulación de SSB. Y lo mismo se puede decir para cualquier otro tipo de modulación que se haya aplicado a la señal recibida en antena para poder demodularla.

Por ello, dado que en los receptores SDR hay muy poco procesado analógico de la señal, el método de conversión de frecuencia y detección empleado y la posibilidad de gobernar muchos parámetros, es de esperar que un equipo de estas características superará las prestaciones de los mejores transceptores disponibles hoy en día, al poder implementar formas de onda sintéticas con gran facilidad y con los límites que establezcan las Matemáticas.

2.3 Sistemas de Radiocomunicaciones basados en software.

2.3.1 Introducción.

Los receptores SDR o Receptores Definidos por Software (Software Defined Radio) son equipos de bajo coste (desde 10 euros a 700 euros) que no disponen de paneles de mando, sino de una única interfaz Ethernet o USB para ser configurados y utilizados mediante ordenador.

Su diferencia fundamental con los receptores normales es que el proceso de la señal se realiza por software, no disponen de filtro de frecuencia intermedia y procesan la señal IQ recibida mediante la tarjeta de sonido del ordenador asociado al equipo.

La sintonización de la señal a recibir la hace el operador, bien de forma local o remota, a su vez se puede obtener a distancia la señal recibida, así como sintonizar la frecuencia deseada.

Los receptores utilizados en el servicio móvil marítimo sólo permiten trabajar en banda superestrecha, es decir 3 KHz para fonía H3E o J3E y 500 Hz para radioteletipo y DSC F1B/J2B. Además hacen imprescindible la instalación de una infraestructura que podemos considerar como obsoleta, al ser operada, fundamentalmente en modo local.



Figura 2.3.1 Transceptor de MF/HF del Servicio Móvil Marítimo.

La capacidad de seleccionar una entre múltiples frecuencias disponibles de manera remota permite abaratar costes de instalación, ya que el supervisor de la red autoriza al

usuario local la monitorización de las frecuencias de interés en tiempo real y vía red de área local, es decir, “comparte sus antenas y receptores”.

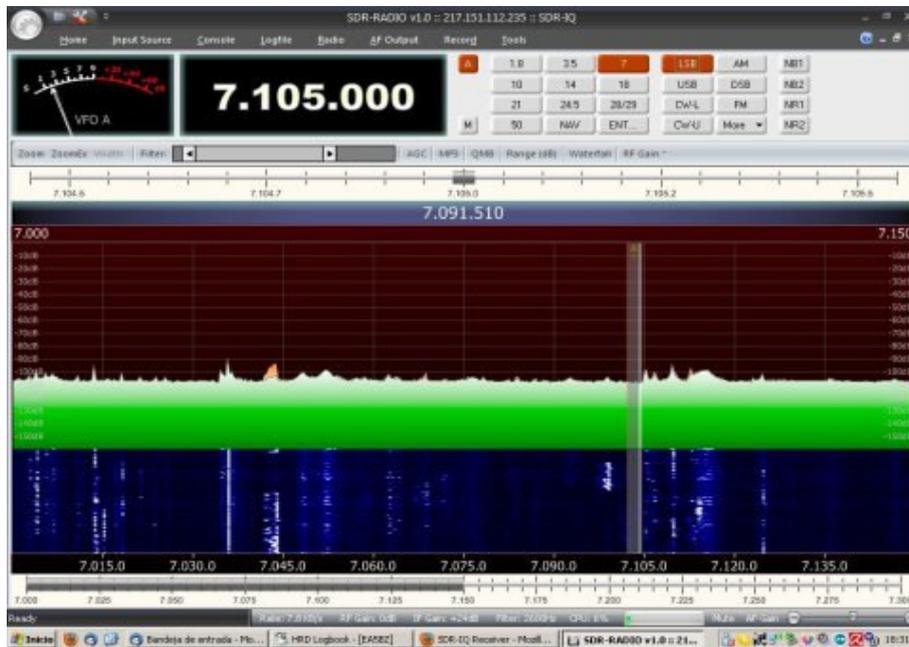


Figura 2.3.2 Selección de una frecuencia entre 7,00 MHz y 7,20 MHz.

Los receptores SDR disponibles en el mercado permiten la visualización de señales de hasta 2 MHz de anchura de banda desde 100 KHz hasta 30 MHz de modo directo y 30 MHz seleccionables, en pasos de 2 MHz, desde 30 MHz hasta 2 GHz o más si se le dota de un convertidor de frecuencias de bajo ruido (LNB o Low Noise Brancher).

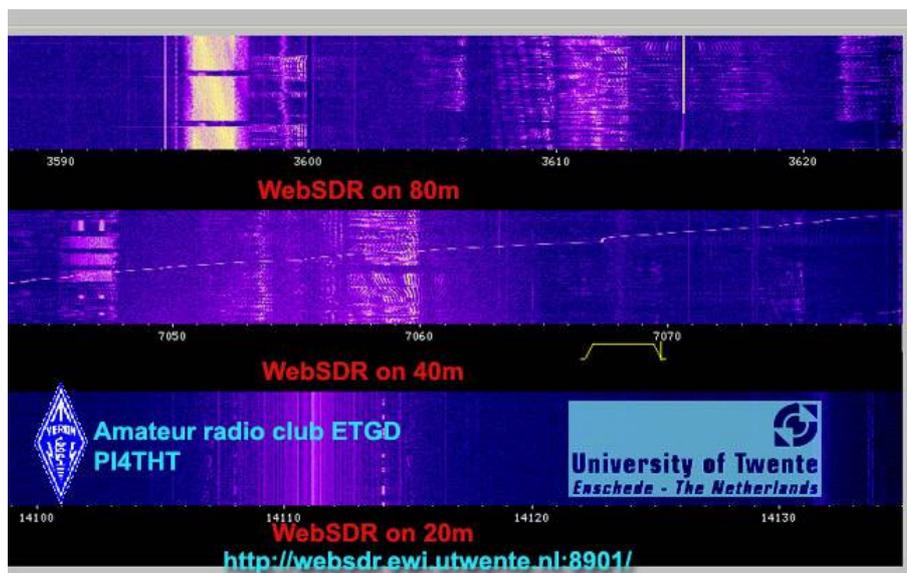


Figura 2.3.3 Pantalla de conjunto de receptores SDR (Universidad de Twente - 2010).



Figura 2.3.4 Pantalla del receptor Perseus.

En la figura 2.3.4 puede verse una imagen de la interfaz hombre-máquina del receptor Perseus.

2.3.2 Bloques funcionales de SDR.

El concepto SDR ha ido evolucionando con los años, pero los avances conseguidos han partido esencialmente de la misma configuración básica que se muestra en la Fig. 2.3.5, la cual se compone de tres bloques funcionales: sección de RF, sección de IF y sección Banda Base, de las cuales RF casi siempre trabaja con hardware analógico mientras que las secciones de IF y Banda Base se implementan con módulos hardware digitales.

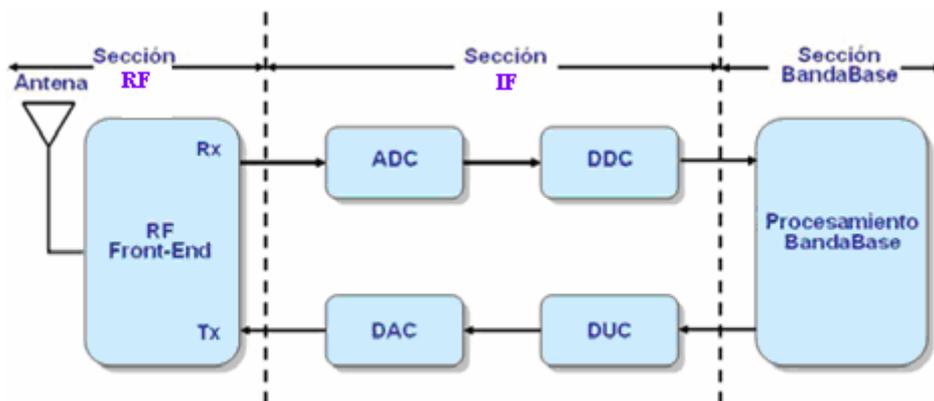


Figura 2.3.5 Diagrama de bloques funcionales de SDR.

La sección de RF – también llamada RF Front-End – es la responsable de transmitir/recibir las señales de radio frecuencia para adecuarlas y convertirlas en frecuencias intermedias – en el caso de la recepción – o amplificar y modular las señales de IF adecuándolas para la transmisión en el aire (en el caso de la transmisión).

La sección de frecuencia intermedia FI, o en inglés IF, es la encargada de pasar la señal de IF a banda base y digitalizarla (en el caso de la recepción) o pasar la señal de banda base a IF y hacer la conversión digital-analógica de la señal (en el caso de la transmisión). Las encargadas de la conversión analógica-digital o digital-analógica de la señal son los módulos ADC/DAC. Los módulos DDC/DUC (Digital Down Converter / Digital Up Converter) son los encargados de bajar digitalmente la señal de IF a Banda Base o subir de Banda Base a IF respectivamente.

La sección de Banda Base es la encargada de todo el procesamiento en banda base de la señal – como frequency hopping, establecimiento de sesión, ecualización, manejo de tiempos de bit, entre otros – y en algunos casos de la implementación de protocolos del nivel de enlace del modelo OSI12.

En esta configuración de SDR, los módulos DDC/DUC y la sección de Banda Base son los que más uso hacen de los recursos del ordenador, motivo por el cual normalmente son implementados en dispositivos de propósito general.

2.3.3 Arquitectura SDR.

La Figura 2.3.6 presenta la arquitectura básica ideal y resumida de SDR. La parte analógica es la encargada de realizar todas aquellas operaciones que no pueden ser efectuadas en el dominio digital, como alimentación a la antena, filtrado y combinación en RF, preamplificación, amplificación, y generación de la frecuencia de referencia.

La idea de la arquitectura es que las etapas de conversión analógico/digital estén lo más cercanas posible a la antena, de hecho, la separación de portadoras y la conversión de frecuencias de subida o bajada son desempeñadas por los recursos de procesamiento digital, al igual que la codificación de canal y las modulaciones.

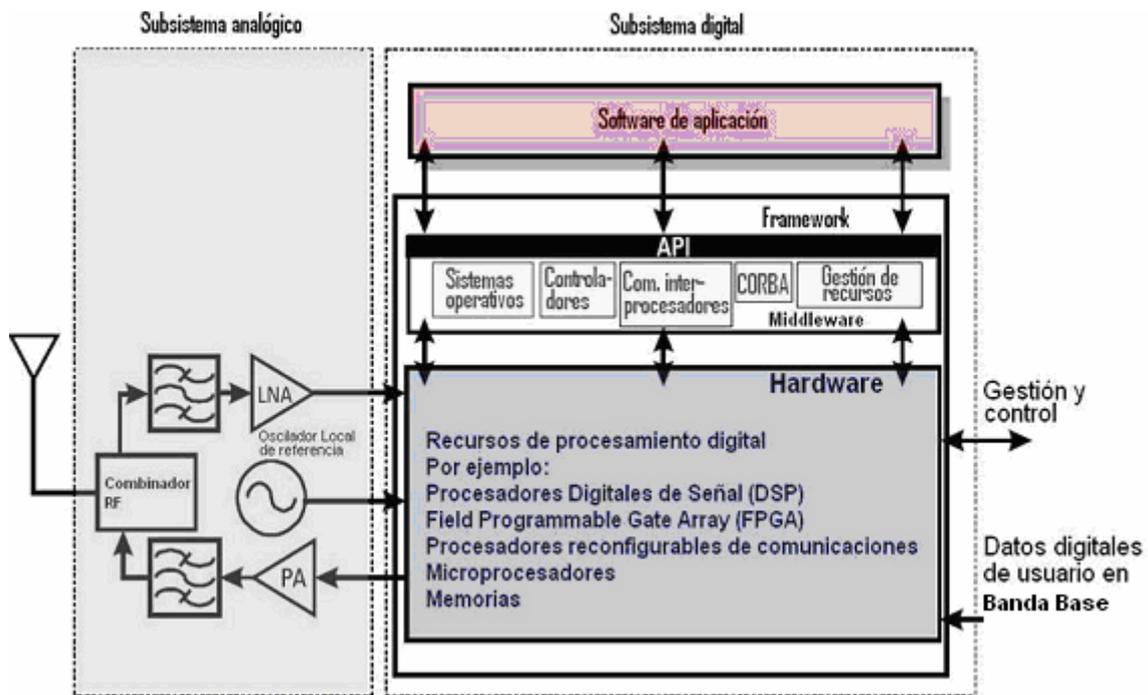


Figura 2.3.6 Arquitectura resumida de un SDR.

El software del sistema está estratificado en capas de la misma manera que sucede en muchas otras arquitecturas, con el objetivo de hacerlo modular y adaptable al hardware sobre el cual operará.

El resto de componentes software son los comunes a los equipos terminales y a los equipos de transmisión e interconexión (sistemas operativos, las API, controladores, entre otros), que facilitan el desarrollo de aplicaciones por parte de los programadores.

El hardware existente dentro del subsistema digital es el encargado de proporcionar toda la flexibilidad y reconfigurabilidad.

Normalmente este hardware está constituido por dispositivos DSP, pero cada vez son más frecuentes las implementaciones que combinan DSP con FPGA y ASIC (Application Specific Integrated Circuit) [MITO92] para el desarrollo de las diversas funciones que desempeña este subsistema, para las cuales cada tipo de dispositivo ofrece ventajas y desventajas significativas en su elección como plataforma hardware de implementación.

Algunos autores, como Mitola (considerado el padre de SDR), diferencian dos tipos de soporte software, denominados software on-line y software off-line. En términos generales el software online es aquel encargado de proporcionar a los equipos la capacidad para realizar las operaciones y funciones básicas que son ejecutadas frecuentemente (intervalos de segundos o minutos), proporcionando reconfigurabilidad; mientras que el software off-line es el encargado de soportar aquellas funciones que se ejecutan con poca frecuencia (intervalos de horas, días o semanas), proporcionando flexibilidad al sistema para actualizaciones que permitan prestar nuevos servicios o mejorar los existentes sin necesidad de cambiar los equipos.

La configuración/reconfiguración del hardware presente en el subsistema digital se realiza utilizando software, que de cualquier manera constituye código de programación; este software es desarrollado utilizando diversas metodologías y usando herramientas para escritura de código y simulación de sistemas para efectos de pruebas y validación de los componentes desarrollados.

Normalmente el lenguaje de escritura es VHDL (Very High Speed Integrated Circuit Hardware Description Language) o un análogo. No obstante, la arquitectura presentada en la Fig. 2.3.6 se llama “ideal” porque sólo es comercialmente aplicable en casos en los cuales las tasas de transmisión son bajas y se trabaja en bandas HF/VHF, pero se vuelve poco práctica para otro tipo de sistemas modernos como los desarrollados para acceso fijo y móvil de banda ancha, sistemas móviles de tercera generación, y redes inalámbricas de área regional.

2.3.4 Ventajas y desventajas de los sistemas de comunicaciones basados en SDR.

Las ventajas que estas arquitecturas presentan (especialmente para los operadores y proveedores de servicios) van desde facilidades para el roaming global hasta la versatilidad del software para proporcionar nuevos y mejores servicios sin necesidad de cambiar los terminales u otros equipos relacionados.

Desde el punto de vista legal, muchos aspectos regulatorios también pueden verse facilitados, especialmente la certificación de terminales teniendo en cuenta que es sólo el software lo que debe certificarse.

Para los usuarios finales hay también muchas ventajas relacionadas con la calidad de los servicios de soporte por parte de los operadores, ya que las actualizaciones, corrección de fallos, adición de servicios y seguimiento a los usuarios, se realizarán de manera más eficiente.

Sin embargo, hay ciertos problemas relacionados con la implementación actual de estas tecnologías:

- El volumen de software descargado para los dispositivos es cada vez mayor y exige complejidad en los componentes, y como primera consecuencia los tiempos de descarga aumentan considerablemente.
- El tiempo de configuración de los dispositivos hardware también aumenta con la complejidad.
- Los procesos de normalización van a un ritmo más lento que los procesos de investigación y desarrollo.
- El hecho de que el sistema de radio esté basado en software aumenta la vulnerabilidad de los sistemas frente a virus y otras amenazas.
- Apenas se están empezando a considerar aspectos como métodos de gestión para varias unidades SDR en un mismo terminal, y terminales de múltiples antenas ajustadas para cada tecnología/modo específico, los cuales pueden ser requisitos de los usuarios y del mercado en general.

Finalmente, gran parte del futuro que tiene SDR depende de los avances que se den en otros campos como la microelectrónica, especialmente en términos de miniaturización, incremento en la capacidad, reducción en consumo de potencia, y modularidad de componentes, así como de factores regulatorios y comerciales que normalmente difieren de un país a otro y afectan de manera distinta a cada segmento del mercado.

2.3.5 Plataformas hardware para su implementación.

Para realizar el procesamiento de las señales basándose en software existen diversas alternativas diferenciadas principalmente por dos parámetros: consumo de potencia y coste; aunque el tamaño de los componentes también puede ser una variable importante a considerar en ciertas aplicaciones.

Especialmente el factor coste es bastante polifacético, pues es percibido de manera diferente por los operadores y por los usuarios, y no siempre la valoración del coste se resume a considerar el “precio” del dispositivo sobre el cual se ejecutará el código.

Kenington [KENI05] presenta algunos de los factores más determinantes en el coste de los elementos digitales de un sistema SDR:

- El coste directo de los dispositivos en sí mismos.
- El coste asociado a los componentes adicionales y complementarios.
- Costes no recurrentes (asociados más con soluciones basadas en ASIC).
- Inversiones en herramientas y entrenamiento para el desarrollo.
- Costes relacionados con consumo de potencia, refrigeración y tecnologías de alimentación.
- El ciclo de vida del producto SDR y sus diferencias con el ciclo de vida de las aplicaciones/servicios sobre él implementadas.
- Grado de flexibilidad requerido.

Todos estos aspectos imponen retos a cada una de las opciones para la implementación física de SDR.

El paradigma tradicional para funciones en banda base y de controladores está basado en procesadores de conjuntos simples de instrucciones optimizados para cómputo aritmético de alta velocidad, llamados DSP (Procesadores Digitales de Señal).

De esta manera, los actuales requisitos de flexibilidad en el procesamiento multimodo y multibanda imponen serios retos a los esquemas de implementación física de SDR.

2.3.6 DSP.

DSP fue la primera tecnología en ser considerada para implementación de SDR, especialmente porque las aplicaciones militares en Estados Unidos tienen limitaciones bajas en cuanto a coste [KENI05].

Los DSP tienen la ventaja de dotar de amplia flexibilidad al sistema, con amplia gama de aplicaciones que se soporta en el fuerte conocimiento que existe en el desarrollo de software para este tipo de dispositivos.

Adicionalmente, gozan de economías de escala por su amplia aplicación en otras áreas diferentes a las comunicaciones, pero por sus características de coste y procesamiento son más aptos para prototipos y primeros volúmenes de producción que para producción en serie de dispositivos orientados al usuario final.

En el caso específico de SDR, los DSP se utilizan más en formas de procesamiento de señal menos intensivas que en aplicaciones front-end de alta velocidad.

2.3.7 Arquitectura GNU Radio.

2.3.7.1 Descripción general.

GNU Radio es un software de desarrollo de herramientas de código libre y abierto que proporciona bloques de procesamiento de señales para implementar software radio.

GNU Radio se puede usar con hardware de RF externo de bajo coste para crear radios definidas por software o sin hardware utilizando el entorno de simulación.

Dentro de esta Tesis doctoral se ha investigado e instalado un servidor con GNU Radio y el receptor RTL-SDR, dado su interés para la enseñanza de las radiocomunicaciones y el bajo coste de este equipo.

A su vez, este equipo puede ser utilizado con el receptor Funcube Dongle.

La explicación que aparece a continuación hace referencia a los bloques e instalación de este sistema con su receptor original, la tarjeta USRP de la compañía ETTUS Research que ofrece una característica de recepción y transmisión mucho más amplia que el receptor RTL-SDR, pero su precio es realmente elevado.

GNU Radio es ampliamente utilizado en entornos de aficionados, académicos y comercial para contribuir en la investigación de comunicaciones inalámbricas y sistemas de radio del mundo real [CHAV05].

GNU Radio está licenciado bajo la GNU General Public License (GPL) versión 3. Todo el código es propiedad de la Fundación del Software Libre.

Las aplicaciones de GNU Radio inicialmente se escribieron utilizando el lenguaje de programación Python, mientras que los bloques críticos de procesamiento de señales que requieren alta velocidad de proceso se implementan en C++.

Se pueden generar de forma simple, usando bloques constructivos, radios de alto rendimiento que funcionan en tiempo real. Para ello, se usan las funciones desarrolladas en GNU Radio, siendo éstas parametrizables. El uso de variables, gráficas, menús y opciones gráficas permiten un control en línea de los parámetros definidos así como una presentación en tiempo real de los resultados.

Aunque no es una herramienta principalmente de simulación, GNU Radio implementa el desarrollo de algoritmos de procesamiento de señales a partir de datos previamente grabados o generados, evitando la necesidad de hardware de RF.

Es de destacar que las últimas versiones de GNU Radio permiten ya el uso de dispositivos de recepción de bajo coste como puede ser el RTL-SDR, lo cual evita cuando no sea necesario, la inversión en la tarjeta USRP, desarrollada por Ettus Research y que tiene un costo aproximado de 3000 euros.

2.3.7.1.1 Qué hace GNU Radio.

GNU Radio realiza todo el procesamiento de la señal.

Se puede usar para escribir aplicaciones que procesen cadenas digitales de datos, tanto para transmitir o recibir, estos se transmiten usando el hardware adecuado.

GNU radio tiene filtros, códigos de canal, elementos de sincronización, ecualizadores, demoduladores, decodificadores y muchos otros elementos llamados bloques, que se pueden encontrar en los típicos sistemas de radio.

Además se implementa un método de conectar los bloques y manejar la forma en que los datos se pasan de un bloque a otro.

Como GNU radio está basado en software, sólo puede manejar datos digitales, usualmente se usa muestreo en banda base para la recepción y la transmisión.

Se usa hardware analógico para desplazar (mezclar) la señal a la frecuencia deseada. Cualquier tipo de datos se puede pasar de un bloque a otro.

2.3.7.1.2 Formas de instalar GNU Radio.

Hay dos formas de instalar GNU Radio:

- Usando paquetes binarios precompilados.
- Compilando los códigos fuente en el ordenador de instalación.

Si queremos obtener alguna de las siguientes opciones:

- Tener el código más actualizado
- Seguir de forma cercana el desarrollo de GNU radio
- Modificar GNU radio por uno mismo.

Es necesario descargar las fuentes desde su repositorio e instalarlos.

Normalmente hay scripts (archivos de comandos) para descargar las Fuentes y ejecutar las acciones más complicadas.

A continuación, se detalla una secuencia normal de descarga e instalación de GNU Radio. Se ha de comentar que como software de código libre las secuencias de instalación pueden variar de una distribución a otra. Además, se requiere una correcta configuración del sistema operativo a usar. Se encuentra en anexo la salida de la instalación del sistema GNU Radio Marconi. En él se compilan las fuentes y se instalan las dependencias y librerías necesarias para una correcta instalación del GNU Radio.

Secuencias de comandos GNU Radio:

El proceso manual para instalar GNU Radio sería:

Abrir Terminal.

Ir al directorio de instalación.

Ejecutar los siguientes comandos.

```
$ wget http://www.sbrac.org/files/build-gnuradio && chmod a+x ./build-gnuradio && ./build-gnuradio
```

Este proceso descarga el instalador y lo hace ejecutable.

Descarga e instala todas las dependencias.

Descarga UHD y GNU radio desde Git.

A continuación se ejecuta el make e instala la aplicación.

El código fuente permanece en el ordenador para modificaciones.

Usando binarios precompilados:

Éstos no suelen tener las últimas modificaciones recogidas en las fuentes Git.

Para instalar paquetes binarios en Ubuntu, se usa la siguiente instrucción:

```
$ apt-get install gnuradio
```

2.3.7.1.3 GNU.

El proyecto GNU se inició en 1983 como una forma de promover el espíritu de cooperación que prevalecía en la comunidad informática en aquellas fechas, es decir, hacer la cooperación posible al eliminar los obstáculos impuestos por los propietarios de software de pago.

En 1971, cuando Richard Stallman comenzó su carrera en el MIT (Instituto de Tecnología de Massachusetts), trabajó en un grupo que usaba exclusivamente software libre. Incluso algunas compañías informáticas distribuían frecuentemente software libre. Los programadores eran libres de cooperar unos con otros, y frecuentemente lo hacían [CHAV05]. En los 80, casi todo el software era privado, lo cual significaba que tenían propietarios que prohibían e impedían la cooperación entre usuarios. Esto hizo necesario el Proyecto GNU.

Cada usuario de ordenador necesita un sistema operativo; si no existe un sistema operativo libre, no se puede ni siquiera comenzar a usar un ordenador sin recurrir a un software de pago.

Así que el primer elemento en la agenda del software libre era crear un sistema operativo libre.

Un sistema operativo no es sólo el núcleo; sino que también tiene las aplicaciones que incluyen compiladores, editores, procesadores de texto, software de correo y muchas otras aplicaciones. Por todo ello, escribir un sistema operativo con aplicaciones preinstaladas es un trabajo complicado.

2.3.7.1.4 GNU/LINUX.

Al completarse el objetivo inicial de un sistema operativo libre parecido al Unix, ya se tenían escritos los componentes principales, excepto uno: el núcleo.

Linus Torvalds desarrolló el núcleo del sistema operativo, Linux.

Combinando Linux con el ya casi completo sistema GNU se consiguió un sistema operativo completo: un sistema GNU basado en Linux.

Se estima que hay cientos de miles de personas que ahora usan sistemas GNU basados en Linux, incluyendo distribuciones como Debian, Red Hat, Centos, Ubuntu y otros.

Existe una gran confusión entre un sistema operativo Linux y un núcleo Linux. Linux no es un sistema operativo, es el núcleo del sistema operativo, este núcleo se encarga de asignarle al ordenador los recursos para ejecutar los programas. El sistema operativo por lo tanto, en este caso, se debe llamar GNU/Linux y no solamente Linux.

2.3.7.1.5 Distribuciones Linux.

Una distribución no es otra cosa que el núcleo Linux con un conjunto de programas seleccionados, con herramientas específicas de configuración, empaquetamiento, documentación, etc.

Prácticamente todas las distribuciones son comerciales, salvo Debian, Hay muchísimas distribuciones alrededor del mundo, pero sólo unas pocas se usan ampliamente, estas distribuciones son:

Fedora: Ésta es la mejor opción para principiantes, es uno de los sistemas que más se ha esforzado en llegar al usuario. Algunas versiones traen una utilidad llamada “supermount”, la cual permite montar y desmontar, disquetes y CDs según se necesite en forma automática. Fedora está basada en Red Hat.

La desventaja es que requiere un hardware más actualizado que otras distribuciones.

SuSE: Distribución que se caracteriza por contar con gran cantidad de software y documentación. Es bastante amigable y posee herramientas de configuración general llamadas YaST y YaST2 que están muy bien logradas. Hay que tener en cuenta que no sigue el modelo de RedHat y algunos comandos y ubicaciones de archivos son algo diferentes a otros sistemas Linux.

RedHat: De alguna manera es la primera distribución de Linux que logró un tipo de instalación más sencilla en su época. La empresa RedHat es una de las más grandes dentro del mercado del software libre. La distribución posee una herramienta de configuración que facilita las tareas del sistema llamada “Linuxconf”. Muchas distribuciones se basan en ella.

Slackware: Esta distribución es la más parecida al Sistema Operativo Unix. En Slackware todo o casi todo está para ser configurado mediante la edición de archivos de texto o por medio de aplicaciones para terminales. Si bien es algo compleja de instalar para el usuario nuevo, para los expertos es una excelente opción ya que es sumamente flexible. Trae un sistema de instalación basado en paquetes .tar.gz el cual si bien es más personalizable no posee un sistema de chequeo de dependencias. De todas formas, como todas las distribuciones cuentan con entornos gráficos que reducen considerablemente la dificultad para operarla.

Debian (Usada dentro del proyecto Marconi / UCA para el servidor de Teamspeak): Considerada la distribución más dura para instalar y configurar. Posee su propio sistema de instalación de paquetes en formato “.deb”. Para muchos es la distribución más robusta, ideal para servidores. Debian se adhiere fervorosamente a la causa del software libre, tal es así que los paquetes que no tengan la licencia GNU GPL se ponen en una categoría totalmente separada del resto. Las versiones de los programas se examinan rigurosamente, y cuentan con las actualizaciones más estables y no precisamente las más nuevas.

Ubuntu (Usada dentro del proyecto Marconi / UCA para servidores SDR así como para el servidor de GNU Radio y FLdigi): Es un sistema operativo basado completamente en Linux, libre. La comunidad Ubuntu está construida sobre los principios del Manifiesto Ubuntu. Éste dice que el software debe estar disponible gratuitamente y sus herramientas deben ser utilizables por cualquier persona, independientemente de su lengua nativa y de cualquier tipo de discapacidad física que pueda tener. Estas personas deben tener la libertad de modificar el software de acuerdo a sus necesidades particulares cualesquiera que sean éstas.

Esta libertad es fundamental para hacer de Ubuntu un software diferente de cualquier software privativo. No sólo las herramientas necesarias son gratuitas, sino también

tenemos los derechos para modificarlas de forma que cumplan con nuestras necesidades.

Ubuntu incluye más de 16000 piezas de software, aunque el sistema operativo cabe en un solo CD. Ubuntu cubre cada una de las aplicaciones de escritorio, desde procesadores de texto y hojas de cálculo, hasta aplicaciones de acceso a Internet, software para servidores web, correo electrónico, lenguajes de programación, además de algunos juegos.

Ubuntu introduce en cada uno de sus lanzamientos la última versión de Kernel y Gnome.

CentOS (Usada dentro del proyecto Marconi / UCA para el servidor de PBX/Tribos y el Firewall IPCop):

CentOS (Community ENTERprise Operating System) es una bifurcación a nivel binario de la distribución Linux “Red Hat Enterprise Linux” (RHEL), compilado por voluntarios a partir del código fuente liberado por Red Hat.

Red Hat Enterprise Linux se compone de software libre y código abierto, pero se publica en formato binario (CD-ROM o DVD-ROM) solamente para los suscriptores de pago. Como es requerido, Red Hat libera todo el código fuente del producto de forma pública bajo los términos de la Licencia pública general de GNU. Los desarrolladores de CentOS usan ese código fuente para crear un producto final que es muy similar al “Red Hat Enterprise Linux” y está libremente disponible para ser descargado y usado por el público.

2.3.7.1.6 GNU Radio.

GNU Radio es un conjunto de archivos y aplicaciones que proveen las librerías necesarias de procesamiento digital de señales para poder manipular señales de radio. GNU Radio corre sobre sistemas GNU/Linux como Ubuntu, el cual debe estar previamente instalado

El programador de GNU Radio necesita, para construir un sistema de radio, crear un grafo, donde los vértices son bloques de procesamiento de señales, y los bordes representan el flujo de datos entre ellos.

Los bloques de procesamiento de señal están implementados en C++. Conceptualmente un bloque procesa señales continuamente desde puertos de entrada hasta puertos de salida. Las partes de un bloque son: el número de puertos de entrada, el número de puertos de salida, y el tipo de datos que fluirán de uno al otro. Los tipos de datos más comúnmente utilizados son: “short”, “float” y “complex”.

Algunos bloques tienen únicamente puertos de salida o puertos de entrada. Éstos sirven como fuente de datos y señales en una gráfica. Existen fuentes que leen datos de un archivo o del ADC, y señales que escriben datos a un archivo, al DAC o a un display gráfico. GNU Radio cuenta con aproximadamente 100 bloques.

Las gráficas para GNU Radio se ejecutan y están construidas en Python.

Actualmente, Python se desarrolla como un proyecto de código abierto, administrado por la Python Software Foundation. Contiene módulos, clases, tipos de datos de alto nivel y escritura dinámica. Tiene interfaces para diversos sistemas y librerías. También puede utilizarse como un lenguaje de extensión para aplicaciones que necesitan una interfaz programable. Otra ventaja es su portabilidad, funcionando en sistemas Unix y derivados, Windows, Dos, Mac y otros.

Hay factores que distinguen a Python de otros lenguajes de programación orientados a objetos:

- 1) No hay necesidad de compilar código en Python antes de ejecutarlo, por lo que se convierte en un lenguaje de comandos (interpretado).
- 2) El lenguaje busca tener un código más fácil de utilizar (y son reutilizables para otras aplicaciones).

2.3.7.1.7 Definiciones generales.

ADCs: Un convertidor analógico a digital tiene tres características primarias:

- Frecuencia de muestreo (Sample Rate)
- Rango dinámico (Dynamic Range)
- Tiempo de conversión

Frecuencia de muestreo es el número de veces por segundo que el ADC mide una señal analógica.

Rango dinámico se refiere a la diferencia entre la señal más pequeña y la más grande que puede ser distinguida por el ADC, está relacionada con el número de bits de salida del ADC.

Tiempo de conversión se refiere al tiempo (T_c) que necesita el ADC para obtener un dato digital a partir de recibir un dato analógico con el fin de que al existir una variación en la entrada no surja una confusión en la conversión.

Teoría de Nyquist: La teoría de Nyquist determina que para evitar el efecto de aliasing cuando convertimos de analógico a digital, la frecuencia de muestreo del ADC debe ser de, al menos, dos veces el ancho de banda de la banda de interés.

RF front end: Un RF front end es la configuración hardware del receptor, que provoca el desplazamiento de las frecuencias a decodificar.

2.3.7.2 La tarjeta USRP

El Universal Software Radio Peripheral es un periférico diseñado para trabajar en conjunto con una computadora a través de un FPGA y permite la realización de SDR.

Este periférico realiza las funciones de llevar la señal de RF a banda base a través de la sección de IF, y viceversa, para un sistema común de radiocomunicaciones, como se muestra en la Figura 2.3.7

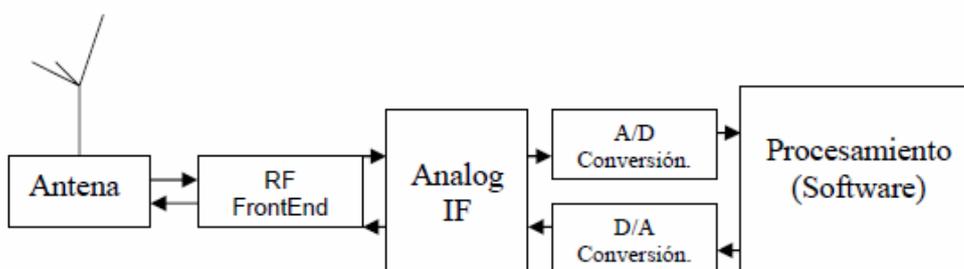


Figura 2.3.7 Bloques principales en este sistema de comunicaciones de Software Radio

Características de la Tarjeta USRP:

La tarjeta USRP cuenta con 2 niveles de tarjetas como se muestra en la Figura 2.3.8 El primero es la tarjeta base en donde se encuentran el FPGA, los convertidores ADC's y DAC's, la alimentación y la conexión vía USB. El segundo nivel se compone de tarjetas secundarias. Estas tarjetas sirven para la transmisión y recepción y son las que trabajan como RF front-end y llevan la señal hasta la banda de RF deseada.

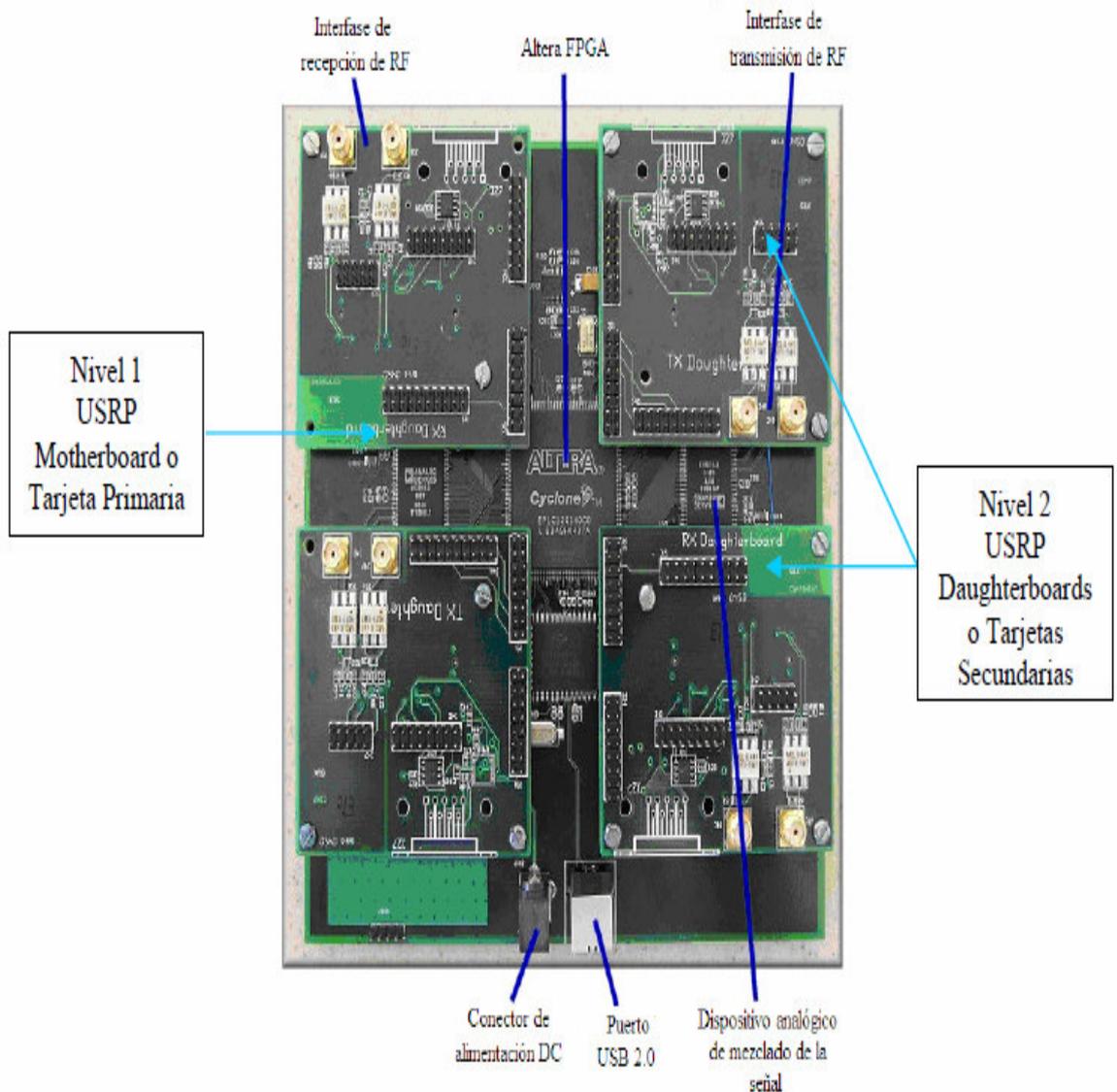


Figura 2.3.8 Universal Software Radio Peripheral

La Figura 2.3.9 muestra un diagrama de bloques del Universal Software Radio Peripheral, donde podemos ver la disposición de los convertidores, el procesador y la interfaz USB.

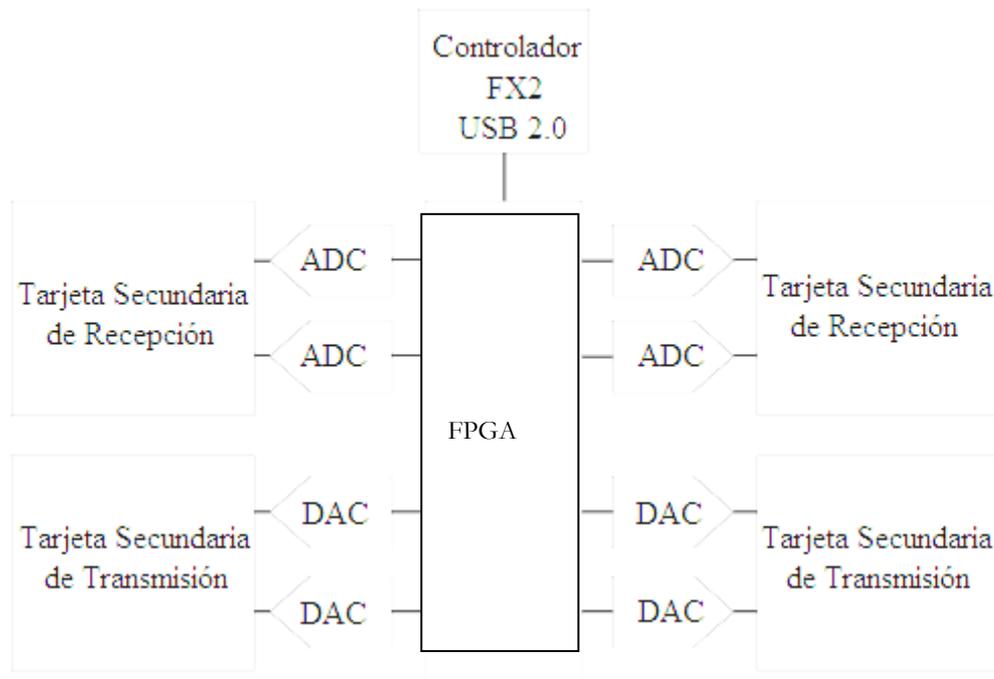


Figura 2.3.9 Diagrama de bloques del Universal Software Radio Peripheral

Primer nivel: convertidores, procesador e interfaz.

Convertidores AD

La tarjeta USRP tiene 4 convertidores ADC de alta velocidad, cada uno a 12 bits, y 64 millones de muestras por segundo con lo que se puede digitalizar una banda de 32MHz.

El rango completo del ADC es de 2V pico a pico y tiene una entrada diferencial de 50Ω , lo que significa 16dBm o 40mW. Existe un amplificador de potencia programable (PGA) antes del ADC para amplificar la señal de entrada y utilizar el rango completo en el caso que la señal sea débil.

Convertidores DA

La tarjeta USRP tiene 4 convertidores DAC de alta velocidad para transmisión, cada uno a 14 bits y 128 millones de muestras por segundo, por lo que la frecuencia de Nyquist es de 64MHz.

Los DAC's pueden suministrar 1V pico a pico a una carga diferencial de 50Ω , esto es 10dBm o 10mW. Se cuenta también con un PGA conectado después del DAC para aumentar la ganancia a 20dB o más.

Los PGA utilizados para transmisión y recepción son programables.

Procesador e Interfaces

El procesador utilizado por la tarjeta USRP es un FPGA Altera Cyclone EP1C12 .

Comprender el funcionamiento del FPGA es parte importante para utilizar GNU Radio. Los ADC y DAC's están conectados al FPGA. La tarea de éste es realizar procesos matemáticos de las señales en la banda que necesitamos, y reducir las tasas de muestreo de datos para enviar datos a través de una interfaz USB 2.0.

EL FPGA se conecta a un chip de interfaz USB 2.0, este chip es el Ciprés FX2. La configuración estándar del FPGA incluye 4 convertidores digitales de bajada (DDC), esto permite 1, 2 o 4 canales independientes de recepción, cada DDC tiene dos entradas I (en fase) y Q (en cuadratura). Cada ADC puede ser ruteado a cualquiera de las entradas I y Q de los 4 DDC.

Para la transmisión hay convertidores digitales de subida (DUC) contenidos fuera del FPGA en el chip AD9862 CODEC .

Los canales múltiples para la recepción deben tener todos la misma velocidad de transmisión de datos (data rate), lo mismo para los canales de transmisión que deben tener la misma velocidad, la cual debe ser distinta de la recepción [CHAV05].

La figura 2.3.10 muestra un diagrama a bloques del camino de recepción desde la antena hasta el DDC.

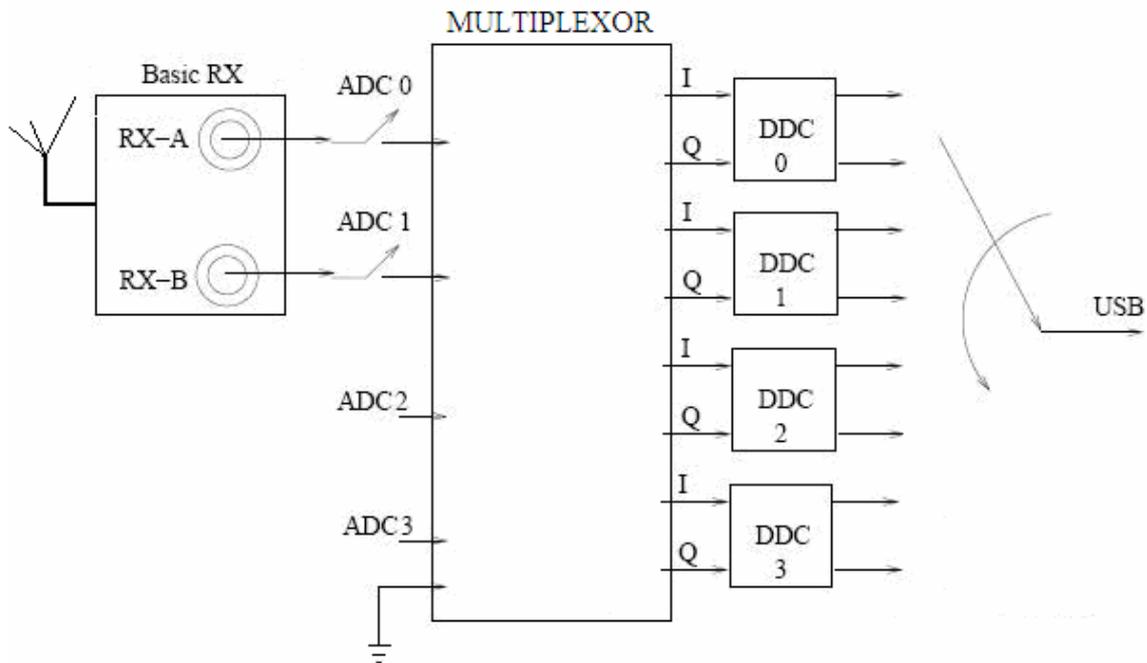


Figura 2.3.10 Multiplexor en el USRP, donde I es la señal en fase y Q la señal en cuadratura.

El multiplexor (MUX) en la Figura 2.3.10 es un circuito seleccionador, que determina que ADC se conecta a cada DDC.

Para la transmisión todo sucede de forma similar, con la diferencia de que el proceso es a la inversa.

Enviamos una señal I y Q en banda base de la computadora al USRP. El convertidor digital de subida (DUC) interpola la señal convirtiéndola a una frecuencia superior para la banda de IF y por último la envía a través del DAC.

Segundo nivel: tarjetas secundarias (antena, sección de RF e IF).

Tarjetas Secundarias:

Estas tarjetas se muestran en la Figura 2.3.11 y realizan la tarea de la antena, la sección de RF y la sección de IF.

Se conectan al primer nivel que cuenta con 4 entradas, 2 para recepción y 2 para transmisión. Para la recepción se utiliza la Tarjeta Secundaria llamada Basic RX y para la transmisión la Basic TX.

Existen 2 ranuras para tarjetas de transmisión etiquetadas como TXA y TXB y sus dos correspondientes para recepción etiquetadas también RXA y RXB. Cada ranura para tarjeta tiene acceso a 2 de los 4 convertidores de datos de alta velocidad (Salidas de DACs para TX y entradas de ADC para RX).

Esto permite a cada tarjeta tener dos secciones independientes de RF y dos antenas (4 para todo el sistema). Si utilizamos un muestreo en cuadratura (IQ) de la señal analógica, cada tarjeta puede soportar una sola sección de RF, con un total de dos para el sistema completo.

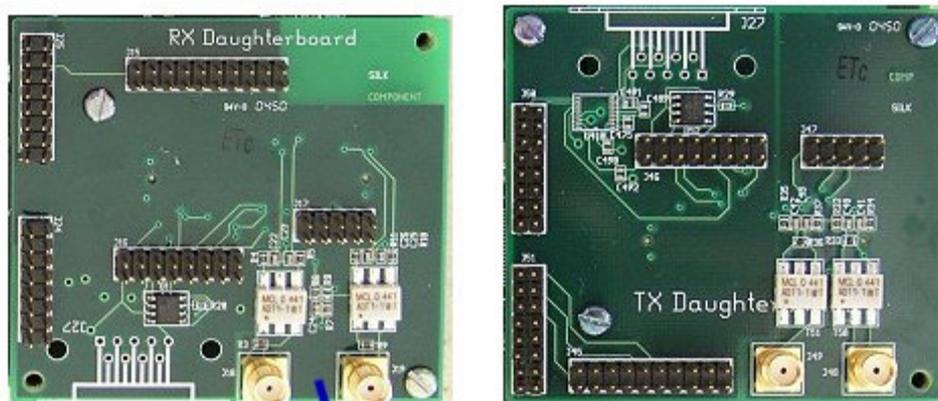


Figura 2.3.11 Imagen de las tarjetas secundarias Basic RX y Basic TX

Instalación de la tarjeta USRP

La instalación de la tarjeta USRP es bastante sencilla, es necesario haber instalado GNU Radio, y hacer la conexión correcta de la tarjeta, por lo que sólo se necesita conectar el USRP a la corriente y con el cable USB al ordenador.

2.3.8 Arquitectura open HPSDR

2.3.8.1 Introducción.

HPSDR (High Performance Software Defined Radio, Radio Definido por software de altas prestaciones) es un proyecto para la creación de una nueva generación de equipos de radio de HF de tecnología SDR para su uso por los radioaficionados y los radioescuchas de la Onda Corta (SWL's).

Los equipos HPSDR son de característica modular, y su desarrollo incluye tanto la parte física o hardware, como el software o programas que lo hacen funcionar. El software es de licencia "open source", esto es, software de código abierto, y por lo tanto, el código de los programas es de conocimiento público, y de libre uso incluso para su estudio y modificación por el usuario.



Figura 2.3.12 Logotipo de los equipos compatibles con HPSDR

El desarrollo del proyecto HPSDR lo iniciaron un grupo de entusiastas norteamericanos de la entonces nueva tecnología SDR, entre los cuales se pueden citar como miembros más destacados a los radioaficionados norteamericanos Ray Anderson WB6TPU, Steve Bible N7HPR, Phil Covington N8VB, Rick Hambly W2GPS, Phil Harman VK6APH, Lyle Johnson KK7P, Ulrich Rohde N1UL, y Bill Tracey KD5TFD.

El proyecto está abierto a todo aquel de todo el mundo que quiera colaborar, y comenzó a debatirse en foros de discusión de Internet en marzo de 2006.

Muy pronto surgieron las primeras ideas prácticas y el 20 de mayo de 2006 el proyecto fue presentado por Johnson KK7P ante una gran audiencia en la Convención anual de radioaficionados norteamericanos de Dayton del 2006.

El desarrollo hardware de un equipo HPSDR es de tipo modular.

El equipo estaría constituido por una serie de módulos, cada uno de los cuales está diseñado para realizar una función determinada, y la interconexión entre los distintos módulos del equipo se realiza a través de un bus de datos común soportado sobre una placa base, la cual dispone de los conectores necesarios para enchufar los distintos módulos.

Físicamente, esto es bastante similar a la estructura de un moderno ordenador PC, donde hay una placa base o "motherboard", con una serie de conectores a los cuales se enchufan distintas tarjetas de ampliación de funciones del sistema (tarjeta de vídeo, de sonido, módem telefónico, tarjeta de red, etc).

2.3.8.2 Módulos.

Esta estructura modular permite que el usuario incorpore en su equipo HPSDR los módulos que le sean de interés, y a los desarrolladores, les permite desarrollar variantes de cada módulo buscando mejorar las características y prestaciones de éstos, y que pueden reemplazar a los ya existentes.

Los módulos pueden variar en complejidad, desde sencillos módulos de filtro pasobanda o de interfaz I/O (entrada/salida), hasta módulos mucho más complejos, como módulos de funciones DSP.

Los módulos han sido denominados cada uno con un nombre para su fácil identificación cuando se habla o escribe acerca de ellos.

Algunos módulos han sido diseñados para ser utilizados conjuntamente con otros para realizar su función, mientras que otros tienen un funcionamiento más autónomo.

Cada módulo está realizado en una placa impresa rectangular de tamaño 100 mm por 120 a 220 mm, a excepción de la placa base donde se conectan.

Los conectores empleados para los módulos y la placa base son conectores de tipo DIN41612 de 64 ó 96 pines. Cada uno de ellos está programado con el firmware (software interno) necesario para su funcionamiento, firmware que está almacenado en uno de los circuitos integrados del módulo.

El desarrollo lo está haciendo el grupo HPSDR, cuyo sitio web tiene una página dedicada a cada módulo desarrollado o en desarrollo.

El esquema de bloques de un equipo HPSDR básico consta de 4 módulos, además de la placa base de soporte e interconexión de los módulos. En el proyecto HPSDR hay desarrollados más módulos, así como otros que, externos al propio equipo HPSDR, complementan a éstos.

Los módulos en desarrollo se describen a continuación.

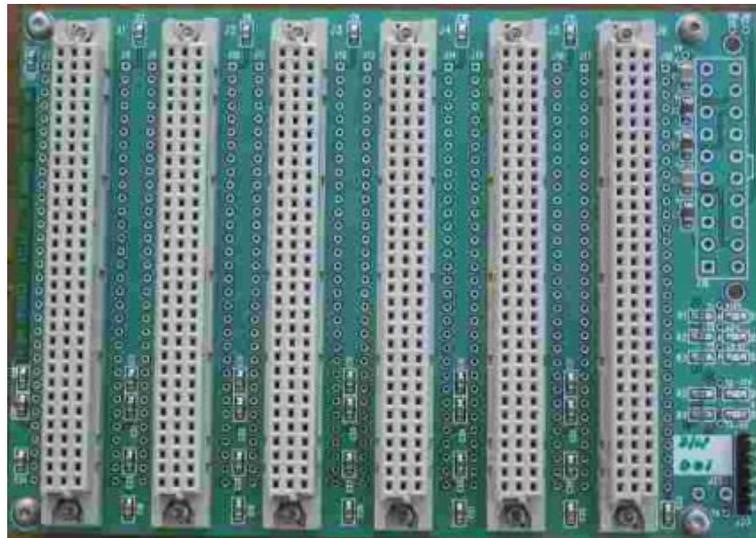


Figura 2.3.13 Placa Atlas, versión 1ª

2.3.8.2.1 Módulo ATLAS.

Placa base pasiva que incluye hasta 6 conectores DIN41612 para conectar los módulos del sistema y un conector de alimentación (alimentación de 12, 5 y 3,3 V proporcionada por una fuente de alimentación externa de PC tipo ATX).

La placa contiene un bus de señales pasivo que pasa por todos los conectores, para proporcionar la interconexión entre todos ellos.

El líder de este proyecto es Phil Covington (N8VB).

2.3.8.2.2 Módulo OZYMANDIAS (OZY).

Módulo de interfaz con el exterior, basado en un chip FPGA de la firma Altera.

Proporciona líneas de entrada y salida con sus correspondientes conectores, para conexión del sistema HPSDR con el mundo exterior.

Un interfaz serie (RS232), un interfaz paralelo, y un interfaz USB 2.0 de alta velocidad para conexión a un PC de control, que puede ser empleado para procesar las señales muestreadas por el módulo Mercury (receptor) si no se utiliza el módulo Sasquatch (módulo DSP).

El líder de este proyecto es Phil Covington (N8VB).

2.3.8.2.3 Módulo MERCURY.

Módulo de muestreo de gran ancho de banda para la etapa frontal del receptor, capaz de muestrear directamente el espectro de frecuencias comprendido entre 0 y 65 MHz.

Utiliza un convertidor A/D de 16 bits LTC2208 de Linear Technology, e incorpora su propia FPGA para generar una señal digital de hasta 250 Kbps a partir de las muestras proporcionadas por el convertidor A/D, que es enviada a través del bus de la placa Atlas al interfaz USB del módulo OZY para su procesamiento por un PC externo.

El líder de este proyecto es Phil Harman (VK6APH).

2.3.8.2.4 Módulo SASQUATCH.

Módulo DSP completo para el procesamiento de las señales I,Q procedentes del módulo Mercury (receptor) o para generar las señales I,Q para el módulo Penélope (transmisor), que evita el uso de un PC para realizar esta función. Ha sido desarrollado por Lyle Johnson (KK7P).

2.3.8.2.5 Módulo PENELOPE.

Placa excitadora o transmisora de baja potencia para HF, de 0,5 W de potencia de salida. Procesa las señales I,Q procedentes del PC o de la placa Sasquatch (módulo DSP del sistema) para generar la señal a transmitir.

El líder de este proyecto es Phil Harman (VK6APH).

2.3.8.2.6 Módulo JANUS.

Módulo convertidor A/D y D/A de funcionamiento full-dúplex, y cuyas características son similares a la de los convertidores A/D y D/A de las mejores tarjetas de sonido para ordenador.

El líder de este proyecto es Phil Harman (VK6APH) y Bill Tracey (KD5TFD).



Figura 2.3.14 Placa Janus. Prototipo versión alfa 2

2.3.8.2.7 Módulo PINOCCHIO.

Placa de extensión de la placa base Atlas que permite la conexión de un módulo que se quiera probar y hacer medidas en él.

Es una placa pasiva de tamaño mayor a la de los demás módulos, que prolonga el bus de la placa Atlas, y que incluye puntos de medida de cada señal del bus en su parte superior (a una altura superior a la del resto de módulos del sistema, para que los puntos de medida sean accesibles), y que incluye en su extremo superior un conector para enchufar el módulo a probar.

El líder de este proyecto es Lyle Johnson (KK7P).



Figura 2.3.15 Conjunto de placas Atlas, Janus (en primer plano) y Pinocchio (al fondo)

2.3.8.2.8 Módulo EPIMETHEUS (EPI).

Placa de entrada/salida de propósito general que se conecta a la placa Atlas.

Incluye entradas-salidas actuadas por relés miniatura, un conector cruzado digital de alta velocidad, entradas digitales opto aisladas y salidas digitales en colector abierto. Un chip CPLD conecta entradas y salidas al bus de datos de la placa Atlas.

El líder de este proyecto es Philip Harman (VK6APH).

2.3.8.2.9 Módulo ALEXIARES (ALEX).

Placa con un conjunto de filtros paso banda de entrada para su uso con el módulo Mercury o con cualquier otro receptor SDR.

El líder de este proyecto es G. Graham (KE9H).

2.3.8.2.10 Módulo PANDORA.

Proyecto de caja destinada a albergar todo el conjunto HPSDR. Alberga la placa base Atlas, las tarjetas enchufadas a ésta, un par de ventiladores de refrigeración y en una de sus caras, están dispuestas ventanas alargadas para que asomen los conectores de los distintos módulos del sistema (similares a las cajas de los ordenadores PC, por las que asoman los conectores de las placas de expansión del sistema). Ha sido desarrollado por Ben Hall (KD5BYB).

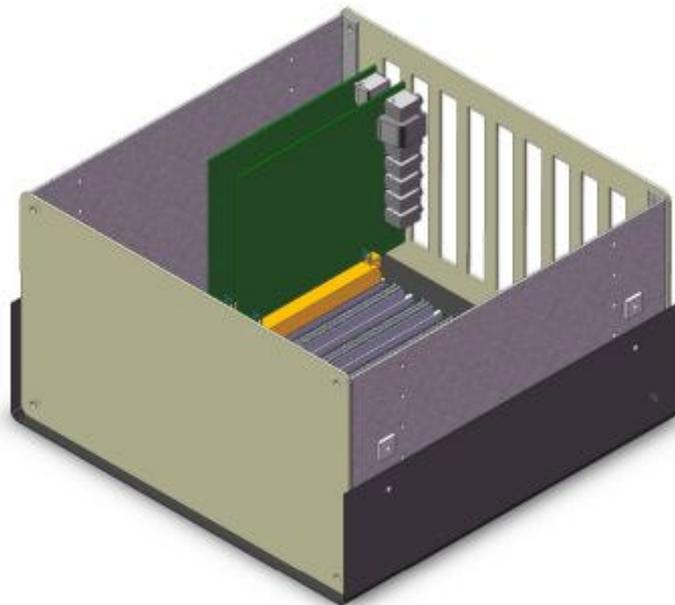


Figura 2.3.16 Caja Pandora modelo BK959. Aloja las placas Atlas, Ozy, y Janus

2.3.8.2.11 Módulo DEMETER.

Fuente o unidad de alimentación externa especialmente pensada para el proyecto HPSDR y desarrollada por Jeroen Bastemeijer (PE1RGE).

Proporciona una serie de tensiones de alimentación al sistema HPSDR (+5, +12 y -12 V principalmente) a partir de la tensión alterna de red (110 ó 230 Volts) o de una batería o fuente de alimentación de 13,8 Voltios.

2.3.8.2.12 Módulo PROTEUS.

Tarjeta de prueba de prototipos destinada para experimentadores que deseen probar cualquier aplicación no recogida por los módulos del proyecto HPSDR.

El líder de este proyecto es Phil Covington (N8VB).

2.3.8.2.13 Módulo THOR.

Amplificador de potencia eficiente de HF que utiliza las técnicas ERR (Envelope Elimination and Restoration). Ha sido desarrollado por Chuck Clark (AF5Z)

Es alimentado por la salida de señal del módulo de baja potencia Penélope. Es, pues, un módulo externo al equipo HPSDR.

2.3.8.2.14 Módulo GIBRALTAR.

Módulo generador de frecuencia estándar precisa de 10000 MHz y 1 Hz para el proyecto HPSDR a partir de las señales GPS captadas por un módulo receptor GPS con toma de antena exterior incluida en su circuitería. Ha sido desarrollado por Rick Hambly (W2GPS) y Steve Bible (N7HPR).

Las señales de 10000 MHz y de 1 Hz se envían a través de las líneas correspondientes del bus Atlas a los módulos que las necesitan, así como a conectores en el módulo, desde donde se pueden tomar las señales.

2.3.8.2.15 Módulo PHOENIX.

Módulo que incluye un receptor HF basado en un ISD (Integrating Sampling Detector), un excitador HF basado en un QSE (Quadrature Sampling Exciter), y un sintetizador.

El líder de este proyecto es Ray Anderson (WB6TPU).

2.3.8.2.16 Módulo CYCLOPS.

Módulo analizador de espectro 0-1 GHz y generador de frecuencia de seguimiento (Tracking Generator). Basado en el módulo Mercury.

El líder de este proyecto es Phil Harman (VK6APH).

2.3.8.3 Desarrollo del proyecto HPSDR.

El proyecto HPSDR (High Performance SDR) se comenzó en octubre del 2005 por Phil Covington [JOHN06]. En noviembre de 2005 se formó otro pequeño grupo basado en la tarjeta Xylo (Tarjeta FPGA). El 1 de marzo del 2006 se unieron los dos grupos en el grupo HPSDR.

Este grupo tiene miembros de diferentes áreas (hardware, software, firmware,..) que participan en el desarrollo de la aplicación HPSDR.

Actualmente están en desarrollo módulos como filtros paso banda, un convertidor I/Q de banda estrecha para complementar al convertidor A/D Mercury, y un módulo de transmisión de baja potencia, entre otros.

Como ejemplo de todo esto se muestra a continuación el transceptor HPSDR que realizó el radioaficionado australiano Phil Harman (VK6APH) en 2007, con el que realizó sus primeros contactos.

Los módulos que lo constituyen son, de izquierda a derecha (fig. 2.3.17), un prototipo Mercury constituido por una placa Ozy y una placa de evaluación del chip LT2208 (chip convertidor A/D de 16 bits), una placa Penélope (con su conector de RF), y una placa Ozy, montadas sobre una placa de bus Atlas.



Figura 2.3.17 Transceptor HPSDR de Phil Harman, VK6APH, año 2007

2.3.9 Arquitectura WEBSDR.

2.3.9.1 Descripción general.

WebSDR es un proyecto liderado por el Dr. Pieter-Tjerk de Boer [BOER13] del departamento de “Diseño y Análisis de los Sistemas de Comunicación” (Facultad de Ingeniería Eléctrica, Matemáticas y Ciencias de los Computadores, Universidad de Twente) en colaboración con el Radio Club de esta Universidad.

WebSDR consiste en varios receptores Radio definidos por software conectados a Internet, permitiendo a muchos usuarios receptores escuchar y sintonizar simultáneamente independientemente de su ubicación.

Evidentemente el ancho de banda entregado es aquél que permite la tarjeta de sonido del PC servidor.

La sintonización por cada uno de los usuarios está limitada por el ancho de banda provisto por la tarjeta de sonido del PC Servidor (48, 96, 192 KHz).

Esta característica de permitir la sintonización individual y simultánea por los usuarios, es la diferencia principal del SDR con los receptores clásicos que se pueden usar hoy en Internet.

WebSDR se desarrolló como una forma de hacer accesible el radiotelescopio de 25 m en Dwingeloo para poder ser usado por muchos radioaficionados para la recepción de comunicaciones Tierra-Luna-Tierra.

Para probar una versión preliminar del software, sin usar el plato de 25 metros, se hizo una versión reducida del software en el radio club de la universidad de Twente, durante la navidad del 2007.

Después de varios ensayos, se publicaron los resultados en Abril del 2008.

El interés por el proyecto ha crecido desde entonces, y muchos grupos de aficionados y universidades han mostrado interés por montar un servidor Websdr.

En noviembre de 2008 se empezaron los ensayos con unas pocas estaciones.

La Universidad de Cádiz ha instalado dos receptores SDR dentro del proyecto WebSDR como uno de los objetivos de esta Tesis Doctoral (Marconi.uca.es:8901).

Básicamente un servidor WebSDR consta de un PC con Linux. Este sistema operativo se escoge por su buena capacidad de multiproceso, la facilidad de instalar el software requerido, sin tener que cargar gran cantidad de paquetes adicionales, así como la gran integración de Linux dentro de entornos tcp-udp/ip.

El PC ha de tener instalado una distribución Linux (Ubuntu), así como por motivos de administración, debe tener instalado Java y Navegador de Internet.

La conexión a Internet debe ser rápida, ya que se necesita alrededor 100 Kbit/s de subida por usuario.

La aplicación WebSDR se distribuye de forma individual, en un fichero zip comprimido, por solicitud a su desarrollador Peter de Boer, al que se le debe explicar el uso que se le va a dar.

El software SDR básicamente trata el ancho de banda recibido a través de la tarjeta de sonido del PC, alcanzándose actualmente anchos de banda de 192 KHz.

La señal de entrada va a través de la entrada estéreo de la tarjeta de sonido.

Por un canal se recibe la señal I y por el otro la señal Q, siendo estas señales comunes al tratamiento de todos los equipos SDR.

Los equipos que se han usado para la entrada de sonido son:

- Softrock (40 m).
- RTL-SDR.

El equipo RTL-SDR tienen interfaz USB y para su uso hay que compilar su interfaz, mientras que los equipos Softrock, hacen su entrada a través de la línea estéreo de la tarjeta de sonido del PC .

A continuación, se muestra la pantalla típica de equipos wedsdr.org donde se listan los equipos que están usando el software WebSDR (Figura 2.3.18):

A WebSDR is a Software-Defined Radio receiver connected to the internet, allowing many listeners to listen and tune it simultaneously. SDR technology makes it possible that all listeners tune independently, and thus listen to *different* signals; this is in contrast to the many classical receivers that are already available via the internet.

More background information is available [here](#) and frequently asked questions are answered [here](#). Questions and comments can be sent to [PA3FWM](#), the author of the WebSDR software and maintainer of this site.

WebSDR servers can register themselves automatically on this site, leading to the below list of currently active WebSDR servers.

Currently there are 55 servers active, with 187 users and 76010 kHz of radio spectrum.

Filter band: and region:

Location and URL	Frequency range	Antenna
 WebSDR at the University of Twente, Enschede, NL http://websdr.ewi.utwente.nl:8901/ JO32KF; 88 users	0.000 - 29.160 MHz	Mini-Whip
 W4AX Softrock 160, 80, 40 and 20M. Atlanta, GA, USA http://W4AX.com/ EM74uc; 26 users	1.799 - 1.991 MHz	Inverted L at 100 ft.
	3.549 - 3.741 MHz	Clifton Labs Active Antenna and Norton Preamp
	3.744 - 3.936 MHz	
	7.018 - 7.210 MHz	
 160m, 60m, 80m and 40m SDR's from Nantwich in Cheshire http://hackgreen.no-ip.org:8901/ IO63RA; 11 users	14.046 - 14.238 MHz	204 foot long double size G5RV
	1.852 - 1.948 MHz	
	3.600 - 3.792 MHz	
	5.237 - 5.428 MHz	BB Loop
	7.022 - 7.214 MHz	
	1.804 - 1.996 MHz	80m Loop
	3.500 - 3.692 MHz	

Figura 2.3.18 Descripción de WebSDR y detalle de localización de servidores.

La Universidad de Cádiz pone un equipo Softrock Lite y un equipo RTL-SDR a la disposición de este proyecto, estando sintonizado el Softrock en la banda de 40 metros y el RTL-SDR en la banda de aficionados de 2 m.

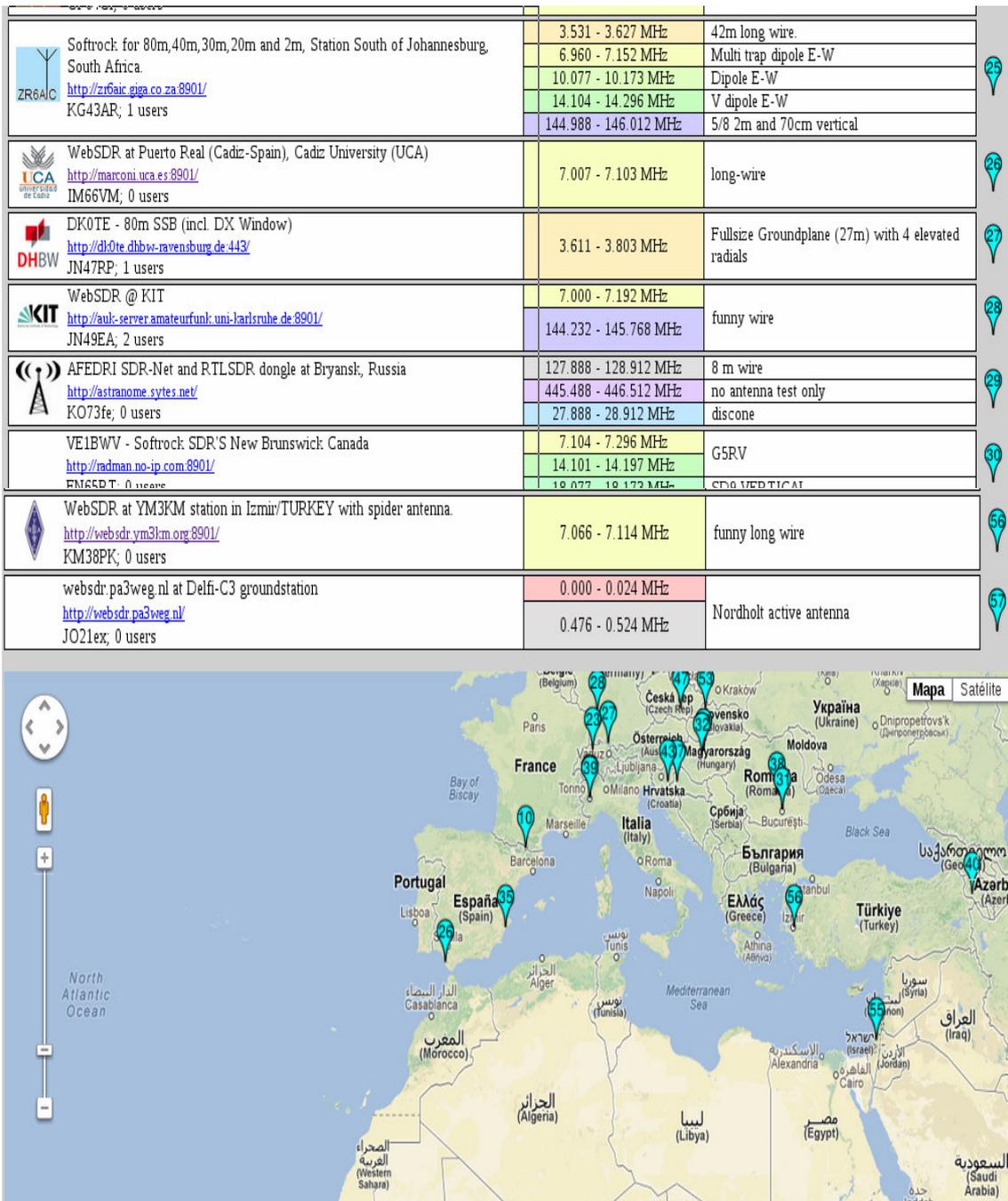


Figura 2.3.19 Relación de servidores donde se observa su localización y posición en el mapa.

Cada estación WebSDR se identifica con la siguiente información:

Descripción de la estación.

Ubicación de la estación.

Coordenadas usando el sistema locator.

Icono de la estación.

Número de usuarios usando el receptor o grupo de receptores de la estación.

Dentro de la aplicación WebSDR se distinguen las siguientes secciones:

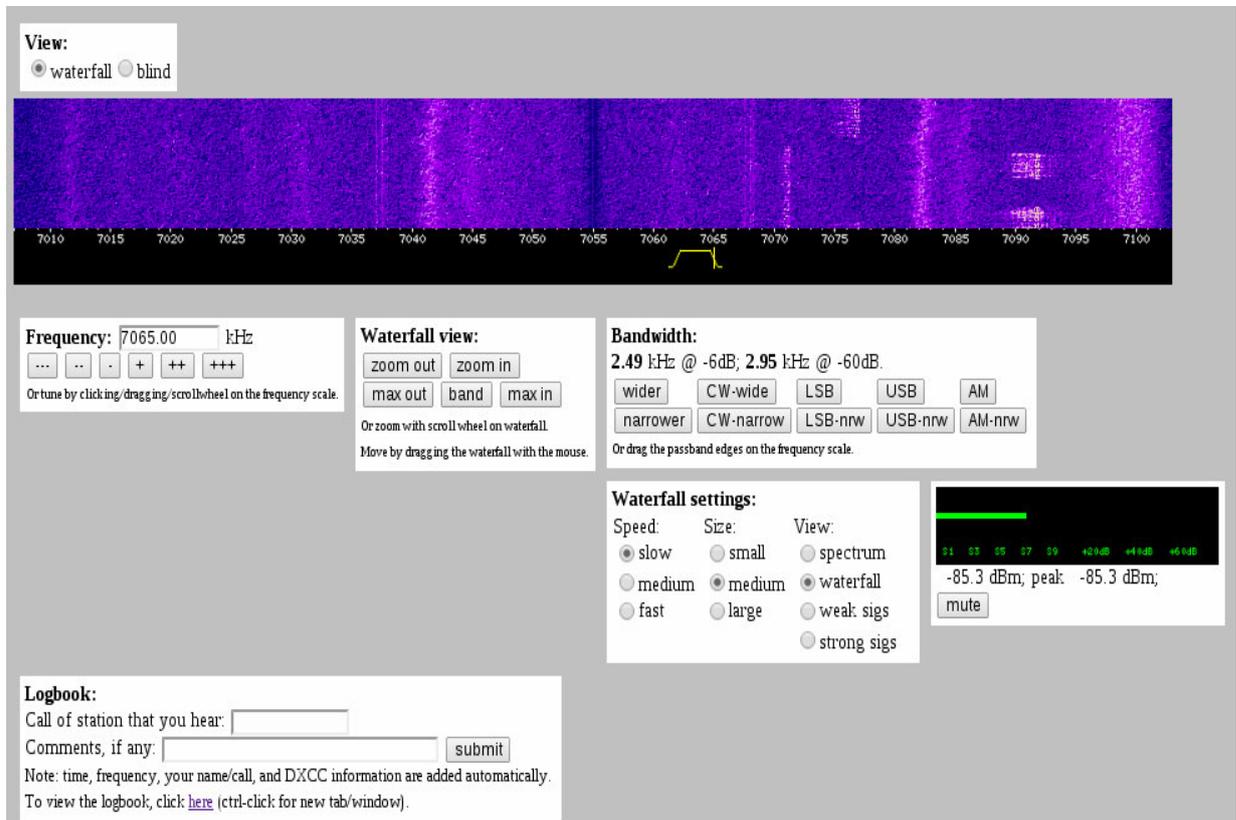


Figura 2.3.20 Componentes de la página WebSDR.

Área de presentación del espectro y búsqueda de señal (Figura 2.3.20), con los siguientes apartados:

- Waterfall o cascada: Donde se hace un seguimiento de la señal en el dominio del tiempo y la frecuencia codificando la amplitud en colores normalizados, así como representación del espectro recibido.
- Frequency: Selección de la frecuencia a sintonizar.
- Waterfall view: Configuración de la cascada, permitiendo aumentar y disminuir la escala de frecuencias o el espectro.
- Bandwidth: Permite seleccionar el modo de recepción, así como el ancho del filtro.

- Morse (CW)
 - Modulación en amplitud (AM)
 - Banda lateral única superior (USB)
 - Banda lateral única inferior (LSB)
 - Modulación en frecuencia (FM)
- Waterfall settings: Permite ajustar el tamaño, velocidad y modo de representación de la cascada, así como atenuar las señales para sintonizar las más fuertes o amplificar las señales para poder visualizar las más débiles.
 - Mute: Representación de la ganancia en forma de medidor de señal o del enmudecimiento de los altavoces.
 - Logbook: Permite anotar comentarios para el administrador, o mantener conversaciones escritas entre usuarios.



Figura 2.3.21 WebSDR. Chat para establecer comunicación entre usuarios.

Área de comunicación:

Permite establecer un diálogo entre los participantes.

Cada comentario viene precedido de la hora y el identificador de la persona que lo hace.

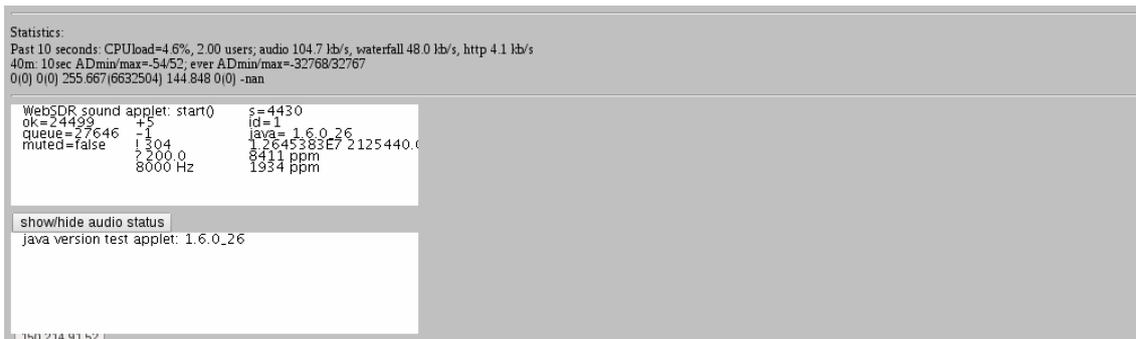


Figura 2.3.22 WebSDR. Información de datos recibidos/enviados.

Área de estadísticas:

Contiene información sobre uso de la aplicación:

- Carga de CPU.
- Número de usuarios.
- Ancho de banda usado. Dentro del ancho de banda usado se tiene :
 - Ancho de banda usado por el audio transmitido
 - Ancho de banda usado por los datos FTT de la cascada.
 - Ancho de banda ocupado por los comandos http

2.3.9.2 Instalación WebSDR.

Como se ha dicho anteriormente, para su instalación se debe usar una máquina Linux (Debian o Ubuntu). Además se debe instalar el entorno LAMP y comprobar que el servidor Web funciona adecuadamente dentro del entorno donde se ha instalado. Para ello se ha de poseer o adquirir conocimientos en esta área de las comunicaciones e informática.

El servidor Marconi se ha instalado en una máquina Linux-Ubuntu.

El ordenador ha de tener comunicación IP, a ser posible usar un procesador I5, aunque los requisitos principales vendrán dados por el ancho de banda necesario para entregar los datos a través de Internet.

Es importante disponer de una tarjeta de sonido estéreo de buena calidad que soporte al menos un ancho de banda de 96 KHz. Es aún mejor si se dispone de 192 KHz. Entonces se ha de hacer un afinado correcto de los parámetros audio del ordenador y asegurarse de que la aplicación “Pulse Audio” funcione adecuadamente (en el caso del sistema operativo Ubuntu).

Para la instalación del software necesario para el funcionamiento del receptor RTL-SDR se ha de seguir el desarrollo del grupo Osmocom, e implementar y compilar los drivers necesarios para este receptor. Se recomienda tener buenos conocimientos de Linux. En caso contrario, se pueden desconfigurar las aplicaciones del equipo.

Como consideración adicional, hemos de tener en cuenta que el servidor ha de poder comunicarse con el puerto seleccionado. Para ello se ha de hacer un chequeo de las comunicaciones del equipo y asegurarse de que tanto el firewall local como los que pueda haber externamente, permitan la comunicación hacia el exterior. Como recomendación se puede usar el comando Linux “netstat -ie” así como hacer un seguimiento en la aplicación de firewall local del PC. Caso de desconocer estas nociones básicas sobre comunicaciones, se puede ir a los puntos de esta tesis donde se habla de VOIP y Telefonía.

En el caso de marconi.uca.es:8901, se usa el puerto 8901 para acceder al servidor Marconi. Dicho puerto tuvo que ser abierto por el CITI de la UCA para la realización de esta Tesis Doctoral, al igual que el puerto 80, ya que lo utiliza el servidor de la Universidad de Twente para obtener los datos del servidor Marconi y mostrarlos en su dirección websdr.org.

Para configurar los datos a mostrar en el servidor websdr.org ha habido que contactar con el Dr. Pieter de Boer “alma mater” del proyecto WebSDR y solicitar nuestra inclusión. El Dr. de Boer se ha mostrado muy interesado en las enseñanzas de la Ingeniería Radioelectrónica, al ser él mismo radioaficionado.

Como primer paso de la instalación del servidor WebSDR Marconi se ha de crear un usuario para la instalación a fin de evitar problemas con otros programas en uso en el mismo servidor (QtRadio, Web server).

El usuario websdr ha tener permiso de acceso a la tarjeta de sonido:

'audio' group:

adduser websdr audio

En el directorio personal de websdr, se descarga el fichero websdr-dist09.tgz y se extrae.

Se crea la estructura de directorios:

/home/websdr/pub	- ficheros públicos (específicos del sitio).
/home/websdr/cfg	- ficheros de configuración de websdr.
/home/websdr/dist09	- ficheros de la aplicación.
/home/websdr/dist09/doc	- documentación.
/home/websdr/dist09/log	- ficheros log.
/home/websdr/dist09/pub2	- ficheros públicos.
/home/websdr/dist09/pub2/tmp	- ficheros públicos temporales

Se copia el fichero `/home/websdr/dist09/websdr.cfg` al

`/home/websdr/cfg` para su configuración.

Se copia el fichero `/home/websdr/dist09/pub2/index.html` al

`/home/websdr/pub` para su configuración.

El servidor usará la tarjeta de sonido, usando los drivers ALSA.

Es muy importante asegurarnos de que usamos versiones de Ubuntu posteriores a 10.10 como la versión 12.04 de Ubuntu, instalada en el servidor WebSDR de la UCA.

El servidor necesita tener cargadas dos librerías en la distribución implementada:

`libfftw3f` (versión 3 de la librería FFTW, compilada para float)

`libpng12` (versión 1.2 de la librería PNG).

Normalmente viene con la distribución de Ubuntu. Habrá que revisar e instalar usando `synaptics` en caso de no estar instaladas.

En caso de no existir dentro de los repositorios de nuestro ordenador, se pueden cargar usando el siguiente comando como superusuario (usando `sudo`):

```
apt-get install libfftw3-3 libpng12-0
```

2.3.9.3 Configuración WebSDR.

Previamente, veremos algunas generalidades sobre el servidor.

El número máximo de usuarios estimados, según el modelo de ordenador es:

Para un PC Pentium 3 con 1 GHz, unos 40 usuarios.

Para un PC Pentium 4 con 3 GHz, unos 200 usuarios.

La cantidad de usuarios aconsejados depende del número de bandas y de la velocidad de muestreo.

El ancho de banda de la conexión Internet, es un factor totalmente limitante. Se puede calcular sobre unos 100 kbit/s por usuario, lo cual depende del nivel de compresión elegido.

Por defecto se sitúa el nivel máximo de usuarios en 40.

A continuación, se debe elegir el puerto a utilizar. Si queremos evitar usar el número de puerto después de la dirección URL se debe elegir el puerto 80.

Una alternativa común es coger el puerto 8901.

La instrucción en este caso sería (UCA Server):

```
tcpport 8901
```

Si se está usando un PC compartido, donde no se confía en los otros usuarios locales, se debe establecer la opción:

```
Dontrustlocalhost
```

La instrucción para no registrar el servidor en el servidor <http://www.websdr.org> es:

```
Noorgserver
```

La información definida para el websdr es:

- Hostname: Nombre del host en Internet.
Para el servidor UCA

```
hostname marconi.uca.es
```
- Locator: El localizador de la posición del servidor.
Para el servidor UCA

```
org qth IM66VM
```
- Una descripción del servidor:
Para el servidor UCA

```
org servidor UCA
```

El Logo a presentar será una imagen GIF, PNG, JPEG de 48x 48, no animada.

Se debe colocar en el directorio público (pub) del Servidor WebSDR.

La instrucción es (servidor UCA):

```
org logo milogo.gif
```

Se debe configurar al menos una banda.

La instrucción es (servidor UCA):

Band 40m

Hay que definir de qué dispositivo de sonido se debe leer la señal I/Q.

Caso de ser ALSA se debe introducir la instrucción anexa:

Device \$hw:0,0,0

A continuación dar la frecuencia de muestreo de la tarjeta de sonido

48000/96000/192000 .

Samplerate 96000

Y la frecuencia central del equipo receptor de RF (Para el caso de la UCA).

centerfreq 7055

Una breve descripción de la antena para que aparezca en la websdr.org.

antenna long wire

Si se quiere, se puede amplificar o atenuar la señal por un número de db:

gain 20

gain -10

En caso de querer tener más de una banda, se debe repetir la sección “band”.

2.3.9.4 Consideraciones generales sobre el cliente WebSDR.

Las consideraciones generales para el cliente WebSDR, son las siguientes:

Se recomienda usar navegadores Firefox o Chrome, los cuales deben tener instalados los plugins de Java.

La lista de servidores activos está en la dirección:

<http://websdr.org>

El cliente debe tener instalado JAVA, al menos la versión 1.4.2 o posteriores.

Los Smartphone y Tablets no se pueden usar por ahora, ya que estos dispositivos no soportan los applets de JAVA.

Actualmente están en marcha prototipos con HTML5 que funcionan sobre Firefox y Google y que se pueden usar en Tablets y Smartphones.

No se puede decodificar señales DRM (“Digital Radio Mondiale”), ya que tienen un ancho de banda de 9 o 10 KHz que es más ancho que el ancho de banda en streaming (envío de sonido) que envía el servidor al cliente.

2.3.9.5 Consideraciones generales para el servidor WebSDR.

Desde el punto de vista radioeléctrico se debe usar como sintonizador un receptor que provea señal IQ audio a la tarjeta de sonido del PC.

Esto incluye los conocidos Softrock.

También se incluyen los receptores que pueden detectarse como una tarjeta de sonido, caso de Funcube y RTL-SDR (Previa compilación de su driver).

Los equipos SDR como Perseus, SDR-IQ y otros introducen sus señales por el puerto USB o bien mediante Ethernet, por lo que no se soportan dentro de la aplicación WebSDR.

Para el servidor WebSDR de la UCA, se ha construido un receptor de radio SoftRock Lite II Combined Receiver Kit. Aunque el proceso de construcción no es fácil, ya que requiere soldar y chequear los bloques funcionales, se ha llevado a cabo ya que es el equipo básico para las aplicaciones SDR y desarrolla lo expuesto anteriormente sobre los equipos receptores de conversión directa.

Posteriormente, en esta tesis, se podrá ver que se ha utilizado también un receptor “Softrock RX Ensemble II HF Receiver”. Este receptor es más complejo y la frecuencia central es variable, no como en este caso en el que es fija.

En las figuras 2.3.23 y 2.3.24 siguientes se puede ver este receptor (Softrock Lite II).

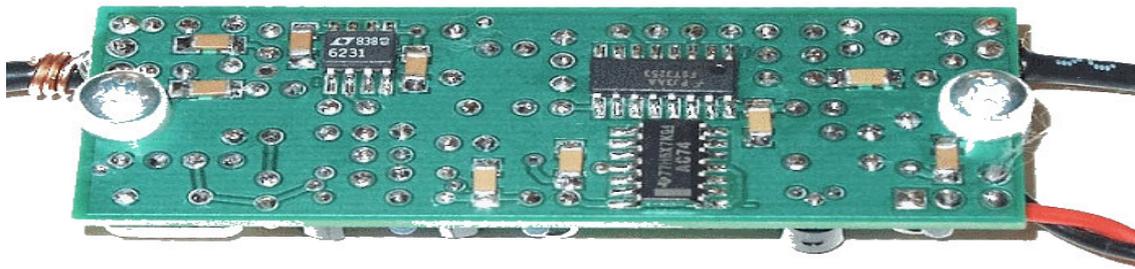


Figura 2.3.23 SoftRock Lite II, cara inferior (Bottom)



Figura 2.3.24 SoftRock Lite II, cara superior (Top)

2.3.9.5.1 El receptor Softrock Lite II.

El Softrock Lite II (40 metros) es un receptor SDR en kit con una frecuencia central fija, mediante oscilador controlado a cristal de cuarzo de 7.056 MHz.

Está diseñado para ser usado con una tarjeta de sonido que pueda muestrear a 96 KHz.

Pudiendo mostrar un ancho de banda 48 KHz por arriba y debajo del punto central.

Los kits traen los componentes necesarios para poder construir el receptor para:

160 metros (frecuencia central 1,843 MHz)

80 metros (frecuencia central 3,522 MHz)

- 40 metros (frecuencia central 7,056 MHz)
- 30 metros (frecuencia central 10,125 MHz)
- 20 metros (frecuencia central 14,047 MHz)

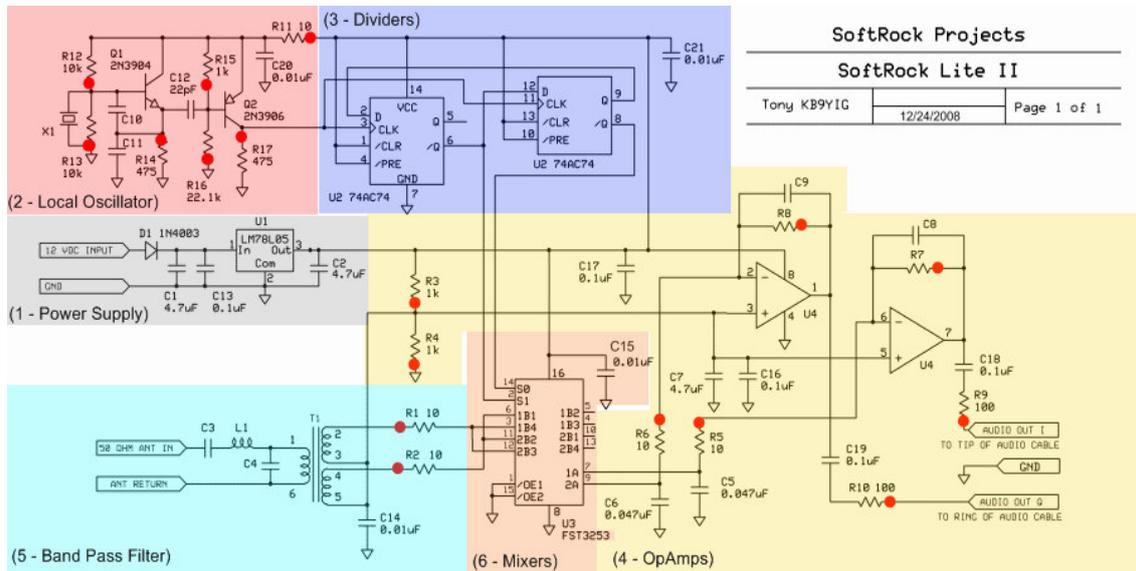


Figura 2.3.25 Diagrama receptor Softrock Lite II (FiveDashInc)

Cada kit consta de las siguientes etapas:

- Fuente de alimentación
- Oscilador local
- Divisor
- Amplificadores operacionales
- Filtro paso banda
- Mezclador
- Conexiones externas

2.3.9.5.1.1 Fuente de alimentación (40m).

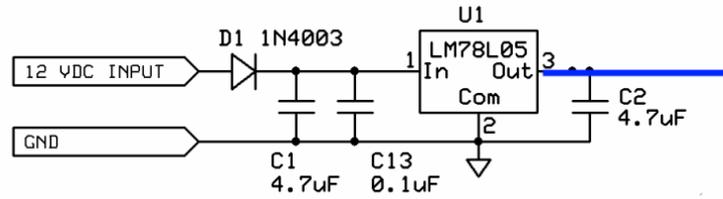


Figura 2.3.26 Fuente de alimentación Softrock lite II (FiveDashInc).

Esta etapa da una tensión de +5v para el receptor.

La tensión de entrada de 9 a 12 voltios, se regula por U1 a 5 voltios (rango 4,5 – 5,1 Vcc).

El diodo D1 sirve para evitar que la polarización inversa del circuito quemé los componentes.

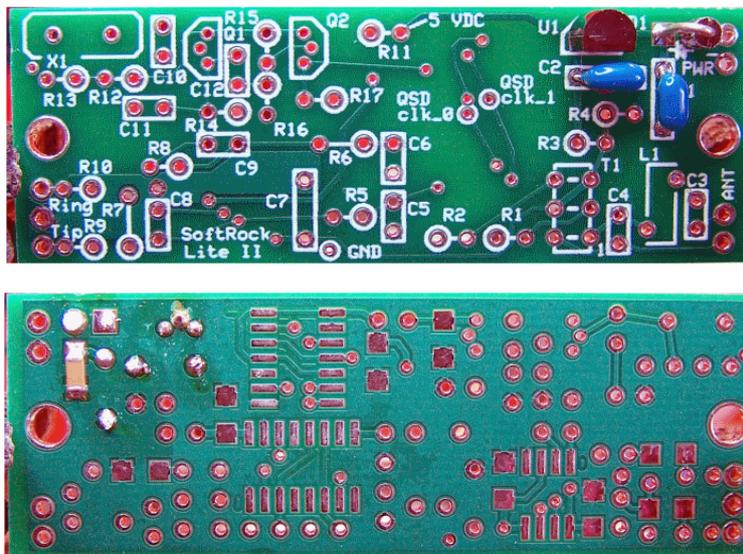


Figura 2.3.27 Detalle de componentes de fuente de alimentación (FiveDashInc).

2.3.9.5.1.2 Oscilador local (40m).

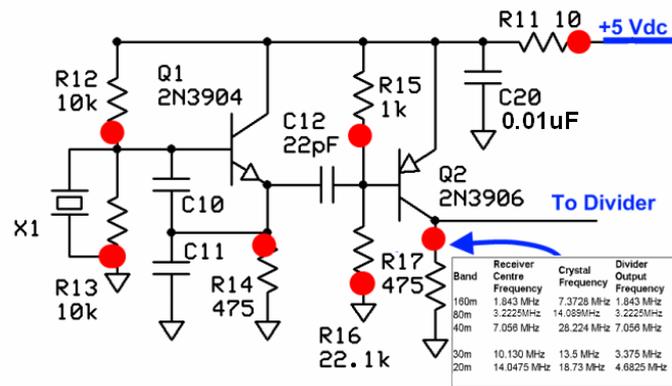


Figura 2.3.28 Oscilador local Softrock lite II (FiveDashInc).

Teoría del funcionamiento:

La etapa del oscilador local tiene un cristal oscilador básico con una etapa amplificadora del nivel de la señal. El oscilador produce una señal que está en la frecuencia fundamental del cristal, 28,224 MHz.

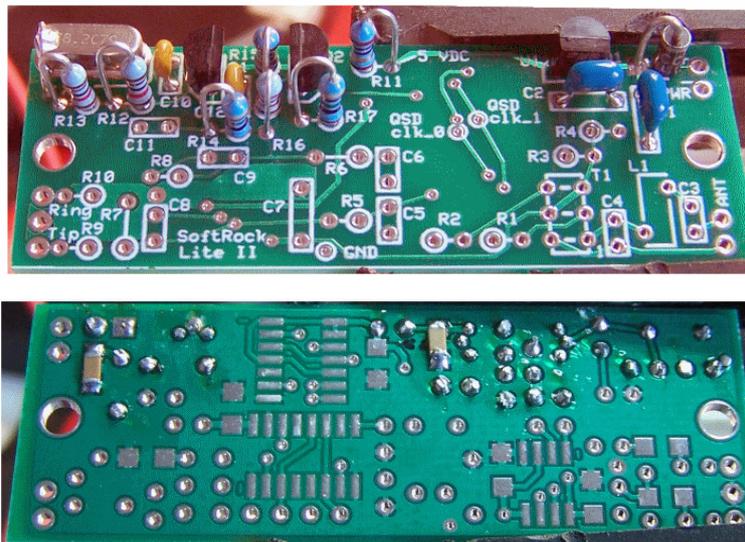


Figura 2.3.29 Detalle de componentes del oscilador local. Softrock Lite II (FiveDashInc).

2.3.9.5.1.3 Divisor (40m).

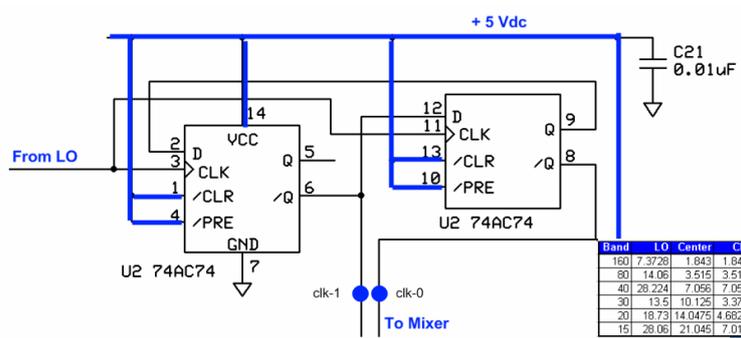


Figura 2.3.30 Divisor Softrock Lite II (FiveDashInc).

Teoría del funcionamiento:

Los divisores toman como entrada la salida del oscilador local y dan dos señales de salida, con una reducción en frecuencia de $\frac{1}{4}$ de la frecuencia de entrada y en cuadratura (90° de desfase entre cada una).

U2 está cableado como un divisor síncrono por 4 de la velocidad del reloj.

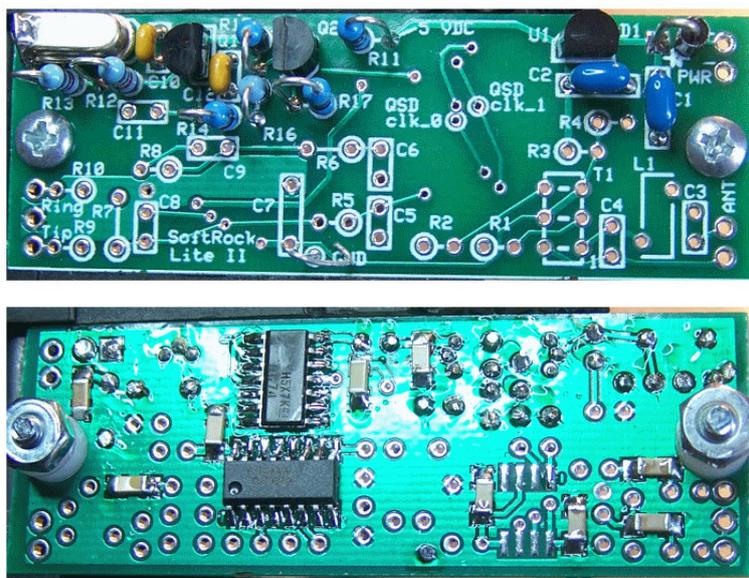


Figura 2.3.31 Detalle de componentes del Divisor Softrock Lite II (FiveDashInc).

2.3.9.5.1.4 Amplificadores operacionales (40m).

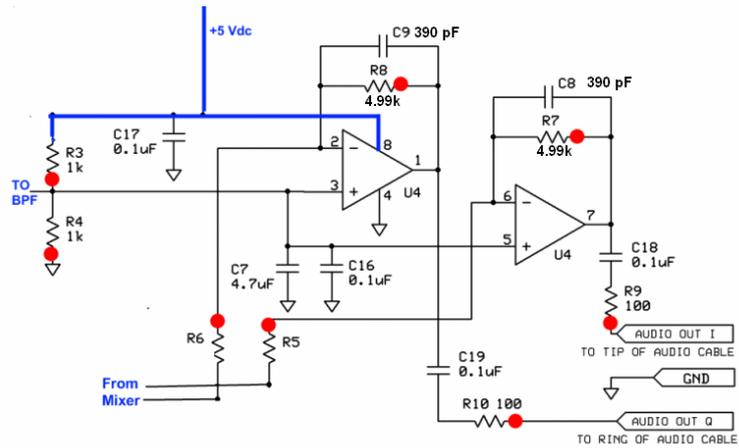


Figura 2.3.32 Amplificadores operacionales. Softrock Lite II (FiveDashInc).

Teoría del funcionamiento:

Los amplificadores operacionales en esta etapa amplifican las señales de banda base I y Q por un factor de aproximadamente 499, para compensar la atenuación como consecuencia de la toma de muestras en cuadratura del subarmónico.

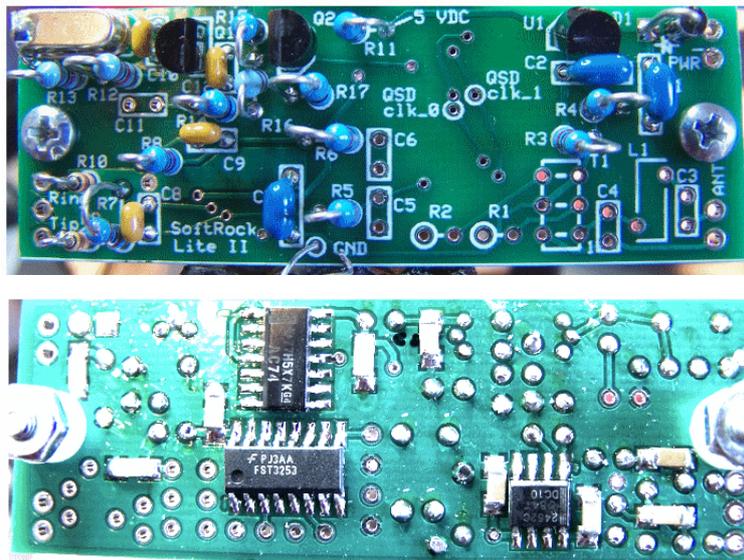


Figura 2.3.33 Amplificadores operacionales. Softrock Lite II (FiveDashInc).

2.3.9.5.1.5 Filtro paso banda (40m).

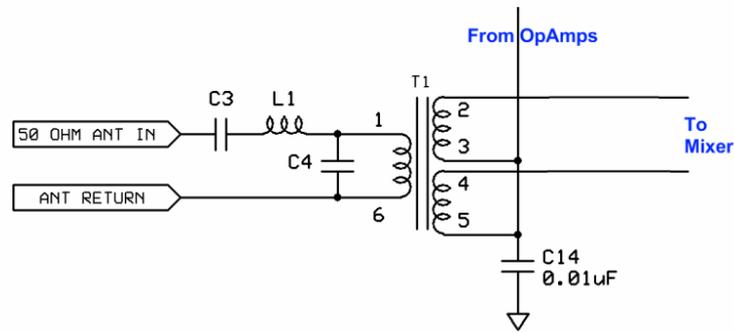


Figura 2.3.34 Filtro paso-banda. Softrock Lite II (FiveDashInc).

Teoría del funcionamiento:

El propósito de esta etapa es pasar las señales de radio frecuencia dentro de la banda de recepción a la etapa mezcladora y atenuar las señales no deseadas que no están dentro de la banda de paso diseñado para el filtro.

Esta atenuación es especialmente importante, ya que permite la toma de muestras del tercer armónico en el mezclador para las bandas superiores.

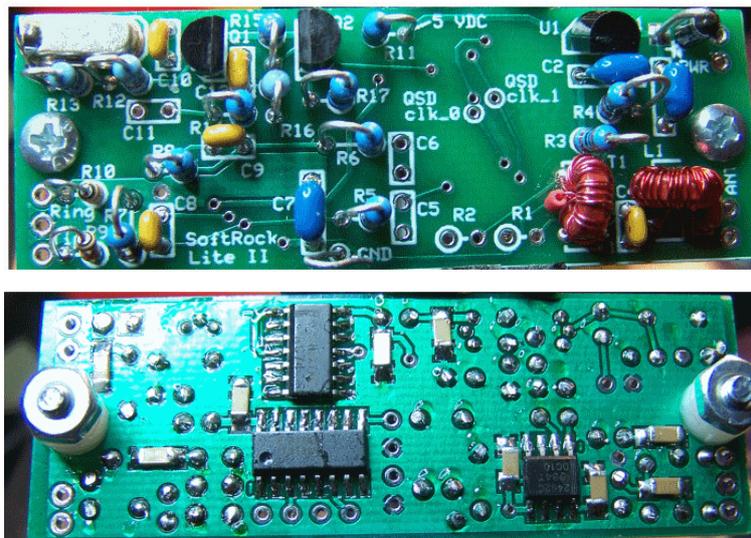


Figura 2.3.35 Filtro paso-banda. Softrock Lite II (FiveDashInc).

2.3.9.5.1.6 Mezclador (40m).

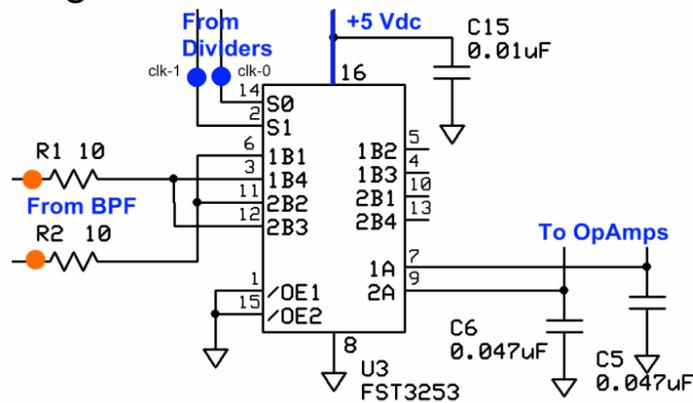


Figura 2.3.36 Mezclador. Softrock Lite II (FiveDashInc).

Teoría del funcionamiento:

La señal de entrada de RF, filtrada por el BPF (Filtro paso-banda) en oposición de fase, se aplica a las entradas 1B y 2B del mezclador U3.

Las dos señales del oscilador local desde la Etapa Divisora operan los conmutadores que conectan R1 a C5 y R2 a C6 durante el primer ciclo de reloj.

Cuando el reloj del oscilador local cambia (90 grados después) la conexión cambia.

R1 se conecta ahora a C6 (Q) y R2 se conecta a C5 (I).

Esta secuencia de conmutación se repite.

La señal de entrada de RF resultante se muestrea sobre los condensadores C5 y C6 como la Frecuencia Intermedia (IF).

En las bandas de 40m los divisores están sincronizados a la frecuencia central deseada, que está en la banda de paso para la RF entrante.



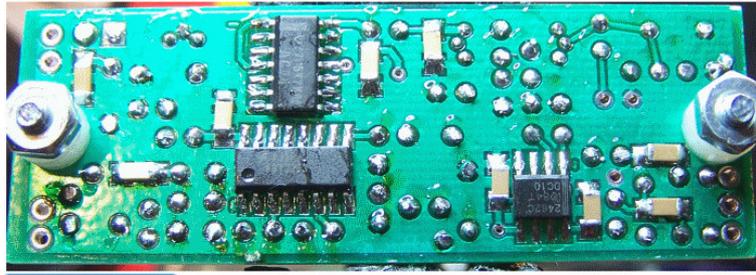


Figura 2.3.37 Mezclador Softrock Lite II (FiveDashInc).

2.3.9.5.1.7 Conexiones externas.



Figura 2.3.38 Conexiones externas. Softrock Lite II (FiveDashInc).

Alimentación: Los cables de alimentación se pueden conectar a una fuente de corriente continua regulada de 9 a 13 Vdc.

RF: Conexión para una antena de 50 ohm sintonizada para la banda especificada.

Salida I / Q: Conexión de las salidas de audio I / Q del receptor al PC a través de la entrada estéreo de su tarjeta de sonido. Normalmente, esto se realizará en el conector "line-in".

En la figura 2.3.39 se observa el diagrama de bloques de los circuitos vistos anteriormente.

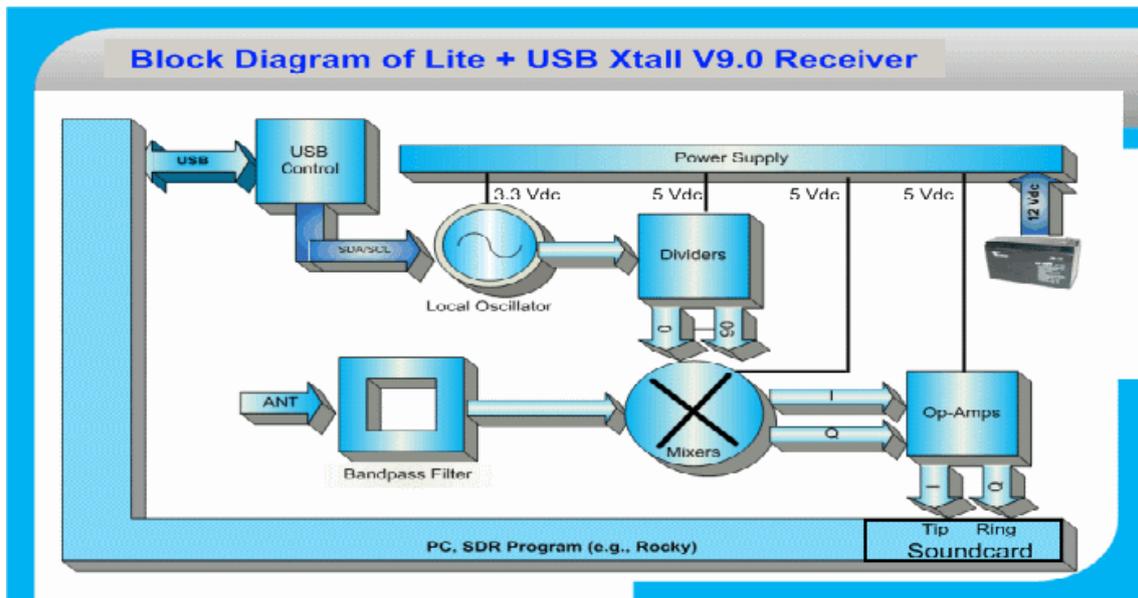


Figura 2.3.39 Diagrama de bloques Softrock Lite II (FiveDashInc).

2.3.9.5.2 Distribuciones Linux del servidor y tarjetas de sonido.

Dentro de la parte informática del servidor, la distribución de Linux no importa, pero debe poder funcionar en micros Intel (compatibles) de 32 o 64 bits (usar Ubuntu o Debian).

Es necesario usar la tarjeta de sonido del ordenador. Algunas tarjetas de sonido tienen 4 canales y una frecuencia de muestreo de 192 KHz. Es importante que la tarjeta de sonido sea de buena calidad.

En el caso de ser necesario se puede colocar más de una tarjeta de sonido. Para ello se ha de asegurar que el ordenador seleccionado disponga de suficientes extensiones para la colocación de éstas.

En el caso de no disponer de extensiones suficientes, se puede recurrir a usar tarjetas de sonido externas. En este caso se ha de tener especial cuidado con su elección, de forma que sean compatibles con el sistema operativo usado.

En el servidor Marconi, debido a la coexistencia en el mismo ordenador de dos usuarios con dos programas diferentes, QtRadio y WebSDR y a la existencia de un solo slot PCI en el PC, se ha provisto de una tarjeta externa estéreo.

Se ha comprobado que esta tarjeta externa ha dado una buena funcionalidad, trabajando de forma adecuada. Es de destacar que esta tarjeta es de 96 KHz.

2.3.9.6 Configuración avanzada de WebSDR.

Toda la configuración del servidor se recoge en el fichero de configuración websdr.cfg .

Cada línea comienza con una palabra seguida de uno o más argumentos, tal y como se detalla a continuación.

Acceso al servidor :

myhost x

Sólo se conecta a esta dirección.

Es útil si el servidor tiene múltiples direcciones IP.

tcpport x

Define el puerto en el cuál trabajará el servidor.

(Ejemplo 8901 en marconi.uca.es:8901)

Por defecto se usa el puerto 80 usado normalmente en los navegadores.

Sin embargo esto requiere correr el servidor como root (administrador).

chroot x

Después de comenzar, coloca el chroot() a x .

Esto se puede usar como una medida de seguridad.

Esto requiere que el setuid de websdr sea root (administrador).

rawpassword x

Coloca una contraseña para acceder al stream (cadena) de datos.

Por defecto no hay contraseña (password).

dontrustlocalhost

Deshabilita la interfaz de administración (sysop.html).

De otra forma sería accesible con cualquier navegador que corriera en la misma máquina.

Esta instrucción es necesaria si en la misma máquina hay usuarios en los cuales no confiamos.

Situación de los ficheros:

chatboxlogfile x

Graba todos los mensajes del chatbox en un fichero llamado x.

logfileinterval x

Coloca el tiempo en segundos entre que un fichero log comienza y el fichero anterior se comprime (zip), por defecto es 3600 segundos.

Si se sitúa el valor en 0, no se escriben los ficheros Log, pero los mensajes Log se envían al stderr (Terminal de errores).

logdir x

Define el directorio Log como x.

Por defecto es Log.

public x

Define el directorio x como el que contiene los ficheros públicos (html, etc).

Por defecto es el pub.

public2 x

Define x como el directorio secundario que contiene los ficheros públicos (html, etc.).

Por defecto pub2.

Configuraciones de proceso:

maxusers x

Limita el número máximo de usuarios a x.

slowdownusers x

Sólo admite cascadas lentas cuando el número de usuarios excede de x.

Es útil para ahorrar uso de cpu en caso de ordenadores lentos o bien ancho de banda en caso de ancho de banda bajo.

slowdownusers2 x

Sitúa la velocidad de la cascada a la mitad de la velocidad normal mínima cuando el número de personas conectadas excede de x,

fftplaneffort x

Sitúa el esfuerzo planeado de la librería FFTW a x, donde x puede ser:

- 0 para FFTW_ESTIMATE
- 1 para FFTW_MEASURE
- 2 para FFTW_PATIENT
- 3 para FFTW_EXHAUSTIVE

Los números mayores hacen comenzar el programa más lento, pero puede reducir la carga de la CPU, porque la librería intenta optimizar el cálculo de la FFT .

Valor por defecto es 0.

waterfallformat x

Define el formato de datos para los paquetes de datos de la cascada.

x=0 da el antiguo formato no comprimido, 8 bits por pixel.

x=8 usa una media de 2.9 bits por pixel.

x=9 es el formato por defecto de compresión entre 1.5 y 2.3 bits por pixel para cpu muy rápidas .

audioformat x

Coloca el modo de compresión para los datos de audio.

x puede ser de 0 a 3.

Cuanto mayor sea x, mayor será la cuantificación del sonido.

Normalmente, x=0 (por defecto) será el mejor; pero si el ancho de banda está realmente limitado es mejor intentar otros.

Configuraciones relativas a la apariencia en la web <http://www.websdr.org> :

noorgserver

Incluyendo estas líneas se evita que el servidor sea listado en la Web

<http://www.websdr.org/>

hostname x

Pone el nombre del hostname en la dirección <http://www.websdr.org> en vez de la dirección de IP.

org

Configura los datos a enviar al servidor websdr.org .

Las posibilidades actuales son :

- | | |
|-------------------|-------------------------------|
| org qth x | - "QTH locutor" del sitio |
| org description x | - Breve descripción del sitio |
| org logo x | - Nombre del fichero del logo |

Instrucciones de configuración por banda:

band x

Comienza la configuración para la banda llamada x.

Debe ser el primer comando a poner para una banda.

device x

Pone el dispositivo de entrada a x, como /dev/dsp0.

Para dispositivos ALSA, también se pueden usar el símbolo '\$', como \$hw:0,0,0

device x y

Alternativa para usar con una tarjeta de sonido multi-entrada.

y es el nombre del device como /dev/dsp0, mientras x es un número de un dígito: 0 para las primeras dos entradas, 1 para las siguientes dos entradas, etc.

Todas las bandas que comparten la misma tarjeta, deben tener la misma frecuencia de muestreo, y ellos no pueden tener la configuración “noniq”. Ni para los dispositivos ALSA.

extrazoom x

Incrementa la máxima profundidad de zoom en x pasos.

Cada paso dobla la resolución máxima. El valor por defecto es 0. Valores mayores, pueden ser útiles para proveer mayor integración para portadoras de larga duración pero WebSDR no está diseñado para esto.

samplerate x

Define la frecuencia de muestreo (en Hz).

centerfreq x

Pone la frecuencia central (en kHz)

noniq

Las entradas se tratan como entrada en banda base (mono), en vez de una entrada I/Q (señal estéreo en cuadratura). Esto es útil para los receptores VLF con la antena conectada directamente a la tarjeta de sonido.

Con “noniq” la configuración “centerfreq” se interpreta no como el centro del paso banda, sino como el lado izquierdo del paso banda.

swapiq

Intercambia las señales de los canales I y Q (izquierda y derecha) .

Se usa si el paso banda aparece como en un espejo.

balance x

Lee los datos de balanceo desde el fichero x. (Ver en doc/iq-balance.txt).

antenna x

Pequeña descripción de la antena que aparecerá en la web <http://www.websdr.org>.

delay x

Artificialmente retrasa la señal x segundo.

gain x

Inserta x dB de ganancia (puede ser negativa).

Esto no hace una diferencia audible, sólo desplaza la escala de color de la cascada y el medidor de señal, de forma que lo podemos usar para calibrar el medidor de señal.

hpf x

Sitúa la frecuencia de corte a la entrada del filtro paso-alto a x Hz.

Este filtro corta el offset DC y posibilita otro ruido de baja frecuencia (ejemplo: zumbidos de red) en la entrada de la tarjeta de sonido.

Esto es justo un filtro de primer orden, así que no es muy abrupto y poniéndolo a cero elimina el filtro completamente. Por defecto es 500 Hz.

Etiquetas de Información de la escala de frecuencias:

Si queremos poner algunas etiquetas en la escala de frecuencias, por ejemplo marcar balizas u otras estaciones permanentes de interés, podemos ponerlas en el fichero llamado "stationinfo.txt" en el directorio /cfg.

El formato de este fichero es muy simple: cada línea tiene una frecuencia en KHz seguido por una breve descripción.

Este fichero se lee cada minuto, así que si se hace un cambio, no se necesita reiniciar el servidor.

2.3.10 Arquitectura QtRadio.

2.3.10.1 Objetivos del proyecto.

Los objetivos del proyecto QtRadio son los siguientes:

1. Uso de Software SDR Código abierto (GPL).
2. Uso de entorno Cliente-Servidor.
3. Desarrollo de un servidor para cada uno de los tipos de servidores hardware. HPSDR, Softrock, UHFSDR, Microtelecom Perseus, RFSPACE SDR-IQ, HiQSDR, Ettus Research USRP, y RTL-SDR DVB-T Dongle.
4. Un solo servidor dspserver que puede conectar a servidores simples. De cualquier forma cada servidor puede tener múltiples canales Recepción y Transmisión.
5. Múltiples clientes QtRadio, pueden estar concurrentemente conectados a un solo dspserver.
6. Completa capacidad de Recepción y Transmisión para hardware capaces de Transmitir.

El servidor de QtRadio se desarrolla en Linux, pudiendo manejar los diversos equipos definidos anteriormente, mientras que el cliente se desarrolla para las diferentes plataformas.

Los clientes se pueden descargar compilados, pero no se mantiene en compilación la última revisión desarrollada.

Es recomendable descargar las fuentes de su dirección git y compilarlo localmente en el servidor, de esta forma aseguramos que el servidor tiene la última versión disponible y que a su vez dispone de los recursos necesarios para correr el programa de QtRadio, software, drivers,..etc.

Posteriormente se describe el proceso de instalación llevado a cabo con el servidor QtRadio de la UCA (Grupo Señales Sistemas y Comunicaciones Navales).

2.3.10.2 Descripción general.

QtRadio es un Software SDR que está siendo desarrollado por el grupo QtRadio y cuyo líder es Alex Lee, trabaja en entorno Linux y permite el acceso a servidores HPSDR, Softrock, Perseus, SDR-IQ, HIQSDR y SDRWidget .

QtRadio está siendo creado en el entorno de desarrollo Qt, usando C++.

Cabe destacar que este software no es un producto comercial, sino que está en fase de investigación y desarrollo. Para acceder a él hay que participar en su foro e interactuar con sus desarrolladores, probando las soluciones y reportando los fallos, acciones que se ha llevado a cabo en esta Tesis Doctoral.

Deseamos también apuntar que su elección se debe a que es el único conocido que permite la transmisión, con el uso de procesos cliente-servidor y SDR. Si bien la transmisión no se ha considerado en esta Tesis Doctoral, será una línea futura de investigación dentro del grupo S2CN de la UCA.

Los procedimientos de instalación requieren un buen conocimiento de Linux así como de seguimiento de su evolución y depuración personal de posibles errores de instalación.

Durante el desarrollo de esta tesis, se ha ido participando en el grupo QtRadio y siguiendo este desarrollo hasta el momento en que ha sido posible obtener un prototipo con las suficientes garantías para ser usado dentro del demostrador de conceptos Marconi.

QtRadio junto con SDR-Radio y WebSDR son aplicaciones que permiten la arquitectura cliente servidor, siendo idóneas para la aplicación del concepto distribuido de acceso a la información.

QtRadio se basa en el software SDR desarrollado por John Melton [G0ORX/N6LYT].

El software original de John se llama ghpsdr3.

La rama actual de desarrollo es la ghpsdr3-alex.

El código fuente se encuentra en git, en la dirección

<https://github.com/alexlee188/ghpsdr3-alex> .

El seguimiento del desarrollo se hace en el grupo sdr-widget

<http://groups.google.com/group/sdr-widget>

Se pueden ver los servidores actualmente on line en la dirección

<http://qtradio.napan.ca/qtradio/qtradio.pl>

La dirección de estos servidores se puede ver en la dirección

<http://www.montefusco.com/ghpsdr3>

Esta Tesis tiene un servidor colocado en la dirección marconi.uca.es. El servidor está puesto a disposición del público por el Grupo de Investigación Señales Sistemas y Comunicaciones Navales de la UCA.

QtRadio actualmente soporta los receptores/transmisores HPSDR, Softrock, Microtelecom Perseus, SDR-IQ, HiQSDR, Ettus Research USRP y dispositivos RTL-SDR.

El software actual tiene tres componentes:

A) Servidor :

Implementa la conexión con cada uno de los modelos de receptores/transmisores empleados. Se encarga de establecer cada uno de los comandos y acciones que hay que hacer para manejar el receptor/transmisor, a su vez se conecta al módulo común dspserver.

Hay un servidor para cada uno de los equipos siguientes:

HPSDR

USRP

Softrock

SDR-IQ

HiQSDR

Perseus

RTL-SDR

Algunos servidores se basan en la tarjeta de sonido/USB y otros se conectan vía ethernet (HPSDR Metis).

B) dspserver :

Éste es el software situado en el medio que procesa los datos IQ desde el servidor (conectado sobre LAN via UDP) usando DttSP. El mismo dspserver se puede conectar a cualquiera de los servidores en A.

C) Clientes:

Ésta es la interfaz GUI. Hay clientes construidos con Qt y funcionan en Windows, Linux, y Mac OSX. Hay clientes funcionando en Android que se ejecutan en Teléfonos y Tablets.

Los clientes se conectan al módulo servidor DSP (dspserver), vía Lan o Internet.

Hay en desarrollo clientes Web basados en Java (bien escritos específicamente o corriendo Qt dentro de un navegador) así como clientes Android. Aunque dentro de la investigación hecha dentro de esta tesis se han seguido e incluso puesto en funcionamiento los clientes, no se ha avanzado más en esta línea por ser un área tan amplia que no daría tiempo a ser cubierta en esta tesis. Como punto para recalcar esto, habría que destacar que el desarrollo de la aplicación glSDR (Cliente Android) está liderada dentro del grupo “sdr-widget” por el profesor Rob Frohne de la Universidad de Walla Walla, siendo sus herramientas de desarrollo Android y su IDE Eclipse.

El otro cliente en desarrollo es JMonitor que actualmente empieza a tener una interfaz estable. Su desarrollador John Melton lo ha desarrollado en Java, requiriendo para ello que el PC donde se use tenga instalada la máquina JAVA. Se considera que el futuro desarrollo de HTML5 evitará el uso de JAVA.

En QtRadio varios clientes pueden acceder de forma concurrente al mismo servidor dspserver y tienen capacidad para recibir y transmitir.

Los clientes QtRadio tienen una interfaz Hamlib y se pueden conectar a gPredict, xLog, y fldigi (para modos digitales).

2.3.10.3 Instalación de QtRadio para Linux-Ubuntu 12.04.

Se recomienda la instalación desde Git, compilando localmente las fuentes.

Existen archivos compilados binarios, pero éstos no servirán en caso de que se quiera investigar sobre SDR.

Los archivos binarios tienen capacidades reducidas y no integran la mayoría de receptores existente, entre éstos el Perseus.

Para llevar a cabo esta instalación se ha de tener un buen conocimiento de Ubuntu, e instalar las librerías requeridas para la compilación del programa. También hay que descargar y compilar los drivers específicos del receptor Perseus, que se pueden encontrar bien en la web de Perseus o del señor Andrea Montefusco. En el caso de no encontrarlos, se tendrá que acudir al grupo de desarrollo “sdr-widget” o bien preguntar dudas a los desarrolladores.

Seguidamente se detalla el proceso seguido para poder instalar localmente el servidor Perseus. Las modificaciones, al ser software en desarrollo, han de seguirse a través del grupo “sdr-widget”. El chequeo del correcto funcionamiento requiere el uso de las herramientas Linux para lo cual hay que formarse en esta área.

En primer lugar, nos aseguramos de tener los compiladores:

Para ello ejecutamos las siguientes instrucciones (hemos de estar seguros de tener conexión a Internet):

```
sudo apt-get install make gcc g++
```

```
sudo apt-get install autoconf automake autotools-dev libtool git subversion
```

A continuación usando el programa Synaptics se instala:

Qt4 (y QtCreator)

Librerías Qt5

Seguidamente se instalan los paquetes de prerequisites:

Para Ubuntu 12.04 y posteriores:

Instalación de portaudio19-dev con jackd2

Portaudio19-dev depende de tener una copia de libjack-dev instalado, si el paquete libkjack-jackd2-dev se ha instalado antes de portaudio19-dev, las dependencias se satisfarán, si no, se debe desinstalar jackd2 e instalar jackd1 (si ya se había instalado previamente jackd2 en el sistema).

Si jackd1 y portaudio19-dev están ya instalados:

Desde Synaptic :

Desinstalar portaudio19-dev

Desinstalar jackd1

Instalar Jackd2

Instalar libjack-jackd2-dev

Instalar portaudio19-dev

Si estamos empezando desde cero:

Desde Synaptic

Instalar Jackd2

Instalar libjack-jackd2-dev

Instalar portaudio19-dev

Con los siguientes comandos instalamos todo el entorno que necesitamos:

```
sudo apt-get install jackd2 libjack-jackd2-dev portaudio19-dev
```

```
sudo apt-get install make gcc g++ autoconf automake autotools-dev libtool git  
subversion
```

```
sudo apt-get install libusb-0.1-4 libusb-dev libfftw3-dev libpulse-dev  
libsamplerate0-dev
```

```
sudo apt-get install libusb-1.0-0-dev libconfig8-dev xdg-utils libortp-dev
```

```
sudo apt-get install libevent-dev
```

Si la máquina no tiene una tarjeta aceleradora de vídeo compatible OpenGL instalamos los siguientes paquetes:

```
sudo apt-get install mesa-utils
```

```
sudo apt-get install mesa-common-dev
```

```
sudo apt-get install libgl1-mesa-dev
```

```
sudo apt-get install x11proto-xext-dev
```

```
sudo apt-get install libxrender-dev
```

Instalación del codec2 de codificación y decodificación de audio a baja relación de bit:

Para descargar y construir la librería codec2 . Ver:

http://www.rowetel.com/blog/?page_id=452

A continuación cogemos codec2 usando Subversion:

```
cd
```

```
svn co https://freetel.svn.sourceforge.net/svnroot/freetel/codec2
```

```
cd codec2
```

```
autoreconf -i
```

```
./configure
```

```
make
```

```
sudo make install
```

```
sudo ldconfig
```

Instalación de Qt5:

Añadimos el ppa qt5 “Canonical Qt5 Edgers” al sistema

En un terminal, copiamos el comando siguiente:

```
sudo add-apt-repository ppa:canonical-qt5-edgers/qt5-daily
```

El sistema ahora cogerá la clave PPA. Esto permitirá a nuestro sistema Ubuntu verificar que el paquete en el PPA no se ha modificado desde que se ha construido.

Posteriormente le indicamos a nuestro sistema que baje la última lista de software incluyendo el PPA que hemos añadido:

```
sudo apt-get update
```

Ahora estamos listos para instalar el software desde el PPA.

Instalación desde el PPA:

El paquete a instalar estará identificado de forma similar a la siguiente:

```
5.0-2012xxx Ubuntu5 Zoltan Balogh (2012-04-05)
```

Abrir Synaptic y seleccionar para instalar cada uno de los paquetes de la siguiente lista:

```
qt-components-qtquick2 5.0-2012xxx Ubuntu5 Zoltan Balogh (2012-04-05)
qt3d
qtbase
qtdeclarative
qtdoc
qtjsbackend
qtmultimedia
qtquick1
qtscript
qtsvg
qttools
qtwebkit
```

qtxmlpatterns

Estos paquetes se instalarán por defecto en el directorio /opt/qt5

Si usamos la línea de comandos, debemos ejecutar los comandos:

```
sudo apt-get install qt-components-qtquick2 qt3d qtbase
```

```
sudo apt-get install qtdeclarative qtdoc qtjsbackend qtmultimedia qtquick1
```

```
sudo apt-get install qtscript qtsvg qttools qtwebkit qtxmlpatterns
```

Colocamos el path a qt5:

Desde el Terminal

```
$ export PATH=/opt/qt5/bin:$PATH
```

Configuración de QtCreator para usar librerías qt5:

Debemos recordar que QtRadio, está creado en el entorno de desarrollo Qt, y que QtCreator es la interfaz para acceder a todo este entorno, en forma similar a Eclipse que es otro entorno de desarrollo parecido.

Si queremos usar QtCreator tendremos que configurarlo para usar Qt5 como una de las opciones:

Desde la línea de menú de QtCreator:

Tools > Options > Build & Run

Seleccionar el panel Qt Versions

Presionar en el botón Add

Ir a /opt/qt5/bin

Presionar en qmake

Presionar en open

Presionar en Apply y finalmente OK

Hay que apuntar al fichero qmake en /opt/qt5/bin

Selección de la librería construida en QtCreator:

Necesitamos importar el proyecto QtRadio y después se puede elegir Qt5 para construir la aplicación.

Puede ser necesario editar el fichero de proyecto (QtRadio.pro) para elegir qué versión de QtCreator estamos usando.

Elección de la última actualización de la librería libevent:

```
$ sudo apt-get install libevent-dev
```

Si la versión es más antigua que la 2.0.5, debemos quitar el libevent de Ubuntu, usando synaptic.

A continuación, descargar la nueva versión desde <http://libevent.org/>:

```
wget https://github.com/downloads/libevent/libevent/libevent-2.0.16-stable.tar.gz
```

Mover el tarball desde el directorio de descarga al directorio de trabajo.

Se sugiere crear un directorio src dentro del directorio home y moverlo allí.

Descomprimir y seguir las instrucciones:

```
cd
tar -zxvf libevent-2.0.16-stable.tar.gz
cd libevent-2.0.16-stable
./configure
make
sudo make install
sudo ldconfig
```

Instalación del codificador y decodificador de audio de baja velocidad de bit codec2:

Con ello se añade un códec de audio de baja velocidad de bit llamado codec2 a QtRadio/dspserver.

Se necesita descargar y construir la librería codec2:

Se encuentra descripción de esta librería en la dirección web indicada a continuación:

http://www.rowetel.com/blog/?page_id=452

Obtenemos codec2 usando Subversion

```
cd
svn co https://freetel.svn.sourceforge.net/svnroot/freetel/codec2-dev codec2-dev
cd codec2
autoreconf -i
./configure
make
sudo make install
sudo ldconfig
```

Ahora vamos a construir la rama máster del proyecto QtRadio. Es la usada en el servidor Marconi de esta Tesis y operativo en el servidor del grupo S2CN de la UCA (Marconi.uca.es, accesible con el cliente de QtRadio).

Se crea el directorio ghpsdr3-alex, donde se sitúan todas las fuentes del programa.

La descripción secuencial de la creación del servidor se encuentra en apéndice anexo. De todas formas, se debe destacar, que la secuencia de compilación e instalación de QtRadio puede variar de la detallada a continuación, al estar este software en desarrollo. Para seguir el último procedimiento se debe ir a la página de QtRadio http://napan.ca/ghpsdr3/index.php/Main_Page .

A continuación se relacionan los comandos a ejecutar en el Terminal:

```
cd
git clone git://github.com/alexlee188/ghpsdr3-alex
cd ghpsdr3-alex
git checkout máster
```

Ponemos el path a la versión de qt5 que estamos usando

```
$ export PATH=/opt/qt5/bin:$PATH
```

Después:

```
autoreconf -i
./configure
make -j4 all
sudo make install
```

Hay ejecutables optimizados en función de la CPU utilizada.

Los scripts optimizados siguientes reemplazarían el ./configure :

<code>./alex-avx-conf.sh</code>	Para CPU's i3,i5 y i7 que soportan Advanced Vector Extensions (AVX).
<code>./alex-openmp-conf.sh</code>	Para CPU's i3,i5 y i7 que soportan OpenMP .
<code>./alex-sse3-conf.sh</code>	Para cpu's con capacidad SSE.
<code>./alex-openmp-sse3-conf.sh</code>	Para cpu's con capacidad SSE y soporte OpenMP.

(Si no queremos instalar ahora los ficheros ejecutables, saltar el `sudo make install` y ejecutar el comando `cd` en los directorios siguientes, ejecutando desde allí los programas).

Terminal 1

```
$ cd ghpsdr3-alex/trunk/src/softrock
$ ./softrock ...options ...
```

Terminal 2

```
$ cd ghpsdr3-alex/trunk/src/dspserver
$ ./dspserver ...options...
```

Terminal 3

```
$ cd ghpsdr3-alex/trunk/src/QtRadio
$ ./QtRadio
```

Al compilar el servidor QtRadio Marconi, faltaban dentro del servidor las siguientes librerías, que se fueron añadiendo y se volvieron a ejecutar los comandos configure y make (al ser un software en desarrollo, futuras compilaciones requerirán ir solucionando posibles errores. El seguimiento del proyecto y soporte se sigue a través del foro sdr-widget en Google):

libffio3, se selecciona en el synaptics y se instala, incluyendo las librerías dev.

A continuación se vuelve a correr el ./configure .

Igual se hizo con las siguientes librerías:

pulse-simple , libpulse-dev .

Libusb-o.1 , libusb-dev .

Libconfig8

samplerate lib y header

ortp lib y header

Después de ejecutar ./configure

Se ejecutan los comandos:

```
make -j4 all
```

```
sudo make installing
```

Construcción de las extensiones Perseus:

Para construir el servidor Microtelecom Perseus debemos ir a la rama máster. Esta rama tiene sólo capacidad de recepción y sus protocolos cliente/servidor no son compatibles con la rama rxtx-event.

Primero descargar, compilar e instalar la librería libperseus-sdr:

```
sudo apt-get install wget
```

```
cd
```

```
wget http://libperseus-sdr.googlecode.com/files/libperseus_sdr-0.6.tar.gz
tar -zxvf libperseus_sdr-0.6.tar.gz
cd libperseus_sdr-0.6/
./configure && make
sudo make install
```

A continuación instalar la librería de prerrequisitos:

```
sudo apt-get install libboost-all-dev
```

Si ya hemos construido otra rama anteriormente, habrá primero que limpiar antes de construir:

```
cd ~/ghpsdr3-alex
make distcheck && sh cleanup.sh
```

En caso contrario, si se ha comenzado de cero, lo primero será clonar el repositorio:

```
git clone git://github.com/alexlee188/ghpsdr3-alex
```

Finalmente cambiamos a la “Rama máster” y reconstruimos cada parte, permitiendo el modo perseus Server :

```
cd ~/ghpsdr3-alex
git checkout máster
autoreconf -i && ./configure --enable-perseus=yes && make -j4
sudo make install
```

Para ejecutar el programa tenemos dos opciones, la primera opción, usando las siguientes líneas de comandos:

```
xterm -e 'perseus_server -d3 -s96000' &
xterm -e 'dspserver --lo 0' &
$(export QT_RADIO_NO_LOCAL_AUDIO=1 ; ./QtRadio )
```

En el menú del receptor de QtRadio, seleccionar: Configure, Pestaña Server. Aquí añadir host local usando la dirección 127.0.0.1. Presionando la tecla C, se puede comenzar el receptor.

La segunda opción es usar el script ghpsdr3-mgr.py (ghpsdr3launcher):

Pulsando en el icono Perseus, los tres procesos comenzarán automáticamente, y QtRadio conecta automáticamente al servidor Perseus.

Para refrescar a una nueva versión:

Bajar la nueva versión a construir:

```
$ git pull
$ make -j4 all
$ sudo make install
```

Si han ocurrido cambios importantes, y obtenemos errores al hacer el make, podemos hacer una construcción limpia, usando los siguientes scripts:

```
$ git pull
$ make distclean
$ sh cleanup.sh
$ autoreconf -i
$ ./configure
$ make -j4 all
$ sudo make install
```

Al ejecutar por primera vez el programa ghpsdr3-mgr.py dentro del servidor Marconi, faltaban las siguientes librerías que hubo que ir instalando con synaptic. Después de esto se recomenzaba el script a partir de ./configure :

```
Python-aptdaemon-gtk
python-central
python-gamin
python-glade2
python-indicate
python-kiwi
python-matplotlib
python-matplotlib-data
python-opengl
python-pygments
```

python-scipy
python-svn
python-telepathy
python-webkit
python-wnck
python-wxgtk2.8

2.3.10.4 Ejecución del programa cliente QtRadio.

QtRadio es una aplicación cliente/servidor que permite conectarse desde un equipo con QtRadio a cualquier servidor Linux, ejecutando el servidor de QtRadio.

Los clientes para las distintas plataformas (Linux, Android, Windows) se encuentran disponibles en la Web de QtRadio:

http://napan.ca/ghpsdr3/index.php/Main_Page

A continuación colocamos la imagen de la página principal de QtRadio.

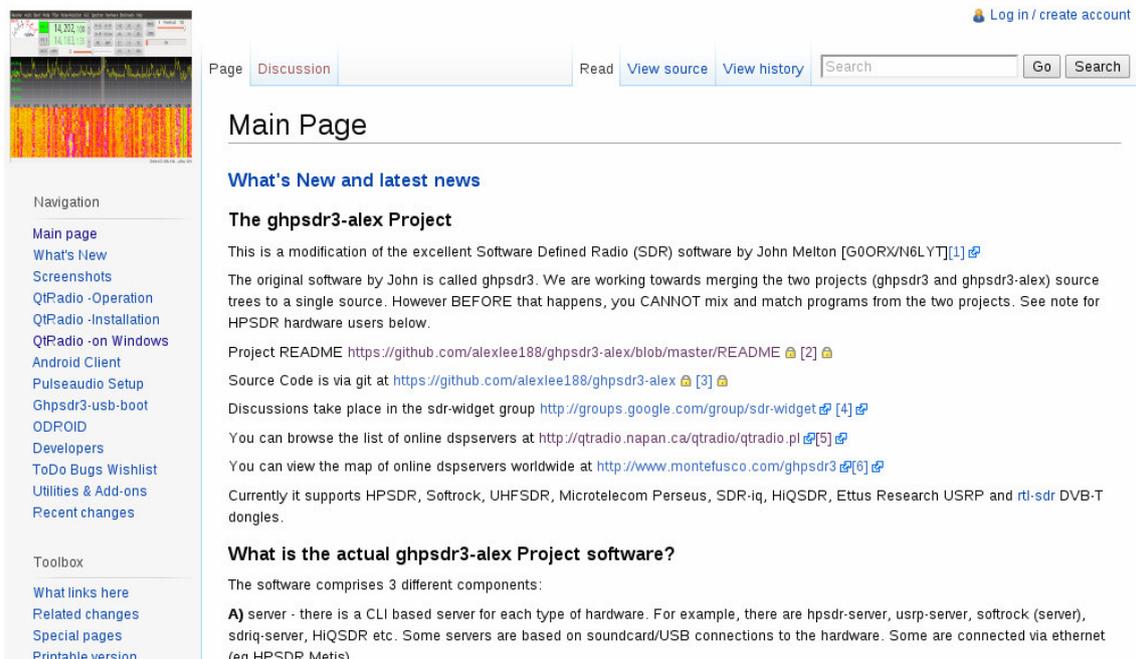
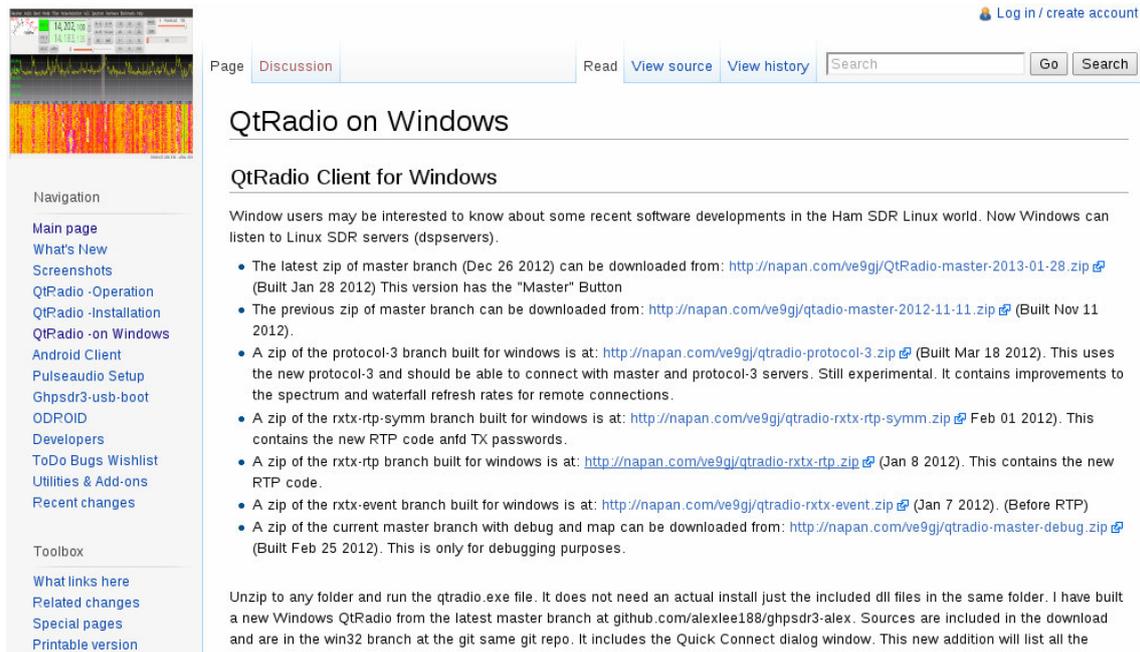


Figura 2.3.40 Página principal de Web QtRadio

Vamos a descargar el cliente QtRadio en Windows.

Para ello vamos a la página siguiente:

http://napan.ca/ghpsdr3/index.php/QtRadio_on_Windows



Log in / create account

Page Discussion Read View source View history Search Go Search

QtRadio on Windows

QtRadio Client for Windows

Window users may be interested to know about some recent software developments in the Ham SDR Linux world. Now Windows can listen to Linux SDR servers (dspervers).

- The latest zip of master branch (Dec 26 2012) can be downloaded from: <http://napan.com/ve9gj/QtRadio-master-2013-01-28.zip> (Built Jan 28 2012) This version has the "Master" Button
- The previous zip of master branch can be downloaded from: <http://napan.com/ve9gj/qtradio-master-2012-11-11.zip> (Built Nov 11 2012).
- A zip of the protocol-3 branch built for windows is at: <http://napan.com/ve9gj/qtradio-protocol-3.zip> (Built Mar 18 2012). This uses the new protocol-3 and should be able to connect with master and protocol-3 servers. Still experimental. It contains improvements to the spectrum and waterfall refresh rates for remote connections.
- A zip of the rtx-rtp-symm branch built for windows is at: <http://napan.com/ve9gj/qtradio-rtx-rtp-symm.zip> Feb 01 2012). This contains the new RTP code and TX passwords.
- A zip of the rtx-rtp branch built for windows is at: <http://napan.com/ve9gj/qtradio-rtx-rtp.zip> (Jan 8 2012). This contains the new RTP code.
- A zip of the rtx-event branch built for windows is at: <http://napan.com/ve9gj/qtradio-rtx-event.zip> (Jan 7 2012). (Before RTP)
- A zip of the current master branch with debug and map can be downloaded from: <http://napan.com/ve9gj/qtradio-master-debug.zip> (Built Feb 25 2012). This is only for debugging purposes.

Unzip to any folder and run the qtradio.exe file. It does not need an actual install just the included dll files in the same folder. I have built a new Windows QtRadio from the latest master branch at github.com/alexlee188/ghpsdr3-alex. Sources are included in the download and are in the win32 branch at the git same git repo. It includes the Quick Connect dialog window. This new addition will list all the

Figura 2.3.41 Página de descarga del cliente QtRadio para Windows.

Se selecciona el último programa desarrollado, y se descarga en el PC.

QtRadio on Windows

QtRadio Client for Windows

Window users may be interested to know about some recent software developments in the Ham SDR Linux world. Now Windows can listen to Linux SDR servers (dspervers).

- The latest zip of master branch (Dec 26 2012) can be downloaded from: <http://napan.com/ve9gj/QtRadio-master-2013-01-28.zip> (Built Jan 28 2012) This version has the "Master" Button

Figura 2.3.42 Descarga del último zip de QtRadio.

Se ha de descomprimir el fichero zip obtenido en el directorio Windows que deseemos, ahí estarán todos los ficheros requeridos.

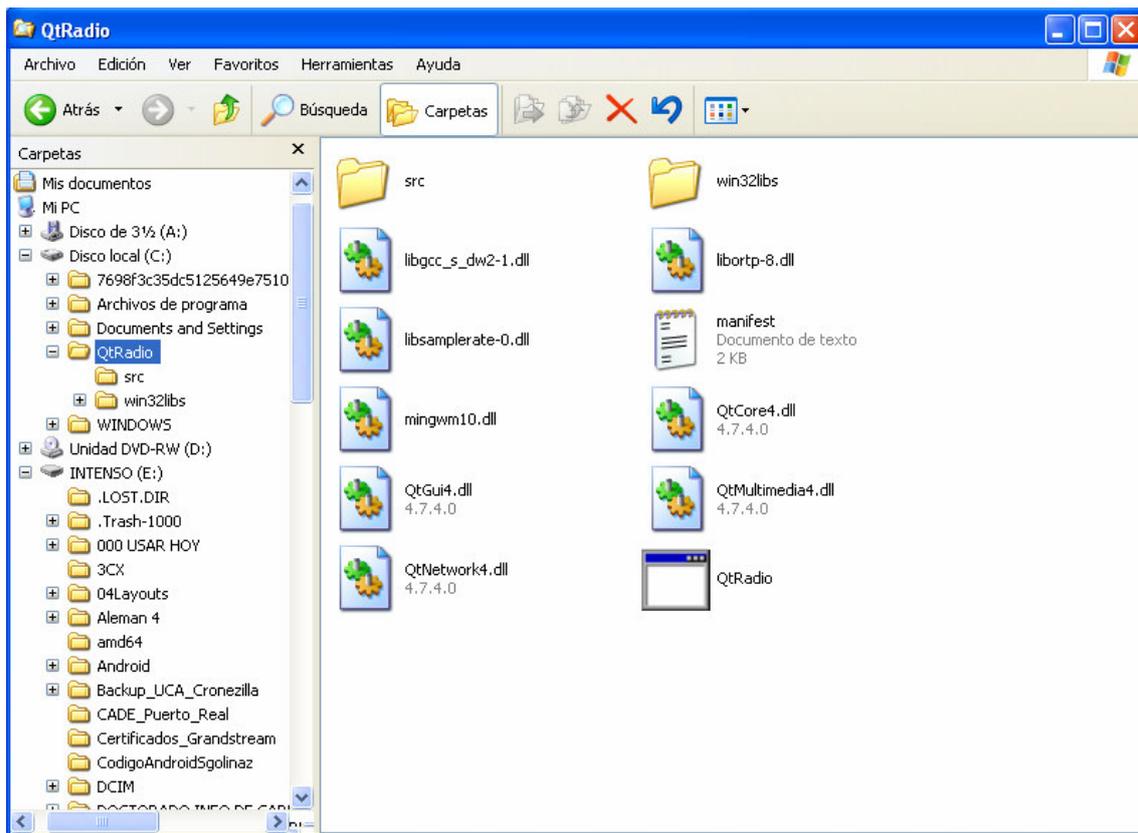


Figura 2.3.43 Directorio QtRadio

Colocar un enlace en el escritorio al programa ejecutable QtRadio.

Con esto hemos terminado la instalación del QtRadio para Windows.

Hemos de tener en cuenta que el PC ha de tener acceso a Internet y el cortafuegos de Windows ha de permitir que el programa QtRadio cliente se ejecute y se comunice con el exterior. Este programa usa el puerto 8000 para comunicarse con el servidor de QtRadio.

A su vez, el puerto 8000 ha sido abierto por el CITI de la UCA en el cortafuego de la Universidad para la dirección Marconi.uca.es para esta Tesis Doctoral con el fin de que se puedan resolver las conexiones cliente/servidor sobre el servidor Marconi.

Picando sobre el icono de QtRadio situado por nosotros en el escritorio se nos abre este programa.

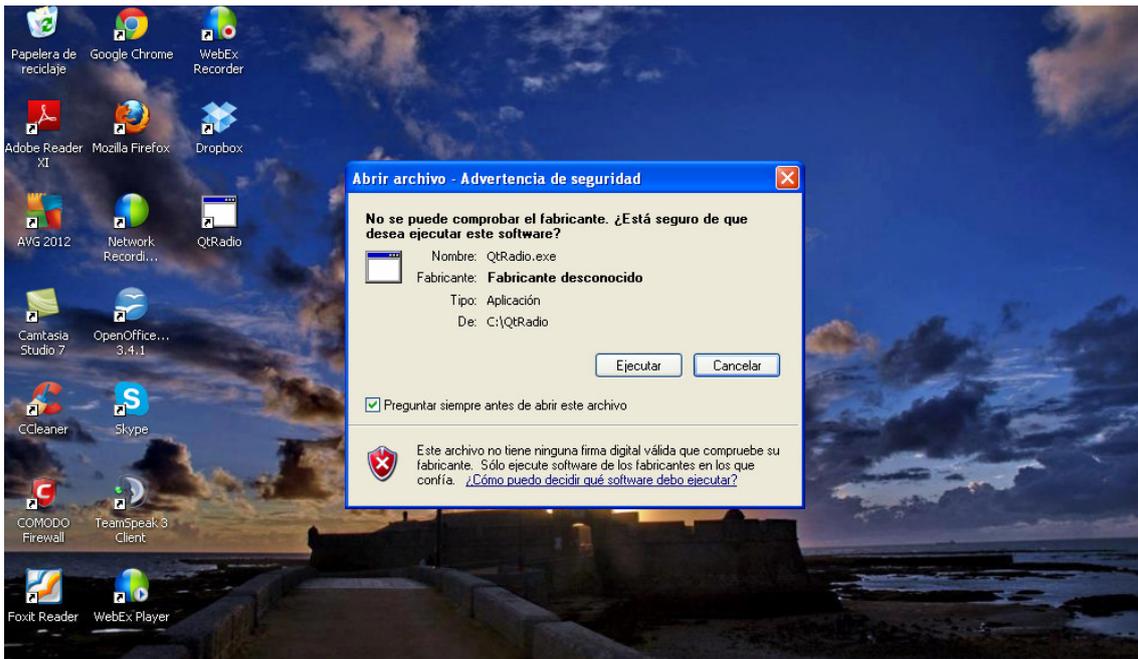


Figura 2.3.44 Programa QtRadio.

A continuación vemos el programa QtRadio conectado al servidor marconi.uca.es del grupo de investigación SSyCN.

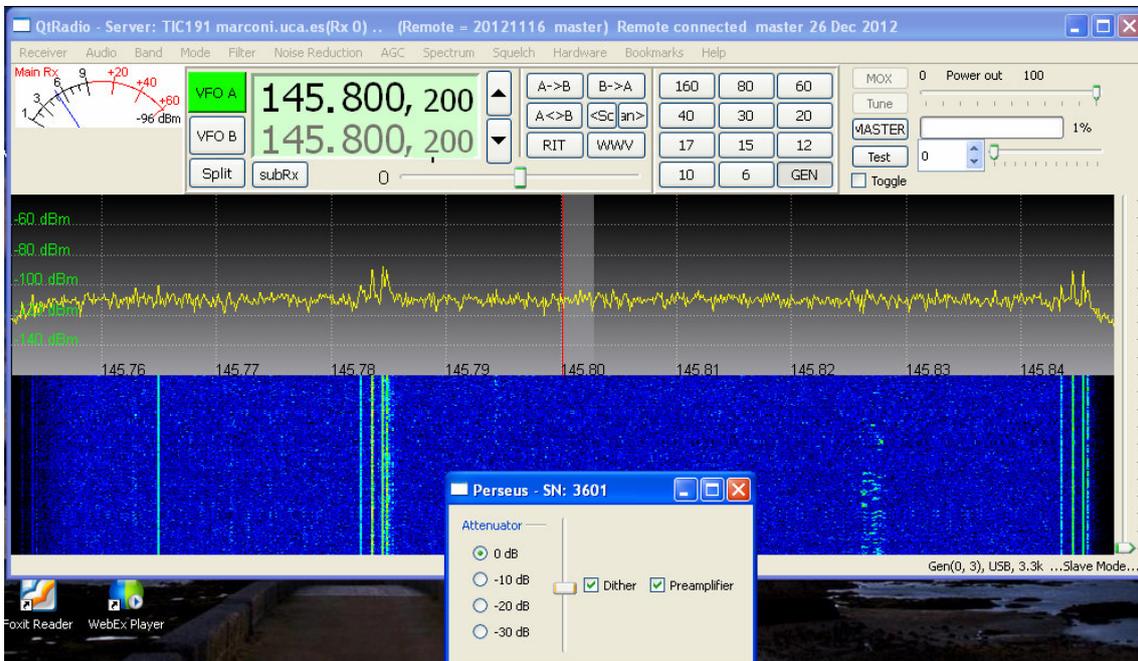


Figura 2.3.45 Pantalla del QtRadio conectado al servidor marconi.uca.es

Como opciones básicas dentro de este programa podemos distinguir en la barra de menú las siguientes:

Receiver:

Connect /Disconnect: para conectarse al servidor.

Quick Server List: despliega el menú con los server disponibles para conectarse.

Configure: permite configurar las opciones del cliente:

Server: Nodo a conectarse

Audio: Configuración de audio del PC

Display: Configuración del display y cascada.

DSP: Condiciones del DSP

Transmit: Configuración para transmitir, usuario/clave

Audio: Condiciones generales de la ganancia de audiofrecuencia.

Band: Selección de bandas de frecuencias.

Mode: Modos de transmisión o recepción (LSB, USB, CW, AM).

Filter: Filtros a seleccionar.

Noise Reduction: Reducción del ruido.

AGC, Spectrum: Configuración del control automático de ganancia y espectro.

Squelch: Habilita o deshabilita el silenciador.

A continuación vemos la ventana de selección de servidores:

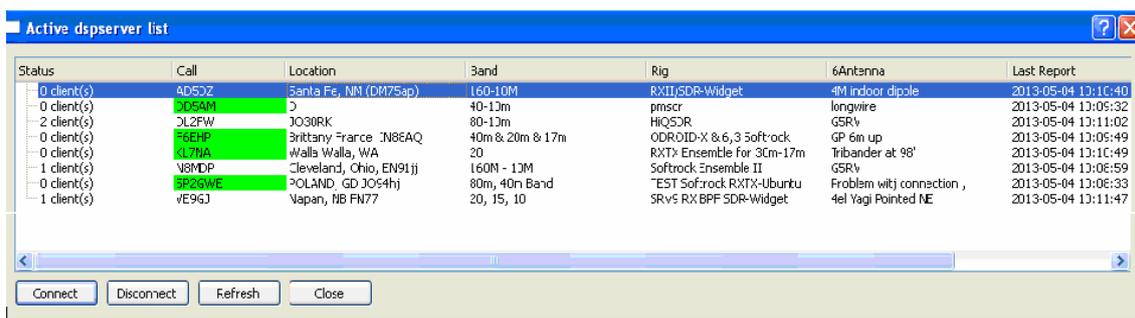


Figura 2.3.46 Ventana de selección de servidores de QtRadio

Una vez seleccionado un servidor remoto, damos a la opción de conectar y el programa QtRadio cliente se conecta al servidor seleccionado.

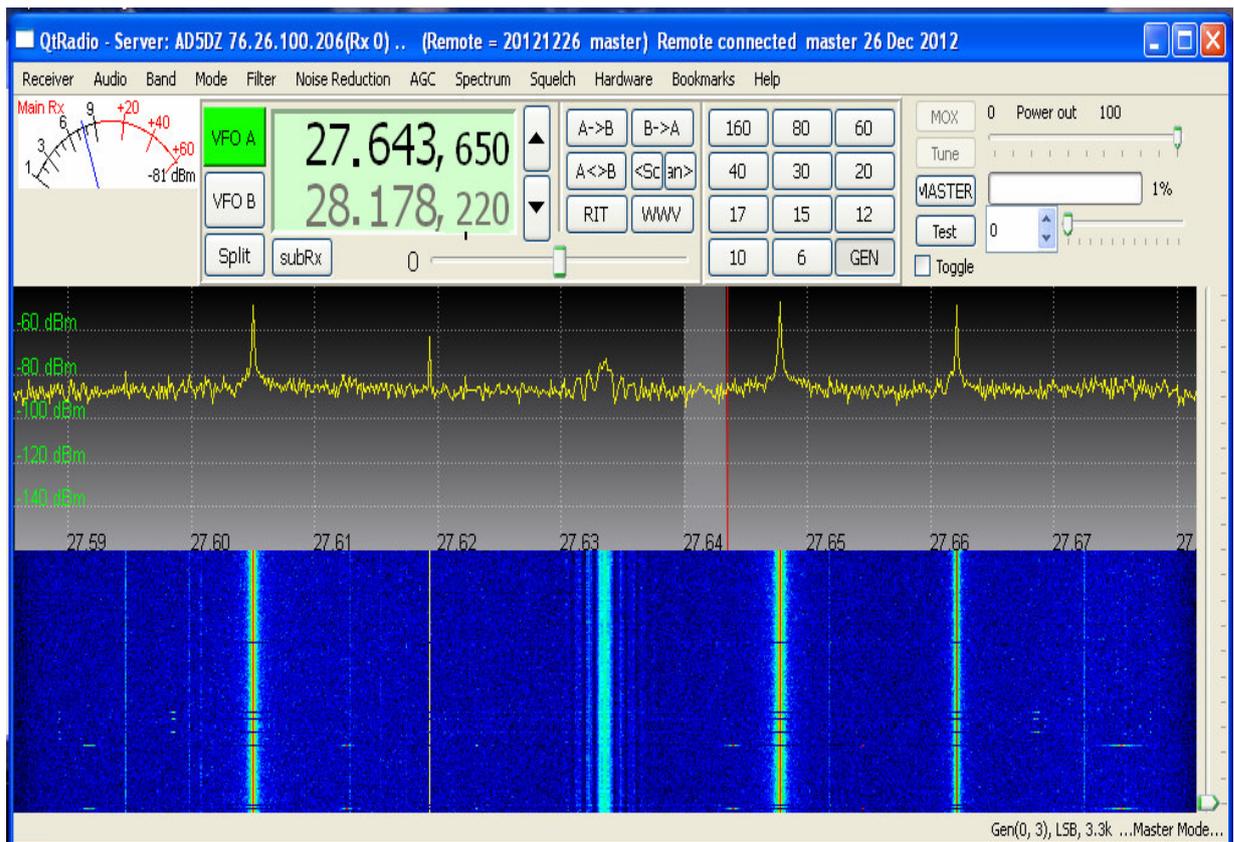


Figura 2.3.47 QtRadio conectado a servidor a través de Internet.

Una vez conectados al servidor, podemos tener preseleccionadas dos estaciones: VFO A/VFO B (según se observa en figura 2.3.47).

A su vez podemos hacer una detección visual de señales emitidas y sintonizarlas.

En la figura 2.3.47 se observa la clásica cascada, típica de todos los programas SDR, así como la amplitud de la señal.

También en la parte superior de la pantalla, tenemos la dirección IP del servidor al que estamos conectados.

2.3.10.5 Ejecución del programa servidor QtRadio.

Una vez hemos instalado QtRadio en nuestro servidor, pasamos a describir la forma de utilizarlo.

Como punto a tener en cuenta dentro del Servidor Marconi es que al ejecutar QtRadio se inician los servidores Perseus, DSPServer y la interfaz gráfica.

La interfaz gráfica necesita tener acceso a la tarjeta de sonido, ya que recibe los datos, los decodifica y los muestra. El servidor Marconi tiene definido por motivos elementales de seguridad, dos usuarios, cada uno de ellos ejecutando una aplicación (WebSDR y QtRadio). Para lograr que ambas aplicaciones puedan ejecutarse, se ha instalado una segunda tarjeta de sonido externa (estéreo).

Se ha comprobado que la aplicación QtRadio es más restrictiva en el uso de la tarjeta de sonido. Por ello, en primer lugar, hay que lanzar la aplicación QtRadio, cogiendo ésta la tarjeta de sonido del bus PCI del PC. En segundo lugar, se lanza la aplicación WebSDR que utiliza la tarjeta de sonido externa. Ambas tarjetas son de 96 KHz.

La aplicación Pulse Audio de Ubuntu, maneja el sonido para ambas aplicaciones, por lo que ha de colocarse el volumen de entrada de sonido, de forma que no sature la representación del espectro en la página de WebSDR.

En apéndices se muestra última instalación para la versión SSL del servidor QtRadio.

Al finalizar la instalación nos debe aparecer el icono CQ en el área de lanzadores de nuestro equipo Ubuntu.

Ver la figura siguiente (2.3.48):



Figura 2.3.48 Detalle de lanzadores de Ubuntu.

Una vez seleccionado CQ, aparece la interfaz gráfica del servidor QtRadio, desarrollada en Python.

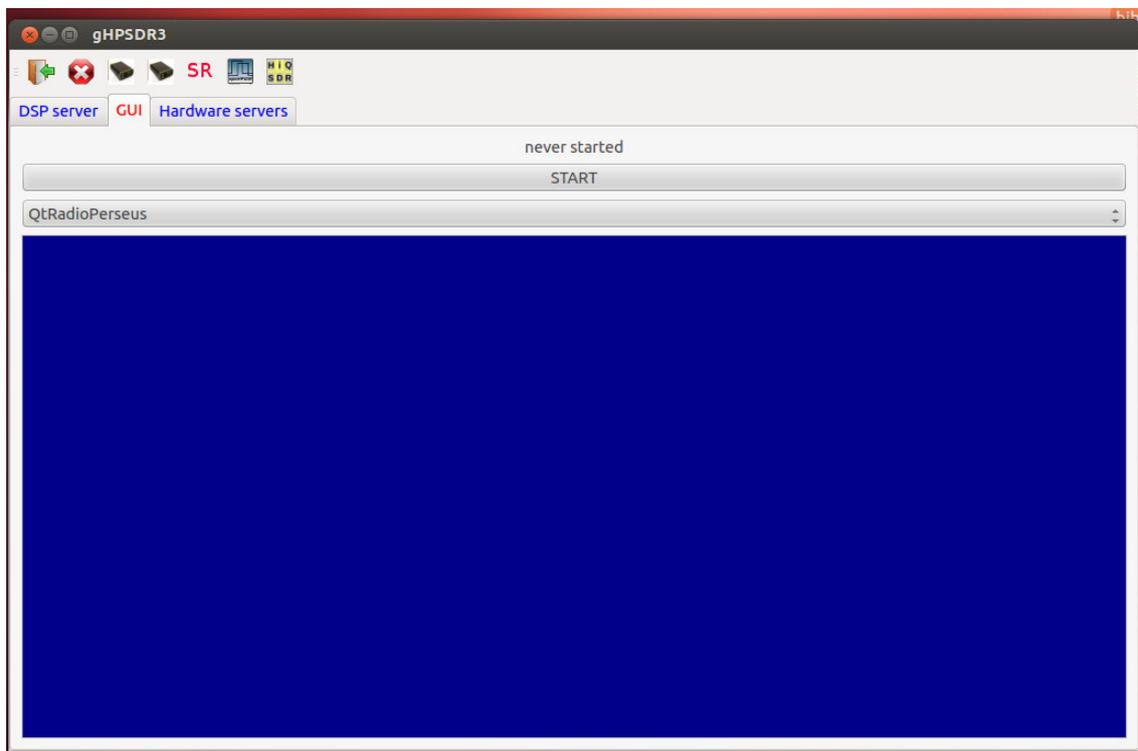


Figura 2.3.49 Pantalla principal servidor QtRadio

En la figura superior (Figura 2.3.49), podemos distinguir la pantalla principal de QtRadio, donde se observan las siguientes partes:

Barra de iconos:

- Opción carpeta para ir a los ficheros.
- Opción de salida del programa.
- Icono receptor SDR-IQ, pulsando sobre él se lanza el servidor de SDR-IQ.
- Icono receptor Perseus, pulsando sobre él se lanza el servidor perseus.
- Icono receptor Softrock, pulsando sobre él se lanza el servidor softrock.
- Icono receptor HPSDR, pulsando sobre él se lanza el servidor HPSDR.
- Icono receptor HiQSDR, pulsando sobre él se lanza el servidor de HiQSDR.

Pestañas de la aplicación:

A continuación, tenemos el área de pestañas de la aplicación, donde se distinguen las

tres partes de QtRadio:

- Hardware servers: Conexión con los distintos servidores.
- DSP Server: Interfaz común para todos los clientes y receptores soportados.
- GUI: Interfaz gráfica del cliente Linux.

Una vez descritas cada una de las partes del servidor, vamos a detallar la forma de lanzamiento, aplicación a aplicación, con objeto de ver cada una de las partes.

Empezaremos lanzando el servidor hardware:

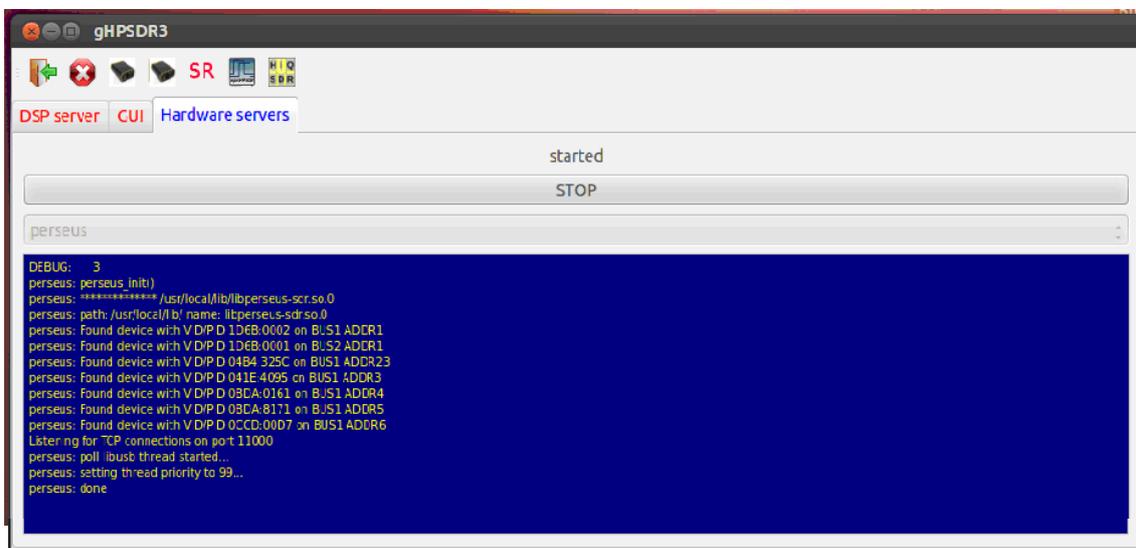


Figura 2.3.50 Detalle del lanzamiento de la interfaz para Perseus.

Para ello vamos a la pestaña “Hardware Servers”, picamos en el área desplegable y seleccionamos Perseus.

A continuación presionamos “start”.

Esto lanzará el programa que conecta con el receptor Perseus. Se distinguen las siguientes líneas si se ha conectado adecuadamente:

```
Listening for TCP connections on port 11000
Perseus: poll libusb thread started .....
Perseus: setting thread priority to 99...
Perseus: done
```

Esto último nos indica que la interfaz Perseus se ha lanzado adecuadamente y que está a la escucha en el puerto 11000.

A continuación lanzamos el servidor dsp, para ello vamos a la pestaña “DSP Server”.

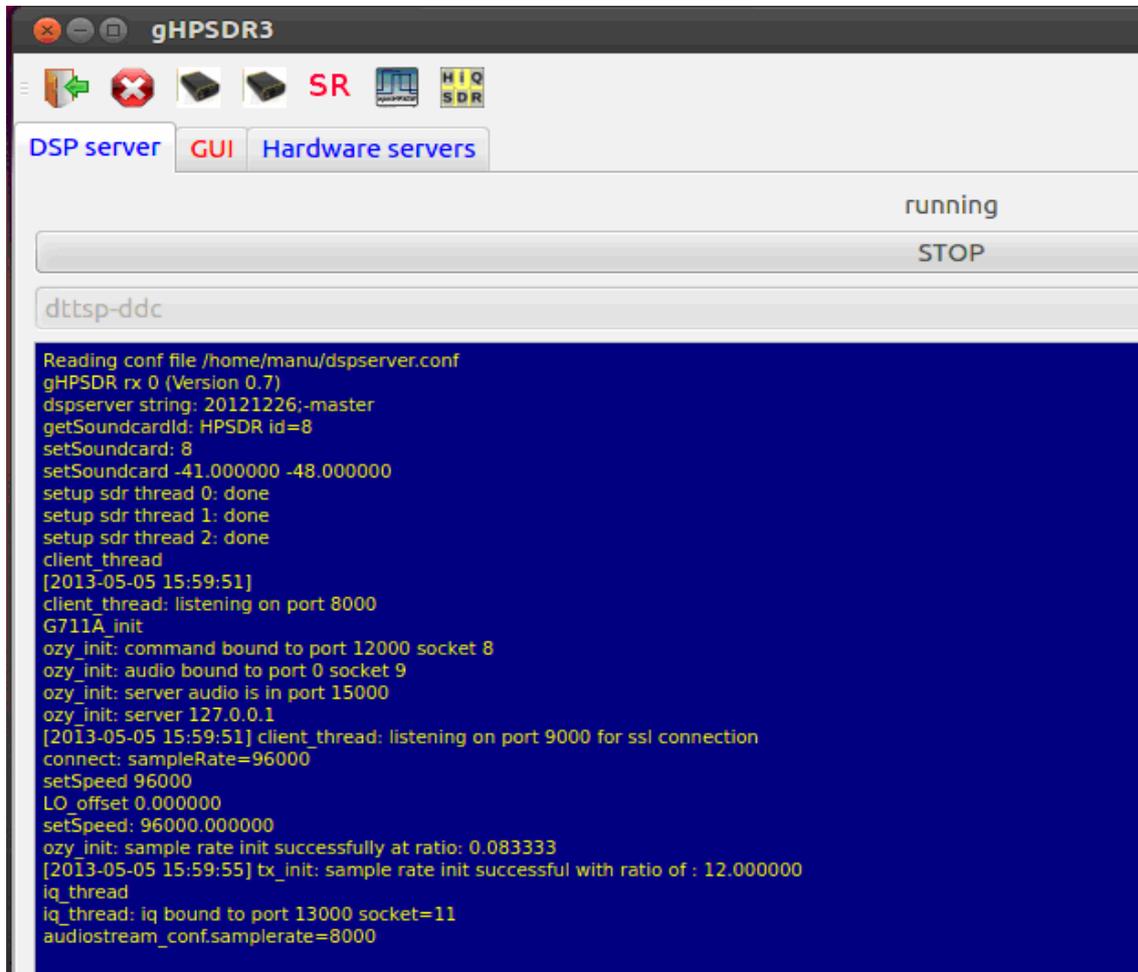


Figura 2.3.51 Servidor DSP de la aplicación servidor de QtRadio

En la pestaña DSP Server, escogemos en el desplegable dttsp-ddc, que corresponde a la opción Perseus, seguidamente, seleccionamos start.

Aparece la figura anterior (2.3.51), en ella veremos que se lee el fichero dspserver.conf y posteriormente aparecen las siguientes líneas que nos indican que se ha lanzado correctamente:

Client_thread: listening on port 8000

Audiostream_conf samplerate = 8000

La primera línea nos indica que se ha lanzando adecuadamente y que se está escuchando el server en el puerto 8000.

La segunda línea nos indica que se envía la cadena de audio con una frecuencia de muestreo de 8000.

Por último lanzamos la interfaz gráfica:



Figura 2.3.52 Lanzamiento de la interfaz gráfica del Perseus.

Para ello seleccionamos la pestaña GUI y la opción en el desplegable QtRadioPerseus. Seleccionamos "Start" y se lanzará la aplicación GUI, apareciendo la interfaz gráfica de QtRadio.

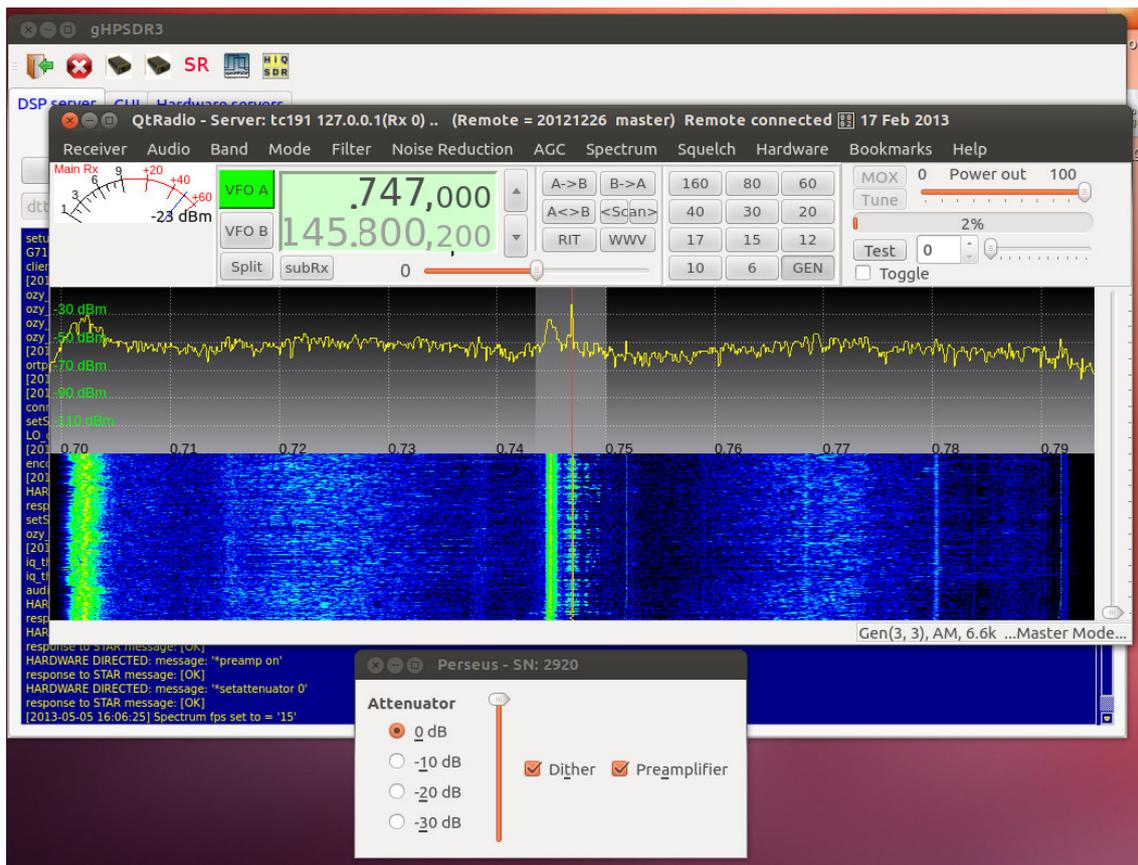


Figura 2.3.53 Interfaz gráfica QtRadio

La interfaz gráfica del QtRadio presenta la clásica cascada y la señal recibida.

En la parte superior se encuentra el servidor al que se conecta el cliente QtRadio.

2.3.10.6 Configuración del servidor QtRadio.

En el directorio personal de instalación del QtRadio, se encuentra el fichero dspserver.conf. Este fichero contiene información básica para el servidor QtRadio.

A continuación se hace una breve descripción de las líneas:

Todas las líneas que comienzan con # son comentarios.

- # Simple config file for ghpsdr3's dspserver.
- # default file is located at ~/dspserver.conf when dsp server is started with --share
- # The information below will be supplied to a web database which will aid QtRadio
- # users find active dspserver to connect to. You may also see the current list at
- # <http://napan.ca/qtradio/qtradio.pl>

```
# valid fields are call, location, band, rig and ant
# lines must end with ; and any characters after a # is a comment and ignored
# field values must be enclosed with " ie: "xxxx"
# This default file will be created if dspserver is started with the --share option
and ~/dspserver.cfg does not exist
# You may also start dspserver with an alternate config file by starting dspserver
with --shareconfig /home/alternate_filename.conf
# Note field names are all lowercase!
```

A continuación las líneas de descripción:

```
call, location, band, rig, ant.
```

Estas líneas aparecen en la página de QtRadio.

```
call = "tic191";
location = "Puerto Real";
band = "NA";
rig = "Perseus";
ant = "NA";
share = "yes"; # Can be yes, no , yes indica que se haga visible el servidor.
lookupcountry = "no"; # Can be yes, no , yes muestra el país
```

A continuación vienen las instrucciones para transmisión:

```
##### Following are new TX options #####

tx = "no"; #Can be: no, yes, password

#ve9gj = "secretpassword"; #add users/passwords one per line (Remove leading
# ! max 20 characters each

groupnames = ["txrules1"]; #add group or ruleset names in ["name1",
"name2"]; format (max 20 characters each

# Add user names in ["call1", "call2"]; format to list members for each
groupname above with an suffix of "_members"
```

```
txrules1_members = ["ve9gj", "call2"];

# Rule sets are defined as group or ruleset name =( ("mode", StartFreq in Mhz,
End Freq in Mhz),("mode", StartFreq in Mhz, End Freq in Mhz) );

# Valid modes are * SSB, CW, AM, DIG, FM, DRM ,SAM, SPEC Where *
means any mode is OK

# The two rules below allow any mode on 20M and CW only on the bottom
100Khz of 80M

# You can make as many rules and rulesets as you wish. The first matching rule
will allow TX

txrules1 = (

    ("*",14.0,14.350), # mode, StartFreq Mhz, EndFreq Mhz

    ("CW",3.5,3.6)

    );
```

2.3.11. Arquitectura SDR-Radio

2.3.11.1 Introducción

La aplicación SDR-Radio se ha tenido en cuenta dentro de esta Tesis por ser una de las pocas que implementan la arquitectura cliente-servidor, además de por ser una de las pocas aplicaciones que desarrollándose en Windows funciona de una forma fluida y permite cumplir con los objetivos de esta tesis (proveer acceso distribuido a los equipos de recepción).

Hay que destacar que esta aplicación no transmite, como es el caso de QtRadio. Además la gestión conjunta de todos los receptores a ella conectada, requiere el uso de la versión beta en desarrollo así como de la investigación dentro de su grupo de trabajo (SDR-Radio), y de la aplicación de interfaces adicionales.

Su documentación describe de forma adecuada cómo se implementa la aplicación en el nivel ISO de presentación y cómo se manejan los equipos de forma remota con los comandos HTML.

Con objeto de cargar de contenido esta tesis se ha participado dentro del grupo de desarrollo que lidera Simon Brown (SDR-Radio) y se ha seguido de forma continua los desarrollos de éste. Se ha implementado dentro del sistema Marconi un servidor con la versión beta (en desarrollo) de este software (ver 2.1).

A su vez, se han tenido que seguir e investigar dentro del grupo Osmocom, así como en el grupo SDR-Radio las configuraciones necesarias para que este software manejara los 4 receptores (Funcube, Softrock, RTL-SDR, SDRIQ) conectados a él. La versión comercial soporta solo el receptor SDRIQ, lo que no cumple con los objetivos perseguidos.

Se ha logrado que este servidor permita dentro de la UCA el acceso a cada uno de estos receptores. Igual que en las otras aplicaciones, el software está en desarrollo, requiriendo su funcionamiento de un seguimiento continuo de los avances dentro del grupo así como en la comunidad que constantemente provee de soluciones innovadoras.

La implementación del modelo aquí descrito requiere del conocimiento básico de los firewall y router del camino a través del que se conecta el cliente al servidor. Todo este

tipo de pruebas se han llevado a cabo usando prototipos locales de cliente-servidor, antes de su instalación en el sistema Marconi.

Siguiendo las descripciones anexas se puede llegar a instalar otro sistema cliente-servidor.

El proyecto SDR-Radio consiste en dos ejecutables:

- La consola, que ofrece acceso directo a:

Interfaz de usuario

A las radios SDR conectadas al mismo ordenador que la consola.

- El servidor, usado por la consola para conectarse a un ordenador remoto.

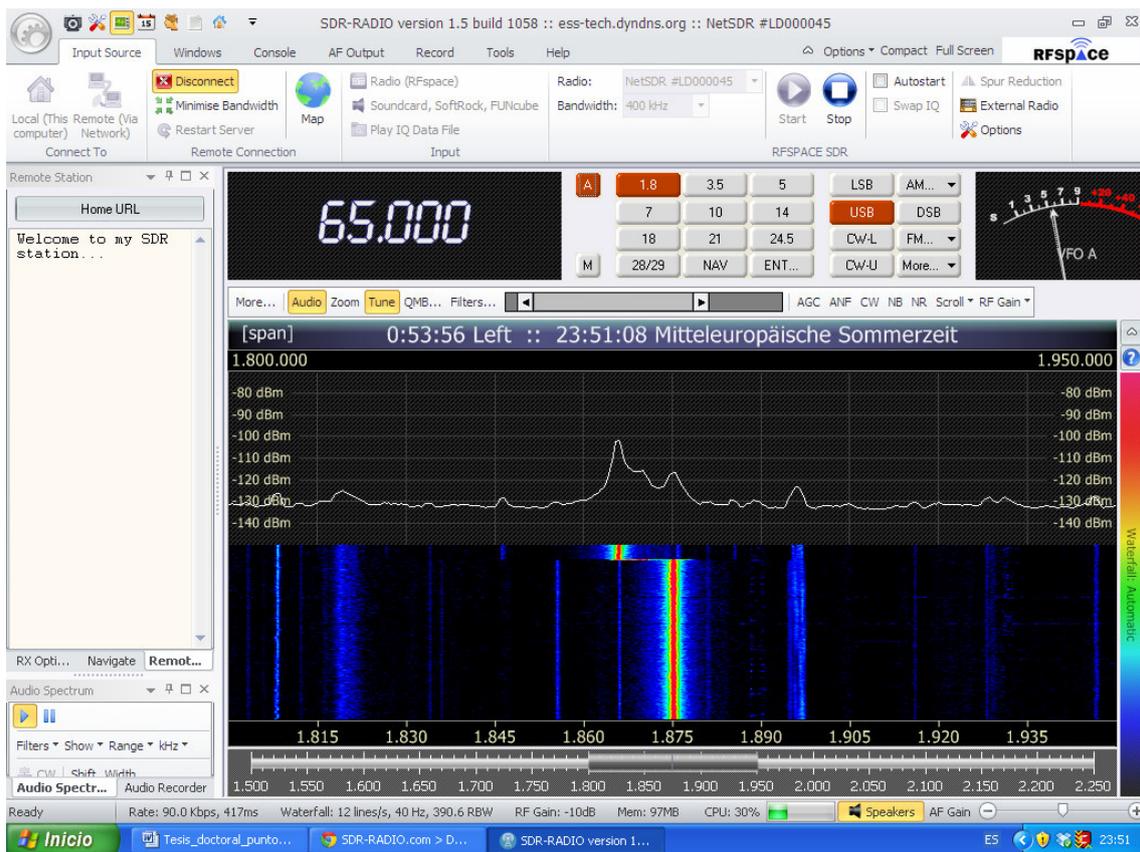


Figura 2.3.54 Consola SDR-Radio

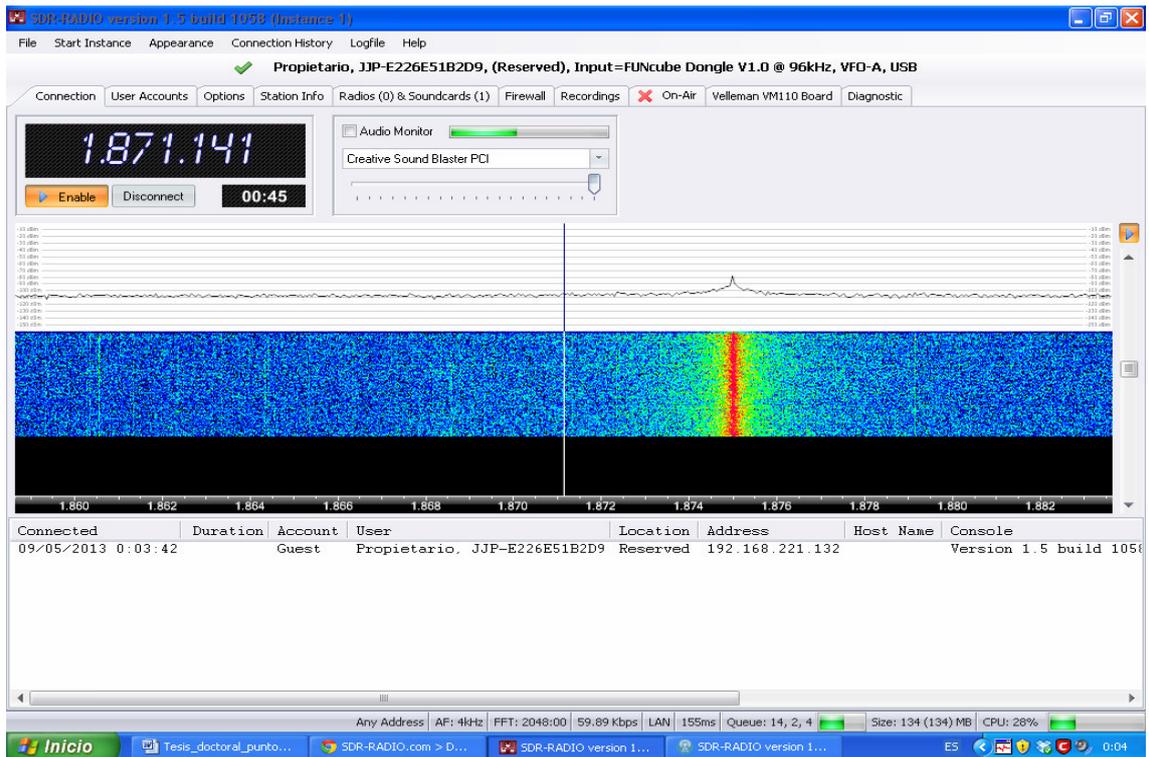


Figura 2.3.55 Servidor de SDR-Radio con Funcube conectado

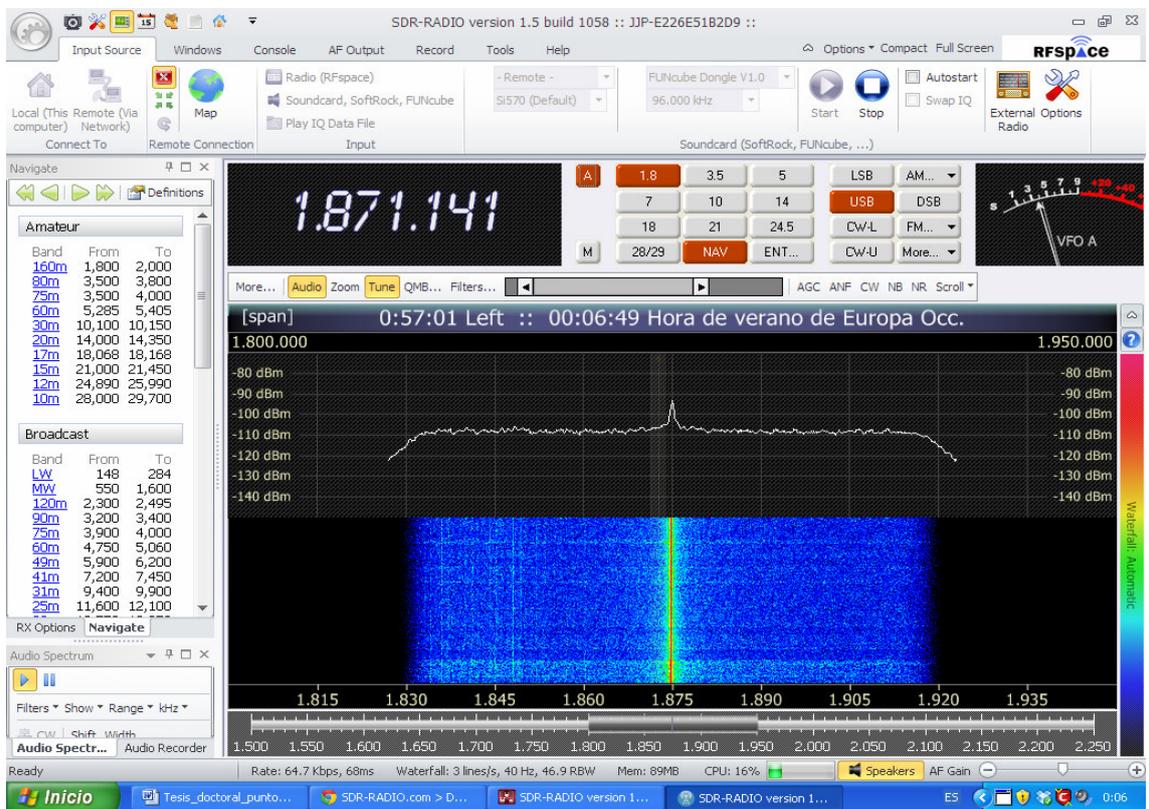


Figura 2.3.56 Consola conectada a servidor de imagen anterior

En las dos imágenes anteriores podemos ver el servidor SDR-Radio con un receptor Funcube Dongle Pro, así como la consola conectada a este servidor.

2.3.11.2 SDR-Radio, conexión cliente-servidor.

El protocolo cliente servidor usado es TCP/IP para todas las comunicaciones. Ambos, la consola y el servidor, usan programación asíncrona de los puertos.

Todos los comandos y respuestas se encapsulan en XML. El audio y datos FTT se envían después como datos.

A continuación, describimos los comandos utilizados en una conexión cliente/servidor.

Primero: Conexión TCP/IP

Como primer paso se establece una conexión TCP/IP entre el cliente y el servidor.

El servidor acepta o rechaza la petición de conexión.

Segundo: Login

El cliente debe ahora hacer login en el servidor. Para ello debe enviar un user id (usuario) y un password (clave) encriptado.

1. Si la conexión TCP/IP se acepta, el cliente envía un comando Start/Login al cliente.
2. El cliente envía un comando ConnectRequest .
3. El servidor envía un comando ConnectUseKey que contiene la key usada por el cliente al encriptar el password (clave).
4. El cliente envía un comando ConnectLogin que contiene el username, password y key (dados anteriormente). El username y password debe coincidir con los definidos dentro del servidor.
5. El servidor manda ConnectLoginStatus con el estado, normalmente OK, a menos que el cliente haya enviado un username/password inválidos.

Tercero: Petición de dispositivos

El cliente pide ahora una lista de dispositivos utilizables .

1. El cliente envía el comando `GetInputDevices`.
2. El servidor responde con `ReturnInputDevices` y una lista de Radios SDR y tarjetas de sonido (las tarjetas de sonido se usan para los receptores Softrock y Funcube).

Cuarto : Comenzar Radio:

El cliente comienza el SDR-Radio.

1. El comando "Start" contiene la radio a comenzar y algunos parámetros para usar.
2. El Servidor devuelve el estado en el comando "ServerStatus".

Quinto: FFT y Audio

El servidor manda ahora datos FFT y audio.

La descripción detallada de estos pasos se encuentra en la sección "Apéndices".

2.3.11.3 Servidor de SDR-Radio.

El servidor no requiere mucha potencia de CPU – un Intel Atom a 1.6 GHz que se puede encontrar en los pequeños portátiles es suficiente, un microprocesador i3 (o superior) asegurará el soporte para las capacidades avanzadas.

El requisito típico de ancho de banda es menos de 30 kbytes / segundo; con una conexión ADSL estándar es suficiente.

Se necesita una dirección IP pública, si no tenemos una dirección IP estática, se puede usar una solución como no-ip.com (servidor de direcciones virtuales).

Se están aceptando entradas TCP / IP así que se debe configurar el firewall, tanto el firewall de Windows como el firewall externo.

La aplicación SDR-Radio, tiene dos links principales como se muestra en la figura 2.3.57.

Console (o consola de la aplicación).

Remote Server (o servidor remoto).

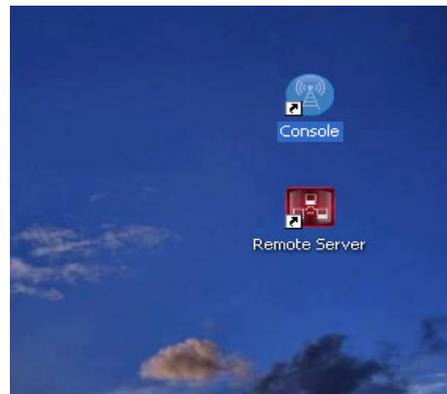


Figura 2.3.57 Iconos de lanzamiento de las aplicaciones SDR-Radio.

Para comenzar el servidor se ejecuta la aplicación “Remote Server” y una vez abierta ésta, picando sobre el icono, nos aparece la pantalla descrita en la figura 2.3.58.

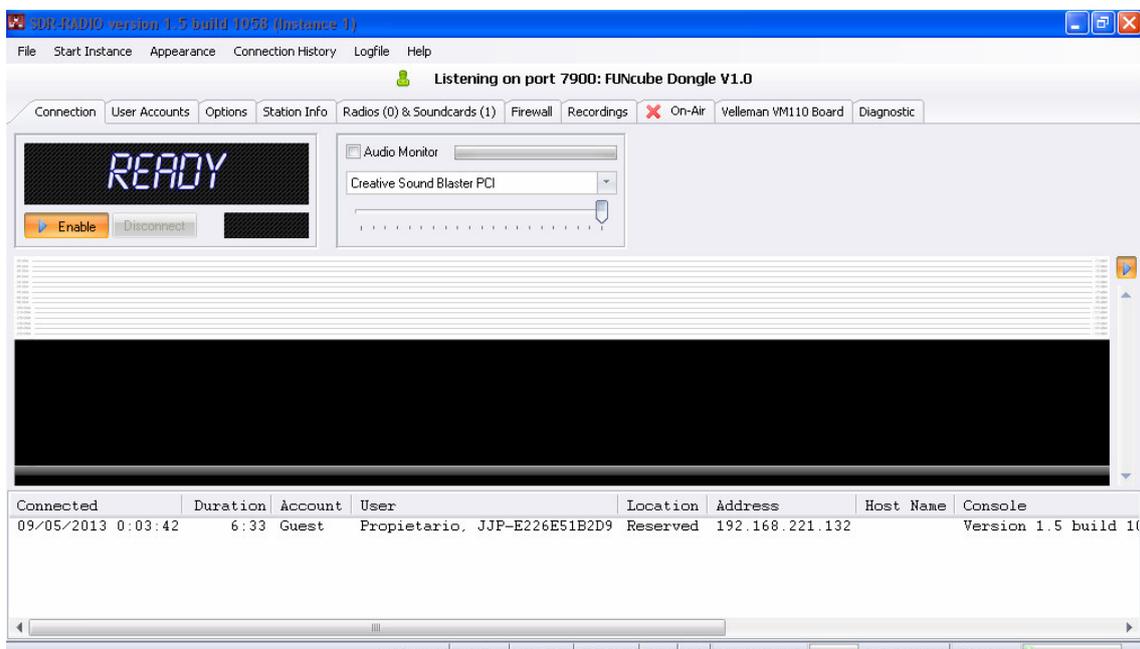


Figura 2.3.58 Pantalla principal del servidor SDR-Radio.

En ella se pueden distinguir las siguientes áreas:

Barra superior:

Compuesta por las opciones: File, Start Instance, Appearance, Connection History, Logfile, Help.

Las opciones File, Appearance, Connection History, Logfile y Help, son opciones de tipo general. La opción “Start Instance” permite abrir una nueva conexión a unos de los dispositivos conectados al servidor.

La versión beta de este producto (versión 2), trabaja de una forma diferente, ofreciendo en una sola conexión, todos los dispositivos a elección de la persona que se conecta al servidor.

Pestañas de configuración:

Pestaña “Connection”:

Presenta la información de los clientes conectados al servidor.

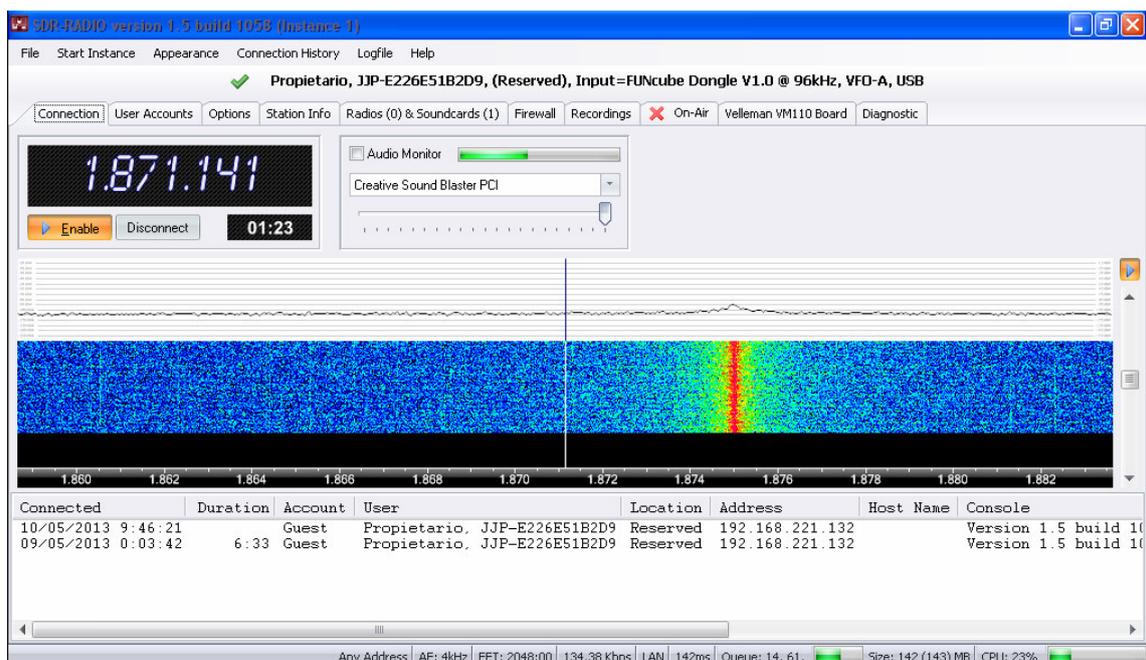


Figura 2.3.59 Pantalla connection con un cliente conectado (Funcube).

En la figura 2.3.59 se ve la imagen de la pestaña de conexiones, donde el cliente conectado es “Guest” o huésped, se conecta desde el equipo con el nombre JJP-E226E51B2D9, la dirección de conexión es la 192.168.221.132.

Se observa que el cliente obtiene conexión al dispositivo SDR Funcube dongle con un ancho de banda de 96 KHz, a su vez este dispositivo se encuentra conectado al servidor por USB.

La cascada muestra la información transmitida al cliente “Guest”.

Pestaña “User Accounts”:

Permite definir los usuarios, así como las características de éstos.

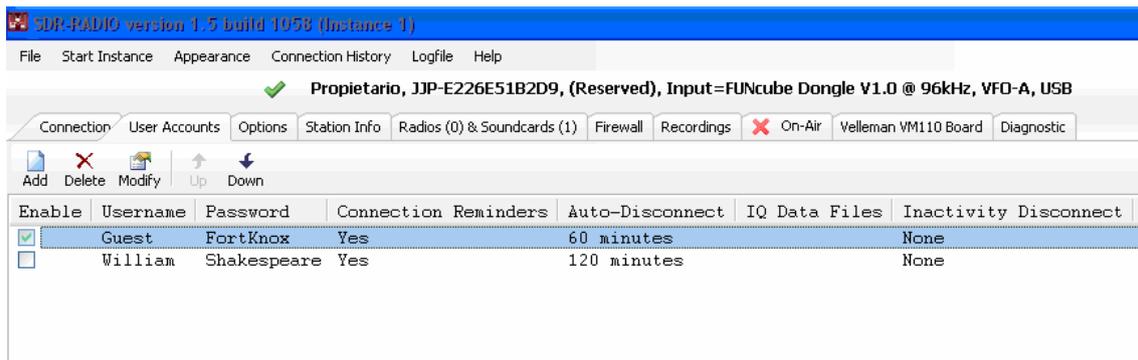


Figura 2.3.60 Configuración de usuarios. Pestaña user accounts.

La pestaña de configuración de usuarios permite hacer las siguientes funciones:

Add (añadir)

Delete (borrar)

Modify (modificar)

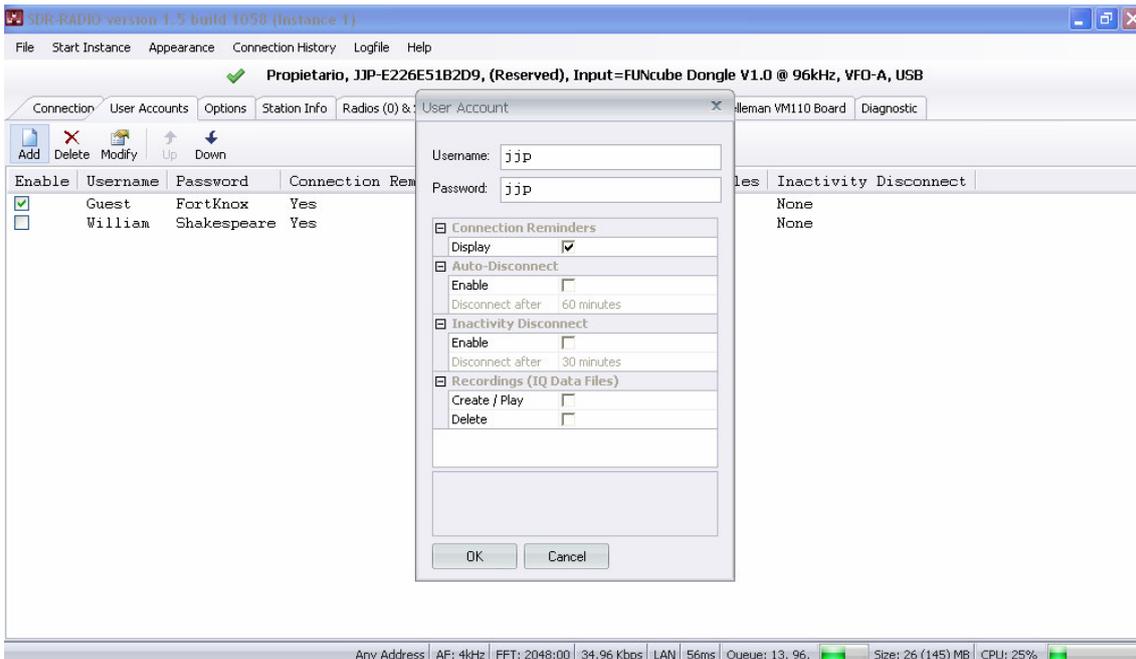


Figura 2.3.61 Creación de un nuevo usuario.

Al añadir un nuevo usuario, introducimos su nombre y contraseña, así como otras opciones que veremos a continuación dentro de la opción de modificación de la cuenta de usuario.

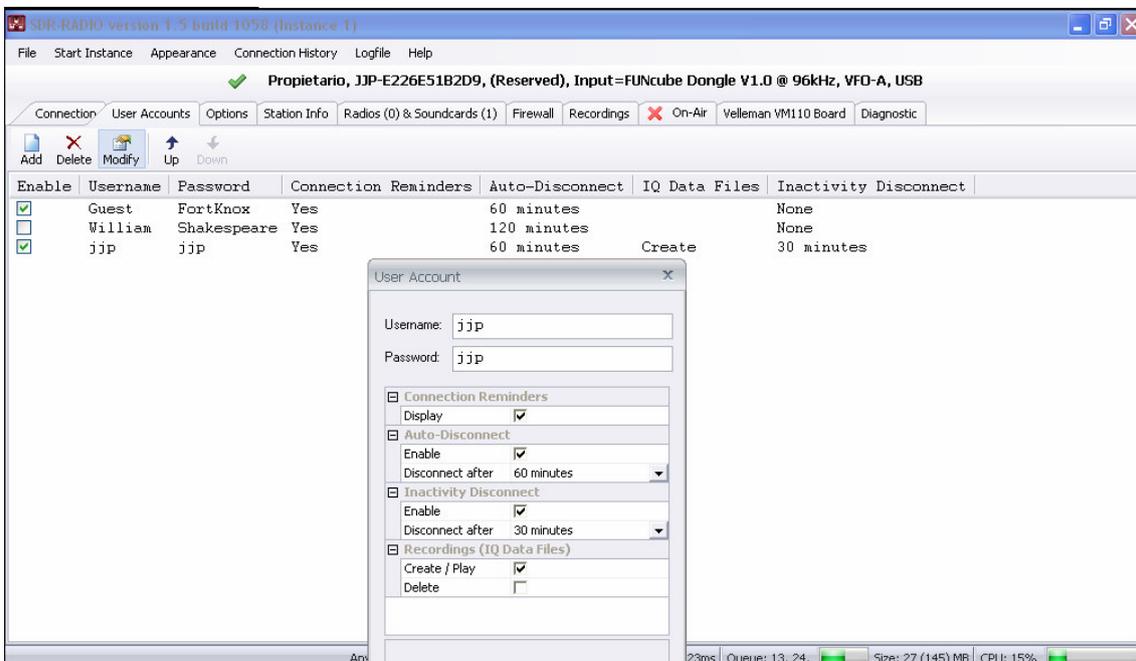


Figura 2.3.62 Modificación de la cuenta de usuario.

Al seleccionar la opción “modify”, podemos modificar las características de la cuenta de los usuarios.

En el ejemplo de la figura para el usuario jjp, hemos definido:

- Recordatorio del tiempo de conexión
- Desconexión después de, en este caso 60 minutos de conexión.
- Desconexión después de, en este caso 30 minutos de inactividad.
- Posibilidad de grabar o reproducir los datos I/Q recibidos.

Además de las opciones de crear y modificar, también existen en esta pestaña las opciones de borrar y subir o bajar el usuario en la cola de conexión.

Pestaña “options”:

La pestaña options permite definir la opciones principales del servidor SDR. Es la pestaña principal.

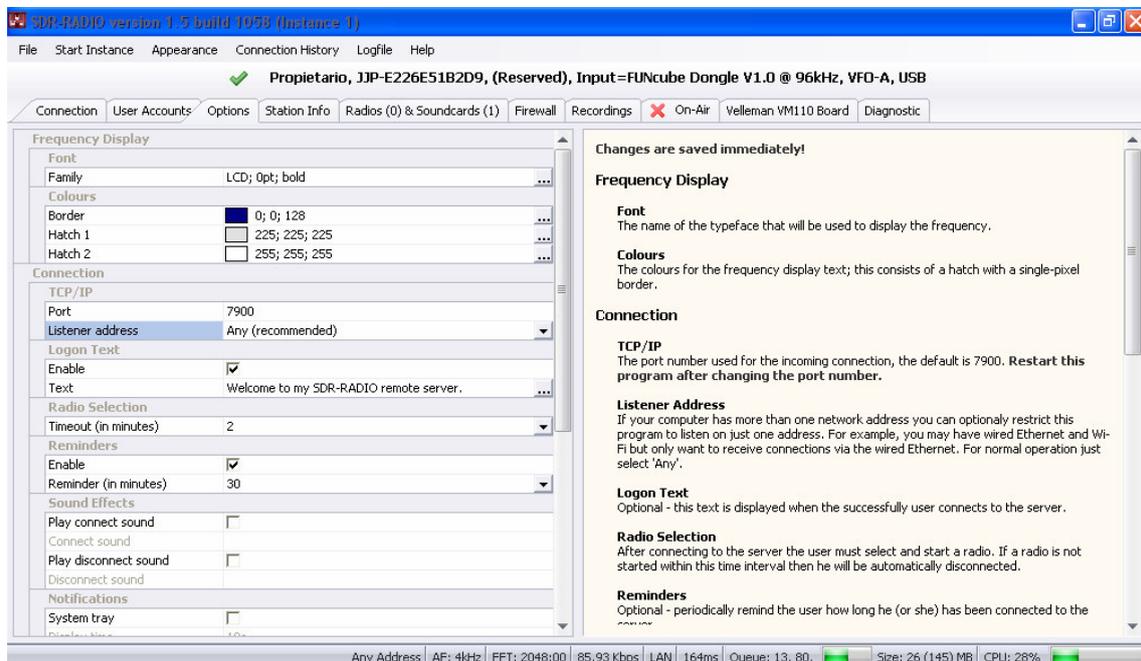


Figura 2.3.63 Pestaña options. Configuración del servidor SDR.

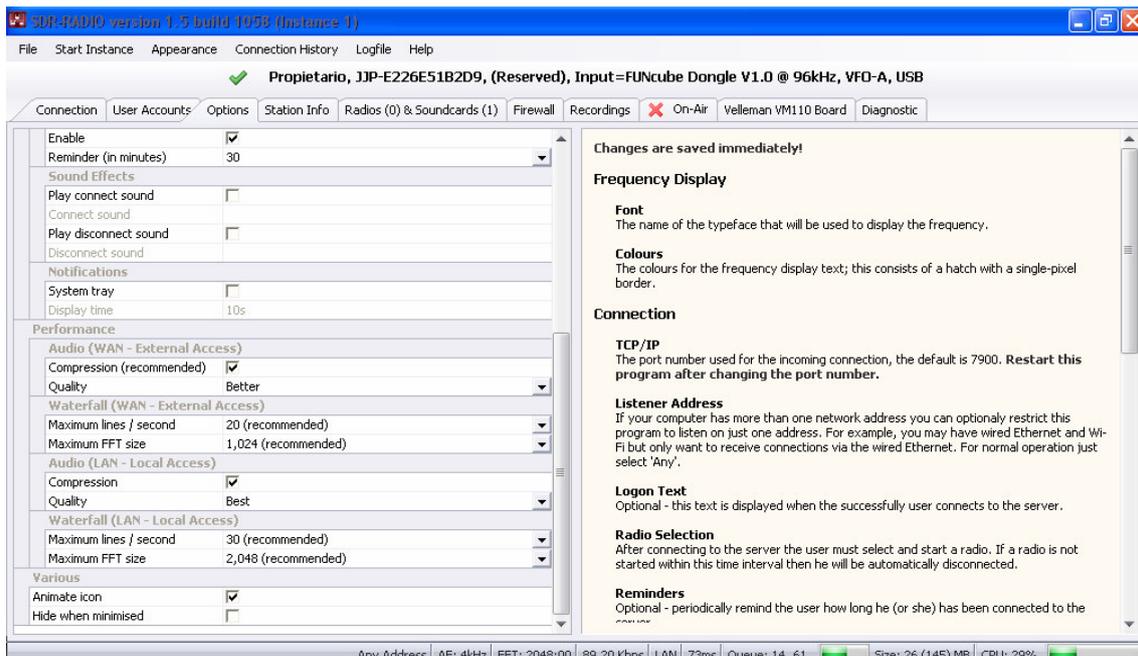


Figura 2.3.64 Pantalla de configuración “options”

Se distinguen las siguientes áreas principales:

Connection:

TCP/IP

Port: define el puerto donde se va a transmitir.

Listener Address: define la dirección donde se transmite. La opción por defecto es any: escucha en cada dirección definida en el PC.

Performance:

Audio: define la compresión del audio.

Waterfall: define las características del waterfall.

Pestaña “Station Information”:

Tiene información de carácter general que el servidor mostrará, en caso de que se habilite a los clientes.



Figura 2.3.65 Pantalla de información que muestra el servidor.

Pestaña “Radios & Soundcards”:

La pestaña Radios y Soundcards, muestra los equipos conectados al servidor SDR. Éstos pueden ser tarjetas de sonido o dispositivos USB.

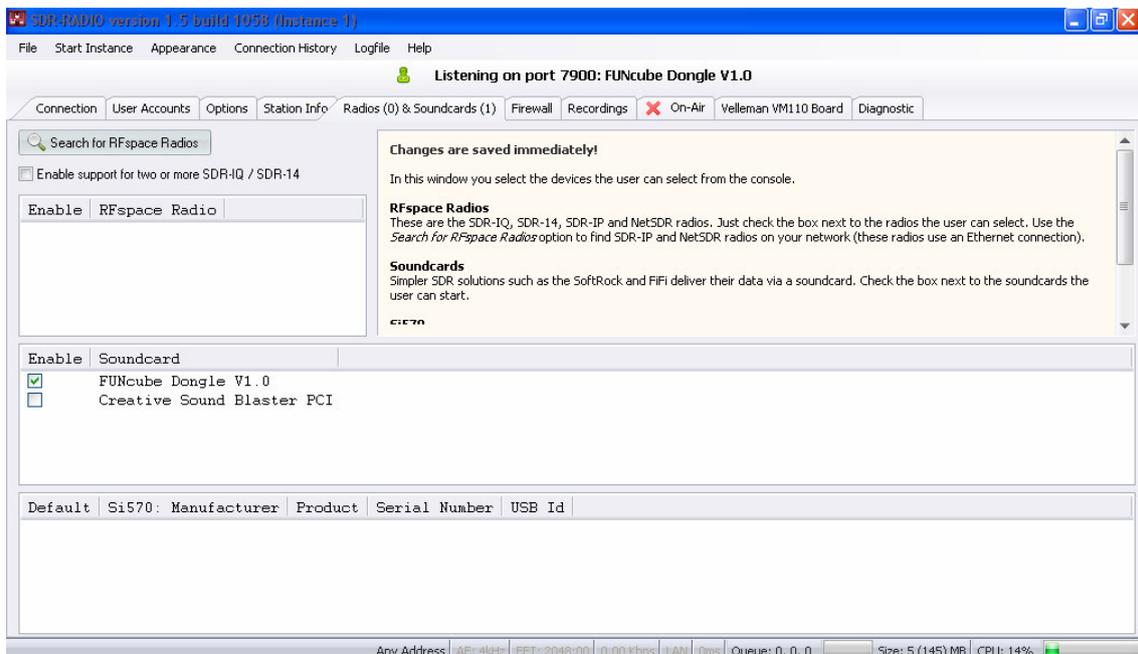


Figura 2.3.66 Pantalla de selección de equipos y tarjetas de sonido.

Desde esta pantalla se pueden habilitar o deshabilitar los dispositivos dentro del servidor de SDR-Radio.

Pestaña “on-air”:

La pantalla on air permite que el servidor se registre dentro de la Web de SDR-Radio, y que de esta forma esté accesible para que los clientes se conecten al servidor usando Internet.

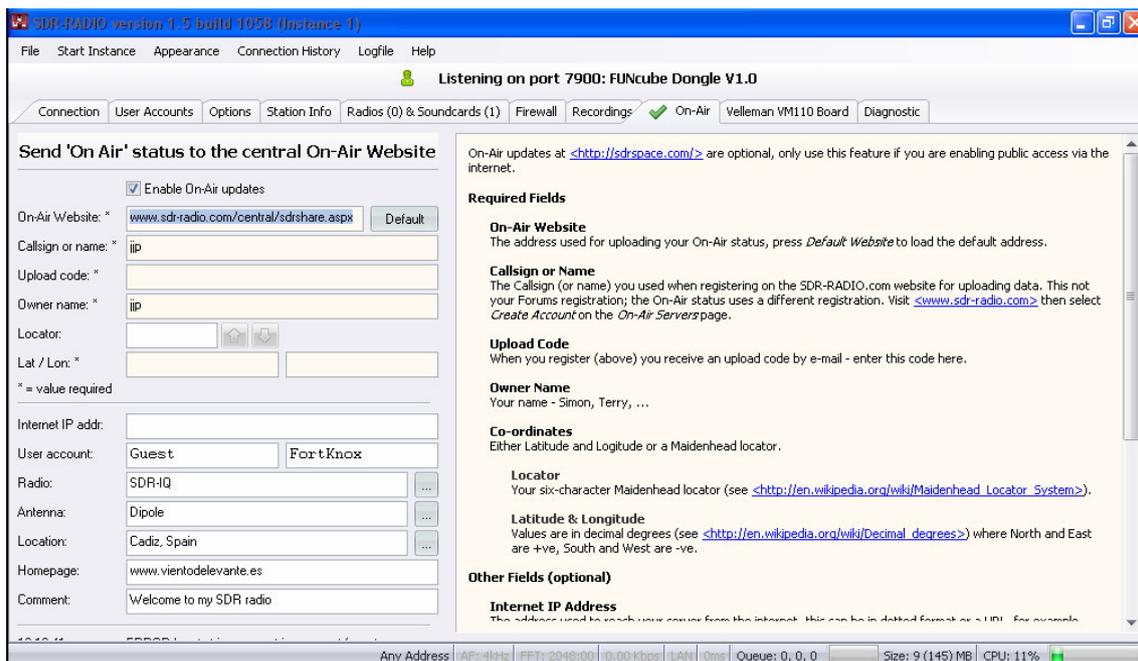


Figura 2.3.67 Opciones de publicación del servidor en la Internet.

Esta opción permite que el servidor de SDR-Radio sea accesible a través del cliente SDR-Radio o de programas como Sigmira que pueden acceder a los servidores SDR-Radio.

En la imagen siguiente, se puede observar la situación de los servidores de SDR-Radio accesibles en Internet. El acceso a esa información se hace en la dirección de Internet <http://www.SDR-Radio.com/central/>.

> Home

Welcome to SDR-Radio over internet

Shared SDR radio

Main menu

Create an account.

If you aren't a member and you want to share your SDR radio, create your account.

Register Now >

Manage your account
View shared SDR list

Statistics

- We have 930 users online
- Shared radio: 45/67
- Used radio: 5
- Registered users: 779

Links

- Forum
- SDP-RADIO.com

Figura 2.3.68 Situación global de los servidores de SDR-Radio

SDR list

Owner	Radio/Antenna	On Air	Freq.
DB7MJ	NET-SDR Wellbrook ALA-1530+	Hans Peter, HB9EHP, Affoltern am Albis	7,200,000
DJ9NZ	FIFI-SDR G5RV		
EA1FAQ	SDR-IQ Doublet dipole 45m long 19m up. Tuned for wideband.		
F5VMH	SDR-4 + 0.1-30Mhz Inverted V		
G0HWC	SDR-IQ #HV001046 Click for SDR info www.g0hwc.com New Hustler 6BTV 6 band vertical 10m to 80m Reports welcome via my website please.		
G4HYG	Cross Country Wireless SDR-4+ Cross Country Wireless Active Antenna		
GI4MUN	AFEDRI SDR-Net Dipole		
HB9DHG	NetSDR 20 mins Guest max connection Pixel Technologies RF PRO-1B	2nd CTO GP, 2NDCTOGP-PC	7,065,500
HB9DRT	SDR-IQ + Antenna switch - No antenna connected - read instructions after connecting and before starting the radio ! Clifton Lab. Z1501D, AL 1530L		
HB9DRV	NetSDR 1 hour max connection Pixel Technologies RF PRO-1B		
	NetSDR-X2		

Ham Radio Deluxe
HPDLOG.net

Bookmarks

- Delicious
- Send to a friend

Figura 2.3.69 Listado de estaciones SDR-Radio

2.3.11.4 Cliente SDR-Radio.

Para acceder al cliente de SDR-Radio, ejecutamos el programa Console. Se nos abrirá el programa cliente de SDR-Radio.

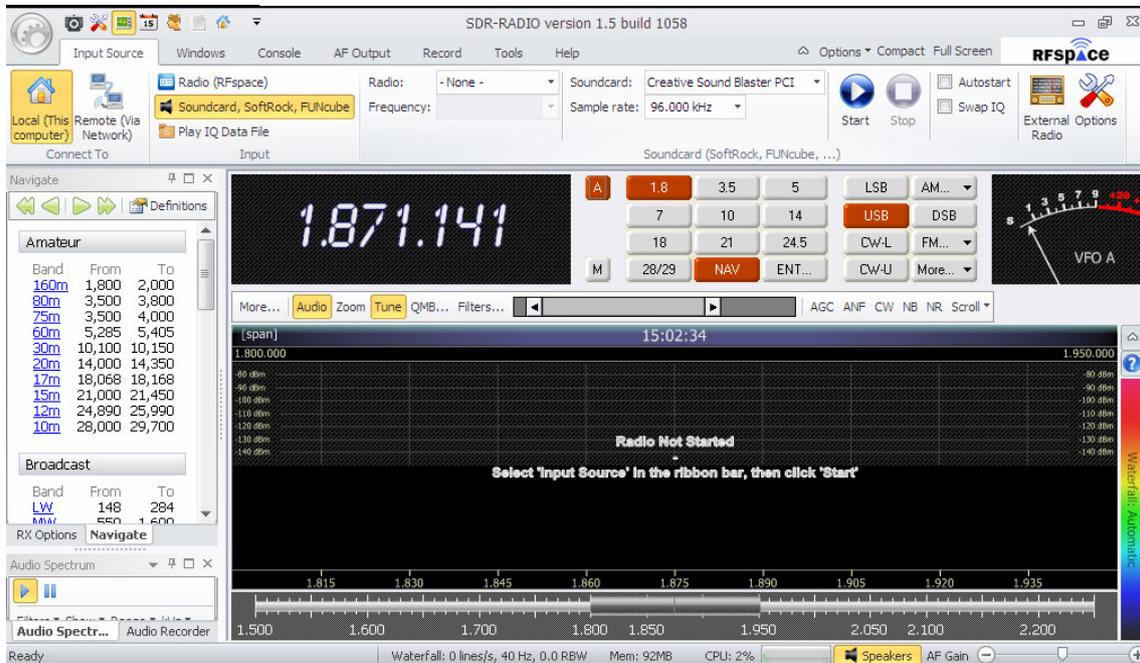


Figura 2.3.70 Consola principal de SDR-Radio.

En la barra principal está la opción “Input sources”.

Como fuente de dato, se pueden escoger equipos conectados al PC localmente o equipos remotos vía Internet.

Al seleccionar los equipos locales, se muestran los conectados al PC.

Una vez seleccionado, se presiona la opción “Start” y comenzará la reproducción.

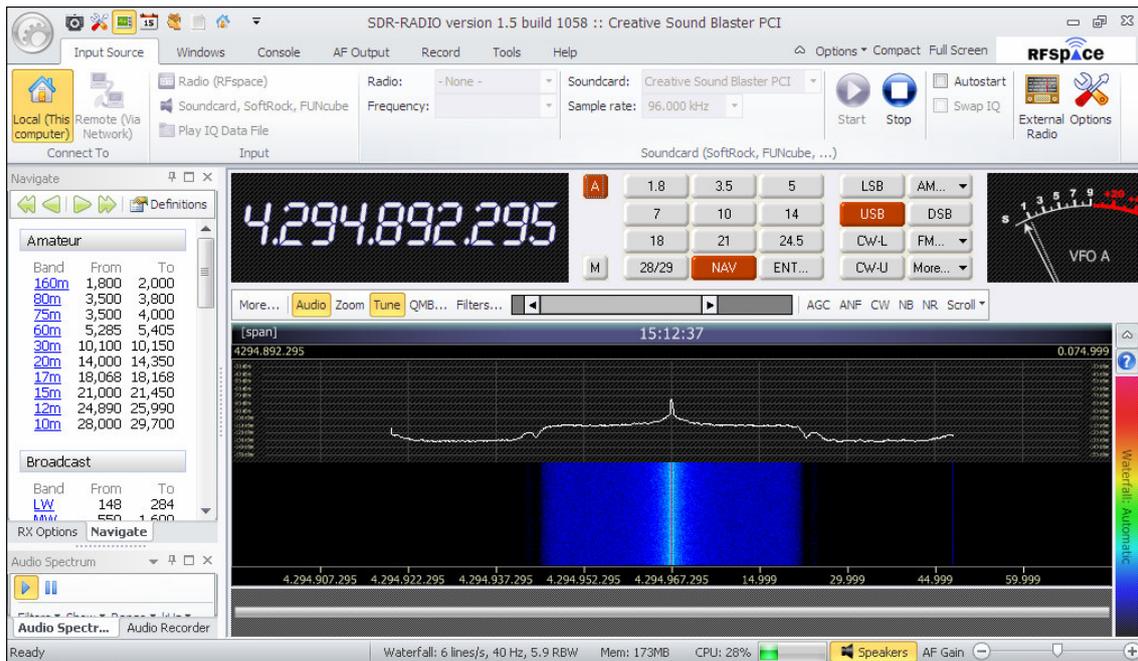


Figura 2.3.71 SDR-Radio conectados localmente a Funcube.

Para conectar a equipo en red, se selecciona la opción, “input source” y a continuación “Remote (via network)”, aparecerá la pantalla para seleccionar el servidor.

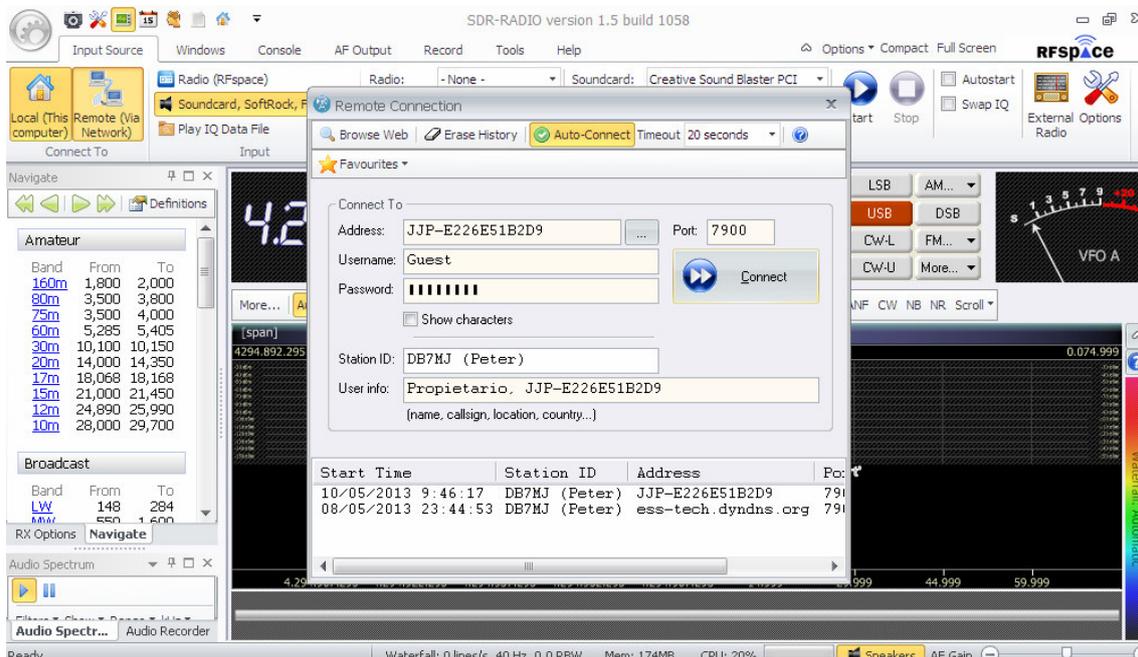


Figura 2.3.72 Pantalla de selección de servidor remoto.

Si se desea conectarse a un servidor público dentro de la red SDR-Radio, se presiona la opción: Browse web.

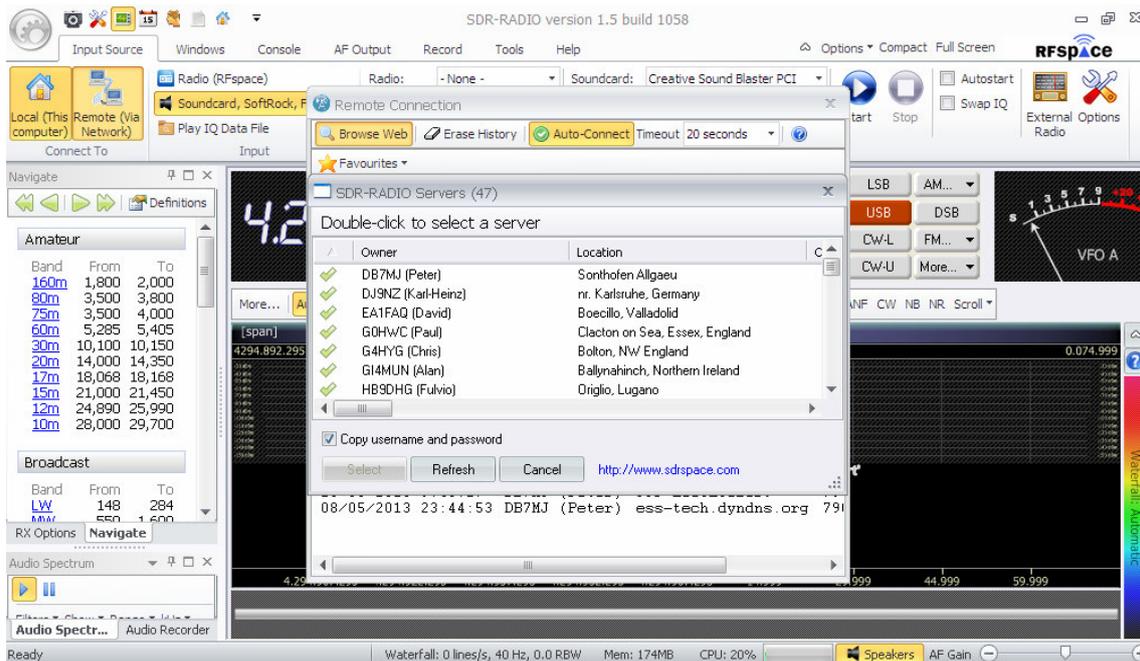


Figura 2.3.73 Relación de servidores remoto SDR-Radio.

Dentro de la relación de servidores activos, se selecciona uno y se conecta.

Dentro de la pantalla de SDR-Radio se presiona la opción start.

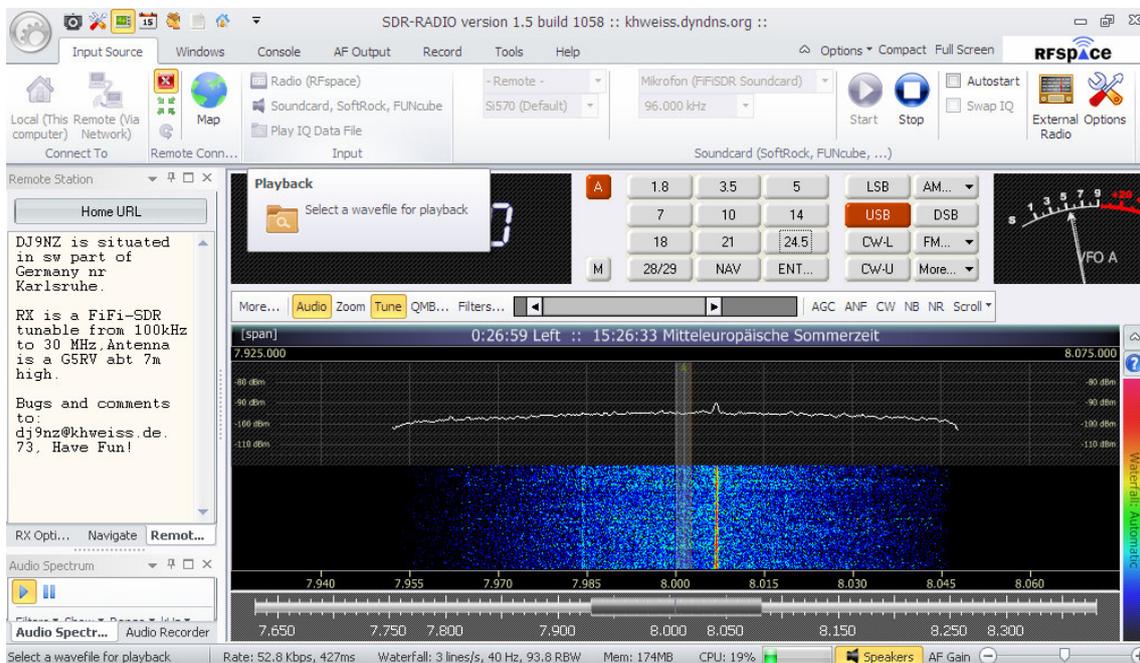


Figura 2.3.74 Consola SDR-Radio conectado a servidor DJ9NZ (Karlsruhe)

El protocolo de comunicación entre cliente y servidor sigue los parámetros descritos anteriormente, donde los comandos están dentro de ficheros HTML.

Como última opción en caso de conectarnos a un servidor remoto en nuestra propia red, tendríamos las pantallas descritas a continuación.

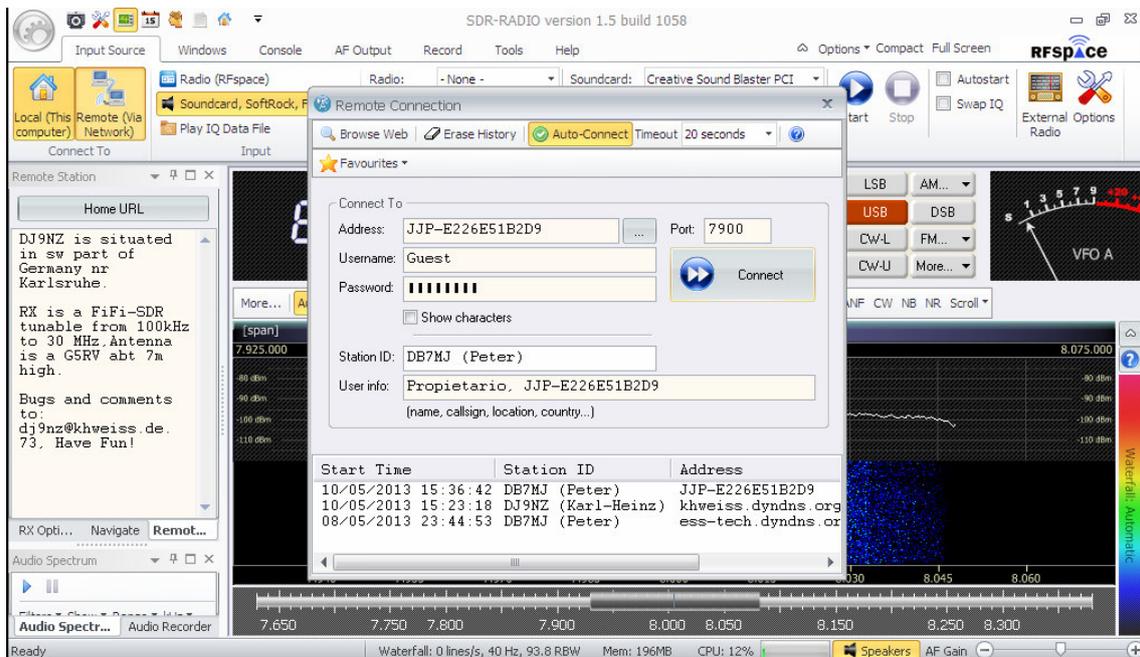


Figura 2.3.75 Conexión a servidor en red local.

Una vez conectado al servidor local, el servidor muestra el waterfall (cascada) enviado y los equipos conectados.

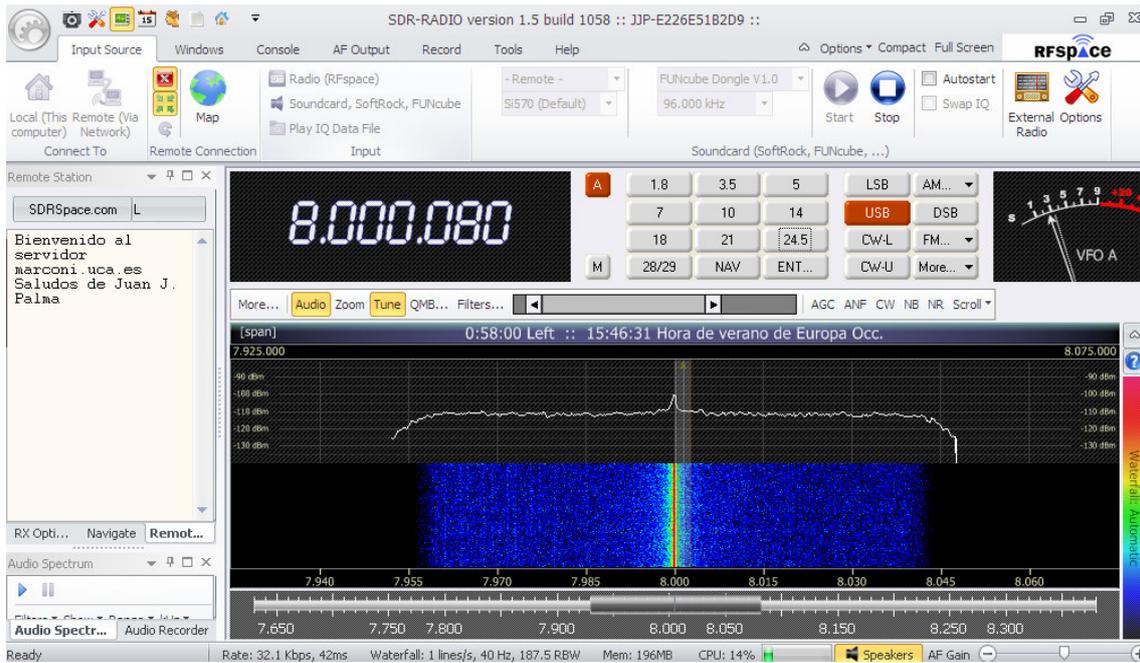


Figura 2.3.76 Conexión a servidor local de SDR-Radio.

El servidor de SDR-Radio contiene la misma cascada que envía y una descripción del equipo que tiene conectado.

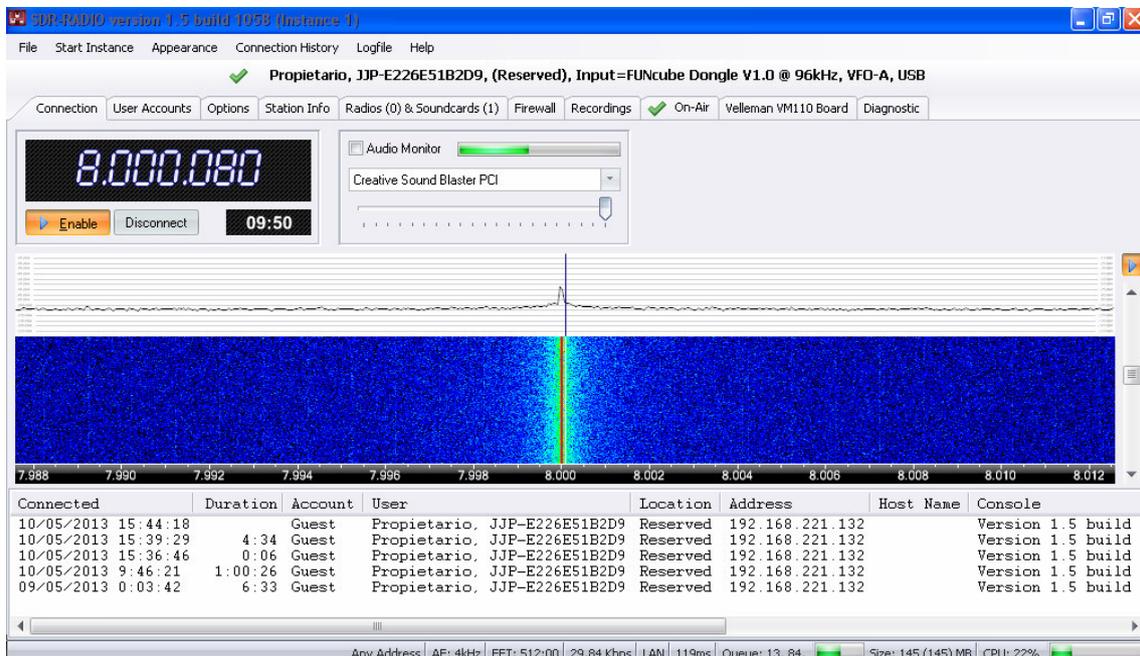


Figura 2.3.77 Servidor SDR-Radio.

El mismo waterfall (cascada) que sale en el cliente aparece en el servidor, así como los controles básicos.

- Usuario conectado.
- Tarjeta de sonido.
- Dirección de conexión.
- FTT.
- Ancho de banda.
- Uso de CPU.

2.4 Voz sobre IP.

2.4.1 Introducción.

La voz sobre IP permite optimizar la infraestructura dentro de las empresas o buques al tener una red única para datos y telefonía.

Las aplicaciones que están creciendo de forma importante en las aplicaciones IP son:

1 Call-Centers:

Las principales herramientas de Asterisk para esto son (aparte de las normales de marcado, grabación y facturación):

- Llamadas en espera.
- Grabación.
- Respuesta automática.
- Sintetización y reconocimiento de voz y cobro avanzado.
- Interfaz API (TAPI para Microsoft) .

Asterisk permite tener personas distantes conectadas al call center (personas en distintos países, distintos idiomas, respondiendo a preguntas).

2 Sistema de mensajería unificado: e-mail y fax integrados en la central telefónica.

3 Llamadas por tarjetas: Establece un número al que el usuario llama, introduce el número de su tarjeta, y puede hablar con cualquier parte del mundo.

Dentro del Demostrador de Conceptos Marconi, se han implementado e integrado con objetos evidentemente educativos los Hardphones y Softphone. Esta investigación ha ido en la dirección de probar las aplicaciones Softphone actualmente en el mercado y aquéllas soportadas por grupos de desarrollo como pueden ser Liphone, SFLphone y Csipsimple.

La línea general de la investigación siempre ha sido el acceso desde distintas plataformas a los servicios distribuidos.

Dentro de las aplicaciones Softphone sobre Linux se ha investigado la automatización de los procesos de llamada, respuesta y direccionamiento. Cabe destacar en este caso la aplicación

Liphone, por su posibilidad de manejo mediante líneas de comandos.

A su vez, se han estudiado e implementado aplicaciones como Teamspeak, acudiendo para su puesta en marcha al grupo de usuarios expertos de Teamspeak así como a la realización de servidores pilotos que permitieran depurar errores de instalación y ejecución, y mejorar los archivos de comandos de lanzamiento e instalación.

2.4.2 Descripción general.

A continuación vemos la distribución de los protocolos VOIP dentro del modelo ISO:

Aplicación	Asterisk
Presentación	G.729/G711/GSM/Speex
Sesión	H323/SIP/MGCP/IAX
Transporte	UDP/RTP/SRTP
Red	IP/CBWFQ/WRED/IP Precedence/Diffserv
Enlace	Frame-Relay/ATM/PPP/Ethernet
Física	Ethernet/V.35/RS-232/xDSL

Si nos fijamos en las capas de Transporte, Sesión, Presentación, Aplicación veremos que:

En la capa de Transporte la mayor parte de los protocolos utilizan el RTP/RTCP. Todos usan UDP para transportar la voz.

En la capa de Sesión entran los protocolos de voz IP H323, SIP, MGCP, IAX, SCCP.

En la capa de Presentación, los códecs definen un formato con sus diferentes variaciones de compresión.

Los protocolos utilizados:

SIP: Estándar abierto (IETF): Es el más usado, sus inconvenientes son: que tiene problemas en el uso de NAT y que el uso del ancho de banda con RTP es alto [DEMP et al. 06].

IAX: Protocolo Asterisk eficiente en el ancho de banda, usa sólo el puerto UDP 4569 para señalización y audio. Esto permite una mejor configuración de Firewalls y NAT's [DEMP et

al. 06]

MGCP: Protocolo para usar con el H323, SIP y IAX. Simplifica mucho la configuración, su manejo es centralizado pero aún se usa poco [DEMP et al. 06].

H323: Usado cuando tenemos routers o gateways de voz de la marca Cisco. Es excelente para videoconferencias, se ha usado durante mucho tiempo, aunque es complicado y poco adoptado en telefonía IP.

Peers, Users y Friends:

Existen tres tipos de clientes SIP/IAX en Asterisk:

User: Puede hacer llamadas a través de Asterisk, pero no puede recibir llamadas del servidor.

Peer: Puede recibir llamadas, pero no puede hacerlas (fax, buzón).

Friend: Puede enviar y recibir llamadas de un servidor. Un teléfono normalmente cae en esta categoría.

Códecs y conversión de códecs:

Los códecs se usan para convertir señal analógica de voz en una versión codificada digitalmente.

Cada servicio, programa, teléfono o gateway, soportan varios códecs, y cuando van a hablar definen el códec a usar.

Asterisk soporta los siguientes códec's:

GSM: 13 Kbps

iLBC: 13.3 Kbps

ITU G.711: 64 Kbps, también conocido como alaw/µlan.

ITU G.723.1: 5.3/6.3 Kbps.

ITU G.726: 16/24/32/40 Kbps.

ITU G.729: 8 Kbps

Speex. 2.15 to 44.2 Kbps

LPC10. 2.5 Kbps

Códec	g.711	g.729	iLBC	GSM 06.10
Ancho de Banda (Kbps)	64	8	13,33	13
Costo	Gratuito	10\$/canal	Gratuito	Gratuito
Resistencia a pérdida de paquetes	Ninguna	3%	5%	3%
MIPS	0,35	13	18	5

Tabla 2.1 Características de los códecs.

A pesar de que los códecs usasen poco ancho de banda, hay una sobrecarga debido a las cabeceras IP, UDP y RTP de los paquetes de voz.

Si estamos en una red Ethernet hay que adicionar el encabezado Ethernet al cálculo, si estamos en una red WAN hay que sumar el encabezamiento Frame-Relay o PPP.

Ejemplos:

CODEC g.711 (64 Kbps) :

Red Ethernet (Ethernet+IP+UDP+RTP+G.711) = 95.2 Kbps

Red PPP (PPP+IP+UDP+RTP+G.711) = 82.4 Kbps

Red Frame-Relay (FR+IP+UDP+RTP+G.711) = 82.8 Kbps

CODEC g.729 (8 Kbps) :

Red Ethernet (Ethernet+IP+UDP+RTP+G.729) = 31.2 Kbps

Red PPP (PPP+IP+UDP+RTP+G.729) = 26.4 Kbps

Red Frame-Relay (FR+IP+UDP+RTP+G.729) = 26.8 Kbps

Estrategia de reducción del ancho de banda:

Se puede reducir el ancho de banda usando compresión de encabezado RTP y el modo IAX trunked.

Este modo sólo se puede usar entre dos servidores Asterisk .

Usando el RTP, la banda usada por el g.729 cae de 26.4 a 11.2 Kbps.

2.4.3 El protocolo SIP.

El protocolo SIP está pensado para iniciar, mantener y terminar sesiones que incluyen voz, vídeo, chat, juegos interactivos y realidad virtual.

El protocolo de señalización de voz SIP posee los siguientes componentes:

UAC (user agent client) – cliente o terminal que inicia la señalización SIP.

UAS (user agent server) – servidor que responde la señalización SIP de un UAC.

UA (user agent) – terminal de red SIP (teléfonos SIP, o gateway para otras redes), contiene UAC y UAS.

Servidor Proxy – Recibe pedidos de conexión de un UA y transfiere éste para otro servidor proxy si la estación en particular no está en su administración.

Servidor de Redireccionamiento – Recibe pedidos de conexión y los envía de vuelta al emisor incluyendo los datos de destino en vez de enviarlos directamente él a la parte llamada.

Servidor de localización – Recibe pedidos de registro de un UA y actualiza la base de datos de terminales con ellos.

Proceso de registro de SIP:

Antes de que un teléfono pueda recibir llamadas, se debe registrar en la base de localización, en este caso lo registramos como 8500.

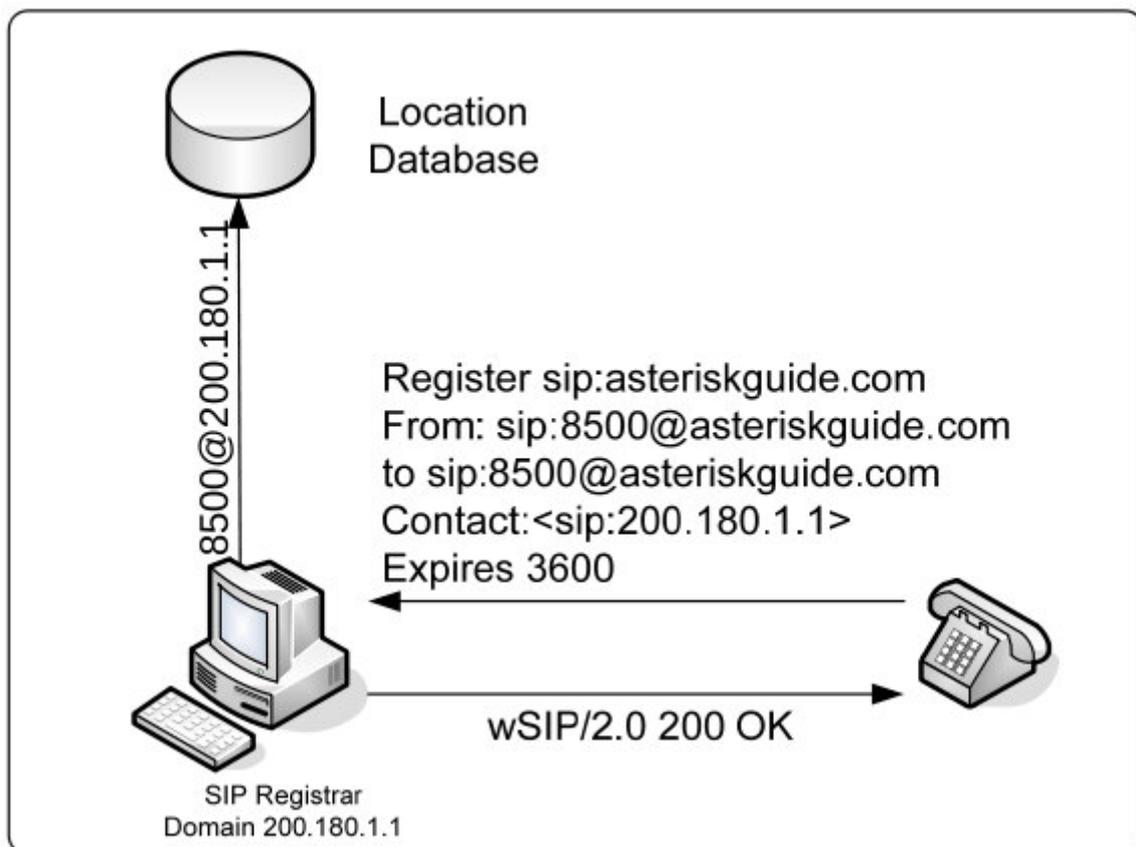


Figura 2.4.1 Registro de un teléfono SIP.

Los mensajes que se envían en un entorno SIP son:

INVITE – pedido de establecimiento de conexión.

ACK – reconocimiento de INVITE por el receptor final del mensaje.

BYE – término de la conexión.

CANCEL – término de una conexión no establecida.

REGISTER – registro de un UA en el SIP proxy.

OPTIONS – pedido de opciones del servidor .

Las respuestas al comando SIP se dan en formato texto como en http. Las principales son :

1XX – mensajes de información (100–intentando, 180–campanilla, 183–en progreso).

2XX – pedido completado con éxito (200 – OK).

3XX – encaminamiento de llamada, el pedido debe ser direccionado para otro lugar. (302 – Moved temporarily movido, 305 – usa proxy).

4XX – error (403 – Prohibido).

5XX – error de servidor (500 – Error interno del servidor, 501 – No implementado).

6XX – falla global (606 – No aceptable).

Archivo de configuración sip.conf:

Cada cliente SIP se identifica dentro de un bloque de texto como el siguiente:

[xxx]

Type=xxx

Parámetro1=valor

Parámetro2=valor

xxx es el nombre de usuario asociado con el cliente SIP, o un nombre usado por otros ficheros de configuración para referirse a este dispositivo.

Si un teléfono tiene la extensión 123, entonces en este archivo sería [123].

Configuración de la sesión general:

Los parámetros más comunes en la sesión [general] del fichero sip.conf son:

bindaddr=0.0.0.0

bindport=5060

Controlan el puerto y la interfaz IP donde Asterisk recibe conexiones.

Bindaddr=0.0.0.0 significa que el servidor recibirá conexiones en todas sus interfaces.

realm=voffice.com.br

Configura el parámetro realm.

srvlookup=yes

Habilita el uso de registros SRV en el DNS .

useragent=Asterisk PBX 1.2.7

Altera el campo useragent de encabezado SIP, identificando el tipo de agente usuario usado.

tos=lowdelay

Los parámetros “tos” colocan bytes correspondientes al tipo de servicio en el encabezado del paquete IP. Este campo le indica a los routers cómo se deben tratar estos paquetes en la red.

Los valores más usados son:

Precedencia 5 (RFC 791, obsoleta)

tos=160

DSCP EF – Express Forwarding (RFC 2474, sustituye a la anterior)

tos=160

Configurar un cliente SIP :

Los pasos son los siguientes:

1. Configurar una entrada en el archivo sip.conf para cada teléfono.
2. Configurar el teléfono.
3. Configurar la extensión en el plan de marcado.

Se detalla en apéndices.

Conexión a un proveedor SIP :

Trixbox (Asterisk) es cliente de un servidor SIP, recibe y coloca llamadas para este servidor.

La recepción de llamada se dirige a una extensión Trixbox (Asterisk).

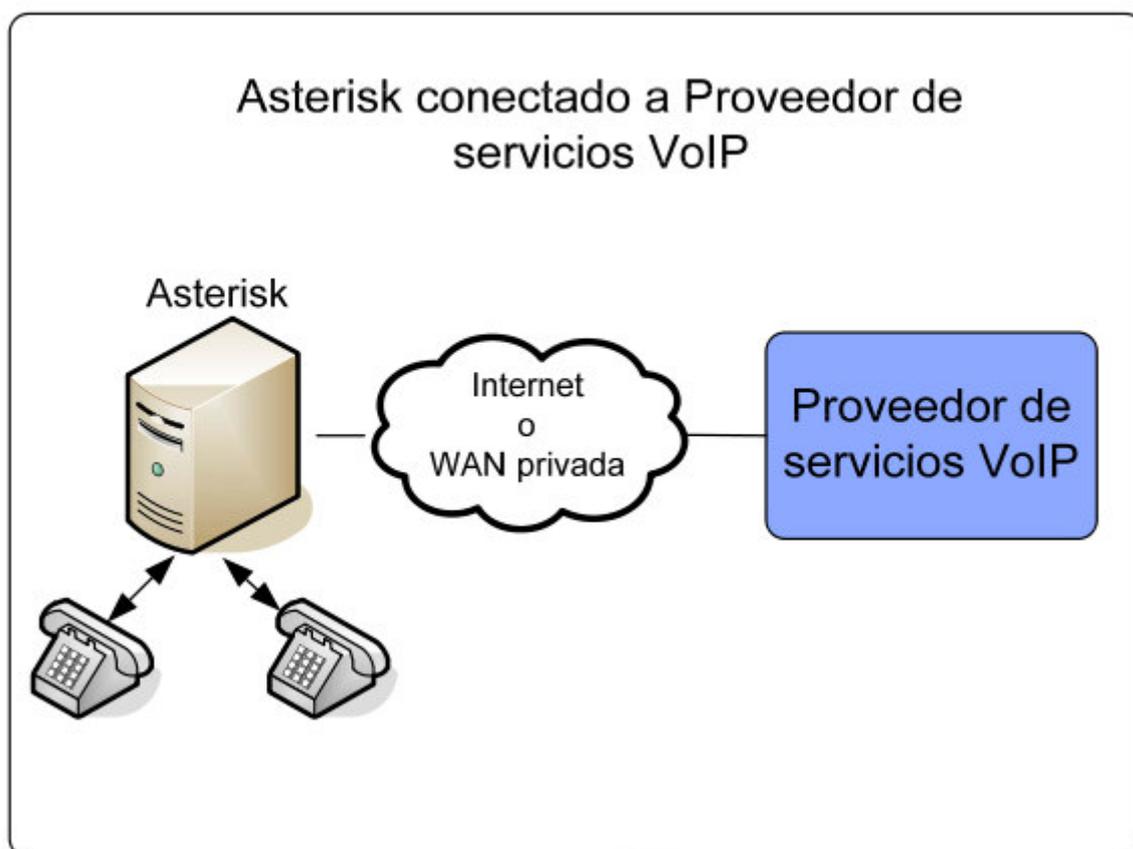


Figura 2.4.2 Trixbox (Asterisk) conectado a servidor VoIP.

Los pasos para configurar un proveedor SIP son:

Paso 1. Dentro de archivo sip.conf, en la sección [general] colocar una línea de registro en proveedor SIP.

Paso 2. Crear una entrada de tipo [peer] para el proveedor para simplificar el marcado.

Paso 3- Colocar una ruta de salida en el plan de marcado.

Se detalla en apéndices.

Nomenclatura de los canales SIP :

El formato del nombre de un canal SIP usado para salir es:

SIP/[exten@]peer[:portno]

peer: Nombre del peer al que se conecta. Puede ser:

peer o friend definido en sip.conf

dirección IP

nombre de dominio

portno: Puerto UDP, por defecto 5060

exten: Pedirá al peer que se conecte a la extensión “exten” .

Ejemplo:

exten=>s,1,Dial(SIP/ipphone)

exten=>s,1,Dial(SIP/info@voffice.com.br)

exten=>s,1,Dial(SIP/192.168.1.8:5060,20)

exten=>s,1,Dial(SIP/8500@sip.com:9876)

Soluciones prácticas para Trixbox:

Podemos tener Trixbox detrás de NAT o clientes detrás de NAT.

Tenemos diferentes tipos de NAT (Full Cone, Restricted Cone, Port Restricted Cone y Simétrico), y tenemos clientes que soportan diferentes tipos de soluciones para NAT (TURN, STUN, ICE, ALG). Esto lo convierte en un problema complejo.

Parámetros Trixbox para atravesar NAT:

Nat:

nat=yes (puede ser true, t, y, 1 y on) . Es la combinación de los modos route y rfc3581

nat=route . Trixbox envía audio al puerto e IP de donde lo recibió, en vez de confiar en las informaciones de los encabezados SIP y SDP .

nat=rfc3581 . Es el estándar, Trixbox va a adicionar “rport” al encabezado SIP, informando al cliente en qué puerto recibió el pedido y va a encaminar el flujo de los mensajes para el cliente en la dirección de donde vino y no en la dirección descrita en los encabezados. El cliente en este punto tiene forma de saber su dirección externa y ahora su puerto externo.

nat=never . Trixbox no va a adicionar el rport en la entre VIA del encabezado como en la rfc3581 .

Qualify:

Esta opción tiene dos funciones: mantener el NAT abierto y asegurarse que Trixbox no intente enviar una llamada a un teléfono inalcanzable .

Qualify=yes, usa el valor estándar de 2 segundos.

Qualify=no, deshabilita el chequeo del peer .

Qualify=x ms, pone un tiempo ms entre chequeos .

Externip:

Opción dentro de la sección [general] del archivo sip.conf, se puede colocar como una ip o como un hostname apuntando para la dirección externa de su dispositivo NAT .

Esta opción se coloca si Trixbox está detrás de NAT, intentando comunicar con dispositivos fuera de NAT .

Ejemplo : externip=200.180.4.168

Localnet:

Opción dentro de la sección [general] del archivo sip.conf, indica la red privada interna donde Trixbox no va a usar la dirección externa provista por externip.

Ejemplo: localnet=10.1.0.0/255.255.0.0

2.4.4 Calidad del servicio.

La calidad de servicio, es la capacidad de la red para poder ofrecer con garantías el servicio ofrecido. El ancho de banda necesaria para el flujo de la comunicación se ha de garantizar, aún cuando exista un alto tráfico de internet por el mismo canal.

Al definir la red de voz hay que optimizar el ancho de banda, controlar las fluctuaciones de red (jitter), y minimizar la latencia.

La latencia mide el tiempo que tarda un paquete en viajar de un punto a otro de la red.

Para ello se le da una mayor prioridad a los paquetes de voz sobre los otros.

Si la comunicación hace uso de un enlace por satélite, se espera una latencia de 3 décimas de segundo, teniendo que optimizar el retardo que ocasiona en las comunicaciones los equipos switch, router, hub de las redes IP.

Se ha de prever la colocación de la centralita de voz en el lugar menos congestionado de la red.

Otro parámetro importante para la mejora de la calidad del servicio es el jitter o la variación del tiempo entre la llegada de los paquetes.

Al viajar por internet, los paquetes de información pueden llegar a su destino de forma descoordinada, pudiendo no llegar en orden.

Una forma de evitar este problema es crear un buffer para jitter. El concepto es fácil. Consiste en hacer que los paquetes esperen un tiempo en cola antes de ser reproducidos. De esta forma da tiempo a que se reorganicen. Esto obviamente introduce un mayor tiempo de latencia, el tamaño del buffer se mide en milisegundos, y es el tiempo que se espera después que han llegado los paquetes.

Se pueden definir dos tipos de buffer jitter, un buffer jitter estático o un buffer jitter dinámico.

La diferencia entre ellos es que el buffer jitter estático lo fija el fabricante y es parte del

equipo.

El buffer jitter dinámico se configura de una forma manual usando algún tipo de programa de administración.

Hay que adoptar soluciones de compromiso, sobre todo teniendo en cuenta que al sobrepasar por encima de 300 ms el retraso puede dificultar la conversación telefónica.

2.5 Telefonía sobre IP.

2.5.1 Introducción.

Para hablar de telefonía sobre IP hemos de hablar de Asterisk.

Asterisk es el primer sistema telefónico desarrollado para IP, con software libre, usando un entorno Linux y tarjetas ZAPATA para conexión a las redes telefónicas tradicionales.

Con la base de Asterisk se han desarrollado distribuciones como Elastic y Trixbox, todas ellas para el mismo entorno Linux.

Existen algunos sistemas telefónicos para Windows menos conocidos como 3CX, no alcanzando el reconocimiento de Asterix por ser sistemas de pago.

Asterisk ha sido desarrollado por Digium, que a su vez fabrica todas las tarjetas de interconexión a los sistemas telefónicos tradicionales.

El proyecto ZAPATA, fue también desarrollado por Digium. Es un hardware abierto y puede ser producido por cualquier empresa, la placa E1/T1 la producen Digium, Ságoma y Varion.

Dentro de esta Tesis Doctoral y con objeto de tener un demostrador de telefonía y VOIP, se han probado, bien de forma virtual (VBox, VMWare) o mediante instalaciones adecuadas, las siguientes centralitas: Asterisk, Trixbox, 3CX. Las centralitas Asterisk, Trixbox y 3CX, se han instalado en discos dedicados.

Por motivos obvios de visibilidad de las soluciones, se ha seleccionado la centralita Trixbox.

Esta centralita se ha descargado de su web (Fonality) y se ha trabajado en la puesta en marcha de la versión gratuita que no es la última en el mercado, por lo que ha habido que solucionar problemas relacionados con su instalación y recompilar drivers (placas de internet, etc.). Para la resolución de problemas se ha acudido al foro Fonality donde se han planteado las dudas.

Como pruebas de la resolución de llamadas IP se ha utilizado el proveedor de internet "Diamondcard", y como tarjeta FX0 se ha seleccionado, probado y depurado problemas de instalación de la tarjeta X100P.

Se ha utilizado previa a la instalación en el demostrador de conceptos Marconi y para depuración de problemas, un par de centralitas situadas externamente con sus firewall IPCop correspondientes.

2.5.2 Descripción general.

La arquitectura general de Asterisk contiene canales, protocolos, códecs (codificador-decodificador) y aplicaciones.

Un canal es equivalente a una línea telefónica en forma de circuito digital. Éste consiste en un sistema POTS, alguna combinación de códec y protocolos (GSM con SIP, Ulaw con IAX).

Para conseguir una optimización del uso del canal usamos los códecs. El códec g.729, permite codificar a 8 kilobits por segundo, y una compresión de 8 a 1.

Otro tipo de códecs son ulaw, alaw, gsm, ilbc y g729.

Asterisk soporta los siguientes códecs:

G.711 ulaw (usado en EUA) – (64 Kbps).

G.711 alaw (usado en Europa y Brasil) – (64 Kbps).

G.723.1 – Modo Pass-through

G.726. 32kbps en Asterisk1.0.3, 16/24/32/40kbps

G.729 – Precisa adquisición de licencia, a menos que esté siendo usado en modo pass-thru.(8Kbps)

GSM – (12-13 Kbps)

iLBC – (15 Kbps)

LPC10. (2.5Kbps)

Speex – (2.15-44.2 Kbps)

Protocolos:

Permiten obtener el camino para establecer la comunicación, así como la señalización de telefonía, tono, tiempo de llamada, identificación de llamada y desconexión.

Asterisk soporta los protocolos SIP, H323, IAXv1, IAXv2, MGCP, SCCP.

Las aplicaciones de Asterisk son las distintas funcionalidades desarrolladas en módulos software, como por ejemplo: mensajería de voz, conferencias, etc.

2.5.3 Primeros pasos con Asterisk.

2.5.3.1 Ficheros de configuración.

Asterisk se controla a través de archivos de configuración localizados en el directorio /etc/Asterisk.

En la sintaxis de estos archivos el “;” es el carácter para los comentarios, los signos “=” y “=>” se usan de forma análoga, y las líneas en blanco se ignoran.

Ejemplo de archivo:

```
;  
; la primera línea sin comentario debe ser el título de una sección.  
;  
[sección]  
llave = valor ; Designación de variable  
[sección 2]  
objeto => valor ; Declaración de objeto
```

2.5.3.2 Configuración de una interfaz con la red pública o un PBX.

Para comunicarse con la red pública se necesita una interfaz tipo FXO, y una línea telefónica. Una línea FXO recibe el tono y una línea FXS da el tono.

Para una configuración básica usaremos una placa X100P, la cual conectaremos en una ranura PCI, y nos aseguraremos tener deshabilitadas todas las interrupciones no usadas.

A continuación hemos de configurar los ficheros zaptel.conf (directorio /etc) y zapata.conf (directorio /etc/Asterisk), que configuran la placa zaptel y los canales zapata de Asterisk respectivamente.

Zaptel.conf

```
Fxsks=1
Loadzone=sp
Defaultzone=sp
Channels=1
```

Zapata.conf

```
[channels]
Context=default
Signalling=fxs_ks
Group=1
Channel => 1
```

Una vez configurados los archivos, se cargan los drivers:

```
modprobe zaptel
ztcfg -vvvvvv
modprobe wcfxo
Asterisk.vvvvvvc
```

2.5.3.3 Configuración de los teléfonos-IP SIP.

Los teléfonos SIP se configuran en el fichero /etc/Asterisk/sip.conf.

```
[general]
bindport = 5060
bindaddr = 10.1.30.45
context = default
disallow = all
allow = ulaw
maxexpirey = 120
defaultexpirey = 80
```

La sección general puede contener los siguientes parámetros:

`allow`: Permite usar un códec

`bindaddr`: Dirección IP donde Asterisk esperará conexiones SIP.

`context`: Contexto general donde se colocarán todos los clientes, al menos que sea sobrescrito en la definición de entidad.

`Disallow`: Prohíbe un códec.

`Bindport`: Puerto en el que Asterisk espera las conexiones de entrada SIP, el más general es el puerto 5060 .

`Maxexpirey`: Tiempo máximo para registro en segundos .

`Register`: Registra Asterisk con otro host . El formato es una dirección SIP, opcionalmente seguido por una barra normal "/" y una extensión .

2.5.3.4 Plan de llamadas:

El plan de llamadas define cómo Asterisk gestiona las llamadas.

La mayor parte de este plan está en el fichero `extensions.conf`, en el directorio `/etc/Asterisk`.

El archivo está separado en cuatro partes:

- Aplicaciones.
- Contextos.
- Extensiones.
- Prioridades.

Dentro de cada contexto se definen diversas extensiones. Estas extensiones pueden ser: literal, estándar o especial.

Ejemplo :

```
exten=>8580,1,Dial(SIP/8580,20)
```

```
exten=>8580,2,voicemail(u8580)
```

```
exten=>8580,101,voicemail(b8580)
```

2.5.3.5 Conceptos generales sobre canales analógicos y digitales.

Canales analógicos:

Interfaces FX (Foreign Exchange / Canal Analógico):

FXO, es una puerta que se comunica directamente con la red pública (emula un terminal telefónico casero) y que requiere de tono de indicación de marcado y de llamada en progreso.

FXS, es equivalente al servicio que provee una compañía telefónica a un teléfono casero. Provee de voltaje, de tono, de detección de descolgado, y de indicación de llamada en progreso.

Canales digitales:

Líneas Digitales E1 / T1, Señalización CAS y CCS:

Al aumentar el número de líneas telefónicas necesarias, se usan canales E1 (2Mbps) con señalización CAS (Channel Associated Signaling) o bien con CCS (Common Channel Signaling).

2.5.3.6 Configuración de un canal de telefonía en Asterisk.

Para cada placa FXO / FXS que usemos, debemos configurar el fichero `zaptel.conf` (directorio `/etc`).

A continuación se deben cargar el módulo `zaptel` y el módulo correspondiente a la placa que se está instalando (se aclara en el manual de la placa).

Placa	Driver	Descripción
TE410P	wct4xxp	4xE1/T1-3.3V PCI
TE405P	wct4xxp	4xE1/T1-5V PCI
TDM400P	Wctdm	4 FXS/FXO
T100P	wct1xxp	1 T1
E100P	Wctlxxp	1 E1
X100P	Wcfxo	1 FXO

2.5.4 Características de la centralita Trixbox:

2.5.4.1 Conceptos generales.

Inicialmente se van a recordar los términos que se emplean posteriormente.

PSTN significa Red telefónica pública conmutada. Es el sistema habitual de telefonía y es controlado por las compañías de teléfonos normales. Esta red cubre todo el mundo y hay diferentes tipos de interfaces en ella.

POTS (Plain Old Telephone Service) es el tipo tradicional de teléfono, el teléfono más usado. Tiene un par de hilos de cobre y es el teléfono más fácil y el más barato.

ISDN (RDSI) es la conexión más conocida y es la que provee de servicios a pequeñas y medianas empresas. Es una forma de conectarse a servicios de voz y datos digitales.

T1/E1 es el sistema más común en las grandes empresas. Es un servicio digital que ofrece más capacidades que el ISDN pues incrementa el ancho de banda, lo que permite tener más líneas telefónicas.

El principal problema del PSTN es que está controlado por las compañías telefónicas por lo cual resulta muy caro.

ISDN/T1/E1 se utiliza como interfaz externa de las compañías. Se dispone de una centralita telefónica para la comunicación interna.

PBX (Private Branch eXchange) provee servicio telefónico dentro de una compañía, permitiendo direccionar internamente las llamadas telefónicas y salir al exterior para realizar las llamadas externas.

Como en una compañía, no todas las líneas telefónicas y llamadas necesitan salir hacia fuera

de ésta, las PBXs se encargan de coordinar el tráfico de llamadas internamente y hacia el exterior, permitiendo tener un número limitado de líneas externas, y de esta forma abaratar los costes.

Los requisitos que se les pide a las PBXs son de mensajería de voz, parking de llamadas, transferencia de llamadas, música en espera, opciones de menú, bajo coste de routing, y distribución automática de llamadas.

Sistema tradicional de PBX:

El sistema tradicional de PBX consiste en un gran armario con infinidad de tarjetas, cada vez que se requiere algún tipo de asistencia hay que acudir a la compañía proveedora o al fabricante del equipo.

Aunque se ahorra una cantidad de dinero importante en las llamadas internas, se necesita el soporte técnico del proveedor, que suele ser la misma compañía que proporciona las líneas exteriores. Habitualmente los teléfonos pertenecen al mismo proveedor de la centralita.

PBXs híbridas:

Combinan la funcionalidad de las centralitas telefónicas habituales con las opciones de VOIP. Permiten tener extensiones telefónicas externas y teléfonos software, permiten guardar las líneas existentes, y añadir VOIP para obtener algún beneficio económico.

Los sistemas de PBX Asterisk, son unos sistemas híbridos, que permiten tener acceso a la red PSTN pública y a VOIP. Las principales funcionalidades son:

1. Uso del estándar SIP.
2. Uso de extensiones remotas, usando Teléfonos SIP y Teléfonos software.
3. Soporte de protocolo IAX para conexión entre equipos Asterisk

VOIP:

VOIP es otra forma barata y fácil de conectarse a PSTN.

A medida que Internet ha ido creciendo, se ha visto que era una forma óptima de conducir no

sólo los datos, sino también la voz.

Una de las limitaciones al transmitir la voz es el tiempo de latencia, ya que es difícil mantener una comunicación telefónica donde la voz se entrecorte o tarde en llegar la respuesta, como pasa en las conversaciones telefónicas por satélite geoestacionario.

A medida que aumenta el ancho de banda y la velocidad de red es mayor, disminuye la latencia, y por tanto la comunicación telefónica por Internet comienza a ser viable.

Por qué elegir VOIP:

Básicamente VOIP, tiene una infraestructura flexible, y al circular sobre IP, cualquiera puede desplazar sus teléfonos, sin la necesidad de hacer inversión en cableado ni terminales.

A su vez el cableado utilizado es el mismo que se usa para la comunicación IP sobre Internet.

Se puede comunicar externamente sobre Internet, sin tener que estar ligado a proveedores PSTN, pudiendo hacer llamadas locales entre distintos países.

Obviamente al desarrollar el servicio hay que evaluar la infraestructura PSTN así como del ISP en nuestra zona, asegurándonos que se cumplen los requisitos necesarios para su implementación.

2.5.4.2 Descripción de Trixbox.

Trixbox, procede del software original [Asterisk@Home](#), y es una solución de VOIP para una pequeña y mediana empresa.

Trixbox básicamente contiene los siguientes elementos:

- CentosOS 4.3: Versión de Red Hat Enterprise Linux.
- Asterisk 1.2: Es el centro de la centralita telefónica.
- FreePBX: Herramienta Web para manejar y mantener Asterisk.

- Flash Operator Panel (FOP): Aplicación Web que permite a un recepcionista ver el estado de las extensiones y circuitos telefónicos.
- Servicios Cisco XML: Aplicaciones para crear las configuraciones necesarias para los teléfonos CISCO.
- SugarCRM: Aplicación que permite seguir ventas y contactos de clientes.
- Herramientas de instalación automática: Las herramientas, el sistema operativo, los scripts, los ficheros de configuración, se configuran automáticamente e instalan automáticamente, para su uso por los scripts de configuración de Trixbox.
- El sistema de voz: Permite pasar un texto a voz.
- Autoconfiguración de tarjetas Digium: Script para configurar las tarjetas del proveedor Digium.

Diferencias entre Asterisk y Trixbox:

Asterisk es el software de la PBX, mientras que Trixbox es un paquete que instala Asterisk, Centos y los elementos que se enumeraron anteriormente.

El interés de Trixbox está en la interfaz Web.

Los requisitos del ordenador para una solución comercial de Trixbox son: Procesador de 2 GB, 1 GB RAM y 100 GB de disco. También debemos tener al menos una tarjeta FXO si deseamos salir a la red telefónica tradicional.

Tarjetas a instalar:

Para empezar a probar, se pueden usar tarjetas para conectar nuestro sistema como teléfonos (POTS – Plain Old Telephone Service lines).

Estas tarjetas FXO o tarjetas X100P se pueden obtener a un buen precio.

Si vamos a conectarnos a servicios de ISP, no se necesita ningún hardware adicional.

Las tarjetas que desarrolla Digium permiten conectarse a una red PSTN, desde multipuertos POTS hasta tarjetas que soportan hasta cuatro líneas T1 (equivalentes a 96 líneas individuales telefónicas).

2.5.4.3 Consideraciones generales para implementar una centralita Trixbox.

Es importante preparar detalladamente el plan de desarrollo de la centralita telefónica para evitar tener que hacer modificaciones durante la explotación del sistema, y consecuentemente tener caídas del servicio.

Aparte de los temas característicos de seguridad, accesibilidad del equipo y calentamiento, tenemos los temas de configuración lógica de los servicios. Estos son:

- Extensiones
- Grupos de llamada
- Colas de llamadas
- Conectividad
 - PSTN
 - VOIP
 - ITSPs
 - Líneas DID (marcación interna directa)
- Teléfonos:
 - Teléfonos electrónicos.
 - Teléfonos software.
- IVR (Interactive Voice Response – repuesta de voz interactiva / menú de opciones)
- Requerimientos de fax

Éstas son las áreas principales de una centralita telefónica.

Extensiones:

Hemos de tener en cuenta varios aspectos:

- Número de usuarios.
- Mensaje de recepción: Debemos establecerlo con un menú de selección de opciones.
- Las extensiones han de evitar empezar con los números seleccionados en las opciones, ejemplo, si tenemos la opción 1, 2, 3, las extensiones deberán empezar con el 4,5.....

La información que habrá que guardar sobre cada registro de usuario sería:

- Nombre a mostrar: aparecerá en el directorio de la compañía.
- Número de extensiones.
- Identificación de la llamada hacia el exterior.
- Grabación de llamada (siempre, nunca, bajo demanda).
- Palabra clave de la extensión.
- Palabra clave de usuario: para extensión con mensajería de voz.
- Dirección de correo.
- Dirección de buscapersonas.
- Permitir anexos en el correo.
- Ver el número que nos llama.
- Ver fecha y hora de la llamada.
- Borrar mensajería de voz.
- Grupo de mensajería de voz.

Grupos de llamada:

Un grupo de llamada es un grupo de extensiones a las que se puede llamar al mismo tiempo. Cuando se llama a cualquiera de ellas, la llamada puede ser respondida desde cualquiera de estas extensiones.

Esto hace posible, por ejemplo, que se pueda llamar a nuestro teléfono de la oficina y a nuestro móvil.

El grupo de llamadas se puede configurar llamando a todas las extensiones a la vez o bien llamando a cada extensión individualmente.

Colas de llamadas:

Es el lugar donde van esperando las llamadas en el caso de que el receptor de la llamada esté ocupado.

Existe la posibilidad de: música en espera, información de tiempo de espera, y otros.

Conectividad:

Una vez sabemos el número de usuarios, debemos estimar el número máximo de llamadas

concurrentes que tendremos.

A continuación, debemos estimar el número de llamadas salientes y cuántas son de larga distancia.

PSTN:

La conexión más básica PSTN es un terminal POTS (Plain Old Telephone Service). Son los terminales telefónicos normales que tenemos en nuestra casa.

Se pueden obtener tarjetas PCI para los PC de 1 a 24 líneas POTS.

La tarjeta TDM400P de Digium puede soportar hasta 4 líneas telefónicas FXO.

La tarjeta TDM2400 puede soportar 24 líneas analógicas.

Con líneas FXO se puede trabajar en instalaciones pequeñas. El siguiente paso sería la utilización de una línea T1.

En la línea T1, las llamadas se multiplexan, permitiendo llegar a 24 líneas telefónicas por T1.

Usando la tecnología de Digium (simple, doble, cuádruple) se puede llegar hasta 96 líneas de teléfono.

Un paso intermedio entre POTS y T1 es RDSI (ISDN), la compañía de telefonía según la zona, nos da uno u otro servicio.

VOIP:

Usando voz sobre IP, las llamadas se hacen sobre internet, usando un proveedor de servicios telefónicos sobre internet (ITSPs). Estos proveedores de servicios telefónicos conectan nuestras llamadas de VOIP a la red pública telefónica (PSTN).

El ancho de banda usado en la comunicación telefónica depende del códec que usemos.

Los requisitos para cada códec son:

G.711 (ulaw): 1 llamada 81,1kbps, 2 llamadas 148 kbps. Llamadas por Mbit 15

ILBC: 1 llamada 28,0 kbps, 2 llamadas 49,3 kbps. Llamadas por Mbit 47

G.729: 1 llamada 30,0 kbps, 2 llamadas 39,7 kbps. Llamadas por Mbit 103

GSM: 1 llamada 35,4 kbps, 2 llamadas 50,2 kbps. Llamadas por Mbit 68

El protocolo más usado es el G729. Para poder usarlo en la central Trixbox hay que pagar licencia.

Comparando con una T1 que puede administrar 24 líneas telefónicas, el códec G.711 puede ofrecer unas 21 líneas concurrentes en 1,5 Mbit.

Es obvio que cuanto más comprima el códec más recursos de cpu se requiere.

Usar 100 líneas con un códec G729, produce una carga importante en la cpu de la PBX.

El ancho de banda ocupado actualmente por la comunicación telefónica no es tan crítico, y hay que estimar sobre todo el ancho de banda compartido por otras aplicaciones, multimedia, etc.

ITSP:

Es importante ver las ofertas que ofrecen los proveedores por internet de una forma cuidadosa, pero hemos de tener en cuenta los códec que soportan. Si vamos a tener una cantidad importante de líneas, debemos dirigirnos al códec G.729, y por lo tanto, buscar un proveedor que soporte este códec.

Líneas DID (Direct Inward Dial-Marcación interna directa):

Es simplemente nuestro número telefónico, al que tienen que llamar para hablarnos.

Obviamente una línea POT, tiene un solo número, pero en una T1 podemos tener muchas líneas, ya que una T1 provee 24 líneas de accesos, pero internamente la centralita telefónica puede manejar muchas más.

Los proveedores ITSPs nos ofrecerán la cantidad de DID que queramos. Al crear los números DID habrá que tener en cuenta las siguientes consideraciones:

- El nombre que le daremos a la troncal, normalmente dado por el proveedor y un número en caso de disponer de más de una troncal.
- La tecnología usada
- El número de canales que da la troncal.
- El orden en que se usará la troncal para llamadas externas.

Ejemplo:

Troncal	Tecnología	Canales	DID	Prioridad de uso
Trocal-T1	PRI	23	956-666-1300 a 956-666-1500	2
PBX-IAX	IAX	10	956-666-8555	1
PSTN-1	PSTN	4	956-666-1800 a 956-666-1804	3

Teléfonos:

Teléfonos electrónicos:

Los Teléfonos electrónicos son teléfonos con un uso y forma parecidos a los normales. Los proveedores de estos teléfonos son: Grandstream, Linksys, Cisco, Snome, Zultys, Polycom, y otros. Se suelen conectar a las centralitas telefónicas por protocolo SIP y disponer de hasta 4 líneas, botones de marcación rápida, transferencia, multiconferencia, LCD y opciones de vídeo, así como de la posibilidad de usar los servicios de internet (skype, google talk, servicios meteorológicos y otros).

Los teléfonos electrónicos también ofrecen la posibilidad de conexión usando tecnología wifi.

Otros tipos de teléfonos electrónicos que se pueden usar, son los teléfonos normales POTs, que usan un dispositivo ATA (adaptador de teléfono analógico).

Teléfonos software:

Son teléfonos basados en software y que se ejecutan en un ordenador, tableta o móvil. Usan los recursos de estos equipos, y corren bajo Windows, Mac, Linux o Android.

Ejemplos de estos teléfonos son:

Ekiga,

Linphone

ExpressTalk

Csipsimple

IVR (Respuesta interactiva de voz):

Usando los menús IVR, se pueden dar opciones para ir a otros menús, al contestador de voz a una extensión o a una aplicación de usuario.

Se deben seguir las siguientes reglas para hacer un buen diseño del sistema IVR:

- Hacerlo fácil, poner pocas opciones de menús, dos o tres. En caso de más opciones, hay que subdividir en submenús.
- Si no se encuentran las opciones de forma fácil, hay que dar acceso a una persona que responda. Es importante no intentar responder todo.
- No preguntar información innecesaria, no repetir las mismas preguntas varias veces.
- Permitir al usuario volver para atrás, en caso de que se haya cometido un error.
- Hacer una grabación de voz de buena calidad y fácilmente entendible.

Requisitos de Fax:

Trixbox puede redireccionar faxes a una extensión, con una máquina de fax conectada o a una dirección de email.

2.5.4.4 Instalación del software Trixbox.

Seguidamente se describen los pasos para instalar la centralita Trixbox.

Obtener el software:

Se descarga la imagen ISO desde la dirección de trixbox <http://www.trixbox.org> seleccionando la opción de download.

Una vez descargada la imagen ISO se copia al disco de instalación.

Instalación:

Se arranca el PC con el disco dentro del CD-ROM y nos aseguramos de que el ordenador arranca desde él.

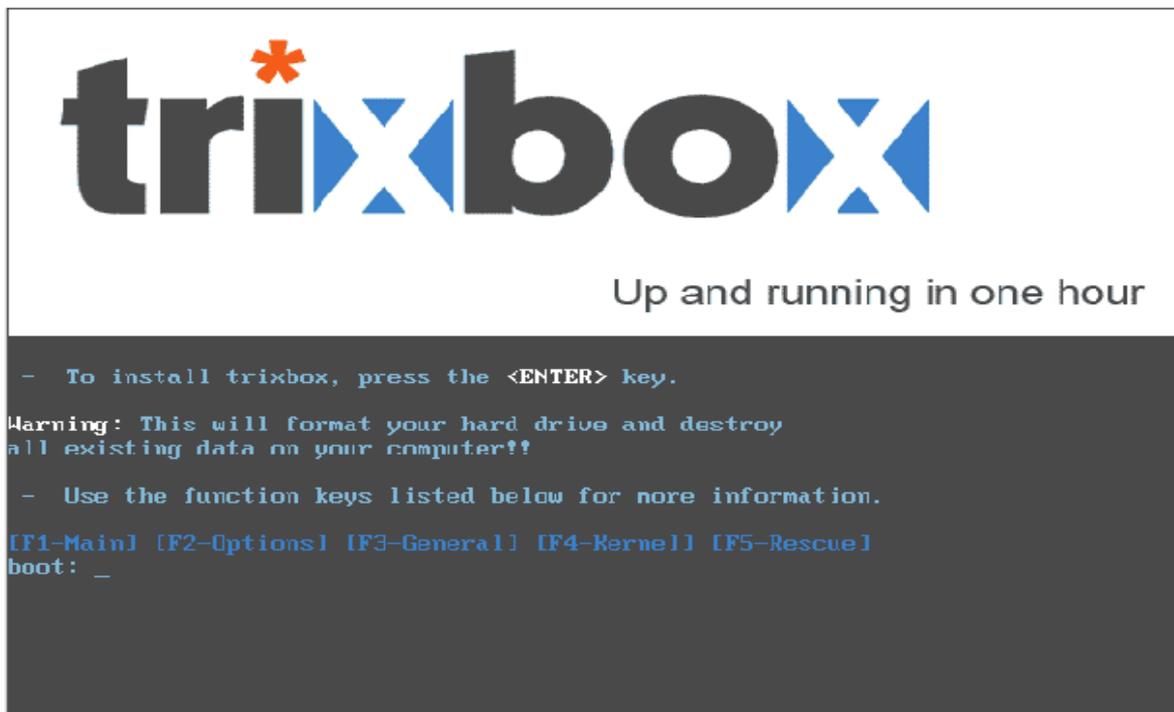


Figura 2.5.1 Instalación Trixbox

F1 Nos devuelve a la página principal anterior.

F2 Nos lleva a un menú para hacer un chequeo del disco y memoria

F3 Se puede ignorar, da opciones para modificar la resolución de pantalla y el sistema de arranque.

F4 Da información sobre parámetros que podemos pasar al kernel, si tenemos problemas durante el boot (arranque).

F5 Nos muestra la opción de rescate que se puede usar para reparar nuestro sistema si tenemos problemas al iniciar Trixbox.

Chequeo del disco:



Figura 2.5.2 Comienzo comprobación CD.

Usando la opción F2, (también se puede hacer tecleando linuxmediacheck), después de uno o dos minutos aparece la pantalla siguiente:



Figura 2.5.3 Comprobación del disco.

Si presionamos Enter, veremos la pantalla siguiente:

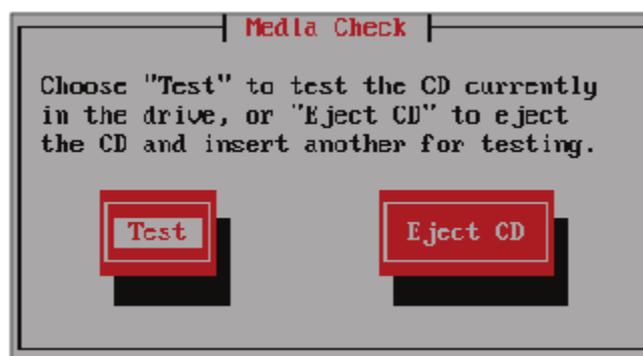


Figura 2.5.4 Confirmación de chequeo de CD

Presionado Enter, con el botón Test seleccionado, comienza el test.

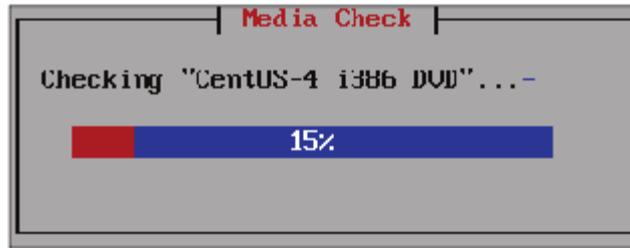


Figura 2.5.5 Comienzo de chequeo.

Si el chequeo es correcto, tenemos:

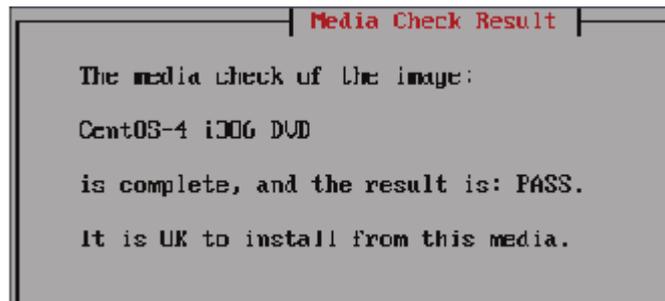


Figura 2.5.6 Chequeo completado.

Esto nos dice que el CD de instalación es correcto.

Instalación automática:

Una vez comenzada la instalación, preguntará por el teclado, seleccionar con la barra espaciadora y Enter.

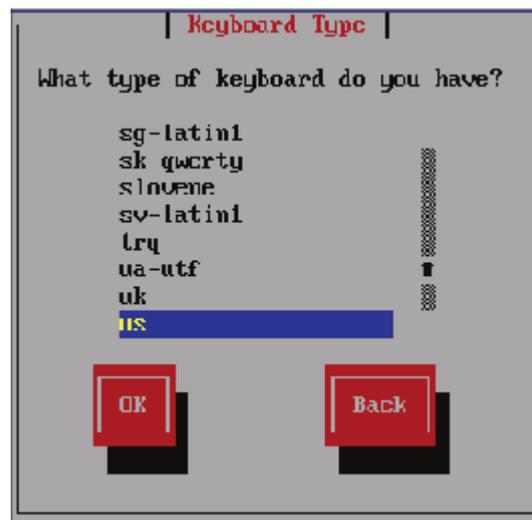


Figura 2.5.7 Selección de teclado.

A continuación, se introduce la clave de Administrador (root).

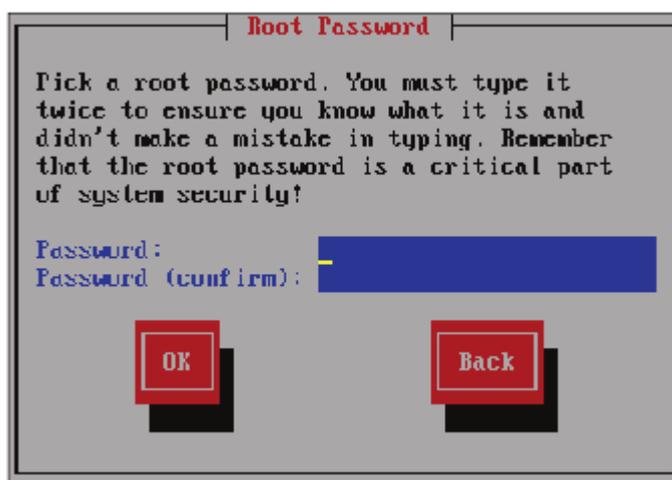


Figura 2.5.8 Selección de clave (password).

A continuación, se introduce la zona horaria.

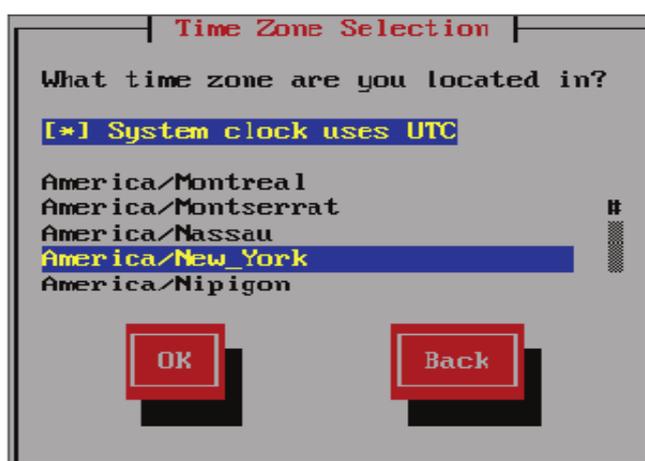


Figura 2.5.9 Selección de zona horaria.

A continuación se instala el sistema operativo y todos los ficheros.

La duración del proceso depende del ordenador usado, llevando desde unos pocos minutos a una hora.

Al terminar el proceso se expulsa el CD, se re-arranca el sistema y cuando el sistema comienza, se compilará y actualizará todo el software y las herramientas que forman parte del sistema Trixbox.

Una vez que todo se ha compilado, el sistema re-arranca de nuevo, y se nos presenta con la consola CentOS del equipo.

Ahora entramos con el usuario root (Administrador) y el password (clave) introducido anteriormente.

Si ahora queremos hacer cambios, usamos el comando setup en CentOS, y obtendremos la pantalla que aparece a continuación:



Figura 2.5.10 Pantalla de configuración

Seleccionamos Network configuration y configuramos las opciones de TCP/IP

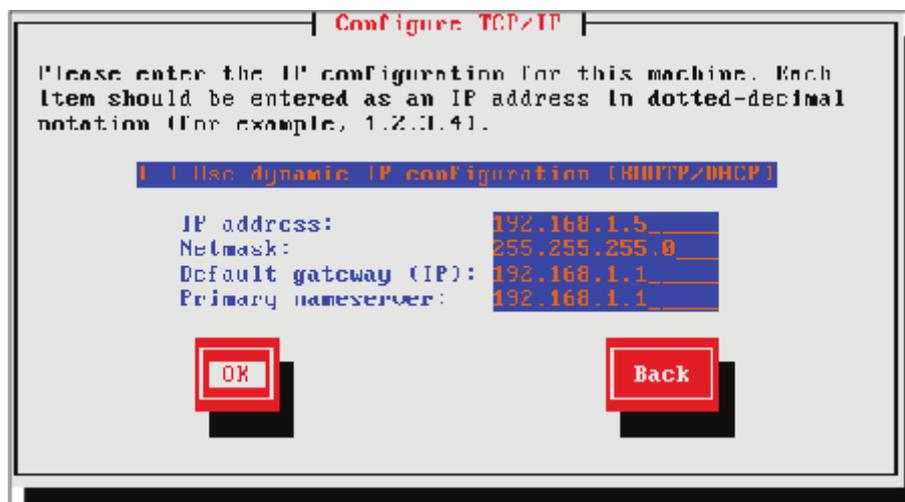


Figura 2.5.11 Pantalla de configuración de red.

Es recomendable escoger una dirección IP fija, a continuación se pulsa tabulador hasta llegar a OK, y Enter para ir a la pantalla de configuración y a continuación Quit para salir de nuevo a la línea de comandos.

Si introducimos el comando reboot, el servidor se reinicia y tienen efectos las nuevas

configuraciones.

Configuración básica del sistema:

Actualización del programa Trixbox:

Para hacer una actualización de Trixbox se usa el comando :

```
trixbox-update.sh update
```

SMTP server, configuración:

Se ejecuta el programa `setup-mail`.

Normalmente hay que editar el fichero `/etc/sendmail.cf` y añadir la información de mail Server de nuestro proveedor de ISP, esto se hace con el siguiente comando:

```
nano /etc/mail/sendmail.cf
```

Una vez que hemos realizado los cambios, salvamos y salimos.

2.5.4.5 Configuración de la centralita Trixbox.

Seguidamente se describen las pantallas de administración de Trixbox y se configura la centralita, creando los servicios necesarios para su funcionamiento.

2.5.4.5.1 Pantallas de configuración de la centralita Trixbox.

La interfaz Web:

Para acceder a ella, vamos al navegador e introducimos en la barra de direcciones la dirección del servidor (ejemplo: dirección Marconi <http://192.169.2.50>), y desde aquí iremos al menú principal de Trixbox (véase figura 2.5.12).

The screenshot displays the Trixbox CE web interface. At the top left is the logo 'trixbox CE' with the tagline 'The Open Platform for Business Telephony'. The top right shows 'Server time: 00:14:31' and 'Admin mode [switch]'. A navigation menu includes 'System Status', 'Packages', 'PBX', 'System', 'Settings', and 'Help'. The main content area is divided into several sections:

- Server Status:** A list of services with their status: Asterisk (Running), web server (Running), cron server (Running), SSH server (Running), and Mysql (Running).
- Announcements:** A large box with the word 'Found' and the message 'The document has moved here.'
- Network Usage:** A table showing data for various network devices.
- Memory Usage:** A table showing the percentage of capacity used by different components.
- trixbox Status:** A sidebar on the right showing system information: Hostname (trixbox1.localdomain), Local IP (192.168.2.50), Public IP, Active Channels (SIP: 1, IAX: 0), Current Registrations (SIP: 2, IAX: 1), SIP Peers (Online: 0, Offline: 6, Unmonitored: 6), and IAX2 Peers (Online: 0).

Device	Received	Sent	Err/Drop
lo	913.35 KB	913.35 KB	0/0
wmaster0	0.00 KB	0.00 KB	0/0
wlan0	0.00 KB	0.00 KB	0/0
eth0	414.37 KB	1.15 MB	0/0
eth1	0.00 KB	0.00 KB	0/0
sit0	0.00 KB	0.00 KB	0/0

Type	Percent Capacity	Free	Used	Size
- Kernel + applications	10%		101.49 MB	
- Buffers	3%		28.85 MB	
- Cached	20%		205.43 MB	

Figura 2.5.12 Interfaz web Trixbox.

Flash operator panel:

El panel de operación, nos permite ver el estado de todas las extensiones, colas y troncales, y también nos permite coger llamadas, picando y arrastrando.

Este panel estará en blanco hasta que configuremos las extensiones, troncales y colas.



Figura 2.5.13 Interfaz web pantalla flash operator panel.

Administración del sistema:

El sistema Trixbox se administra desde su aplicación Web.

Existen básicamente 3 pantallas de administración:

La pantalla PBX – Admin

La pantalla PBX – Reports

La pantalla PBX – Panel



Figura 2.5.14 Pantalla Trixbox Admin.

Editor de configuración:

Para editar los ficheros de configuración del Trixbox, se utiliza la opción,

PBX- Config file editor:

Con esta utilidad se pueden modificar los ficheros de configuración, sin tener que acudir a entrar con una sesión ssh en el servidor Trixbox.



Figura 2.5.15 Pantalla Trixbox edición de ficheros.

Información de Asterisk:

La pantalla PBX status nos da la información de Asterisk, las conexiones activas, las troncales, los teléfonos, las comunicaciones en curso, figura 2.5.16.

The screenshot displays the 'PBX Status' page for a Trixbox CE system. The page is titled 'PBX Status: trixbox1.localdomain ()' and provides a comprehensive overview of the Asterisk PBX system's current state.

System Information:

- Version:** Asterisk 1.6.0.26-FONCORE-r78 built by root @ revisor.trixbox.com on a i686 running Linux on 2010-06-08 22:01:27 UTC
- Uptime:** System uptime: 1 hour, 49 minutes, 53 seconds; Last reload: 15 seconds

Active Channel(s):

Peer	User/ANR	Call ID	Format	Hold	Last Message
46.19.58.41	475725	58d08eaf140fd8e	0x0 (nothing)	No	

1 active SIP dialog

SIP Registry:

Host	Username	Refresh	State	Reg.Time
sip.diamondcard.us:5060	475725	105	Registered	Sun, 09 Dec 2012 17:40:27

1 SIP registrations.

SIP Peers:

Name/username	Host	Dyn	Nat	ACL	Port	Status
Diamond_s/475725	46.19.58.41				5060	Unmonitored
900/900	192.168.2.195	D	N	A	58618	OK (63 ms)
500	(Unspecified)	D	N	A	5060	UNKNOWN
400	(Unspecified)	D	N	A	5060	UNKNOWN
300	(Unspecified)	D	N	A	5060	UNKNOWN
200	(Unspecified)	D	N	A	5060	UNKNOWN
100	(Unspecified)	D	N	A	5060	UNKNOWN

7 sip peers [Monitored: 1 online, 5 offline Unmonitored: 1 online, 0 offline]

IAX2 Registry:

Host	dnsmgr	Username	Perceived	Refresh	State
0 IAX2 registrations.					

IAX2 Peers:

Name/Username	Host	Mask	Port	Status
0 iax2 peers [0 online, 0 offline, 0 unmonitored]				

Subscribe/Notify:

```
-- Registered Asterisk Dial Plan Hints ==
900@ext-local      : SIP/900&Custom:DND90  State:Idle           Watchers 0
500@ext-local      : SIP/500&Custom:DND50  State:Unavailable   Watchers 0
400@ext-local      : SIP/400&Custom:DND40  State:Unavailable   Watchers 0
300@ext-local      : SIP/300&Custom:DND30  State:Unavailable   Watchers 0
200@ext-local      : SIP/200&Custom:DND20  State:Unavailable   Watchers 0
100@ext-local      : SIP/100&Custom:DND10  State:Unavailable   Watchers 0
*76900@ext-dnd-hints : Custom:DEVND900      State:Unavailable   Watchers 0
*76500@ext-dnd-hints : Custom:DEVND500      State:Unavailable   Watchers 0
*76400@ext-dnd-hints : Custom:DEVND400      State:Unavailable   Watchers 0
*76300@ext-dnd-hints : Custom:DEVND300      State:Unavailable   Watchers 0
*76200@ext-dnd-hints : Custom:DEVND200      State:Unavailable   Watchers 0
*76100@ext-dnd-hints : Custom:DEVND100      State:Unavailable   Watchers 0
*21900@ext-findmefollow : Custom:FOLLOWME900  State:Unavailable   Watchers 0
*21500@ext-findmefollow : Custom:FOLLOWME500  State:Unavailable   Watchers 0
*21400@ext-findmefollow : Custom:FOLLOWME400  State:Unavailable   Watchers 0
*21300@ext-findmefollow : Custom:FOLLOWME300  State:Unavailable   Watchers 0
*21200@ext-findmefollow : Custom:FOLLOWME200  State:Unavailable   Watchers 0
*21100@ext-findmefollow : Custom:FOLLOWME100  State:Unavailable   Watchers 0
-----
- 18 hints registered
```

Dahdi driver info:

Chan: ChanType: Context: Language: MOI: Test: ...

Figura 2.5.16 Pantalla Trixbox información.

Gestor de Teléfonos (Endpoint manager):

La pantalla End Point Configuration se usa para crear ficheros de configuración para los teléfonos que se conectan a la centralita telefónica (figura 2.5.17).

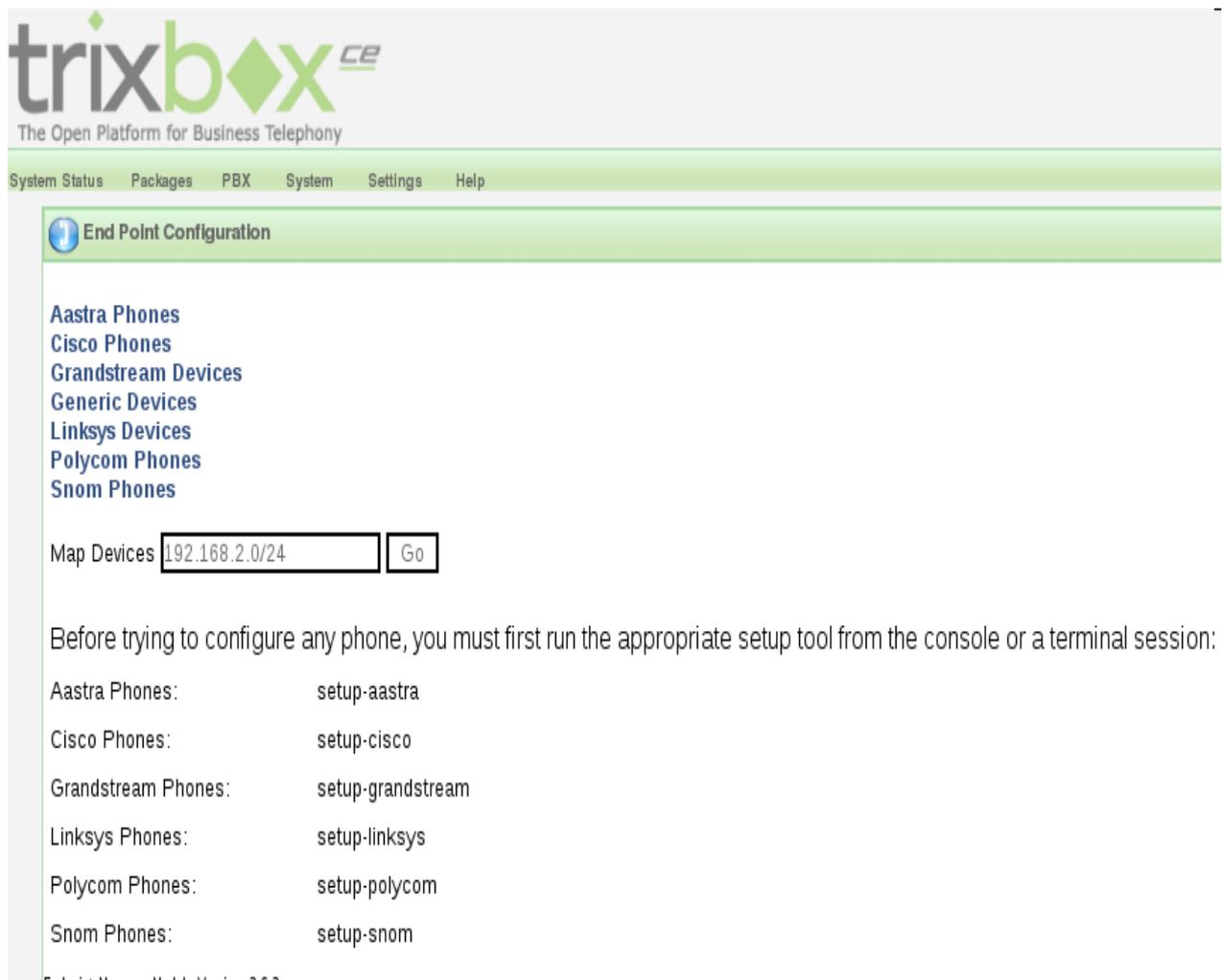


Figura 2.5.17 Gestor de teléfonos.

Información del sistema:

La pantalla de información del sistema da información general sobre los parámetros del sistema, la red y el hardware (figura 2.5.18).

The screenshot displays the Trixbox CE web interface. At the top, the logo 'trixbox CE' is visible with the tagline 'The Open Platform for Business Telephony'. A navigation menu includes 'System Status', 'Packages', 'PBX', 'System', 'Settings', and 'Help'. The main content area is titled 'System information: trixbox1.localdomain (192.168.2.50)'. Below this, there are three sections: 'SYSTEM VITAL', 'NETWORK USAGE', and 'HARDWARE INFORMATION'. The 'SYSTEM VITAL' section lists details like Canonical Hostname, Listening IP, Kernel Version, Distro Name, Uptime, Current Users, and Load Averages. The 'NETWORK USAGE' section is a table showing network interface statistics. The 'HARDWARE INFORMATION' section shows details about the processor and CPU speed.

SYSTEM VITAL	
Canonical Hostname	trixbox1.localdomain
Listening IP	192.168.2.50
Kernel Version	2.6.18-164.11.1.el5 (SMP)
Distro Name	CentOS release 5.5 (Final)
Uptime	2 hours 15 minutes
Current Users	0
Load Averages	0.00 0.00 0.00

NETWORK USAGE					
Device	Received	Sent	Err/Drop		
lo	584.04 KB	584.04 KB	0/0		
eth0	885.69 KB	4.38 MB	0/0		
eth1	0 KB	0 KB	0/0		
wmaster0	NaN KB	0 KB	0/0		
wlan0	NaN KB	0 KB	0/0		
sit0	0 KB	0 KB	0/0		

HARDWARE INFORMATION	
Processors	1
Model	Intel(R) Pentium(R) 4 CPU 2.80GHz
CPU Speed	2.79 GHz

Figura 2.5.18 Información del sistema.

Pantalla de mantenimiento del sistema:

La pantalla de mantenimiento del sistema permite realizar las funciones habituales del sistema (figura 2.5.19):

- Apagado (Shutdown).
- Reinicio (Reboot).
- Reinicialización (Reset).

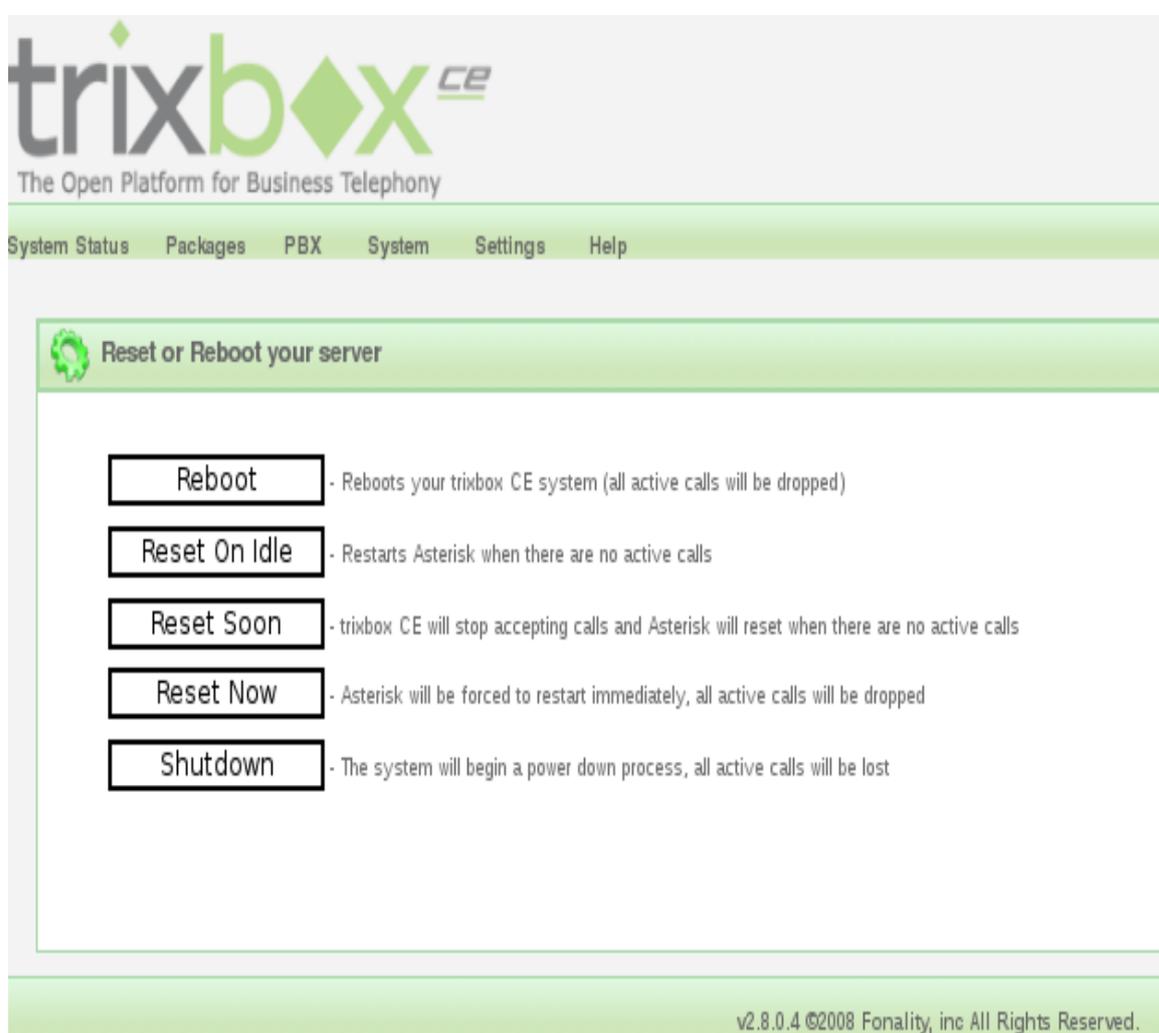


Figura 2.5.19 Mantenimiento del sistema.

Terminal SSH:

Este terminal nos permite entrar en el servidor de Trixbox, sin tener que conectar una consola al equipo, pudiendo correr desde él las aplicaciones de configuración, etc. que se corren desde el servidor (figura 2.5.20).

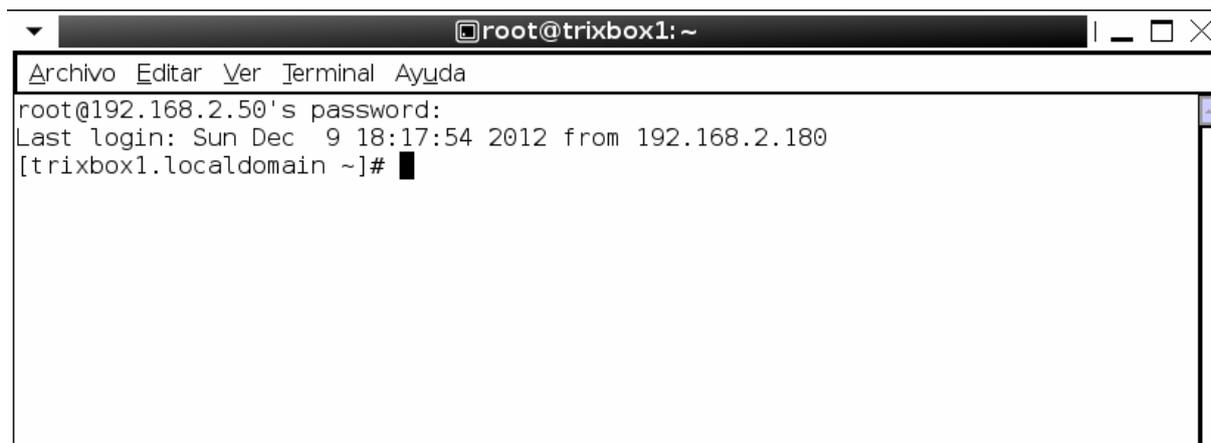


Figura 2.5.20 Terminal de acceso al servidor.

2.5.4.5.2 Pantalla de administración.

Para empezar a configurar Trixbox, hemos de entrar en la pantalla de administración (figura 2.5.21), usando el usuario por defecto maint y clave de entrada password.

Una vez dentro de esta aplicación, vamos a la opción PBX, admin.

The screenshot displays the Trixbox CE administration interface. The top navigation bar includes 'System Status', 'Packages', 'PBX', 'System', 'Settings', and 'Help'. Below this, a secondary bar contains 'Admin', 'Reports', 'Panel', 'Recordings', and 'Help'. A left sidebar menu lists various administrative options such as 'Setup', 'Tools', 'Admin', 'System Status', 'Module Admin', 'Basic', 'Extensions', 'Feature Codes', 'General Settings', 'Outbound Routes', 'Support', 'Trunks', 'Administrators', 'Inbound Call Control', 'Inbound Routes', 'Zap Channel DIDs', 'Announcements', 'Blacklist', 'CallerID Lookup Sources', 'Day/Night Control', and 'Follow Me'. The main content area is titled 'System Status' and is divided into several sections:

- Notices:** A list of system alerts, including '6 extensions/trunks have weak secrets', 'Cronmanager encountered 1 Errors', 'No email address for online update checks', and '53 New modules are available'. A 'show all' link is provided.
- Statistics:** A table showing call and channel statistics:

Total active calls	0
Internal calls	0
External calls	0
Total active channels	0
- Connections:** A table showing online status:

IP Phones Online	1
IP Trunks Online	1
IP Trunk Registrations	1
- System Statistics:** A section for hardware and system metrics:
 - Processor:** Load Average: 0.00, CPU: 0%
 - Memory:** App Memory: 13%, Swap: 0%
 - Disks:** /boot: 20%, /dev/shm: 0%
 - Networks:** eth0 receive/transmit: 0.00 KB/s, eth1 receive/transmit: 0.00 KB/s

Figura 2.5.21 Trixbox pantalla de administración.

2.5.4.5.3 Creación de extensiones.

Para manejar las extensiones, nos vamos al menú de Trixbox, PBX, admin, Setup.

Dentro del apartado Basic, encontramos la opción extensiones (figura 2.5.22).

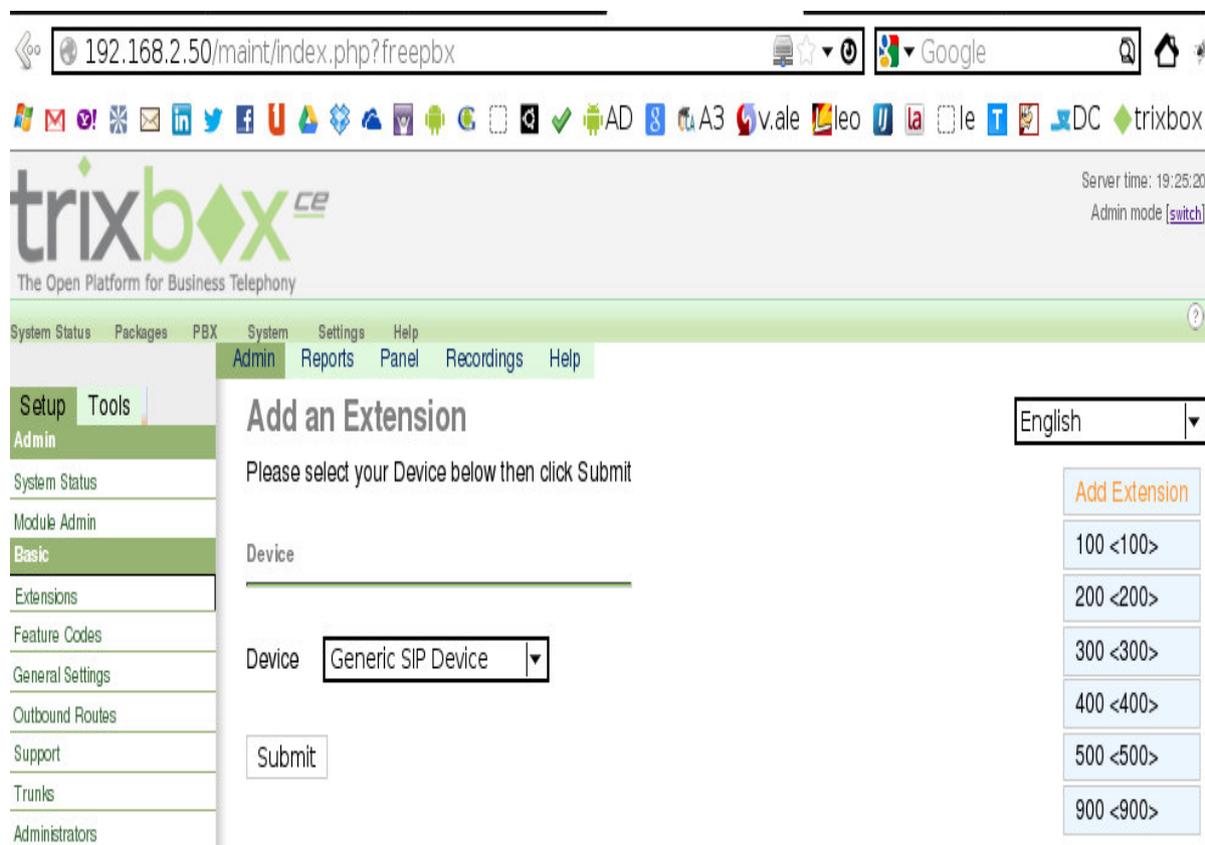


Figura 2.5.22 Trixbox, añadir extensión.

Se pueden añadir 3 tipos de extensiones:

- Extensiones SIP (Session initiation protocol), es la más común.
- Extensiones IAX2, es la extensión propia de Asterisk, la más adecuada si vamos a tener usuarios remotos pues estas extensiones dan menos problemas que las SIP.
- Extensiones ZAP, son las usadas normalmente para las PBX's tradicionales.

Lo primero que hay que hacer es generar una extensión y ver que nos podemos conectar al sistema normalmente. Con esto se tiene una comprobación sencilla de que nuestro sistema

está funcionando.

Add SIP Extension

Add Extension

Extension Number:

Display Name:

Extension Options

Direct DID:

LID Alert Info:

Outbound CID:

Emergency CID:

Record Incoming: On Demand ▼

Record Outgoing: On Demand ▼

Device Options

secret

rtmfmode

Voicemail & Directory: Disabled ▼

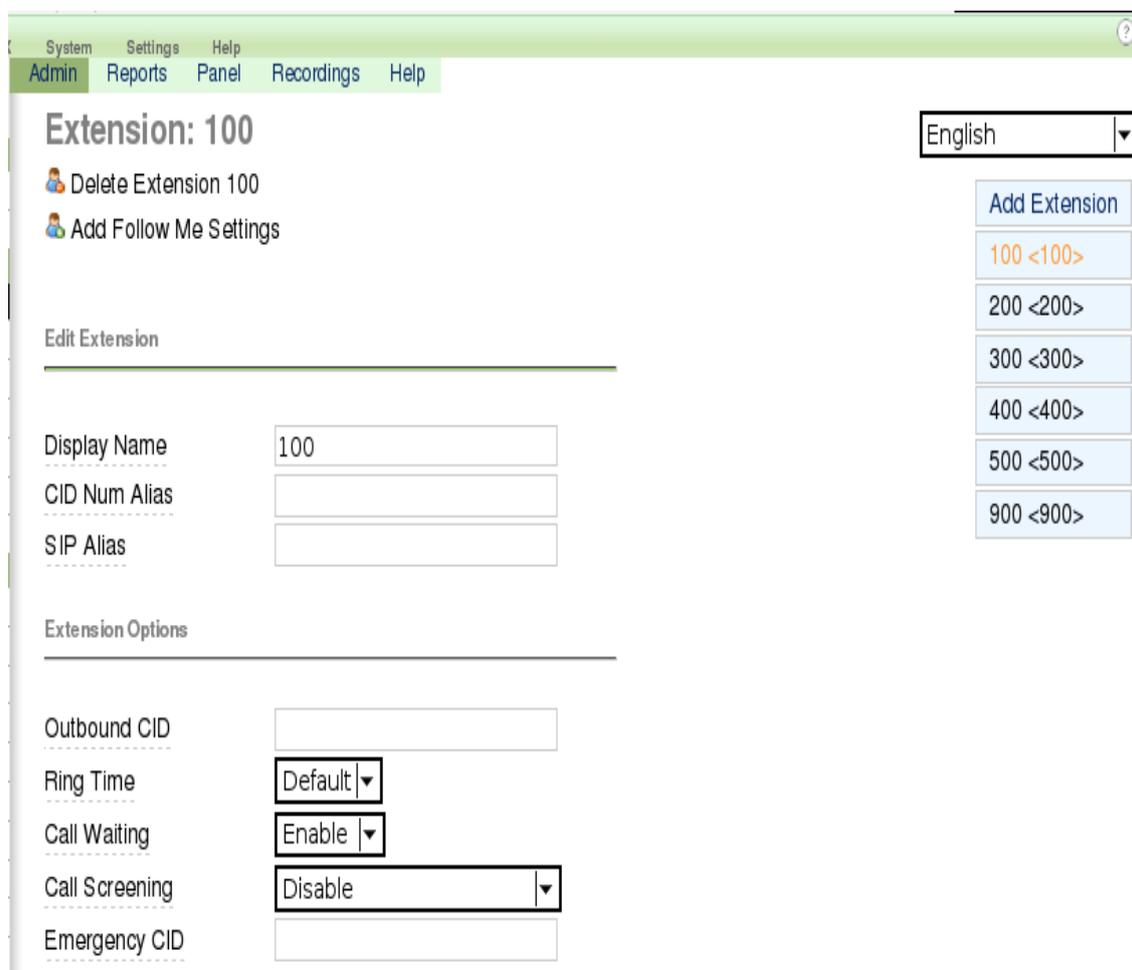
Figura 2.5.23 Trixbox. Definición de extensión.

Sólo habrá que abrir la opción de extensiones y en la opción add extensión, damos a botón de submit.

Para una extensión básica sólo tenemos que introducir los valores para el Número de extensión (Extension number), Nombre a mostrar (Display name) y Número secreto o password (secret).

Editando extensiones:

Para editar una extensión, vamos al menú de extensiones, seleccionamos una extensión y aparecerán bastantes más opciones que las que teníamos cuando se crearon (figura 2.5.24).



The screenshot shows the Trixbox web interface for editing extension 100. The top navigation bar includes 'System', 'Settings', and 'Help'. Below it, a secondary menu has 'Admin', 'Reports', 'Panel', 'Recordings', and 'Help'. The main content area is titled 'Extension: 100' and includes a language dropdown set to 'English'. On the right side, there is a vertical list of buttons for adding extensions: 'Add Extension', '100 <100>', '200 <200>', '300 <300>', '400 <400>', '500 <500>', and '900 <900>'. The main configuration area is divided into two sections: 'Edit Extension' and 'Extension Options'. The 'Edit Extension' section contains three input fields: 'Display Name' (containing '100'), 'CID Num Alias', and 'SIP Alias'. The 'Extension Options' section contains five fields: 'Outbound CID' (input), 'Ring Time' (dropdown set to 'Default'), 'Call Waiting' (dropdown set to 'Enable'), 'Call Screening' (dropdown set to 'Disable'), and 'Emergency CID' (input).

Figura 2.5.24 Trixbox, edición de extensión.

En la línea nat: never, tenemos que cambiar ésta a nat: yes. Esto es obligado sobre todo para los teléfonos que estén en distinta subset (subred) que el servidor Asterisk, y para aquellos teléfonos que utilicen NAT. Si tenemos problemas como audio en una sola dirección, esto se deberá siempre a problemas de NAT. Si cambiamos esta configuración solucionaremos el problema.

2.5.4.5.4 Creación de Troncales.

Troncales VOIP:

Vamos a definir una troncal VOIP usando un proveedor de servicios VOIP, en este caso Diamondcard..

Para comenzar tenemos que hacer Login a “Trixbox Web admin center” y cambiar a modo "Admin" (user maint).

Añadiendo troncal:

Para configurar la troncal, desde el menú se escoge:

Setup->Trunks->Add SIP Trunk

Trunk Name:

diamondcard

PEER Details:

type=peer

username=Tu_CuentaDiamondcard

fromuser=Tu_CuentaDiamondcard

secret=Tu_Codigo_PIN

host=sip.diamondcard.us

disallow=all

allow=gsm,ulaw

insecure=invite

context=from-diamondcard

Si estamos planeando usar números personales de Diamondcard (DIDs), deberemos configurar también:

USER Context:

from-diamondcard

USER Details:

Dejarlo en blanco

Register String:

Tu_CuentaDiamondcard: Tu_Codigo_PIN @diamondcard

Rutas salientes:

Para hacer llamadas salientes, deberemos definir ruta de salida. Esto asociará el prefijo de salida con ciertas troncales. En este ejemplo, usamos el prefijo 011 ligado con la troncal SIP "diamondcard". Esto significa que cuando en una PBX una extensión marca cualquier número que comience por 011, se mandará a través del troncal de Diamondcard.

Ejemplo, si la extensión marca el 011442088461122 la PBX mandará la llamada al número 442088461122 (Reino Unido) vía Diamondcard.

Podemos usar otros prefijos. Nos debemos asegurar que el número enviado a Diamondcard esté en formato internacional. El símbolo "+" se debe usar para cortar el prefijo marcado.

Para añadir ruta saliente, elegimos desde el menú:

Setup->Outbound Routes->Add Route

Route Name:

Diamondcard

Dial Patterns:

011.

Trunk Sequence:

SIP/diamondcard

Extensiones:

Debemos configurar al menos una extensión, que hará las llamadas. La configuración de las extensiones, depende del equipo VOIP que vamos a usar con la PBX.

Realización de llamadas:

Los números marcados que siguen los prefijos de las "Rutas salientes" se enviarán a través de la troncal (canal) configurada.

Definiendo una troncal PSTN:

Si hemos comprado una tarjeta X100P, o una tarjeta Digium TDM400, podemos configurar una troncal PSTN.

Para ello hemos de instalar y configurar la tarjeta X100P dentro del equipo que aloja el servidor Trixbbox.

Una vez hecho esto debemos definir la troncal PSTN: para ello vamos a menú PBX, Admin, Basic, Trunks y allí seleccionamos una troncal Zap Trunk (DAHDI compatibility mode).

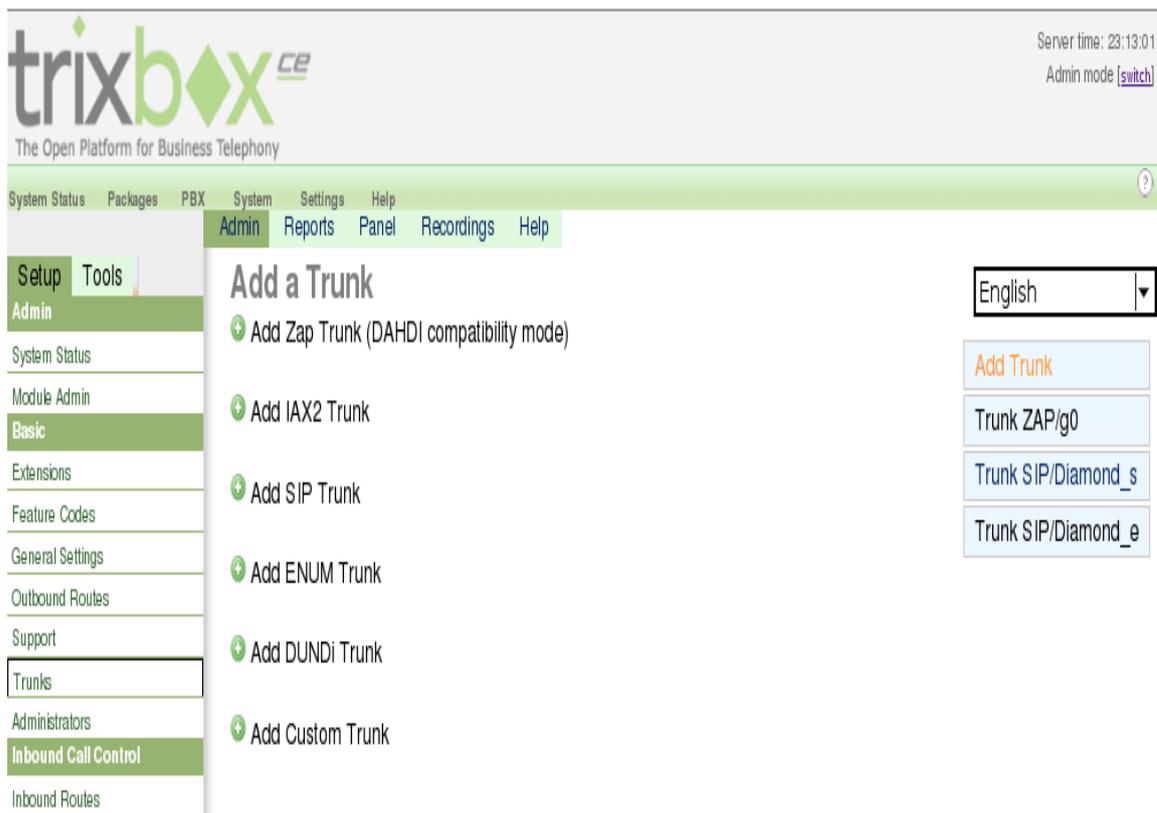


Figura 2.5.25 Trixbbox. Añadir troncal.

Al seleccionar Add ZAP trunk, tendremos la siguiente pantalla:

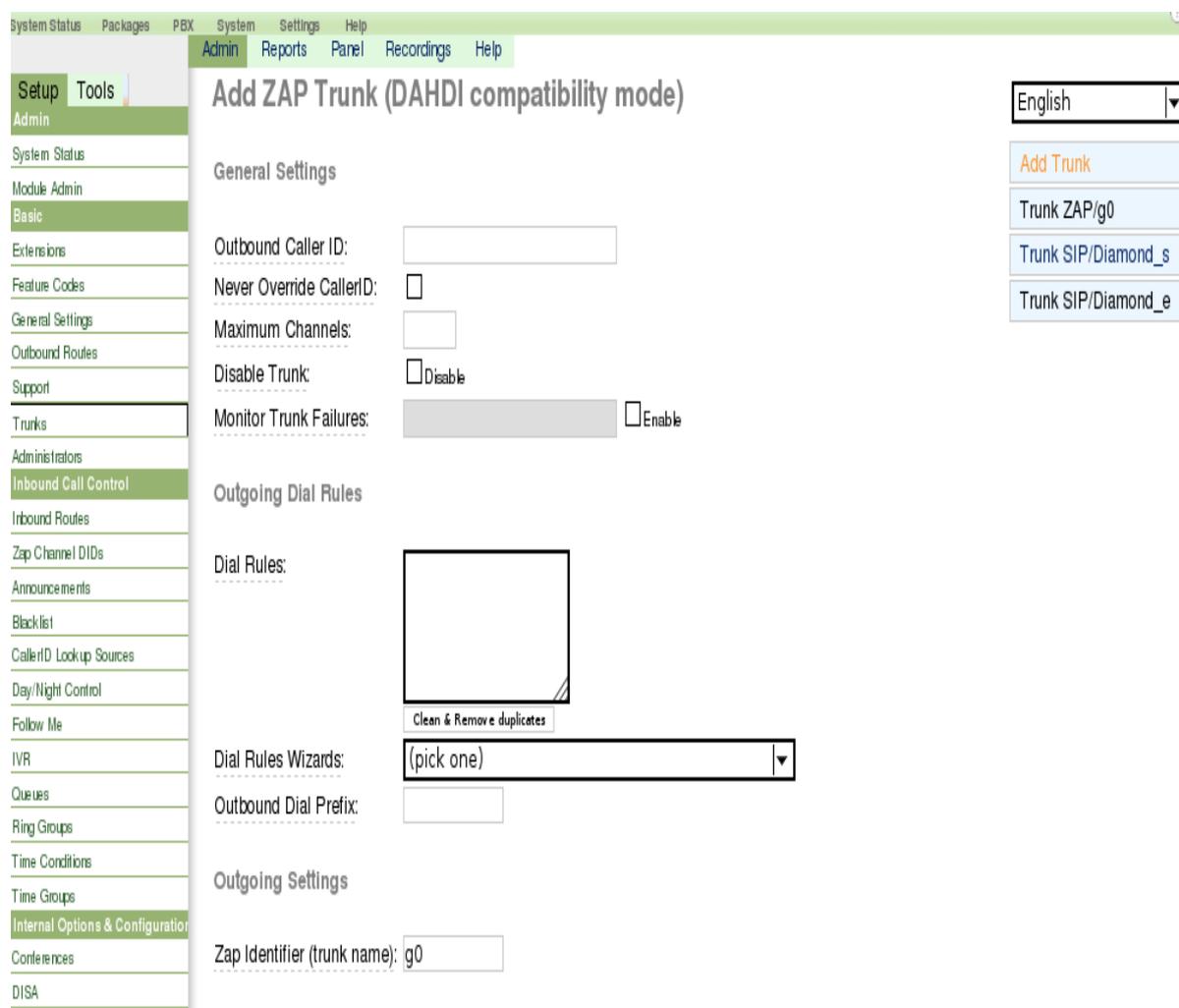


Figura 2.5.26 Trixbox. Añadir troncal zaptel.

Se selecciona la línea existente, y se guardan los cambios, presionando en el botón submit que aparece en la pantalla.

2.5.4.5.5 Creación de rutas.

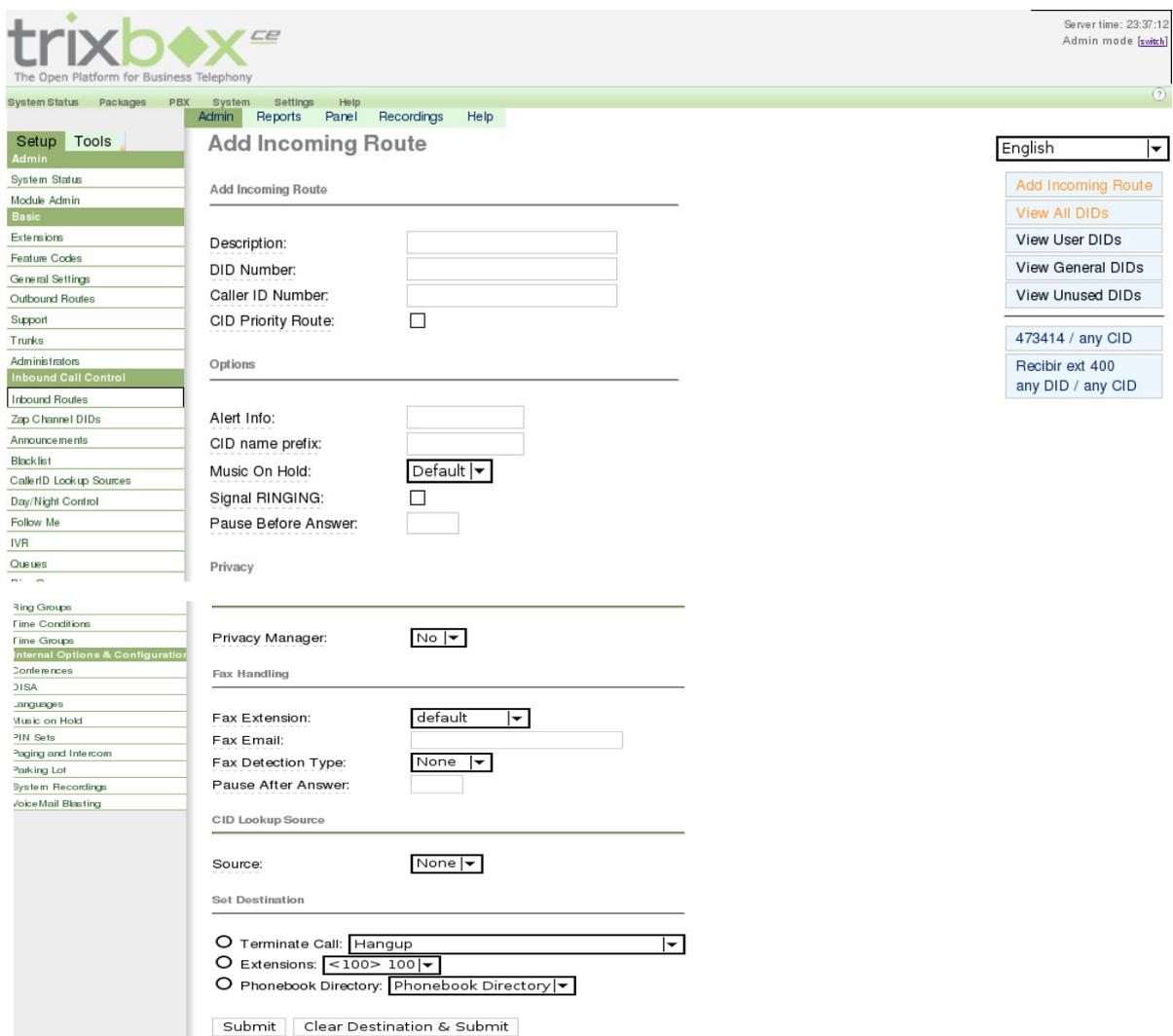
Una vez definidas las troncales, y las extensiones hemos de definir la rutas, o cómo se dirigen la llamadas.

Empezaremos con las rutas entrantes.

Rutas de entrada:

Las rutas de entrada tomarán las llamadas entrantes y las direccionarán a su destino. (DID. Direct Inward Dialing).

Basándose en el número al que se está llamando, se determinará a quién va la llamada. Al menos habrá una ruta de entrada. En configuraciones grandes, los departamentos o usuarios tienen su propio número DID.



The screenshot displays the Trixbox CE administration interface. The main heading is "Add Incoming Route". The interface includes a navigation menu on the left with categories like "Setup", "Tools", "Admin", "Basic", "Administrators", and "Internal Options & Configuration". The "Add Incoming Route" form contains several sections:

- Add Incoming Route:** Fields for "Description:", "DID Number:", "Caller ID Number:", and "CID Priority Route:" (checkbox).
- Options:** Fields for "Alert Info:", "CID name prefix:", "Music On Hold:" (dropdown menu set to "Default"), "Signal RINGING:" (checkbox), and "Pause Before Answer:".
- Privacy:** "Privacy Manager:" dropdown menu set to "No".
- Fax Handling:** Fields for "Fax Extension:" (dropdown menu set to "default"), "Fax Email:", "Fax Detection Type:" (dropdown menu set to "None"), and "Pause After Answer:".
- CID Lookup Source:** "Source:" dropdown menu set to "None".
- Set Destination:** Radio buttons for "Terminate Call:" (dropdown menu set to "Hangup"), "Extensions:" (dropdown menu set to "<100> 100"), and "Phonebook Directory:" (dropdown menu set to "Phonebook Directory").

At the bottom of the form are "Submit" and "Clear Destination & Submit" buttons. On the right side of the page, there is a language dropdown menu set to "English" and a list of buttons: "Add Incoming Route", "View All DIDs", "View User DIDs", "View General DIDs", "View Unused DIDs", "473414 / any CID", and "Recibir ext 400 any DID / any CID". The top right corner shows "Server time: 23:37:12" and "Admin mode [switch]".

Figura 2.5.27 Trixbox. Adición de la ruta de entrada.

La configuración básica de una ruta de entrada es:

- Número DID: Identifica el número al que se llama internamente.
- Número Caller-ID: Se basa en el número de la persona que está llamando. Esto es bueno para poder direccionar, por ejemplo, colas de personas VIP.
- Extensiones de FAX: Podemos configurar que los faxes vayan a alguna extensión o bien al sistema de extensiones de fax.
- Fax Email: Si la dirección de envío de los faxes está en blanco, se usa la dirección de e-mail definida en las configuraciones generales.
- Detección del tipo de fax: Detecta los faxes, chequeando las llamadas entrantes.
- Pausa después de respuesta: Si estamos usando la detección de fax, necesitamos poner la opción del número de segundos a esperar después de responder una llamada y determinar si hay tonos de fax presentes.
- Privacy manager: Cuando está activado el privacy manager, y no hay identificación del llamante (caller-id), el sistema preguntará al llamante por sus 10-dígitos del número telefónico antes de continuar.
- Información de alerta: Se utiliza para modificar el tono empleado con dispositivos SIP.
- Establecer destino (set destination): El paso final es definir el destino de la llamada. El destino puede ser un recepcionista digital, una extensión, una mensajería, un grupo de llamada, una cola, una aplicación, una condición de tiempo o cualquier otro destino posible.

Rutas de salida:

Es una de las principales opciones de la configuración de Trixbox, ya que con ella se puede ahorrar una buena cantidad de dinero, usando, aquella ruta más económica en función de la localización de la persona llamada.

Por defecto, Trixbox dirige cada llamada saliente que comienza con el prefijo 9 a través de la troncal Zaptel.

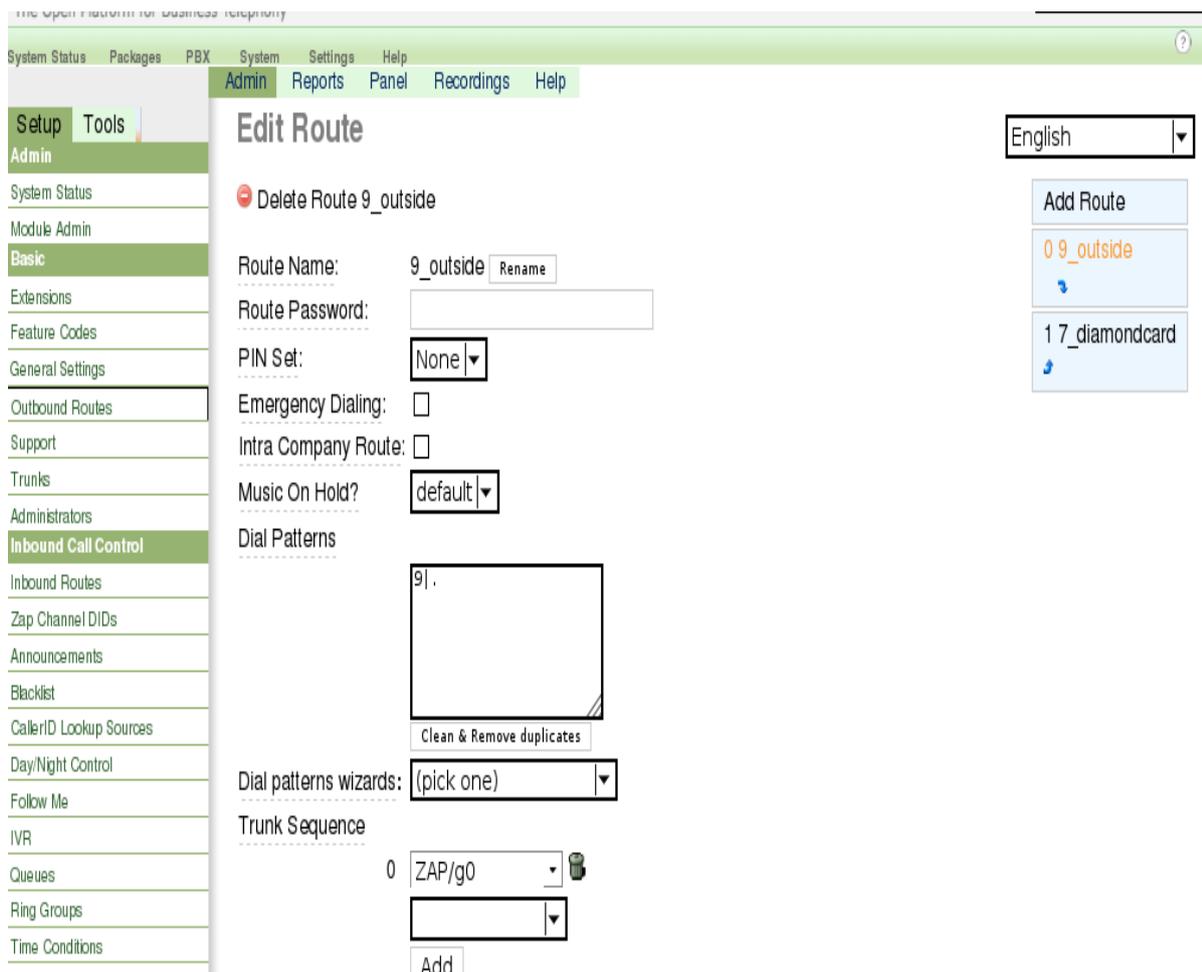


Figura 2.5.28 Trixbox. Edición de la ruta de salida.

La definición de la ruta tiene varios componentes:

- Nombre de la ruta (Route Name).
- Password de la ruta: Pregunta al usuario por clave para acceder a las líneas de pago.
- Modelos de plan de marcado: Define qué llamadas van por la ruta.
- Secuencia de troncales, la secuencia en la que se asignan las troncales dentro de la ruta.

2.5.4.5.6 Planes de marcado.

Debemos crear un plan de marcado que asegure que nuestras llamadas se manejan adecuadamente.

Los caracteres válidos, para un plan de marcado, se muestran en la tabla anexa:

Caracteres de marcado	Significado
X	Cualquier dígito de 0-9
Z	Cualquier dígito de 1-9
N	Cualquier dígito de 2-9
[1237-9]	Cualquier dígito o letra en los paréntesis (en este ejemplo 1,2,3,7,8,9)
.	Comodines: uno o más caracteres .
	Separa prefijos de marcado de números.
+	Añade dígitos precedentes al número.

Por ejemplo, si tenemos en los modelos de marcado (Dial Patterns) 9| y marcamos el número 91714555212, el sistema separará el 9 del número y mandará por la troncal ZAP/g0 el resto de números.

2.5.4.5.7 Grupos de llamada.

Si queremos que una opción de un menú guiado de acceso vaya a un grupo, por ejemplo, el grupo de soporte, hemos de crear un grupo de marcado.

La estrategia más fácil es que suenen todos los números del grupo de marcado al mismo tiempo.

Si queremos que se llame a una extensión externa, por ejemplo, un teléfono móvil, pondremos el número del móvil seguido del símbolo de la libra (Pound), al final.

Si ponemos algo en el campo "CID name prefix", el sistema pondrá este prefijo a la información del llamante, pudiendo conocer nosotros en qué grupo entra.

Con la estrategia de llamada, “hunt”, el sistema va llamando a cada extensión, mantiene una jerarquía de llamada, y mantiene cada teléfono en llamada.

La estrategia memoryhunt, llama primero al primer teléfono de la lista, después al primero y al segundo, luego primero segundo y tercero y así sucesivamente.

The screenshot displays the 'Add Ring Group' configuration page in Trixbox. On the left is a navigation menu with categories like 'Setup' and 'Tools'. The main content area is titled 'Add Ring Group' and contains the following fields:

- Ring-Group Number:** 600
- Group Description:** (empty text field)
- Ring Strategy:** ringall (dropdown menu)
- Ring Time (max 60 sec):** 20
- Extension List:** (empty text area)
- Extension Quick Pick:** (pick extension) (dropdown menu)
- Announcement:** None (dropdown menu)
- Play Music On Hold?:** Ring (dropdown menu)
- CID Name Prefix:** (empty text field)
- Alert Info:** (empty text field)
- Ignore CF Settings:**
- Skip Busy Agent:**
- Confirm Calls:**
- Remote Announce:** Default (dropdown menu)
- Too-Late Announce:** Default (dropdown menu)

Below these fields is the 'Destination if no answer:' section with radio button options:

- IVR: Unnamed (dropdown menu)
- Terminate Call: Hangup (dropdown menu)
- Extensions: <100> 100 (dropdown menu)
- Phonebook Directory: Phonebook Directory (dropdown menu)

A 'Submit Changes' button is located at the bottom of the form.

Figura 2.5.29 Trixbox. Grupo de llamada.

2.5.4.5.8 Colas de llamadas.

Cuando una llamada llega a la centralita, si el destinatario está ocupado, la llamada entra en una cola, la persona puede recibir música e incluso información de tiempo en espera o la posición en la cola.

Las funciones que una cola de llamadas puede tener son:

- Número de la cola: Con este número se puede acceder a la cola. Por ejemplo, se accedería a la cola marcando 123* y se dejaría la cola marcando 123**.
- Nombre de la cola: Se le puede asignar un nombre para identificarla de forma visual.
- Clave de la cola: Se puede activar un password de acceso a la cola.
- Nombre de prefijo CID: Si se activa, compara el CID del teléfono que llama con el de la cola.
- Agentes estáticos: Si algunas extensiones se suponen que van a estar siempre en las colas, a esas extensiones se les llama agentes estáticos.

Aparte de estas opciones existen otras más como son:

- Mensaje antes de transferencia a la cola.
- Tipos de música en espera (on hold).
- Máximo tiempo de espera.
- Número máximo de llamadas.
- Unirse a colas vacías.
- Salir cuando la cola esta vacía.
- Estrategia de llamada.
- Reintentos.
- Grabación.

System Status Packages PBX System Settings Help

Admin Reports Panel Recordings Help

Setup Tools

Admin

System Status

Module Admin

Basic

Extensions

Feature Codes

General Settings

Outbound Routes

Support

Trunks

Administrators

Inbound Call Control

Inbound Routes

Zap Channel DIDs

Announcements

Blacklist

Caller ID Lookup Sources

Day/Night Control

Follow Me

IVR

Queues

Ring Groups

Time Conditions

Time Groups

Internal Options & Configuration

Conferences

DISA

Languages

Music on Hold

PIN Sets

Paging and Intercom

Parking Lot

System Restrictions

Add Queue

Add Queue

Queue Number:

Queue Name:

Queue Password:

CID Name Prefix:

Wait Time Prefix:

Alert Info:

Static Agents:

Extension Quick Pick:

Queue Options

Agent Announcement:

Join Announcement:

Music on Hold Class:

Ringing Instead of MoH:

Max Wait Time:

Max Callers:

Join Empty:

Figura 2.5.30 Trixbox. Adición de colas.

2.5.4.5.9 Configuraciones generales.

Las centralitas Trixbox tienen las opciones que se detallan a continuación:

Opción de marcado (Dialing Options):

Se puede configurar a cualquier combinación de las siguientes:

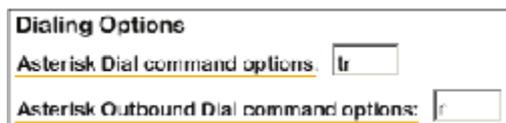
t, permite transferir la llamada usando la tecla #

T, permite al usuario que hace la llamada transferirla.

r, pone un tono al usuario que realiza la llamada.

w, permite grabar la llamada recibida, presionando *1.

W, permite grabar al usuario que realiza la llamada.



Dialing Options

Asterisk Dial command options:

Asterisk Outbound Dial command options:

Figura 2.5.31 Trixbox. Opciones de marcado.

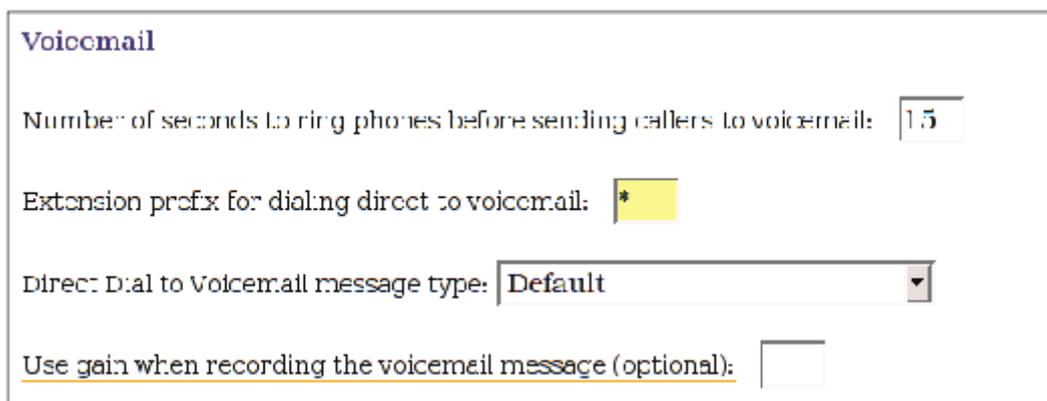
Mensajería de Voz (Voicemail):

Se puede configurar:

Número de segundos de espera para que entre el buzón de voz.

Prefijo para llamar directamente al buzón de voz.

Mensaje recibido al llamar al buzón de voz.



Voicemail

Number of seconds to ring phones before sending callers to voicemail:

Extension prefix for dialing direct to voicemail:

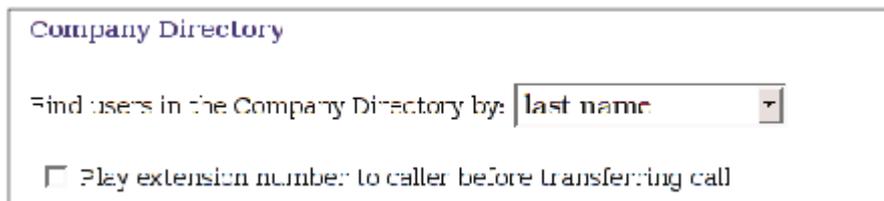
Direct Dial to Voicemail message type:

Use gain when recording the voicemail message (optional):

Figura 2.5.32 Trixbox. Buzón de voz.

Directorio de la compañía (Company Directory):

Se puede configurar para encontrar usuarios por el apellido o el nombre.



Company Directory

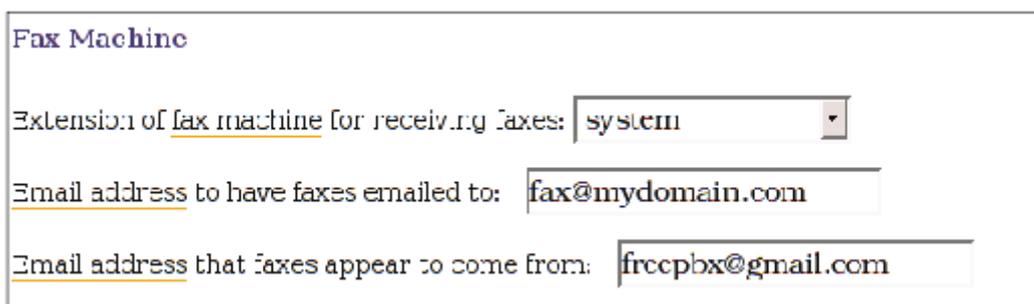
Find users in the Company Directory by:

Play extension number to caller before transferring call

Figura 2.5.33 Trixbox. Directorio.

Configuración de fax (Fax setting):

Los faxes se pueden enviar a una extensión de destino, los faxes también se pueden enviar a una dirección de correo.



Fax Machine

Extension of fax machine for receiving faxes:

Email address to have faxes emailed to:

Email address that faxes appear to come from:

Figura 2.5.34 Trixbox. Configuración de fax.

Configuración internacional (International settings):

Aquí debemos elegir el país correcto y la región .

Configuración de seguridad (Security Settings):

Si no queremos que las personas nos llamen usando el protocolo SIP, debemos configurar esta opción en “no”. Cambiando a “yes”, permitimos a las personas llamar a las extensiones SIP de la centralita, sin haberse conectado primero con un usuario nombre y clave (sin estar conectadas previamente a la PBX).



Security Settings

Allow Anonymous Inbound SIP Calls?

Figura 2.5.35 Trixbox. Configuración de seguridad.

Opciones estándar en una centralita (PBX):

Redirección de llamada:

La más popular es la redirección de llamada cuando no se contesta, en este caso se puede llamar a un grupo de llamada con números alternativos, tales como número de móviles o números fijos.

Las funciones que se pueden obtener con los códigos correspondientes son:

- *72 redirección de llamada, todo activado
- *73 redirección de llamada, todo desactivado
- *74 redirección de llamada, todo los prompt desactivado
- *90 redirección de llamada, ocupado activado
- *91 redirección de llamada, ocupado desactivado
- *92 redirección de llamada, prompt ocupado desactivado
- *52 redirección de llamada, no respuesta / no disponible activado
- *53 redirección de llamada, no respuesta / no disponible desactivado

Llamada en espera:

Es el beep que se oye cuando tenemos una segunda llamada. Asterisk permite que varias llamadas vengan a un teléfono, si el teléfono en sí lo soporta.

Por defecto, la llamada en espera está deshabilitada en cada extensión, este comportamiento se puede cambiar editando el fichero `/etc/ampportal.conf` y cambiando `ENABLECW` a `yes`.

Llamada en espera activada, marcar *70.

Llamada en espera desactivada, marcar *71.

Core:

Con esta opción podemos simular llamadas entrantes, para ver cómo se gestionan.

Simular llamadas entrantes, marcar código 7777.

Simular fax entrantes, marcar código 666.

No molestar (DND . Do-Not-Disturb) :

Simplemente redirecciona todas las llamadas entrantes a nuestro buzón de voz .

Activar DND , marcar *78.

Desactivar DND, marcar *79.

Información de servicios:

Son herramientas para ver que nuestra central está funcionando adecuadamente.

Seguimiento de llamada, *69.

Directorio, #.

Test de eco, *43.

Hablar a nuestro número de extensión, *65.

Hablando al reloj, *60.

Grabación:

Este módulo nos ayuda a grabar y reproducir la grabación que estamos haciendo.

Prueba la grabación, *77.

Salva la grabación, *99.

Control de dispositivos:

La función longon y logoff, nos permite cuando trabajamos con dispositivos, tomar el control de éstos. La función ZapBarge, nos permite escuchar, una vez que las llamadas están sobre los canales Zap .

Logoff del usuario, *12.

Logon de usuario, *11.

ZapBarge, 888.

Códigos activos durante las llamadas (Active-Call Codes):

Durante las llamadas hay una serie de códigos activos:

- Entrar en transferencia de llamada, #.
- Aparcar la llamada en curso, #70.
- Empezar grabando la llamada, *1.
- Terminar grabación de llamada, *2.
- Transferir directamente al buzón de voz de la extensión llamada, *+Extensión.

Sistema de marcado rápido (System-Wide Speed Dialling):

El sistema usa las extensiones 300 a 399 para el marcado rápido, si necesitamos usar extensiones en el rango 3xx debemos deshabilitar el sistema de marcado rápido en el fichero `extensions_custom.conf`.

Los comandos para el marcado rápido son:

- Añadir marcado rápido a extensiones 300 a 399, 3003xx+Número de teléfono.
- Hablar a la configuración de marcado rápido, *3xx.
- Marcar el número de marcado rápido especificado, 3xx.

Buzón de voz:

Hay dos formas de acceder al buzón de voz:

“Llamar al Buzón de voz”, marcando *97 desde una extensión, preguntará el password para la extensión.

Marcando *98, preguntará por la extensión y password.

Las opciones del buzón de voz son:

- 1, escuchar nuevos mensajes.
- 2, cambiar de grupo de opciones.
- 0, opciones del buzón.
- *, Ayuda.
- #, Salir.

Las opciones de escuchar mensajes, tiene subopciones:

- 0, Nuevos mensajes.
- 1, Mensajes viejos.
- 2, Mensajes de trabajo.
- 3, Mensajes familiares.
- 4, Mensajes de amigos.
- 5, Repetir mensaje.
- 6, Oír siguiente mensaje.
- 7, Borrar mensaje.
- 8, Redirigir a otro usuario.
- 9, Salvar mensaje.
- *, Ayuda.
- #, Salir al menú principal.

2.5.4.5.10 Panel rápido del operador (Flash Operator Panel).

Es un panel en el cual podemos ver de una forma rápida qué está sucediendo en la centralita.

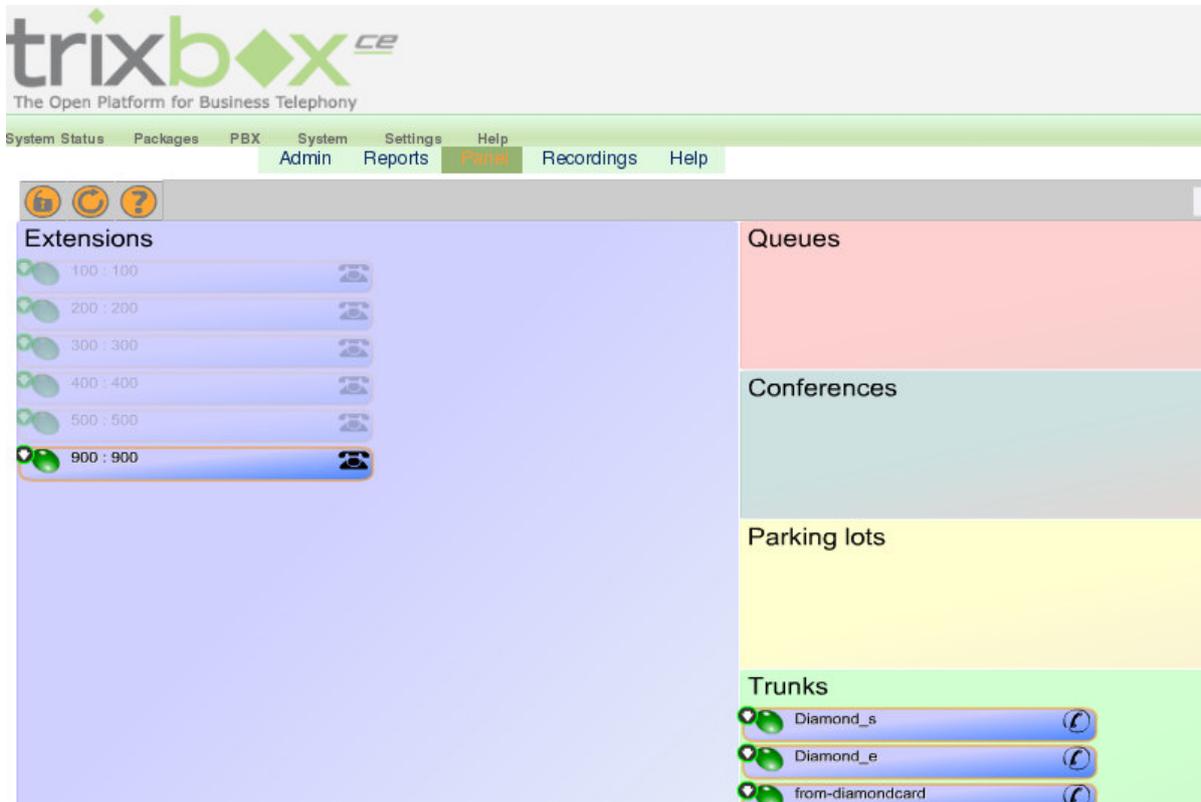


Figura 2.5.36 Trixbox. Flash operator panel.

El panel muestra:

- Qué extensiones están ocupadas, llamando o libres.
- Quién está llamando y a quién (CID, contexto, prioridad).
- Estado de registro SIP y IAX y si nos podemos conectar.
- Estado de sala de reuniones (número de participantes).
- Estado de las colas (número de usuarios esperando).
- Indicador de mensajes esperando y cuenta.
- Canales aparcados.
- Agentes conectados.

Cuando una extensión o troncal está en uso, el óvalo pasa de verde a rojo. Un óvalo parpadeando, indica que el canal está llamando.

Si hay actividad en algún botón y movemos el ratón sobre él, vemos información adicional en la barra de estado. Para cada sesión, la primera vez que intentamos hacer alguna acción, se nos preguntará por la contraseña (passw0rd).

Las acciones que se pueden hacer desde el panel son:

- Colgar un canal.
- Transferir una llamada arrastrando y soltando.
- Originar llamadas arrastrando y soltando.
- Poner el id del llamante cuando transferimos u originamos una llamada.
- Automáticamente sacar página Web con detalles del cliente.
- Picar para llamar desde una página Web.
- Silencio / Hablar para los participantes en una reunión.

2.5.4.6 Configuración avanzada de la centralita Trixbox.

Debido al uso de NAT y una sola dirección de internet provista por nuestro ISP de internet, es frecuente encontrar problemas de falta de sonido y de iniciación de llamadas, ya que el protocolo SIP no es muy amigable para utilizarlo en una NAT.

Para evitar esto tenemos que abrir algunos puertos en el Firewall hacia nuestro sistema Trixbox.

Puertos	Protocolo	Descripción
4569	UDP	IAX2 Soporte
5060-5061	UDP	SIP Puertos
10000-20000	UDP	SIP Soporte

Normalmente se hace “Port Forwarding” en nuestro router o firewall.

Consideraciones NAT:

Cuando tenemos un sistema NAT entre nuestro proveedor y la centralita Trixbox, podemos solucionar los problemas que tengamos acudiendo a los ficheros de configuración, y haciendo cambios en éstos desde la interfaz Web.

Para ello, vamos a la opción PBX del menú principal.

The screenshot displays the Trixbox web interface. At the top left is the Trixbox logo with the tagline "The Open Platform for Business Telephony". The top right corner shows "Server time: Admin mod". A navigation menu includes "System Status", "Packages", "PBX", "System", "Settings", and "Help".

Server Status: Asterisk, web server, cron server, SSH server, and Mysql are all shown as "Running".

Helpful Links: Forum, Recent Posts, HUD Lite, Video Tutorials, Documentation, FtOCC, Buy Support.

Announcements: A message titled "Found" states "The document has moved here."

Network Usage:

Device	Received	Sent	Err/Drop
lo	2.09 MB	2.09 MB	0/0
eth0	2.38 MB	4.07 MB	0/0
eth1	0.00 KB	0.00 KB	0/0
wmaster0	0.00 KB	0.00 KB	0/0
wlan0	0.00 KB	0.00 KB	0/0
sit0	0.00 KB	0.00 KB	0/0

Memory Usage:

Type	Percent Capacity	Free	Used	Size
- Kernel + applications	15%		148.59 MB	
- Buffers	9%		91.64 MB	
- Cached	21%		214.00 MB	
Disk Swap	0%	760.88 MB	0.00 KB	760.88 MB

Mounted Filesystems:

Mount	Type	Partition	Percent Capacity	Free	Used	Size
/	ext3	/dev/sda2	46% (18%)	21.06 GB	1.51 GB	25.28 GB

trixbox Status:

- Hostname: trixbox1.localdomain
- Local IP: 192.168.2.50
- Public IP:
- Active Channels: SIP: 0, IAX: 0
- Current Registrations: SIP: 2, IAX: 1
- SIP Peers: Online: 1, Offline: 5, Unmonitored: 5
- IAX2 Peers: Online: 0, Offline: 0, Unmonitored: 0
- Extensions DND

Figura 2.5.37 Trixbox. Pantalla principal.

Y desde allí vamos a la opción "config file editor"

phpconfig for Asterisk PBX

/etc/asterisk	/var/www/html/panel	/etc	/ftpboot	Re-Read Configs
adsis.conf				
adtranvoipr.conf				
agents.conf				
alarmreceiver.conf				
alsa.conf				
amd.conf				
asterisk.conf				
cbmysql.conf				
cdr_custom.conf				
cdr_manager.conf				
cdr_mysql.conf				
cdr_pgsql.conf				
cdr_tds.conf				
chan_dahdi.conf				
chan_dahdi_additional.conf				
cli.conf				
codecs.conf				
console.conf				

Figura 2.5.38 Trixbox. Configuración de ficheros.

En esta pantalla vemos los ficheros principales de configuración de Trixbox, que podemos cambiar manualmente.

Si queremos modificar el fichero sip_nat.conf, picamos en él y lo mostramos.

El fichero mostrará algo como lo indicado a continuación:

```
Edit: sip_nat.conf  
[general]  
externip=55.66.77.88 ; Change to match our external IP address  
localnet=192.168.1.0/255.255.255.0 ; Change to match our network settings
```

Figura 2.5.39 Trixbox. Fichero sip_nat.conf.

Debemos añadir la siguiente información:

[general]

externip=55.66.77.88 ; Cambiar a nuestra dirección internet externa.

localnet=192.168.1.0/255.255.255.0; Cambiar a la configuración de nuestra red interna.

externip debe ser la dirección de IP que nos da nuestro proveedor externo de internet . Si no la conocemos podemos usar la aplicación <http://whatismyip.com> .

Si no tuviéramos una IP estática, y nuestro proveedor de IP nos la cambia, entonces cualquier usuario remoto o conexión ITSP fallará hasta que no cambiemos el externip y reiniciemos Asterisk.

Ésta es una de las razones por las que usar una IP estática es especialmente útil, sobre todo usando VoIP.

La segunda línea localnet, le dice a Asterisk el rango IP sobre el que está el servidor. Localnet se debe poner en el intervalo IP de nuestra red de área local.

Para algunas extensiones remotas, debemos editar la extensión y cambiar la configuración para la NAT a “yes”. Esto le dice a Asterisk que la extensión no está en la misma red que el servidor.

Como sea que una vez que tengamos el sistema arriba y corriendo, si sólo tenemos audio en un lado de la conversación, esto es casi siempre un problema con la configuración de la NAT.

Configurando las tarjetas ZAPTEL:

Normalmente para comenzar a utilizar tarjetas telefónicas, se usan tarjetas Zaptel, modelos X100P o TDM400. Para instalar estas tarjetas, se corre el script genzaptelconf.

```
[root@Asterisk1 ~]# genzaptelconf

STOPPING ASTERISK

Disconnected from Asterisk server

Asterisk Stopped

STOPPING FOP SERVER

FOP Server Stopped

Generating '/etc/zaptel.conf'

Generating '/etc/Asterisk/zapata-auto.conf'

Unloading zaptel hardware drivers:

Unloading ztdummy:           [ OK ]
```

Una descripción más detallada de la instalación de la tarjeta, se puede ver en el manual de la tarjeta a instalar.

El script genera un fichero llamado zaptel.conf que se puede encontrar en el directorio /etc/.

Al ver el fichero para una tarjeta TDM400 con 4 módulos FXO tenemos lo siguiente:

```
# Span 1: WCTDM/0 "Wildcard TDM400P Board 1"

fxsks=1

fxsks=2

fxsks=3

fxsks=4
```

Configuración de las tarjetas T1 / E1:

La configuración de estas tarjetas en el sistema hay que hacerlas manualmente, modificando los ficheros de configuración.

A continuación configuramos una tarjeta TE110P de Digium:

Editamos el fichero zaptel.conf

```
[root@Asterisk1 ~]# nano /etc/zaptel.conf
```

Una vez dentro del fichero añadimos la siguiente información:

```
# TE110P T1 Card / Typical US Settings  
span=1,1,0,esf,b8zs  
bchan=1-23  
dchan=24
```

Después Ctrl+O y Ctrl+X para salir del editor nano, salvando.

La información anterior define la tarjeta y cómo ésta comunica con el exterior.

A continuación definimos los canales y cómo éstos funcionan con Asterisk.

```
[root@Asterisk1 ~]# nano /etc/Asterisk/zapata.conf
```

Añadimos lo siguiente:

```
[channels]  
switchtype=national  
context=from-pstn  
signalling=pri_cpe  
group=1  
channel => 1-23  
language=en  
context=from-pstn  
signalling=fxs_ks
```

Esto define el número del grupo, y los canales que pertenecen a ese grupo.

Anuncios a través de la centralita (Overhead Paging):

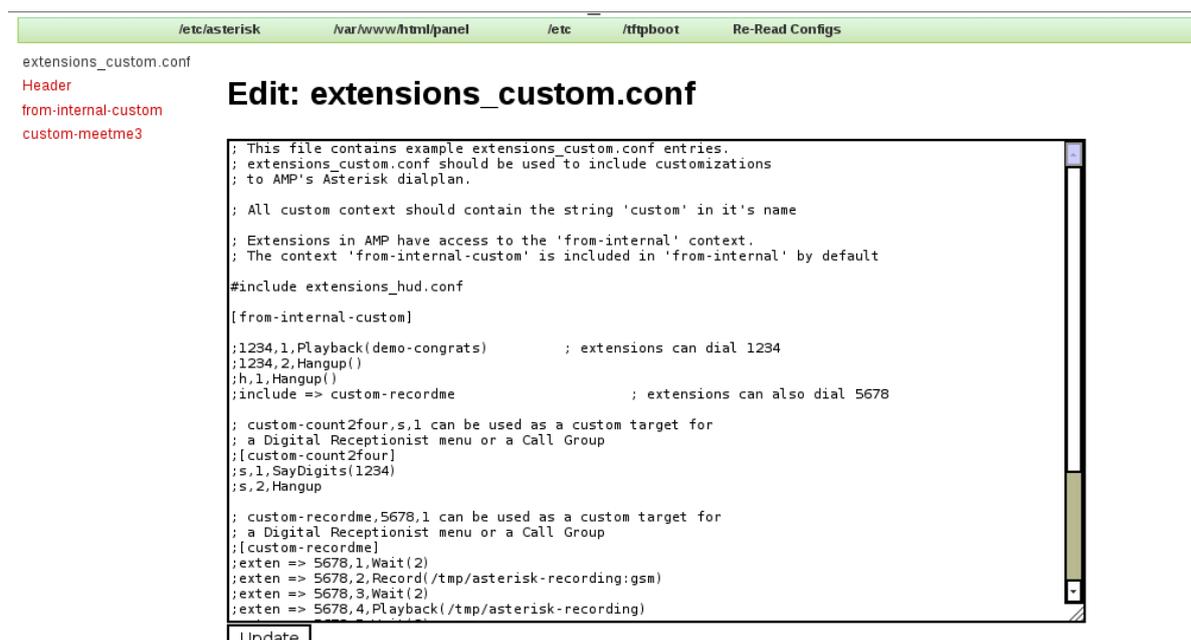
Es una funcionalidad de las centralitas como las utilizadas en los supermercados, aeropuertos, estaciones de trenes, etc, para hacer anuncios.

En el sistema Trixbox esto se hace, creando una extensión donde podemos llamar para hacer anuncios.

Para esto el PC de la Trixbox debe tener una tarjeta de sonido.

Hay dos pasos para hacer esto: Primero, definir las líneas de avisos en el plan de marcado y segundo, habilitar la tarjeta de sonido en Linux.

Para ello hemos de ir a la página Config File Editor y editar el fichero `extensions_custom.conf`.



```
extensions_custom.conf
Header
from-internal-custom
custom-meetme3

; This file contains example extensions_custom.conf entries.
; extensions_custom.conf should be used to include customizations
; to AMP's Asterisk dialplan.

; All custom context should contain the string 'custom' in it's name

; Extensions in AMP have access to the 'from-internal' context.
; The context 'from-internal-custom' is included in 'from-internal' by default
#include extensions_hud.conf
[from-internal-custom]
;1234,1,Playback(demo-congrats)      ; extensions can dial 1234
;1234,2,Hangup()
;h,1,Hangup()
;include => custom-recordme        ; extensions can also dial 5678

; custom-count2four,s,1 can be used as a custom target for
; a Digital Receptionist menu or a Call Group
;[custom-count2four]
;s,1,SayDigits(1234)
;s,2,Hangup

; custom-recordme,5678,1 can be used as a custom target for
; a Digital Receptionist menu or a Call Group
;[custom-recordme]
;exten => 5678,1,Wait(2)
;exten => 5678,2,Record(/tmp/asterisk-recording:gsm)
;exten => 5678,3,Wait(2)
;exten => 5678,4,Playback(/tmp/asterisk-recording)

Update
```

Figura 2.5.40 Trixbox. Fichero `extensions_customs.conf`.

Añadimos en el área `[from-internal-custom]` las siguientes líneas:

```
exten => *52,1,Dial(console/dsp)
```

```
exten => *52,2,Hangup()
```

Para habilitar la tarjeta de sonido se ejecuta el comando:

```
[root@Asterisk1 ~]# alsamixer
```

Podemos acceder al sistema de Alertas, tecleando *52 y hablando a continuación.

Podemos acceder a controlar la voz usando el programa alxamixer.

Dentro de este programa podemos “un-mute” el “Master Volume” y PCM, usando la tecla M del teclado, y para controlar el volumen podemos usar las teclas arriba, abajo del teclado.

Bloquear identidad del llamante (Caller-id):

Si estamos usando una red PSTN y queremos bloquear la identificación de identidad, se puede usar el siguiente script para enviar el código *67 a la línea telefónica:

```
exten => *67,1,Dial(ZAP/1/*67 )
```

```
exten => *67,2,Wait(3)
```

```
exten => *67,3,SoftHangup(ZAP/1)
```

```
exten => *67,5,DISA(no-password|from-internal)
```

2.6 Equipos y Sistemas de comunicaciones IP.

2.6.1 Introducción.

Para poder utilizar los medios disponibles para la comunicación sobre VOIP, se utilizan sistemas de distintas tecnologías. Cabe destacar los siguientes:

Sistemas o aplicaciones VOIP basadas en software:

- Softphones (Teléfonos basados en software)
- Aplicaciones sobre internet para compartir recursos de voz.
 - Teamspeak
 - Podcast
 - Radionomy
 - Icecast
 - Aplicaciones Streaming (Transmisión de voz o vídeo)

Sistemas hardware VOIP:

- Hard Phone (Teléfonos basados en electrónica)
- Cámaras IP
- PBX
- Switches/Gateways

Dentro del Demostrador de Conceptos Marconi se ha implementado la solución Teamspeak, implementando un Servidor detrás del firewall IPCop y del firewall de la Universidad.

Para depuración de errores e instalaciones se ha recurrido al foro de Teamspeak, y concretamente al área de “Servidores”, donde se han ido solucionando los errores aparecidos durante la instalación y puesta en marcha. La información de instalación se ha recopilado de Internet así como de la programación de la aplicación en si, siendo esta información muy escasa y requiriendo la depuración en un PC piloto antes de la instalación de la aplicación en el servidor Marconi.

Hay que destacar que se han instalado pocos servidores locales, recurriéndose más al

uso del Teamspeak en Server profesionales ya operativos.

Para la integración de equipos tipo cámara o teléfono, se ha recurrido a la compra de teléfonos y cámaras Grandstream, que se han configurado dentro de la centralita Trixbox de la red Marconi. Se han configurado estos teléfonos y cámaras y se ha recurrido al distribuidor de Grandstream Avanzada 7 y a la propia Grandstream y su foro para implementar la capacidad de streaming de vídeo y voz usando navegadores como firefox o Chrome.

Como punto a remarcar dentro de la configuración de los teléfonos Grandstream es que es necesario configurar los planes de marcado y tener cargados los códecs de vídeo y audio a usar durante las llamadas.

Dentro de las aplicaciones Softphone se han analizado las existentes en el mercado, seleccionando Exprextalk, Zoiper, Linphone y Csipsimple como clientes dentro del demostrador de conceptos Marconi. La elección se ha debido principalmente a su buen funcionamiento, facilidad de configuración y buena respuesta tanto en voz como en vídeo.

Naturalmente Skype como aplicación VOIP está suficientemente probada y funciona perfectamente en todos los entornos, siendo en casos específicos como Debian y algunos otros, donde las aplicaciones son beta y hay que recurrir a consultar en los foros adecuados.

Todas las aplicaciones apuntadas en los puntos siguientes han sido probadas y están accesibles en el Demostrador de Conceptos Marconi.

2.6.2 Equipos Softphones (Teléfonos Software).

Un Softphone es un software que es utilizado para realizar llamadas a otros softphones o a otros teléfonos convencionales usando VOIP.

Normalmente, un Softphone es parte de un entorno VOIP y puede estar basado en el estándar SIP/H.323. También soportan protocolos propios de Asterisk, AIX.

Hay muchas implementaciones disponibles, las más conocidas son:

- Skype
- Messenger

- Googletalk

De estas últimas Skype y Googletalk se usan principalmente en sistemas operativos Windows, existiendo su equivalente para otros sistemas operativos como el Linux.

Entre éstas están:

- Empathy para Debian y Ubuntu
- SFLphone para Debian
- Ekiga para Debian y Ubuntu
- Linphone para Debian y Ubuntu

Como aplicaciones de telefonía para Windows tenemos:

- Zoiper
- Expresstalk
- CounterPath
- 3CX (usado por la Universidad de Cádiz desde 2012)

También se dispone de aplicaciones de telefonía para Android:

- Ekiga
- csipsimple
- 3CX

Estas últimas han sido obtenidas a través de Android Market.

Los Softphones (teléfonos software) funcionan correctamente con la mayoría de los proveedores de ISP (Proveedores de Servicios de Telefonía por Internet). Caso de encontrar dificultades en algunos es debido a la deshabilitación de algunas de las opciones necesarias en el router para usar el protocolo SIP. En este caso habrá que recurrir a analizar el problema acudiendo a los foros expertos en el tema como por ejemplo bandaancha.eu.

Los teléfonos software se pueden definir como una extensión en las centralitas telefónicas Asterisk, saliendo hacia el exterior a través de los canales de éstas.

La conexión de los teléfonos a las centralitas telefónicas se hace usando el protocolo SIP o bien IAX , éste a su vez establece la conexión sobre TCP/IP.

El muy popular Skype no es simplemente un softphone sino un servicio P2P VOIP (punto a punto VOIP).

Algunos softphones están implementados completamente en software, y se comunican con las PBX a través de la Red de área Local - TCP/IP para controlar y marcar a través del teléfono físico.

Generalmente la llamada se hace a través de un entorno de centro de llamadas, para comunicarse desde un directorio de clientes o para recibir llamadas. En estos casos la información del cliente aparece en la pantalla de la computadora cuando el teléfono suena, dando a los agentes del centro de llamadas determinada información sobre quién está llamando y cómo recibirlo y dirigirse a esa persona.

2.6.3 Equipos IP-Phones.

Los teléfonos IP se conectan directamente a la centralita telefónica o a servidores telefónicos de internet.

Existen una gran variedad de fabricantes de estos teléfonos. Los principales son:

- Cisco
- Grandstream
- Polycom
- Snom
- Avaya
- Siemens

Los teléfonos actuales IP-Phones, tienen una funcionalidad ampliada. Tienen las siguientes funciones:

- Soporte de códecs audio
- Soporte de códecs vídeo H.264/H.263/H.263+

- Navegación por internet
- Acceso a radio sobre internet
- Acceso a las redes sociales como Facebook o Twiter
- Acceso a sitios de streaming sobre internet como Youtube.
- Acceso a Yahoo / MSN / Talk

Independientemente de estas opciones, estos teléfonos soportan envío de vídeo, bien directamente o a través de la PBX.



Figura 2.6.1 Vídeo Teléfono Grandstream

Las características de los teléfonos actuales son:

Compatibilidad con estándares abiertos:

- SIP 2.0
- TCP/IP/UDP
- RTP/RTCP
- HTTP/HTTPS
- ARP/RARP
- ICMP
- DNS
- DHCP
- PPPoE
- TFTP
- NTP

Interfaces:

- Puertos 10/100Mbps Ethernet.

- Puertos USB 2.0.
- Puertos 3.5mm (Auriculares estéreo).

Audio de calidad:

- DSP avanzado para audio
- Supresión de silencio
- VAD
- CNG
- AEC
- AGC.

Vídeo de Calidad:

- Real-time vídeo H.263/H.263+
- H.264 códec

Modos avanzados de vídeo:

- TFT color LCD con resolución 480x272

Funcionalidades estándar:

- Identificación del que llama
- Llamada en espera
- Llamada retenida
- Transferencia de llamada
- Opción de no molestar
- Sin sonido (mute)
- Marcado automático
- Navegador Web
- Noticias por RSS
- Actualización del estado de la bolsa
- Pronóstico del tiempo
- Internet Radio
- Reproductor vídeo
- Reproductor de imágenes
- Juegos
- Calculadora
- Alarma de reloj
- IM chat
- Manejador de archivos
- Youtube
- Facebook
- Twitter

Funcionalidades avanzadas:

- Soporte para varias líneas
- Conferencia
- Manos libres
- Intercomunicador
- AES encriptado

Puerto Ethernet:

- Dual conmutado 10M/100M auto-detección puertos Ethernet.

LCD:

- 4.3 digital TFT color LCD

Cámara:

- 1.3M pixel CMOS (VGA).

Puertos auxiliares:

- RCA estéreo puerto de salida
- 3.5 mm puerto estéreo para cascos
- Puerto USB
- Lector de tarjeta SD.

A continuación, se muestran las características generales de un teléfono Grandstream GXV3140:



Figura 2.6.2 Detalle de teléfono Grandstream GXV3140

1	Ranura de tarjeta SD	Para guardar datos o imágenes
2	Puerto USB	Para conexión de : Pendrive USB Teclado USB Ratón USB
3	Conector para cascos	Puerto de conexión 3.5mm para cascos estéreos
4	Conectores RCA	Vídeo/Audio para conexión a TV
5	Conector RJ11	Para conexión de auricular
6	Puerto Ethernet	PC para conexión al PC.
7	Puerto de red Ethernet	Para conexión a Ethernet
8	Alimentación 12V	

Figura 2.6.3 Características de teléfono Grandstream GXV3140.

La pantalla del teléfono se encuentra dividida en 3 partes:

Zona A: Información de tipo general, así como vídeo recibido.

Zona B: Zona de vídeo enviado.

Zona de información general.



Figura 2.6.4 Display de teléfono Grandstream GXV3140

La pantalla normal del teléfono muestra las líneas disponibles. En figura 2.6.5 vemos que el teléfono se encuentra conectado a dos líneas, en una de ellas está el servidor IPVideoTalk, provisto por el fabricante Grandstream. En la parte inferior se puede observar la conexión a la centralita local, donde la dirección IP 192.168.1.24, es la

dirección IP dada por el router o firewall local al teléfono.

A su vez, el teléfono se registra dentro de la PBX (centralita telefónica) con un número de extensión.

Para llamar a este teléfono, solo habrá que marcar la extensión dada por la centralita (PBX), esta última se encarga de direccionar las llamadas.



Figura 2.6.5 Teléfono Grandstream GXV314 conectado a centralita remota y centralita local.

En figura 2.6.6, podemos ver las siguientes opciones del teléfono Grandstream accesibles mediante la opción menú:

- Información general.
- Información de mercados y cambios de moneda.
- Información del tiempo.



Figura 2.6.6 Teléfono Grandstream GXV314 Noticias/Stock/Tiempo.

Las figuras 2.6.7 y 2.6.8 muestran las formas de establecer una llamada telefónica.

Estamos usando el proveedor SIP IPVideoTalk.

Las opciones de marcado se pueden hacer, bien marcando el número deseado directamente a través de la línea seleccionada (en este caso IPVideoTalk), o bien seleccionando un número del historial de llamadas o de la agenda.

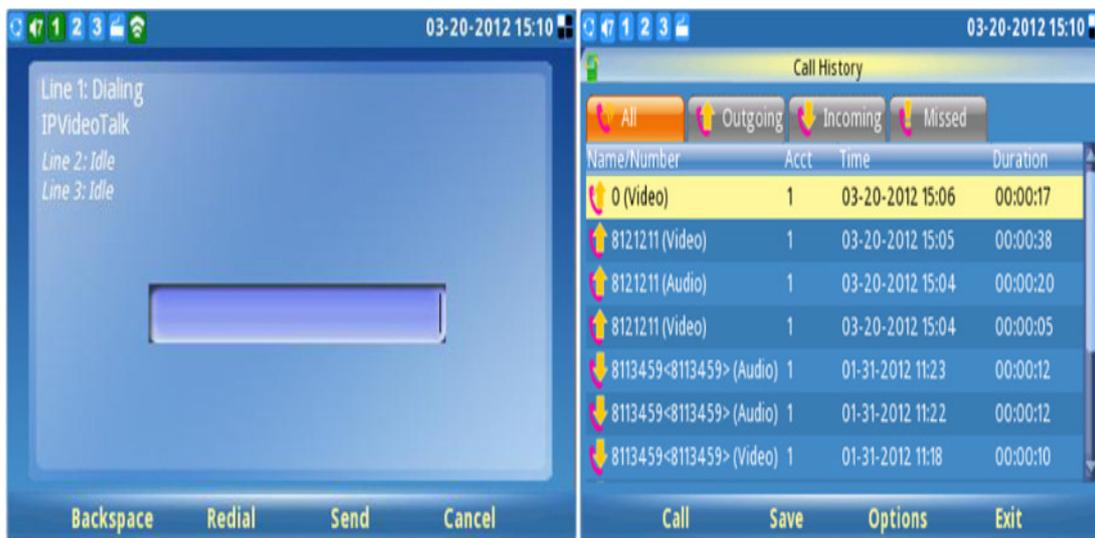


Figura 2.6.7 Teléfono Grandstream GXV314 Marcación y listado de llamadas.



Figura 2.6.8 Teléfono Grandstream GXV314 Directorio, Mensajería.

En la figura 2.6.9 vemos una resolución de llamada: Se observa en la parte inferior de la figura que el teléfono nos da la opción de enviar audio o vídeo.

Sólo aclarar que una vez establecida la llamada al número marcado (en figura el 81212211), la condición última para establecer una llamada correcta es que los teléfonos tengan configurados los mismos códecs de audio y vídeo.

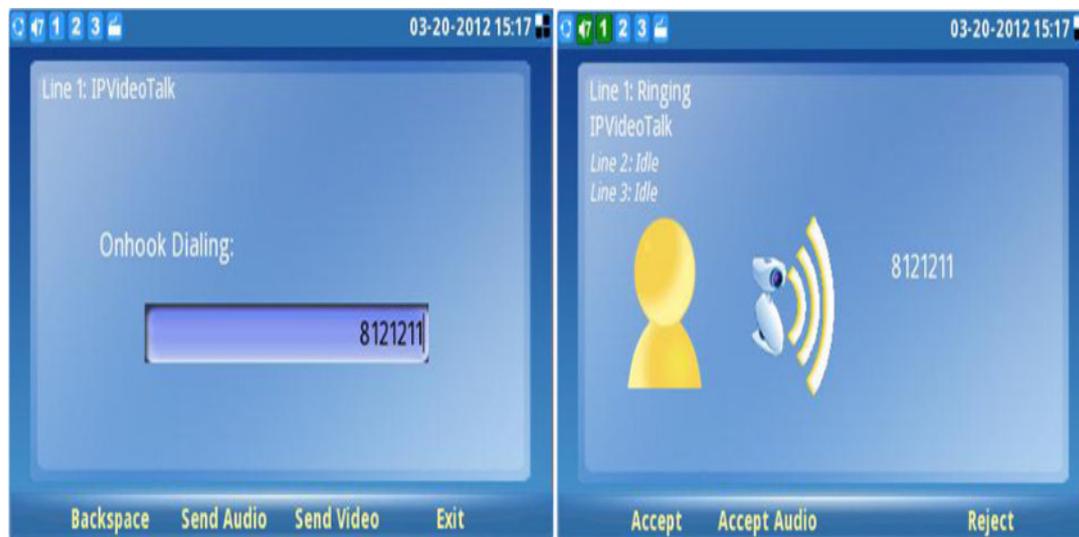


Figura 2.6.9 Teléfono Grandstream GXV314 Marcación y Llamada.

En figura 2.6.10 tenemos el menú de opciones de un teléfono Grandstream, donde habría que destacar las opciones de Social Networks (redes sociales), Webbrowser (Navegador) y Multimedia.

Dentro del entorno de las redes sociales, los teléfonos suelen soportar las aplicaciones normales como Skype, Twitter, Facebook, Google.

Estas últimas opciones aunque inicialmente puedan parecer poco importantes, en el caso de llamadas de larga distancia o duración, pueden ser fundamentales, ya que las llamadas entre estas aplicaciones se producen sobre internet, y obvian tanto el proveedor de telefonía (Vodafone, Orange), como la procedencia y destino de la llamada.

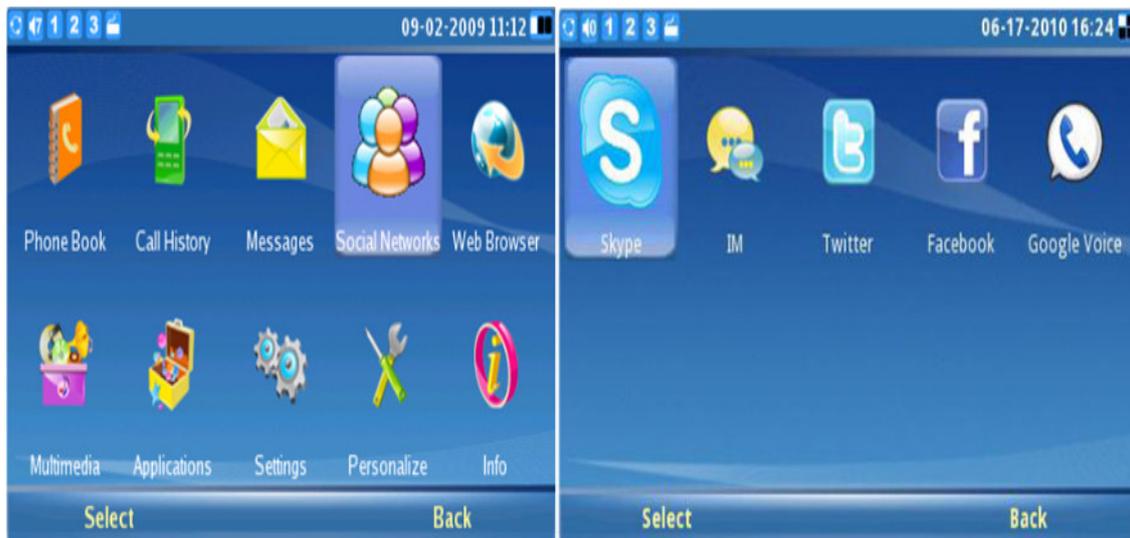


Figura 2.6.10 Teléfono Grandstream GXV314 Opciones y Redes Sociales.

Es de destacar el uso de las aplicaciones P2P, Skype y Google Voice.

Éstas requieren registrarse en los servidores de Skype y Google, siendo necesario obtener una clave de usuario y contraseña.

Las aplicaciones Skype y Google una vez configuradas se inician automáticamente al conectar el teléfono.

Una vez conectados a Skype/Google, se puede, en cada momento y desde estos mismos programas, configurar la opción de estado, permitiendo estar visible en el servidor de Skype/Google, e informando de la disponibilidad para recibir llamadas.

Ver figuras 2.6.11 y 2.6.12.

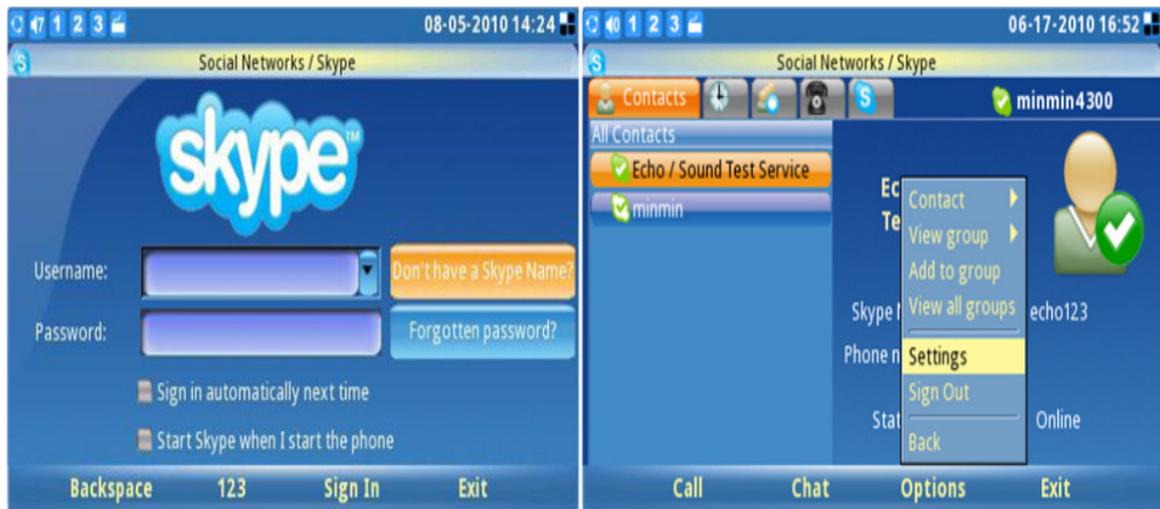


Figura 2.6.11 Teléfono Grandstream GXV314 Skype.



Figura 2.6.12 Teléfono Grandstream GXV314 Skype, llamada.

Igualmente el teléfono provee acceso a Twitter y Facebook, así como la posibilidad de navegación.

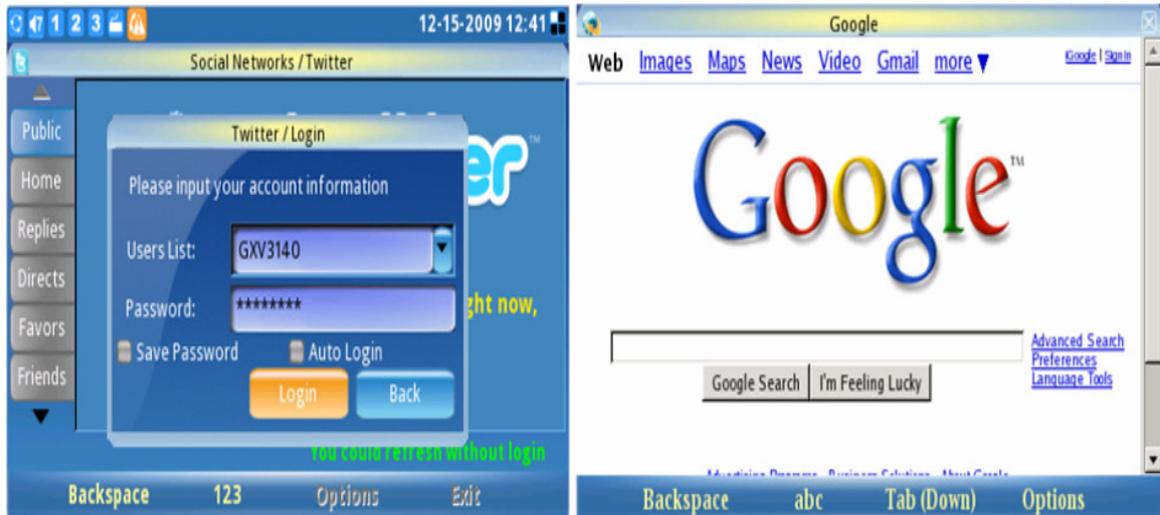


Figura 2.6.13 Teléfono Grandstream GXV314 Twitter y Navegación

En figura 2.6.14 se pueden ver las opciones multimedia de un teléfono Grandstream:

- Radio
- Reproducción de audio
- Reproducción de vídeo



Figura 2.6.14 Teléfono Grandstream GXV314 Radio, Música y Vídeo

Como anotación a la descripción general de los teléfonos Grandstream decir que tienen la posibilidad de marcación directa entre teléfonos SIP, siempre que conozcamos la dirección IP de ellos.

Como punto final a la descripción de teléfonos IP se describe la configuración de un

teléfono IP.

Se verán los aspectos fundamentales de la configuración dentro de una centralita software, en este caso, dentro de la centralita instalada para esta Tesis en el sistema Marconi.

Para acceder al menú de configuración del teléfono ha de disponerse de un PC con navegador, conectado a la misma red donde se encuentra el teléfono.

En la barra de direcciones del navegador, introducimos la dirección IP de nuestro teléfono.

Como anotación de tipo general, aclarar que muchos dispositivos (sean teléfonos, decodificadores u otros), implementan sistemas operativos abiertos, contando con servidor Web, y acceso a través de navegadores.

Una vez que introducimos en el navegador la dirección de nuestro teléfono (como aclaración, indicar que ésta aparece en la pantalla del teléfono) obtendremos una pantalla como la que aparece en figura 2.6.15, donde habremos de introducir el usuario (Username) y la clave (Password). Los valores por defectos de éstos son conocidos y vienen en los manuales de usuario.



Figura 2.6.15 Teléfono Grandstream GXV314 Configuración.

Una vez introducidos el usuario y claves correctos, obtenemos la pantalla principal de estado, donde tenemos las siguientes áreas (ver figura 2.6.16):



Figura 2.6.16 Teléfono Grandstream GXV314. Pantalla de información general

Cabe destacar que un teléfono SIP puede estar conectado a distintos servidores de telefonía, pudiendo por cada línea llamar a través de uno de ellos.

Esta opción permite organizar las llamadas, por usuarios y a su vez por motivos económicos, usando proveedores locales en el caso de llamadas internacionales, y con esto una reducción estimable del coste de las llamadas.

El estado de la red permite obtener información de los parámetros de la red IP.

La información del sistema nos provee de los datos generales del sistema.

La información mostrada en la figura 2.6.16 nos indica que hay 3 cuentas definidas en el teléfono:

Cuenta 1: Con el número 8130456, definido en el servidor sip.ipvideotalk.com, que pertenece a Grandstream.

Cuenta 2: Con el número 900, definido en la centralita Trixbox del sistema de enseñanza Marconi, dirección de la centralita 192.168.1.5 (provista por el

firewall IPCOP de este mismo proyecto), y además vemos que se ha registrado correctamente.

Cuenta 3: Con el número 57195719, definido en el servidor ekiga.net, que pertenece a Ekiga.

Simplemente a modo de ilustración, comentar que las cuentas 1 y 3 no están registradas, según se ve en figura 2.6.16. El motivo de esto es que el proyecto Marconi es una red privada detrás del Firewall IPCOP, y que a su vez este mismo firewall, se encuentra protegido por el firewall de la Universidad de Cádiz.

El firewall de la UCA por motivos de seguridad tiene permitido el acceso sólo a unos pocos puertos, es por esto que las cuentas Ekiga e Ipvideotalk no se pueden registrar

La pantalla de estado de la red (Network Status) nos da la siguiente información (Figura 2.6.17):

- Mac address: Dirección física de la tarjeta de red del teléfono (es única).
- Address Type: Forma de asignación de dirección IP (Fija o DHCP).
- IP address: Dirección de red asignada por el Firewall-Marconi.
- Subnet Mask: Máscara de la dirección IP.
- Default Gateway: La puerta de acceso a internet. En este caso el Firewall Marconi, que tiene la dirección 192.168.1.1
- DNS Server: Servidor de DNS, en este caso el mismo Firewall.
- NAT Type: Tipo de NAT, en este caso indiferente.

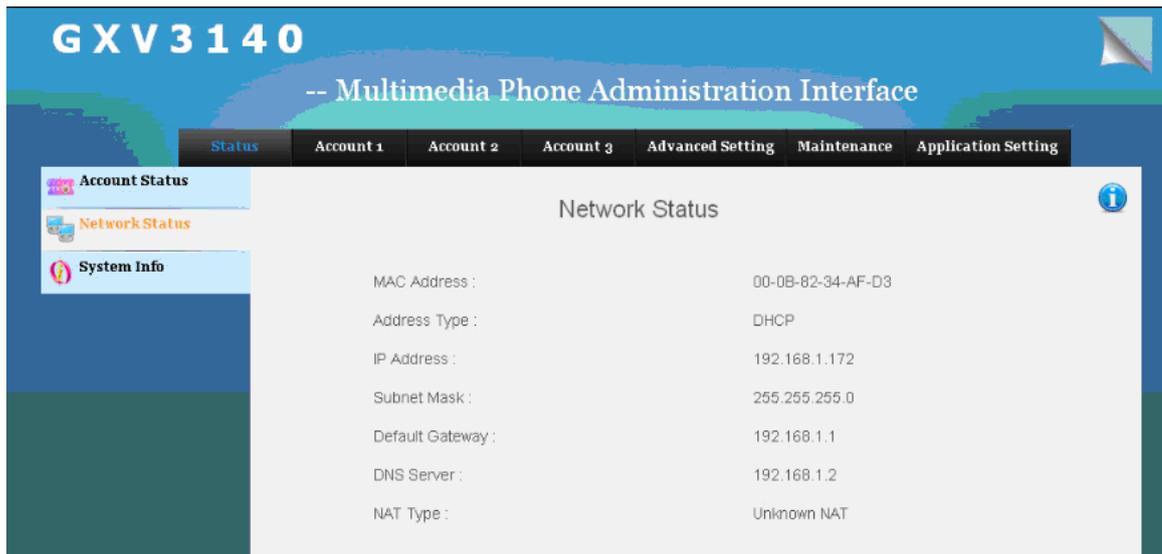


Figura 2.6.17 Teléfono Grandstream GXV314 Pantalla de información de la red.

Seguidamente vamos a detallar la configuración de las cuentas.

La configuración de las cuentas se hace en cinco pestañas:

- Parámetros generales.
- Parámetros de red.
- Parámetros SIP.
- Códecs.
- Parámetros de llamada.

Detalle gráfico de esto se puede observar en figura 2.6.18.

La pestaña de parámetros generales contiene parámetros de activación e identificación de la cuenta. Se ve detalle en figura 2.6.18.



Figura 2.6.18 Teléfono Grandstream GXV314. Configuración de cuenta. Parámetros generales.

Consta de los siguientes campos:

- Account active: Permite activar o desactivar la cuenta.
- Account name: El nombre de la cuenta, que aparece en el teléfono.
- SIP Server: El servidor SIP del teléfono, en el caso de Marconi, es la centralita Trixbox, y corresponde a la dirección IP de ésta.
- SIP user id: Definida dentro de la centralita Trixbox.
- Authenticate id: Definida dentro de la centralita Trixbox.
- Authenticate password: Clave definida en la centralita Trixbox para el user id.

Estos parámetros son los que configuran la conexión al servidor SIP (en este caso centralita Trixbox), y permiten que el teléfono se defina como cliente de ésta, pudiendo llamar por esta línea a todas las extensiones y servicios definidos en la PBX Trixbox (mensajería, llamadas a extensiones, llamadas hacia el exterior por troncales IP o PSTN).

La pestaña de configuración de red, permite definir parámetros generales de conexión

dentro de las redes IP, como son:

- Proxy
- DNS
- NAT

Se pueden ver detalles en figura 2.6.19.

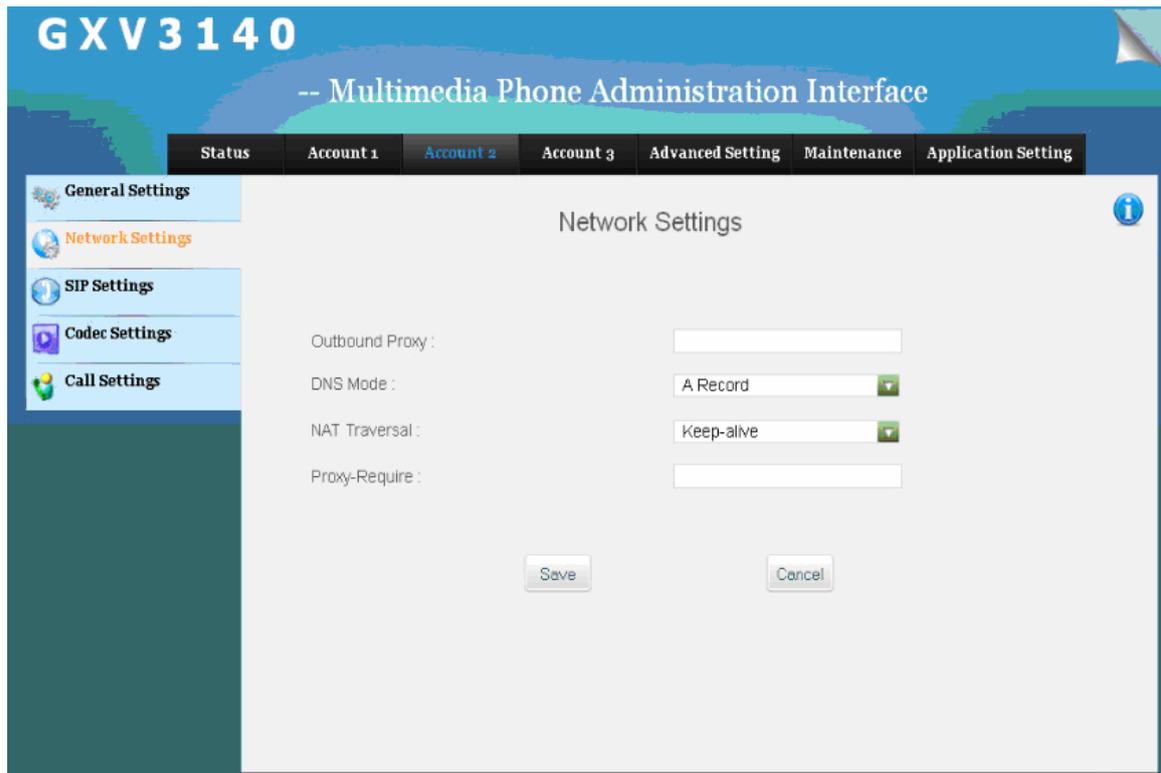


Figura 2.6.19 Teléfono Grandstream GXV314 Configuración de cuenta. Parámetros de red.

La pestaña de configuración SIP configura las características generales del protocolo SIP.

Se puede ver detalle en figura 2.6.20.

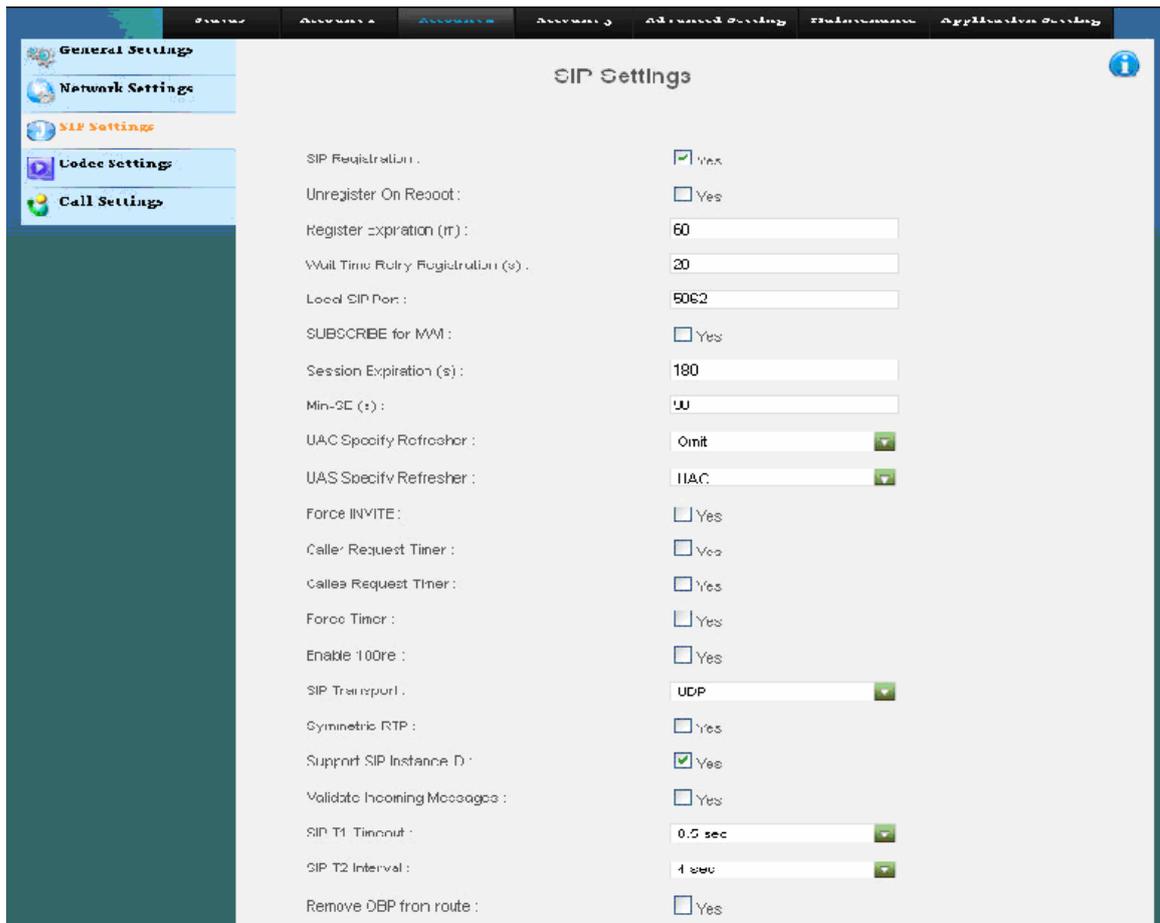


Figura 2.6.20 Teléfono Grandstream GXV314 Configuración de cuenta. Parámetros SIP.

Esta pestaña tiene los siguientes parámetros principales:

- SIP registration: Habilita la conexión o desconexión con el protocolo SIP.
- Local SIP Port: El puerto usado por este protocolo.
- SIP transport: El formato de empaquetado usado (UDP/TCP).

La pestaña de configuración de códecs, es fundamental para lograr que se resuelva adecuadamente la comunicación de datos.

Una vez tenemos establecida la comunicación SIP sobre IP, transmitimos los datos, vídeo o audio. Para lograr un ahorro del ancho de banda se utilizan diferentes formas de empaquetado.

Hemos de tener definidos en el teléfono los códecs necesarios para enviar y recibir el audio y el vídeo. Esto se realiza a través de esta pestaña, de la que vemos detalle en la figura 2.6.21.

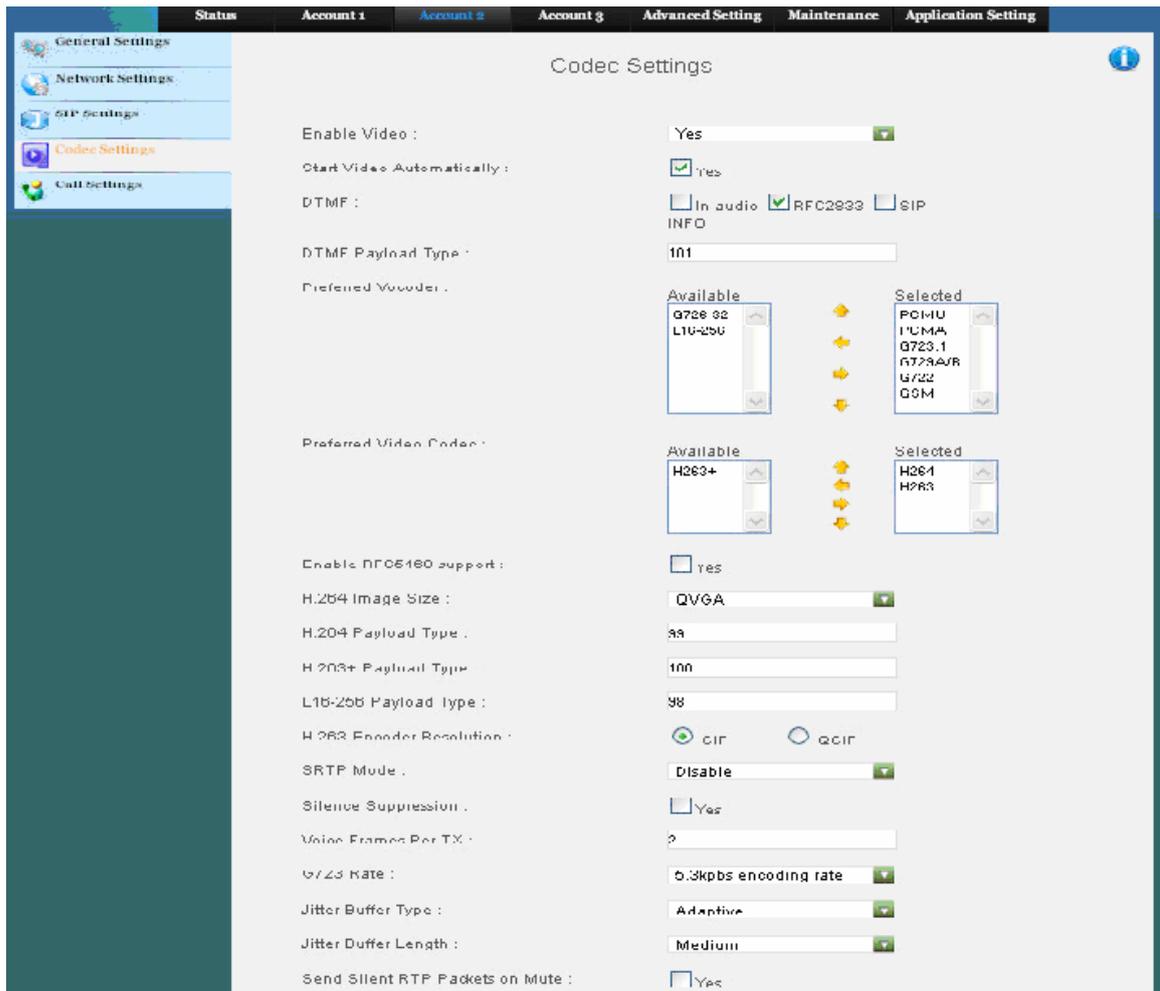


Figura 2.6.21 Teléfono Grandstream GXV314 Configuración de cuenta, CÓDECS.

Dentro de esta pestaña se distinguen los siguientes campos fundamentales:

- Enable Vídeo: Habilita o deshabilita la transmisión de vídeo.
- Preferred Vocoder: Permite seleccionar códecs audio de los disponibles.
- Preferred Vídeo Códec: Permite seleccionar códecs de vídeo de los disponibles.

Se han de tener seleccionados los códecs necesarios para la comunicación, si no, no se podrá transmitir correctamente, bien el sonido o la figura.

En la figura 2.6.21, se encuentran detallados los códecs usados en los videoteléfonos del sistema Marconi, que han resultado ser los requeridos para las cuentas definidas (Ipvideotalk, Ekiga, Skype, Trixbox).

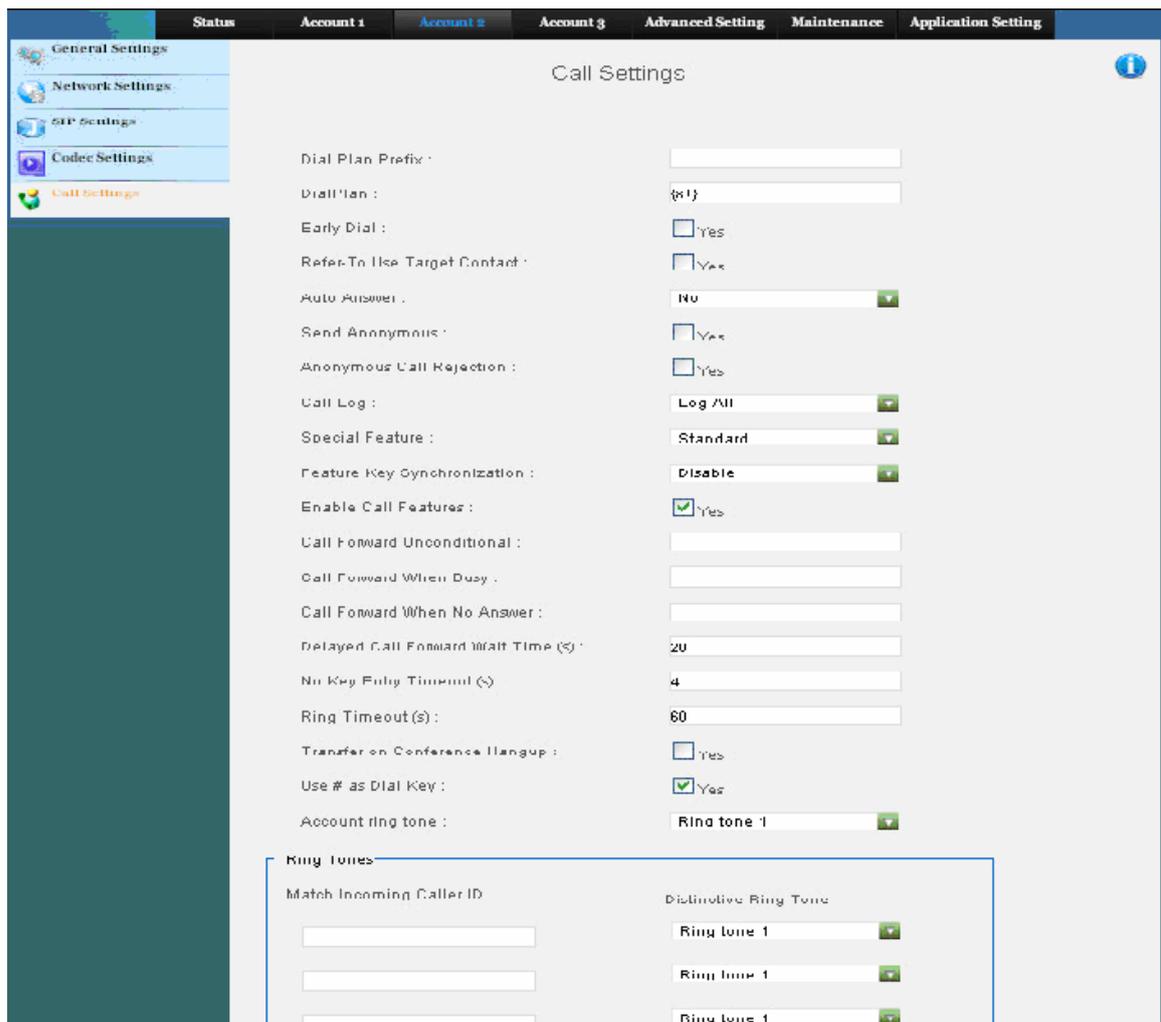


Figura 2.6.22 Teléfono Grandstream GXV314 Configuración de cuenta. Parámetros de llamada.

Como última pestaña de configuración, vamos a definir la pestaña de parámetros de llamada. Esta pestaña es la que nos define cómo interpretar el número marcado, si se usan prefijos, los tiempos de espera después de marcación de prefijo, etc.

El campo fundamental es:

- Dialplan. Éste define el plan de marcado del teléfono. Una descripción más detallada está en el manual de usuarios del teléfono. En el caso del teléfono usado dentro del sistema Marconi se ha colocado la secuencia {x+}, que admite cualquier secuencia de marcado.

Con esto, se ha dado una panorámica general de la configuración llevada a cabo en los videoteléfonos del sistema Marconi.

Para casos puntuales, se recomienda usar el manual de usuario de los videoteléfonos. Seguidamente se describe otra parte fundamental dentro de los sistemas de vídeo y de audio. Éstas son las videocámaras.

Las videocámaras actuales también poseen capacidades de audio y se integran como dispositivos SIP.

2.6.4 Cámaras IP.

Las cámaras IP, se pueden considerar como dispositivos VOIP.

Llevan la opción de transmisión de vídeo, como opción principal, pero su comportamiento es similar a los videoteléfonos IP.

Poseen opciones adicionales como son:

- Detección de humos
- Análisis de imagen / Detección de movimiento.
- Control de dispositivos.
- Autollamada de emergencia.
- Emisión de mensajes.
- Multistreaming.
- Vigilancia IP desde móviles.



Figura 2.6.23 Opciones de videocámaras.

Ilustración cedida por Avanzada7.

Las cámaras de vídeo implementan funciones de vídeo vigilancia y de telefonía sobre IP.

Dentro de las funciones de telefonía IP distinguimos:

- Protocolo SIP
- Llamadas de voz y vídeo IP.
- Códecs de audio y vídeo.
- Entrada y salida de micrófono.

- Auto marcado con alarma.



Figura 2.6.24 Arquitectura Vídeo IP-SIP (1).

Ilustración cedida por Avanzada7.

Se puede observar en la figura 2.6.24 la topología de red implementada con las videocámaras.

Como base está la red IP, bien intranet o internet, y sobre ellas tenemos los dispositivos SIP.

La resolución de llamadas al igual que ocurre con los teléfonos IP, puede ser directa o a través de servidor SIP (PBX).

Desde un teléfono IP bien sea software o teléfono electrónico, se puede establecer una llamada a una videocámara, pudiéndose hablar a través de ella o recibir vídeo.

El direccionamiento de la llamada la realiza la centralita telefónica, en el caso del sistema Marconi, es la centralita Trixbox.

Esta resolución de llamadas también se puede hacer desde teléfonos tradicionales dentro de la red PSTN, siempre y cuando la centralita telefónica disponga de tarjetas adecuadas, tipo X100P, que hacen de puente.

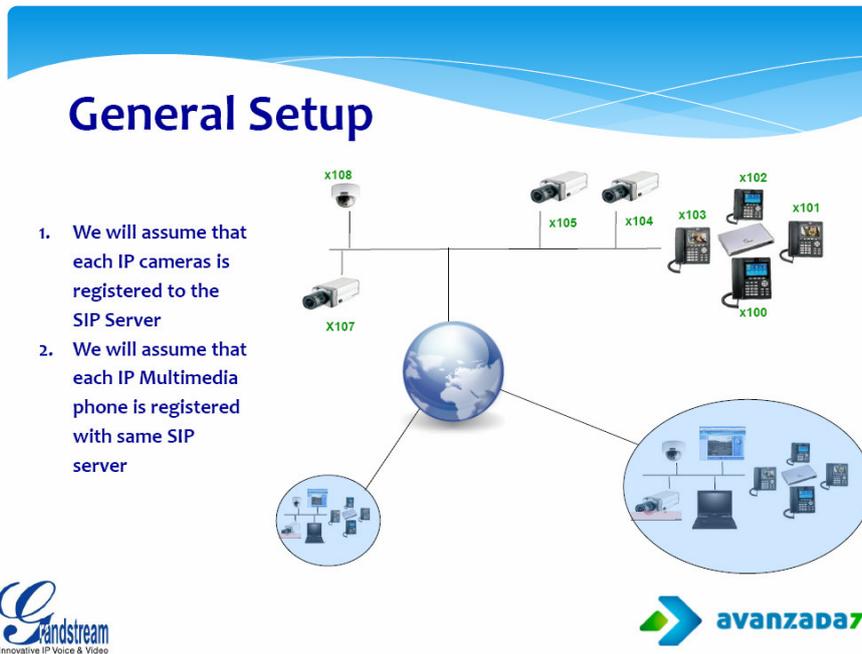


Figura 2.6.25 Arquitectura Vídeo IP-SIP (2).

Ilustración cedida por Avanzada7.

Como condición inicial cada cámara y teléfono se deben conectar al mismo servidor SIP (centralita PBX). De esta forma podemos interconectar con todos, ver figura 2.6.25.

Una vez interconectadas al mismo servidor podemos redireccionar las llamadas y las condiciones de alerta.

De esta forma podremos llamar al teléfono deseado una vez detectemos una alarma de humos o bien movimientos en la zona donde se encuentra el teléfono.

En las figuras siguientes, 2.6.26, 2.6.27, 2.6.28 podemos ver las pantallas de configuración de estas opciones.

Senario 1: Detección de Humo

(solo con el GXV3611HD/LL)

DI Alarm Time Schedule

Smoke Alarm Action

- Record Video From Pre Alarm 0 seconds to After Alarm 0 seconds
- Voice Alarm to SIP Phone
- Alarm Output
- Upload to Alarm Center
- Upload to Alarm HTTP Server
- Email and FTP upload JPEG

Save

Smoke Alarm Time Schedule

Date	Start Time	End Time	
Everyday	00:00	23:59	Add
Everyday	00:00	23:59	<input type="checkbox"/>
			<input type="checkbox"/> Check All Delete

Set the time when Smoke alarm will be enabled. In this example, it's enabled 24/7.

To initiate a SIP call to configured extension when Smoke is detected

avanzada7

Figura 2.6.26 Configuración de teléfono Grandstream. Detección de humos.

Ilustración cedida por Avanzada7.

Senario 2: Análisis de Imagen

Step1: Enable Motion detection and setup regions

Motion Detection Region Settings

Enable Motion Detection

Show Motion Detection Regions

Select a Region

Sensitivity Gain

Edit Remove Save

Step2: Enable Motion detection and setup regions

Alarm Action

- Record Video From Pre Alarm 0 seconds to After Alarm 1 seconds
- Voice Alarm to SIP Phone
- Upload to Alarm Center
- Upload to Alarm HTTP Server
- Email and FTP upload JPEG

Save

Note: When Upload to Alarm Center is checked, the Motion Detection region and icon (in Home page) will blink in red if your camera detects any motion.

Step3: Setup Motion detection Time Schedule

Motion Detection Time Schedule

Region ID	Date	Start Time	End Time	
0	Everyday	00:00	23:59	Add



Figura 2.6.27 Configuración de teléfono Grandstream. Análisis de figura.

Ilustración cedida por Avanzada7.



Figura 2.6.28 Configuración de teléfono Grandstream. Control de cerradura eléctrica. Ilustración cedida por Avanzada7.

Seguidamente vamos a describir las videocámaras usadas en el sistema Marconi.

En la figura 2.6.29 podemos ver el modelo, se distingue:

- Conector de alimentación, hay modelos que toman la alimentación del conector de red.
- Conector de red, las cámaras wifi se pueden conectar por este método, así como por cable de red RJ45, siendo necesario éste antes de configurar la wifi.
- En la parte frontal se encuentran el micrófono y el foco.

Cámara tipo Cube GXV3615WP_HD



Figura 2.6.29 Videocámara Wifi del sistema Marconi.

Ilustración cedida por Avanzada7.

Una vez conectada la cámara al firewall, éste le asigna una dirección IP: vamos al firewall o al router si éste es el caso y vemos la dirección que ha recibido la cámara.

Conocida esta dirección, ya podemos empezar a administrar la cámara.

Para ello abrimos nuestro navegador y en la barra de direcciones le escribimos la dirección IP de nuestra cámara.

Nos aparecerá en el navegador una pantalla similar a la de la figura 2.6.30.

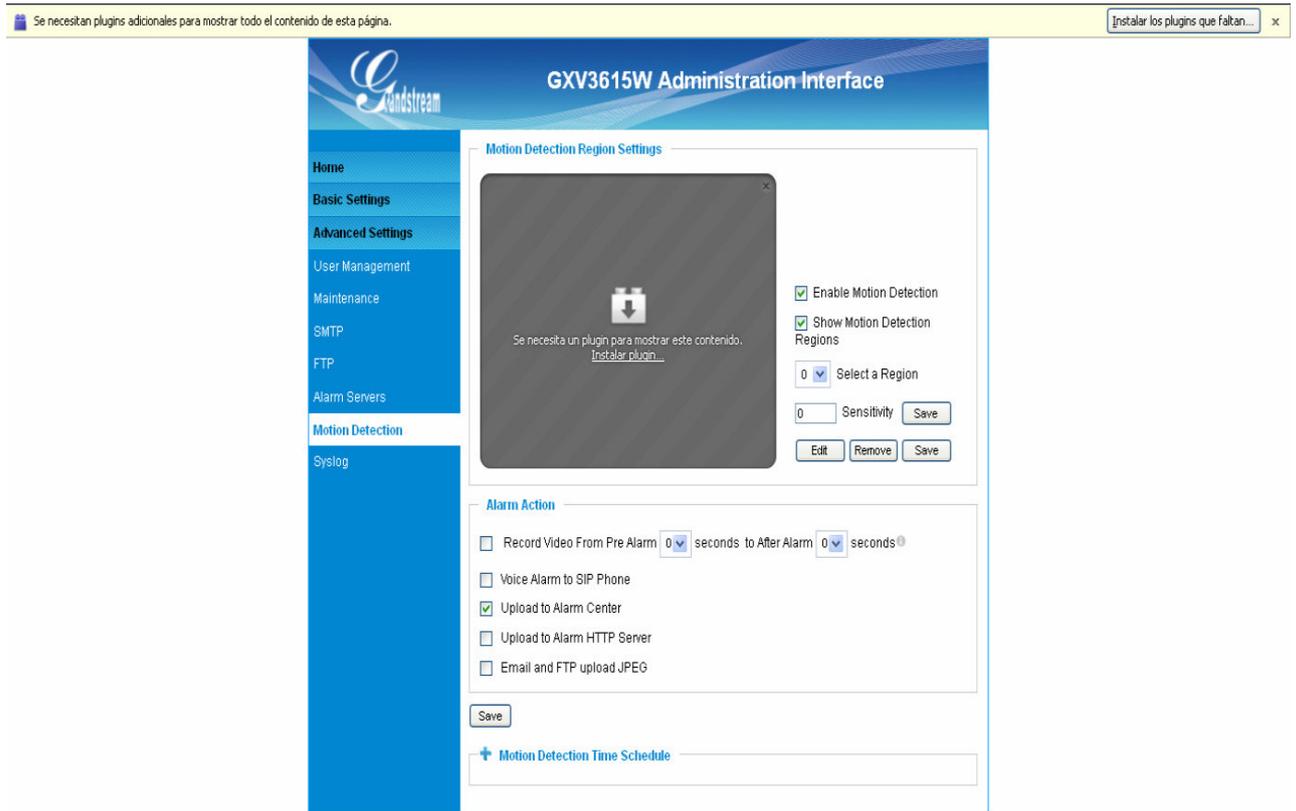


Figura 2.6.30 Cámara GXV3615W. Interfaz de administración.

En esta pantalla tenemos configuraciones básicas y avanzadas.

Dentro de las configuraciones básicas tenemos:

- System: Información de sistema.
- Vídeo Audio: Configuraciones de vídeo y audio.
- Networking: Configuraciones de red.
- Wifi: Configuración wifi.
- DDNS: Configuración DNS.
- SIP: Configuración SIP.
- Status: Información de estado.

Dentro de las configuraciones avanzadas tenemos:

- User management: Manejo de usuarios.
- Maintenance: Acciones generales de mantenimiento.
- SMTP: Configuraciones SMTP.
- FTP: Configuraciones FTP.
- Alarms Servers: Configuración de las alarmas.
- Motion detection: Configuración de la detección del movimiento.
- Syslog: Log del sistema.

Seguidamente se hará una descripción de las configuraciones básicas de las cámaras del sistema Marconi.

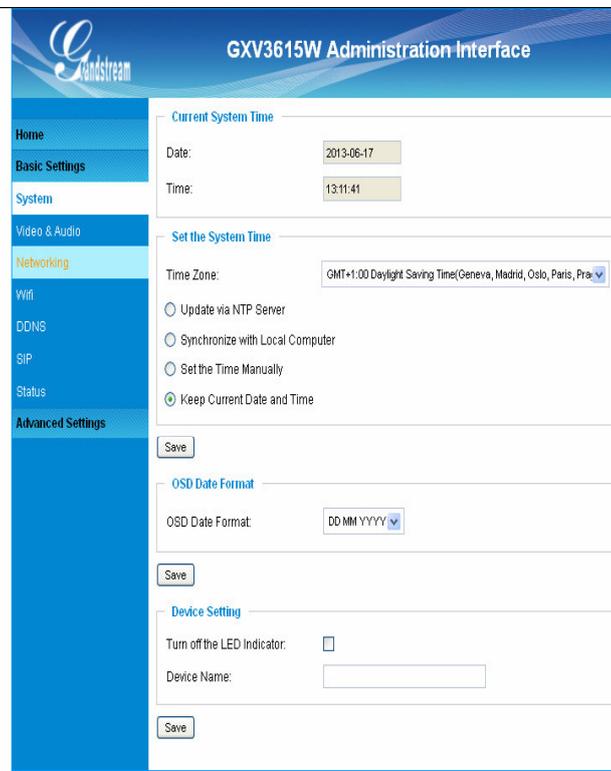


Figura 2.6.31 Cámara GXV3615W. Configuración básica: system.

Dentro de la pestaña system, se incluyen las opciones básicas:

- Fecha y hora.
- Definición de zona horaria.
- Formato de fecha.
- Estado de led indicador.

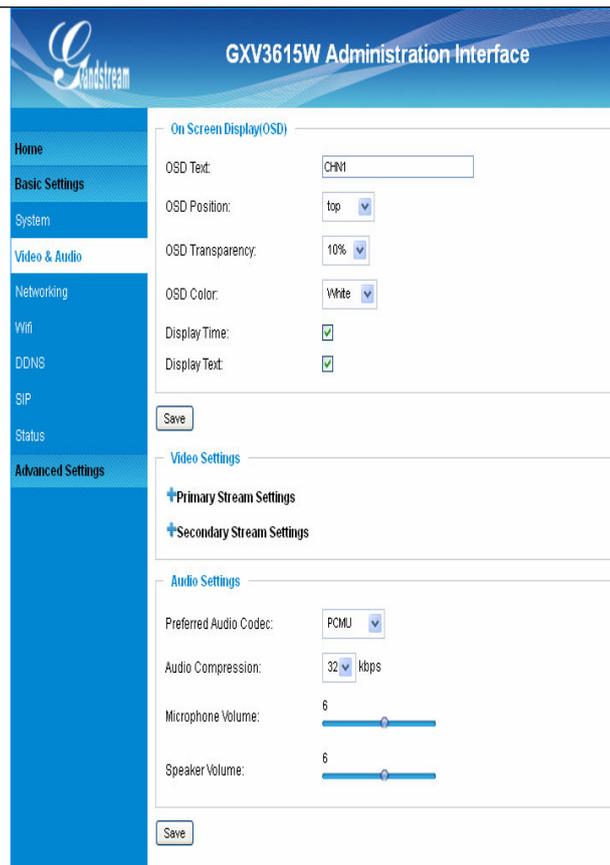


Figura 2.6.32. Cámara GXV3615W. Configuración básica vídeo y audio.

Desde la pestaña vídeo y audio tenemos las opciones siguientes:

- Configuración del texto sobre la pantalla.
- Configuración del vídeo.
- Configuración del audio;
- Códec
- Compresión
- Volumen del micrófono.
- Volumen del altavoz.

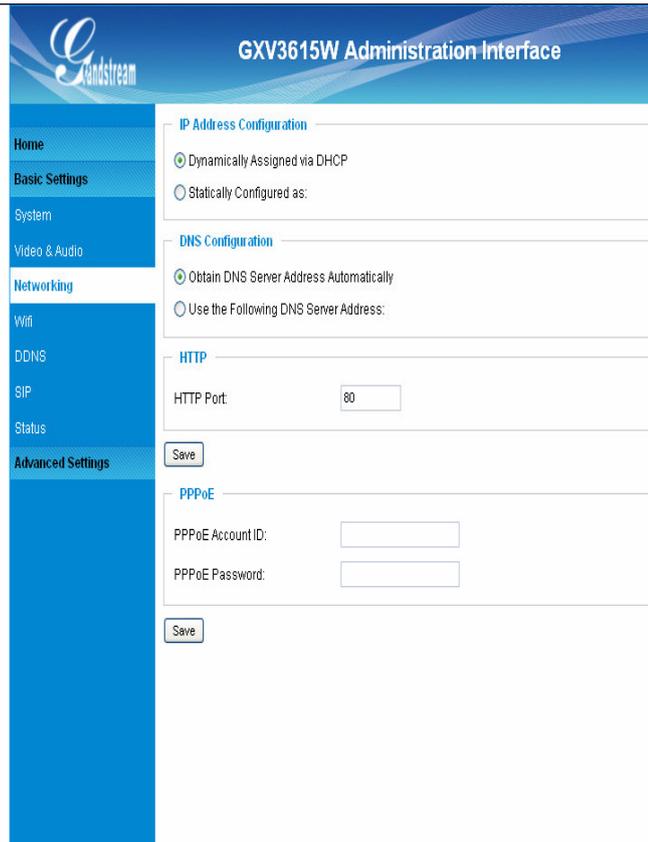


Figura 2.6.33 Cámara GXV3615W. Configuración básica de red.

Desde la pestaña de red tenemos las opciones siguientes:

- Modo de asignación de dirección de red.
- Modo de selección del servidor DNS.
- Puerto asignado a HTTP.
- Configuración del protocolo PPP.

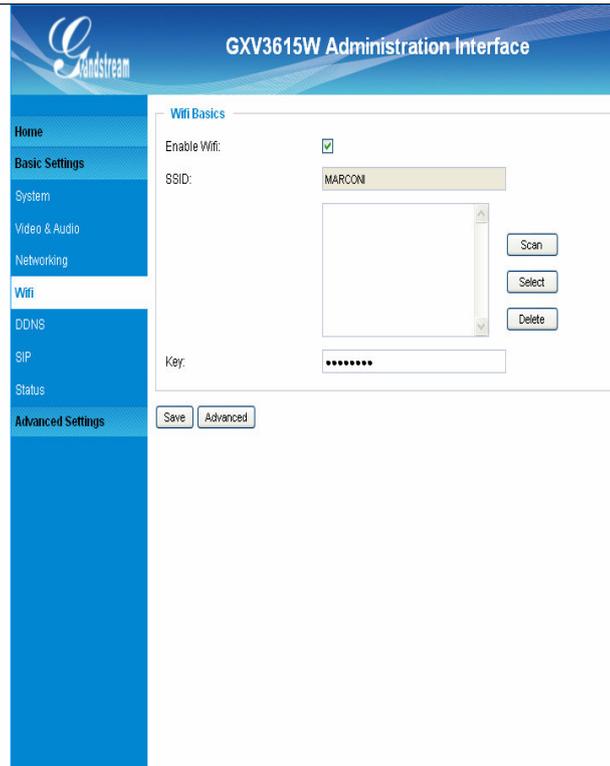


Figura 2.6.34 Cámara GXV3615W. Configuración básica wifi.

Desde la pestaña de red wifi tenemos las opciones siguientes:

- Configuraciones básicas de wifi.
- Habilitar / Deshabilitar wifi.
- SSID, red wifi seleccionada.
- Opciones para buscar las redes existentes, seleccionarlas y borrarlas.
- Clave de acceso a la red seleccionada.

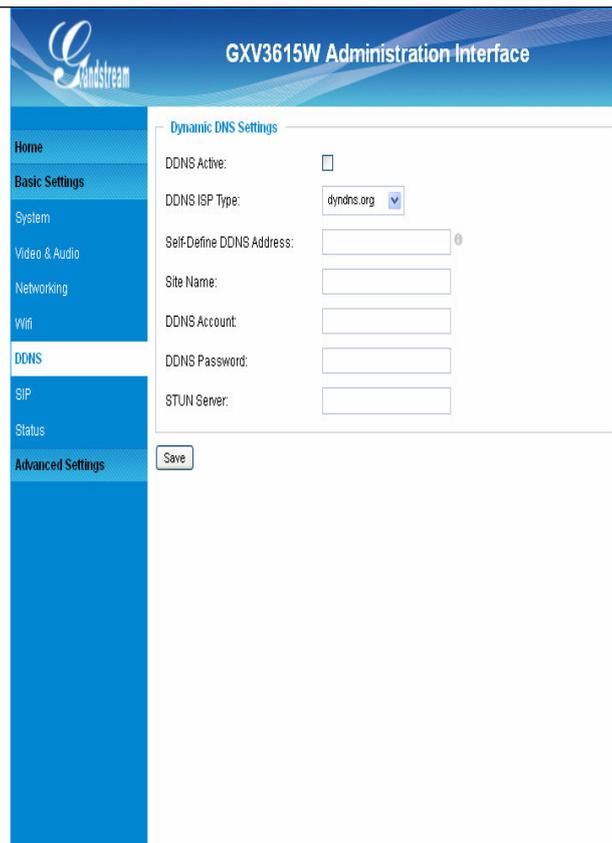


Figura 2.6.35 Cámara GXV3615W. Configuración básica DDNS.

Desde la pestaña de DDN tenemos las opciones siguientes:

- Activación del Dynamyc Domain Server.
- Proveedor DDNS.
- Dirección DDNS.
- Nombre del sitio.
- Cuenta en el servidor DDNS.
- Clave de acceso al servidor DDNS.
- Servidor STUN.



Figura 2.6.36 Cámara GXV3615W. Configuración básica SIP.

Desde la pestaña SIP tenemos las opciones siguientes:

- Habilitar / Deshabilitar SIP.
- Cuenta de usuario en el servidor SIP.
- Dirección del servidor SIP.
- User ID SIP.
- Códecs de voz.
- Puertos SIP y RTP.

The screenshot displays the GXV3615W Administration Interface. On the left is a navigation menu with options: Home, Basic Settings, System, Video & Audio, Networking, Wifi, DDNS, SIP, Status, and Advanced Settings. The main content area is divided into three sections:

- System Statistics:**

Product Model:	GXV3615W
Hardware Version:	V0.2A
Part Number:	9670001902A
Bootloader Version:	1.0.0.0
Core Version:	1.0.4.2
Base Version:	1.0.4.27
Firmware Version:	1.0.4.27
System Up Time Since:	3 days 23 hours 44 minutes
- Network Status:**

MAC Address:	00:0B:82:34:B8:82
LAN IP Address:	192.168.1.251
LAN Subnet Mask:	255.255.255.0
LAN Default Gateway:	192.168.1.1
PPPoE IP Address:	0.0.0.0
PPPoE Status:	Disconnected
DDNS Status:	Disabled
Wifi Status:	Connected
- Camera Type:**

Camera Type:	Aptina, MT9V136 Pixels 680*512
--------------	--------------------------------

Figura 2.6.37 Cámara GXV3615W. Información de estado.

En la pestaña de información de estado tenemos la siguiente información:

- Información del producto, software, firmware, modelo.
- Estado de la red:
- Dirección Mac.
- Información Lan.
- Información PPP.
- Estado de la conexión wifi.
- Tipo de cámara.

2.6.5 Descripción de Teamspeak.

TeamSpeak es un programa de charla por voz sobre IP (Chat), permite a los usuarios hablar en un canal de chat con otros usuarios, tal como se hace en una multiconferencia vía llamada telefónica tradicional.

Los usuarios deben usar un software cliente de TeamSpeak, para conectarse a un servidor TeamSpeak y una vez conectados establecen el canal, a partir de este momento, los usuarios ya pueden hablar.

Los principales usuarios de TeamSpeak son jugadores de videojuegos en red, quienes usan diferentes programas para comunicarse entre jugadores o equipos en modalidad multi-jugador.

La comunicación mediante voz en estos juegos les da una gran ventaja competitiva, dado que permiten a los jugadores una experiencia superior en lo que a diversión se refiere.

Servidor:

Para conectarse a Teamspeak es necesario tener una dirección del servidor Teamspeak, que se puede comprar a empresas que las brindan. Estas empresas cuentan con sistemas seguros de conexión permanente, sistemas de defensa y soporte especializado en el sistema Teamspeak.

El servidor de Teamspeak actualmente funciona en Windows, Mac OS X y Linux.

Los tipos de usuarios del servidor Teamspeak se dividen en seis grupos

- Administrador del servidor.
- Administrador del canal.
- Operador.
- Usuario con Voz.

- Registrado.
- Anónimo.

A cada tipo de usuario le corresponde un nivel de acceso determinado denominado permiso.

Se pueden definir diferentes privilegios para cada uno de estos grupos. Además, en Teamspeak3 se pueden crear grupos específicos personalizando los privilegios.

Extensiones:

En la página Web oficial de Teamspeak, están disponibles para ser descargadas las extensiones desarrolladas por terceros.

Las características más comunes de estas extensiones se orientan a poder administrar o visualizar TeamSpeak desde dentro de algún juego.

Otros programas similares a Teamspeak son :

- Mumble
- Xfire

A continuación se describe de forma resumida los pasos para instalar en un servidor propio Teamspeak (en este caso Teamspeak 3, última versión de este programa).

En Apéndices se encuentra la descripción detallada de toda la instalación.

Como resumen de la instalación, los pasos son los siguientes:

- Crear usuario ts3srv
- Descargar de la página de Teamspeak el tar del software (Ver figuras 2.6.38 y 2.6.39)
- Descomprimir
- Seguir procedimiento de instalación
- Es importante, anotar claves y token dados durante la instalación, pues habrá que utilizarlos posteriormente.
- Ejecutar el servidor y chequear que éste funciona adecuadamente.
- Instalar cliente y ejecutarlo.
- Ejecutar cliente contra servidor local (pedirá claves y token dados durante la instalación del servidor).
- Conectarse a servidor local con cliente y hacerse administrador.

En figuras 2.6.38 y 2.6.39 vemos los sitios de descarga de los programas usados.



Figura 2.6.38 Página principal de Teamspeak <http://www.teamspeak.com/>.

En figura 2.6.38 vemos la página principal (home) de la Web Teamspeak. Destacamos las siguientes áreas:

- Forums: Área donde se debate sobre cada uno de los productos y aplicaciones, así como se resuelven problemas que puedan surgir durante la instalación, configuración o programación del programa Teamspeak.
- Downloads: Área donde se pueden encontrar los programas para su descarga.

Es de destacar dentro de esta Web, las áreas donde existen plug-ins de mejora de la aplicación.

En Figura 2.6.39, se ve detalle de la página downloads, donde se pueden obtener los programas para cada una de las plataformas donde se puede ejecutar Teamspeak.

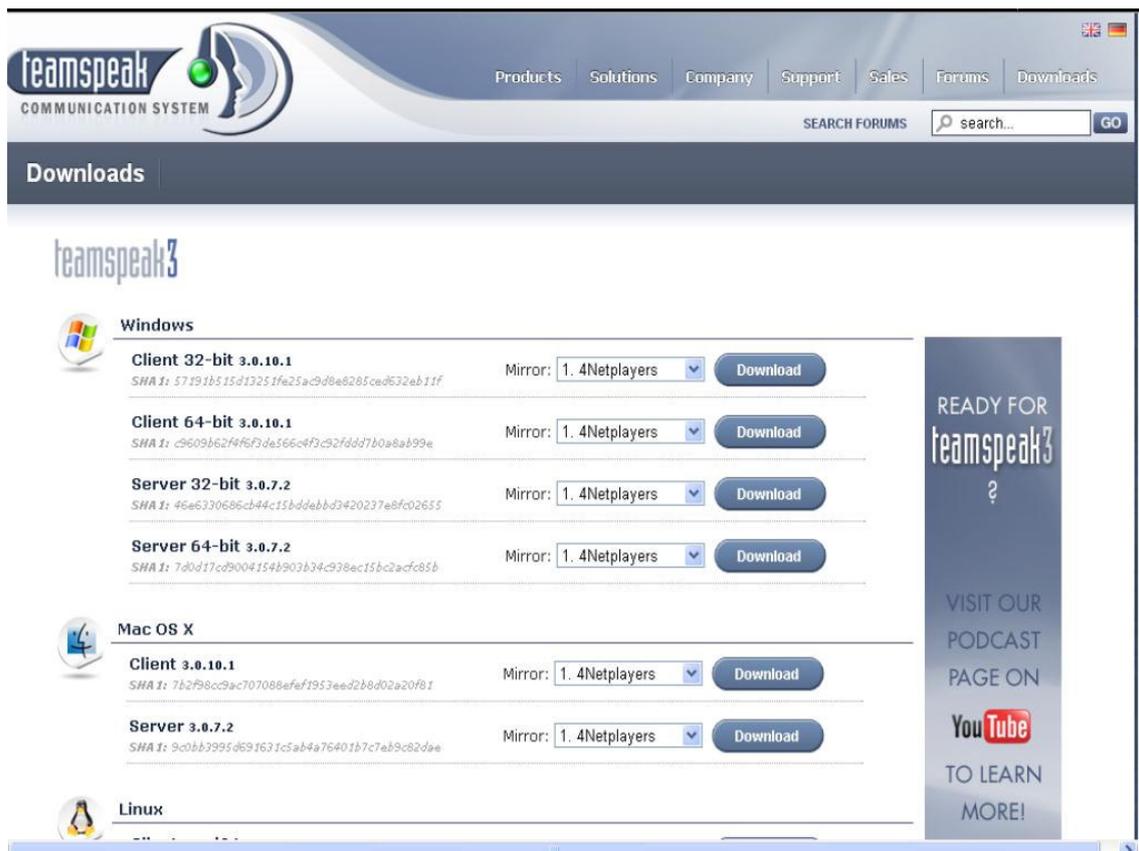


Figura 2.6.39 Página de descarga de Teamspeak.

A continuación se da una pequeña guía de programación de la aplicación, haciendo referencia a cómo se ha hecho en el caso del servidor TeamSpeak Marconi del grupo S2CN de la UCA.

2.6.6 Programación de Teamspeak3 MARCONI.

El servidor Marconi se ejecuta en un equipo servidor localizado en la Universidad de Cádiz. El objetivo de su implementación es divulgar las nuevas tecnologías de las radiocomunicaciones (ROIP) y su aplicación a casos concretos.

El servidor Teamspeak Marconi se encuentra localizado en la dirección `marconi.uca.es`.

Su acceso se realiza a través de un sistema Firewall Linux (IPCOP) que provee los siguientes servicios:

- Proxy.
- Cache.
- VPN.
- IPSec.
- NAT.
- Antivirus/firewall.

Este sistema ha sido implementado dentro de esta Tesis Doctoral, para proveer acceso NAT a los distintos servidores de la red Marconi.

El acceso al servidor Marconi se consigue instalando Teamspeak cliente, y definiendo como servidor la dirección `marconi.uca.es`.

Las claves iniciales de acceso son proporcionadas por el administrador del servidor. A cada usuario, en función de sus características, se le da una serie de privilegios o posibilidades de accesos a canales así como de pertenencia a los grupos definidos dentro de la aplicación Teamspeak.

En la Figura 2.6.40 se observa la pantalla de Teamspeak observada desde un usuario. Cada línea corresponde a un canal, y dentro de él se encuentran los usuarios. Todos los usuarios que están dentro de un canal pueden hablar o escucharse entre ellos, dependiendo del atributo asignado a cada usuario.

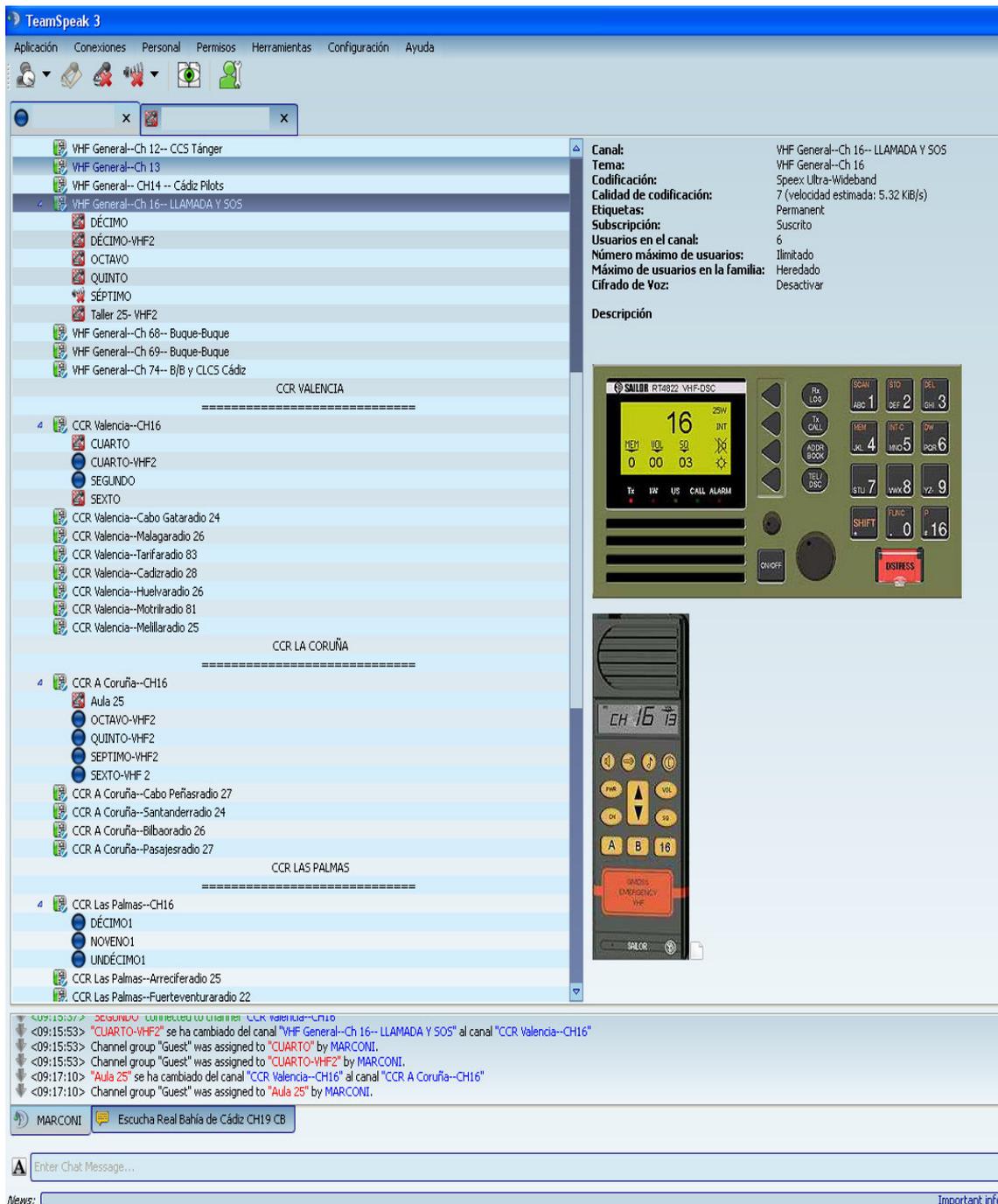


Figura 2.6.40 Servidor Teamspeak Marconi.

A continuación, se definen nociones de programación para realizar la programación y administrador del servidor Teamspeak.

Cada usuario está dentro de un grupo, y en éste tiene unos privilegios.

A su vez, cada usuario puede estar dentro de alguno o algunos de los grupos de cada

canal.

Los grupos de permisos para acceso al sistema (grupos del servidor) son:

Al programar la aplicación en TS3 tenemos que tener en cuenta que existen los llamados grupos del servidor y grupos del canal.

Los grupos del servidor son grupos con ciertos privilegios a los que se les van añadiendo usuarios.

Los grupos del canal corresponden a unos tipos de canal con unos privilegios determinados.

Los grupos del servidor predefinidos por el programa son:

- Server admin: Tienen derecho a todo
- Normal: Tienen algunos privilegios
- Guest: Es un invitado, no tiene privilegios



Figura 2.6.41 Servidor Teamspeak Marconi. Grupos del servidor.

En la aplicación Marconi se crean los Grupos Alumnos y Profesores copiando el grupo normal, y dándole más privilegios en el caso de Profesores y menos en el caso

de Alumnos.

Para ver los privilegios hacer lo siguiente (Figura 2.6.42):

- En barra principal de Teamspeak, picar en Permisos.
- Seleccionar en el desplegable Grupos de servidor.
- Situarse sobre el grupo a modificar y aparecerán las opciones.

Para ver en detalle los permisos, pulsar el botón con la flecha que se ve en el detalle de la figura 2.6.42.



Figura 2.6.42 Servidor Teamspeak Marconi. Permisos asociados a grupos.

Una vez presionada la flecha anterior, aparece un menú como el de la figura 2.6.42.

Se observan los siguientes detalles importantes:

Canal- Acceso- Nivel para entrar al canal, el valor 50.

Ése es el valor para poder entrar a los canales que tienen los usuarios de este grupo. Es muy importante tener en cuenta que al cambiar un valor y guardar, el programa pregunta si se quieren guardar los otros valores que estaban definidos. Decir siempre sí.

Si se quiere cambiar los valores de acceso que se han dado en un canal, se han de hacer en el canal, no en los accesos que se han dado antes al grupo.

Cambio de valores en un canal:

Para cambiar de valor en un canal hay que seguir los siguientes pasos: En la pantalla principal de Teamspeak, pulsar sobre el canal deseado, pulsar el botón derecho del ratón. En el menú desplegable aparecerán las opciones, elegir Editar.

En la solapa Estándar aparece (ver figura 2.6.43):

Nivel requerido para hablar, si es mayor de cero, el canal será moderado. Si un usuario tiene privilegio para entrar por su nivel de acceso, pero no para hablar, tendrá que obtener el permiso del administrador de canal.

También se puede ver en el menú que hay contraseña para entrar al canal, y que el canal es permanente (aunque quede vacío no desaparece. Siempre está).

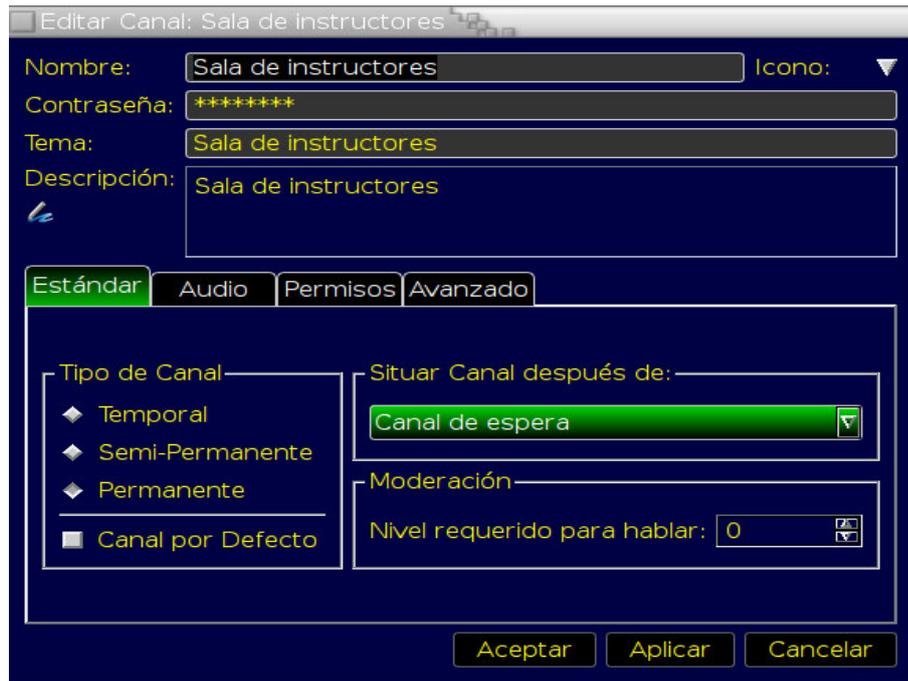


Figura 2.6.43 Servidor Teamspeak Marconi. Programar canal, solapa Estándar. En la solapa Audio, que se ve a continuación en la Figura 2.6.44, se puede cambiar el ancho de banda del canal. Para ello se desplaza el botón tal como se ve en la figura 2.6.44.

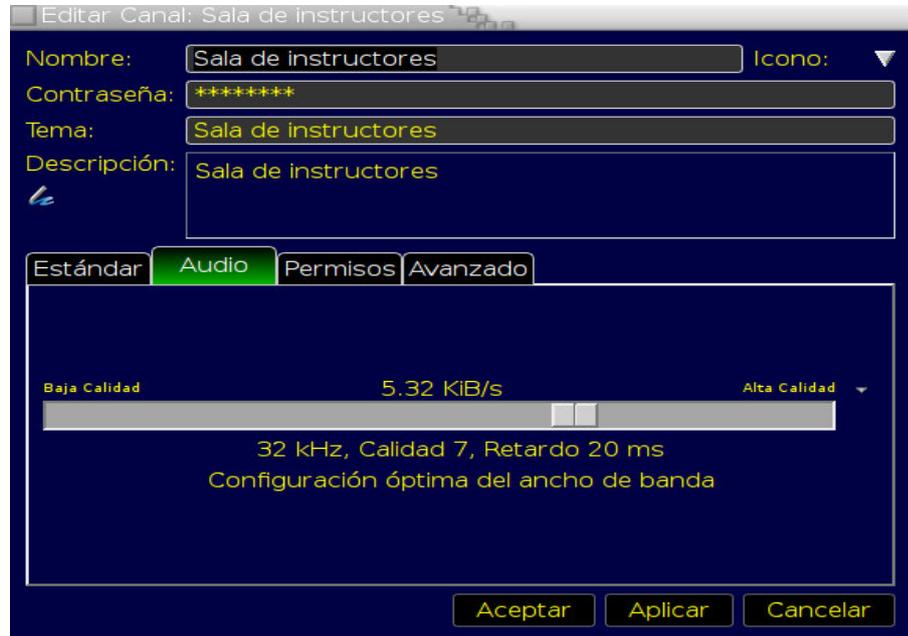


Figura 2.6.44 Servidor Teamspeak Marconi. Programar canal, solapa Audio.

En la solapa permisos, ver figura 2.6.45, se ven los niveles de accesos que hay que tener para cada acción dentro del canal.

El más importante es el nivel de acceso necesario (éste se ha dado anteriormente en los grupos de usuarios).

Para entrar en este canal se debe tener un nivel de acceso igual o superior al definido aquí.



Figura 2.6.45 Servidor Teamspeak Marconi. Programar canal, Solapa Permisos.

Vamos a definir qué son los grupos del canal.

En un grupo de canal se definen los privilegios que ese grupo tiene para manejar un canal:

Vamos a la barra principal, seleccionamos Permisos, y en el menú desplegable, seleccionamos Grupos de canal.

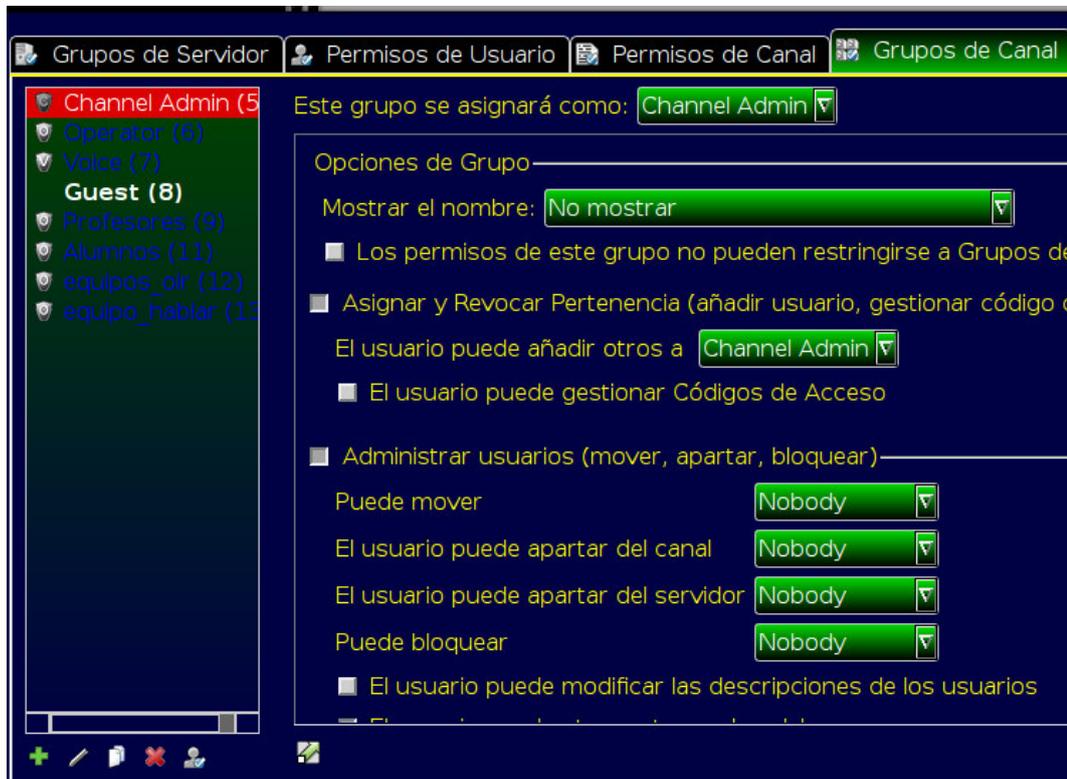


Figura 2.6.46 Servidor Teamspeak Marconi. Grupos definidos en los canales.

Los grupos de canal iniciales dentro del programa Teamspeak son:

- Channel admin: Tiene privilegios de administrador, es para los administradores.
- Operator: Es el usuario administrador del canal.
- Voice: Usuario con permisos básicos para hablar.
- Guest: Cualquier invitado.

En la aplicación Marconi, partiendo del grupo operator, se han creado los grupos:

- Profesores
- Alumnos
- equipos_oir
- equipo_hablar

Vamos a ver qué privilegios tiene cada grupo

Los profesores tienen más privilegios que los alumnos.

A los equipos_hablar y equipos_oir se les han dado los privilegios para poder acceder

al equipo y sólo oír o bien hablar y oír.

Es importante destacar que en los grupos de canal, es donde se da el nivel para hablar al grupo.

Para programar los privilegios necesarios dentro de los grupos del canal, accedemos dentro de la solapa permisos, a la pestaña Grupos de Canal, ver figura 2.6.47.

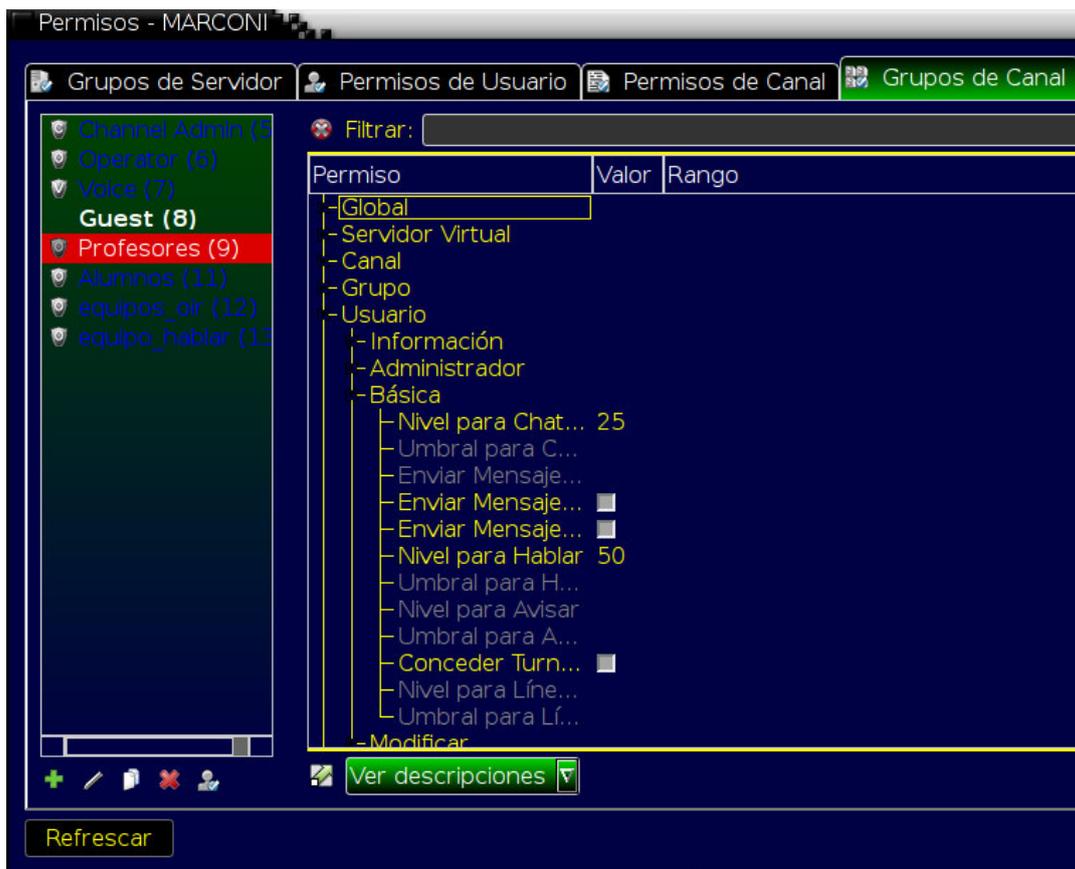


Figura 2.6.47 Servidor Teamspeak Marconi. Grupos del canal .

Desplegando el menú totalmente, presionando la flecha de la parte inferior del menú, modificar la opción Nivel para hablar.

Una vez que tenemos esto, se pueden dar privilegios a los usuarios sobre los canales: Para ello vamos a la opción de la barra Permisos, y allí escogemos Grupos de canal del usuario, ver figura 3.4.48.

Dentro de la pantalla de la figura 2.6.48, podremos distinguir los siguientes campos:

- Identificador de la base de datos, es el número del usuario en la base de datos, aquí se mete el valor del usuario deseado, o se van introduciendo los valores consecutivos hasta que aparezca el valor que se quiere.
- Aparece el identificador único del usuario , y el grupo del canal (conjunto de privilegios definidos). Picando sobre la opción de grupo de canal en el canal deseado, aparece un menú desplegable con las opciones posibles, se selecciona la opción que se desee y se guardan los privilegios del usuario en ese canal.

En el ejemplo se ve que en el canal SSB2182 se requiere el acceso con los privilegios de profesor.



Figura 2.6.48 Servidor Teamspeak Marconi. Grupos de canal del usuario.

Siguiendo estas reglas se ha programado el servidor Marconi del grupo S2CN de la UCA.

2.6.7 Streaming de voz en Linux.

2.6.7.1 Introducción.

Como complemento a lo desarrollado hasta ahora, se van a explicar los programas clásicos para la realización de streaming de voz dentro del entorno Linux. Para ello vamos a utilizar un software libre, como hemos hecho durante esta Tesis Doctoral.

Ices 2 y Icescast 2 son las utilidades más usadas en Linux para transmitir audio en una red TCP / IP.

Icescast es el servidor de audio, siendo Ices el cliente.

Vamos a usar la distribución de Linux Ubuntu para establecer un servidor y unos clientes de audio.

Para ello vamos a hacer la instalación de los paquetes usando apt-get, aunque se puede usar el paquete gráfico synaptic.

Usamos los comandos de consola:

```
$ sudo apt-get install icecast2 ices2
```

Icecast utiliza un archivo de configuración xml donde están todos los parámetros de la emisora además de unos archivos para la interfaz Web. Los archivos predeterminados se instalan con los paquetes. Hay que hacer una copia de ellos en el directorio del usuario.

Primero se crea una estructura de directorios para albergar los archivos de configuración y de registro de icecast.

Desde el directorio home se ejecuta:

```
$ mkdir icecast  
$ cd icecast  
$ mkdir conf  
$ mkdir logs
```

A continuación se copia el archivo de configuración predeterminado de Icecast

/etc/icecast2/icecast.xml a la carpeta que hemos creado.

Si el archivo no está en el lugar indicado, se busca con el comando find:

```
$ find / -name icecast.xml
$ cd conf
$ cp /etc/icecast2/icecast.xml icecast.xml
$ sudo chown user:group icecast.xml
```

Dado que el archivo pertenece al administrador (root), se ha usado la instrucción “chown” para hacer que nos pertenezca.

Ejemplo :

```
sudo chown jjp:jjp icecast.xml
```

Por ahora no se modifica el archivo de configuración. Antes de eso, se copian las carpetas que el Icecast utiliza para la interfaz Web.

Estas carpetas normalmente están en /usr/share/icecast2 pero si no están allí se puede usar el comando find para localizarlas.

```
$ cd ..
$ cp -r -L /usr/share/icecast2/web web
$ cp -r -L /usr/share/icecast2/admin admin
$ sudo chown -R user:group admin
$ sudo chown -R user:group web
```

Donde user y group hay que cambiarlo por nuestro nombre de usuario y de grupo respectivamente.

Ahora se configura Icecast. Para ello se modifica el archivo de configuración copiado:

```
$ cd conf
$ gedit icecast.xml
```

Las líneas en el orden que aparecen en el archivo son:

```
<clients>10</clients>
```

Esta línea fija la cantidad máxima de clientes (reproductores) que se pueden

conectar a la emisora. Normalmente esto se ajusta de acuerdo con la cantidad de ancho de banda de la que se dispone para emitir. Un valor razonable es 50.

```
<source-password>hackme</source-password>
```

En esta línea se coloca la contraseña que las fuentes (en este caso el ices2) utilizan para conectarse al servidor y enviar la música.

Es recomendable cambiar la contraseña predeterminada para evitar que alguien más pueda emitir con nuestro servidor.

```
<relay-password>hackme</relay-password>
```

(no usado)

```
<admin-user>admin</admin-user>
```

```
<admin-password>hackme</admin-password>
```

Nombre de usuario y contraseña para conectarse en el módulo de administración Web de Icecast.

```
<hostname>localhost</hostname>
```

Si la emisora está montada en un servidor al que se accede con un dominio que se ha adquirido, aquí se coloca el nombre del dominio: Por ejemplo miradio.org. Si se está montando un servidor de pruebas, basta con dejar localhost.

```
<port>8000</port>
```

Éste es el puerto por el que se emite. Normalmente no hay que cambiar el puerto a menos que haya restricciones de firewall.

```
<logdir>/home/user/icecast/logs</logdir>
```

```
<webroot>/home/user/icecast/web</webroot>
```

```
<adminroot>/home/user/icecast/admin</adminroot>
```

Estas tres líneas le dicen al Icecast dónde guardar sus registros, dónde están los archivos del panel de interfaz Web y dónde están los archivos del panel de administración de la interfaz Web. (Cambiar user por el nombre de usuario).

A continuación se pone en marcha servidor Icecast .

Para arrancar el servidor se ejecuta el comando siguiente:

```
$ icecast2 -c icecast.xml
```

Si la consola no arroja ningún error, entonces, Icecast ya está corriendo. Ahora se verifican los logs y la interfaz Web para ver que todo está funcionando correctamente.

Nota importante: La consola ha de quedar bloqueada con el Icecast. No se puede cerrar o se caerá el servidor.

Para chequear los procesos se abre otra ventana y se ejecutan desde el directorio “home” los siguientes comandos:

```
$ cd icecast
```

```
$ cd logs
```

```
$ cat error.log
```

Se debe ver algo parecido a :

```
[2010-05-22 13:08:33] INFO main/main Icecast 2.3.2 server started
```

```
[2010-05-22 13:08:33] INFO yp/yp_update_thread YP update thread started
```

```
[2010-05-22 13:08:33] INFO connection/get_ssl_certificate No SSL capability
```

No importa si el log es un tanto diferente. Lo importante es que no haya errores y que aparezca la línea que dice “server started”.

Para ver el estado del servidor se usa el navegador con la dirección

<http://localhost:8000/> éste presenta una página de estado del servidor.

Para entrar en modo administración hay que introducir el nombre de usuario y contraseña especificados en :

```
<admin-user></admin-user> y <admin-password></adminpassword>
```

Icecast es el servidor de emisión. Él se encarga de aceptar las peticiones entrantes y de distribuir el medio a los escuchas que se conecten pero no sabe nada de música ni de listas de reproducción. De ello se encarga el programa al que se denomina “fuente”

(source).

Existen varias fuentes diferentes para el Icecast. Entre ellas, la más básica, es Ices, desarrollada por las mismas personas que desarrollaron Icecast. También se pueden usar otras fuentes diferentes, por ejemplo VLC Player.

Vamos a usar Ices como fuente del audio a escuchar.

Para ello hay que instalar Ices.

Los pasos de instalación son:

Desde el directorio home se ejecuta:

```
$ mkdir ices
```

```
$ cd ices
```

```
$ mkdir conf
```

```
$ mkdir playlists
```

```
$ mkdir logs
```

A continuación se copia el archivo de configuración predeterminado del Ices para usarlo con listas de reproducción.

Si no se encuentra el archivo se puede usar el comando find para localizarlo.

```
$ cd conf
```

```
$ cp /usr/share/doc/ices2/examples/ices-playlist.xml ices.xml
```

```
$ gedit ices.xml
```

Se edita el archivo de configuración de Ices.

```
<logpath>/home/user/ices/logs</logpath>
```

Esta línea le dice a Ices donde guardar los archivos de registro. Reemplazar user con nombre de usuario.

```
<name>Nombre de la emisora</name>
```

```
<genre>Genero</genre>
```

```
<description>Descripción de la emisión</description>
```

Estas tres líneas contienen los datos de la emisora: nombre de la emisora, el género y la descripción.

```
<param name=file>/home/user/ices/playlists/playlist.txt</param>
```

Esta línea le dice a Ices dónde encontrar el archivo de lista de reproducción. Reemplazar user con nombre de usuario.

```
<param name=random>0</param>
```

Esta línea le dice a Ices si se toca la lista de reproducción secuencialmente o aleatoriamente. Con el valor 1 Ices toca las entradas aleatoriamente.

```
<param name=once>0</param>
```

Este parámetro le dice a Ices si debe tocar la lista de reproducción sólo una vez o debe tocarla infinitas veces. Con el valor 1 Ices toca la lista de reproducción una vez y cuando termine se cierra la conexión con el servidor y se finaliza, haciendo que la emisora ya no se pueda escuchar más.

Para que la emisora funcione todo el día y que la gente se conecte cuando quiera se debe establecer el valor en 0 para que Ices repita indefinidamente la lista de reproducción.

```
<hostname>localhost</hostname>
```

```
<port>8000</port>
```

```
<password>hackme</password>
```

```
<mount>/example1.ogg</mount>
```

Ésta es la información del servidor. Aquí se coloca la información del servidor Icecast a través del cual se va a emitir.

Esta información debe coincidir con la información que ya se había configurado en el icecast.xml.

En

```
<password></password>
```

colocamos el password que se había puesto en la etiqueta

```
<source-password></source-password>
```

 de icecast.xml

En

```
<mount></mount>
```

 debemos escoger el mount point que queramos para emitir.

El mount point es la parte final de la URL que se usa para escuchar la emisora,

no debe contener espacios y debe terminar en .ogg ya que es el códec que Icecast e Ices usan para emitir. Si no se pone el .ogg algunos reproductores se quedan colgados sin hacer nada cuando se trata de escuchar la estación.

En `<nominal-bitrate>64000</nominal-bitrate>`

Se define la velocidad con la que emitirá la emisora.

Cuanto mayor sea la velocidad, mejor la calidad del audio, pero se necesita un mayor ancho de banda.

Si no se está muy seguro de qué velocidad usar, se debe dejar la predeterminada, que es de 64kbps. Esta velocidad garantiza una buena calidad para la mayoría de aplicaciones.

Ahora, hay que crear una lista de reproducción.

La lista de reproducción es simplemente un archivo en el que cada línea tiene la ruta de acceso a un archivo .ogg, por ejemplo:

```
/home/user/music/anime/música - rolling star.ogg  
/home/user/music/anime/música - every heart.ogg  
/home/user/music/anime/música - catch you catch me.ogg  
/home/user/music/anime/música graffiti - melissa.ogg
```

Puede resultar un poco tedioso crear la lista a mano, pero se puede usar el comando find para agilizar el trabajo, por ejemplo:

```
$ find /home/user/ -name *.ogg > playlist.txt
```

Este comando escribe en un archivo llamado playlist.txt todos los archivos .ogg que encuentre en la carpeta /home/user/, (reemplazar user con nombre de usuario).

En cuanto se tenga preparada la lista de reproducción (se debe guardar en /home/user/ices/playlists/playlist.txt) se puede poner en marcha Ices.

Para poner en marcha Ices se ejecutan los siguientes comandos:

```
$ cd ices
```

```
$ cd conf
$ ices2 ices.xml
```

Si todo va bien, no debe aparecer ningún mensaje de error.

Para ver el estado de Ices se ejecuta en una nueva consola los comandos (desde home):

```
$ cd ices
$ cd logs
$ cat ices.log
```

Debemos obtener algo similar a:

```
[2010-05-22 18:20:55] INFO ices-core/main IceS 2.0.1
started...
[2010-05-22 18:20:55] INFO signals/signal_usr1_handler
Metadata update requested
[2010-05-22 18:20:55] INFO playlistbasic/
playlist_basic_get_next_filename Loading playlist from
file /home/sergio/ices/playlists/playlist.txt
[2010-05-22 18:20:55] DEBUG playlist-basic/shuffle Playlist has
been shuffled
[2010-05-22 18:20:55] INFO playlist-builtin/playlist_read
Currently playing /home/sergio/Msica/J-Music/YUI/From Me To
You/11.ogg
[2010-05-22 18:20:55] INFO stream/ices_instance_stream
Connected to serverlocalhost:8000/yui.ogg
[2010-05-22 18:20:55] DEBUG reencode/reencode_page
Reinitialising reencoder for new logical stream
[2010-05-22 18:20:55] INFO encode/encode_initialise Encoder
initialising in VBR mode2 channels, 4410Hz, nominal 128000
```

Básicamente lo que se debe buscar es una línea como:

```
[2010-05-22 18:20:55] INFO
playlist-builtin/playlist_read Currently playing /home/sergio/Msica/J-
```

Music/YUI/From Me To You/11.ogg

que indica que Ices leyó y empezó a emitir uno de los archivos que se puso en la lista de reproducción.

También se puede verificar el estado del servidor desde la interfaz Web entrando a <http://localhost:8000>. Allí debe aparecer el punto de montaje (mount point) que se ha configurado con el metadata que se colocó en el archivo.

Para escuchar la música emitida se puede hacer con el programa VLC Player:

En el menú Medio selecciona Abrir volcado de red, en el cuadro que aparece se escribe en el campo de dirección:

`http://localhost:8000/mount.ogg` (reemplazar `mount.ogg` por el mount point que se ha colocado en `<mount></mount>` en el archivo `ices.xml`) y luego hacer clic en el botón Reproducir.

Si todo va bien se escuchará la música que la emisora emite.

Es posible ejecutar Icecast e Ices sin que la consola quede bloqueada. Para ello se deben ejecutar los comandos agregando un ampersand (&) al final:

```
$ icecast2 -c icecast.xml &  
$ ices2 ices.xml &
```

Al hacerlo así, el sistema responde con el pid del proceso:

```
[1] 2923
```

El pid (id de proceso) de Icecast identifica el proceso en el sistema y se necesita para detener la emisión más adelante. Si se pierde el número o la emisora ha estado emitiendo durante mucho tiempo y ya no se recuerda, se puede recuperar usando el comando:

```
$ ps ax | grep icecast2
```

El sistema responde con algo como:

```
2329 pts/1 Rl 0:03 icecast2 -c icecast.xml
```

El primer número que aparece es el pid de Icecast.

Para detener la emisión se debe enviar al proceso la señal de salida (TERM).

Se hace con el comando:

```
$ kill -15 2923
```

Aunque kill puede parece algo drástico, en realidad, ese comando simplemente envía una señal de terminación al proceso, no lo mata forzosamente.

El sistema responde con algo como:

```
[1]+ Hecho icecast2 -c icecast.xml
```

Esta línea indica que el proceso ha terminado.

Cuando se configura una estación de radio con Icecast e Ices, una de las preguntas que primero surge es ¿Cómo convierto un Mp3 a ogg?

Para convertir un Mp3 a ogg en Linux, basta un comando de consola pero antes hemos de tener instalados los paquetes necesarios. Para ello hacemos:

```
$ sudo apt-get install mpg321 vorbis-tools
```

Ahora para convertir un mp3 a ogg:

```
$ mpg321 music.mp3 -w -l oggenc -o music.ogg
```

Con esto ya se puede hacer streaming de voz en Linux , y tener nuestra propia emisora.

2.6.7.2 Cómo calcular el ancho de banda necesario.

Si no estamos muy seguros de cómo saber cuántos oyentes pueden soportar el canal o cuál es la máxima velocidad binaria que se puede usar sin que los oyentes

experimenten problemas se debe hacer una aproximación según se indica a continuación.

El ancho de banda concurrente es la anchura del canal que se necesita para emitir.

Se calcula de la siguiente manera:

$$cb = b * n \quad (2.1)$$

Donde el parámetro cb es el ancho de banda concurrente (en kbps), b es la velocidad binaria a la que se emite (en kbps) y n la cantidad de oyentes que se espera que se conecten simultáneamente.

Así, por ejemplo, si se va a emitir a 128 kbps y se esperan 50 oyentes simultáneos $cb = 128 \text{ kbps} * 50 = 6400 \text{ kbps}$. Así pues se necesita un canal de 6.4 Mbps para soportar 50 oyentes simultáneos si emite a 128 kbps

2.6.7.3 Límite de Tráfico.

Algunos servidores imponen un límite mensual de tráfico y cuando se sobrepasa este límite o bien empiezan a cobrar más o bien no permiten emitir más por el resto del mes. Para saber si el límite mensual es suficiente para soportar la emisora se puede calcular de la siguiente forma:

$$tb = b * n * at * 30 \quad (2.2)$$

Donde tb es el ancho de banda total del mes (en kbits), b es el bitrate (velocidad de transmisión) a la que se emitirá (en kbps), n es la cantidad de oyentes promedio que se espera por día y at es la cantidad promedio de tiempo que un oyente estará conectado al día (en segundos).

Así pues, si se emite a 128 kbps, y se esperan 50 oyentes diarios y cada uno estará conectado un promedio de 5 horas diarias escuchando la emisora, entonces tendremos:

$$tb = 128 * 50 * 18000 * 30 = 3456 \text{ GB}$$

Se necesitará un límite de tráfico mensual de al menos 3500 GB

3. Objetivos de la Tesis Doctoral.

Los Objetivos de esta Tesis Doctoral son :

1.- Crear un Demostrador de Conceptos que ayude a asentar las bases técnicas mediante:

Un sistema integrado de comunicaciones de Radio por Protocolo Internet (ROIP).

Un sistema integrado de comunicaciones interiores que compartiese los mismos datos e infraestructuras que el anterior.

Disponer de acceso al sistema desde el exterior de la UCA.

Disponer de acceso al sistema desde el interior de la UCA.

Integración de vídeo IP dentro del Sistema.

Integración de Tabletas y Smartphones con todo el sistema.

2.- El acceso completo a dicho sistema y equipos desde cualquier parte del Mundo donde tenga una conexión a Internet mediante distintos aparatos electrónicos, tales como, teléfonos IP, tabletas electrónicas o teléfonos tipo “smartphone”, aparte de los ordenadores fijos y portátiles, ya fuera por cable o por radio (WIFI).

3.- La aplicación a las radiocomunicaciones tanto de las nuevas tecnologías de la Información y las Telecomunicaciones, así como los avances en ROIP y SDR. Añadiendo la posibilidad de tener imágenes en “tiempo real” del entorno de las comunicaciones que se están recibiendo, entendiendo por entorno la presentación en una Pantalla Gráfica de las Señales en el dominio de tiempo y de la frecuencia y la posibilidad de recibir dichas radiocomunicaciones en los domicilios particulares para la realización de prácticas o simplemente el asentamiento eficaz de los conocimientos teóricos.

4. Metodología propuesta para el desarrollo de esta Tesis Doctoral.

Para el desarrollo de esta Tesis Doctoral se ha planteado la metodología descrita en la Fig. 4.1.1.

4.1 Enfoques metodológicos.

Para la fase de documentación y conceptualización se ha aplicado un diseño metodológico transversal exploratorio que permita llegar a las fuentes y extraer la información relevante, para ello se ha explorado tanto la documentación existente en el área de la telefonía software (Asterisk, Trixbox, 3CX) como de las comunicaciones sobre IP y SDR. Se ha incidido de forma especial en el entorno GNU, Open Software, con objeto de obtener soluciones aplicables a bajo coste.

Se han explorado los foros de desarrollo existentes en Yahoo y Google, así como se ha interrelacionado con los grupos de investigación y empresas líderes de las distintas áreas.

En concreto se ha participado en los siguientes grupos:

Sdr-widget de QtRadio.

UltraCheapSDR y Osmocom del Receptor RTL-SDR.

Airspy sobre nuevo Receptor Dongle con base en RTL-SDR.

Sdr-radio.com de SDR-Radio.

Sdr-users grupo generalista sobre el uso de SDR.

GNU-Radio del grupo GNU-Radio.

Fonality grupo de la centralita Trixbox.

Perseus-SDR del receptor Perseus.

basicDSP sobre el desarrollo WebSDR,

ShiftEdit sobre IDE basado en entorno Web.

Teamspeak, foro de soporte de Teamspeak.

Grandstream, foro de soporte de las aplicaciones Grandstream.

Se han seguido las siguientes líneas generales:

Investigación en el entorno cliente servidor.

Aplicaciones multiplataformas.

Arquitectura 386 y ARM (esta última para dispositivos de bajo consumo).

Portabilidad de las soluciones y movilidad.

Prioridad de los entornos Linux y Android

Aplicación a la Enseñanza de las soluciones.

En la fase de descripción de la tecnología, se ha empleado un diseño metodológico transversal descriptivo cuyos resultados sirven de base para las implementaciones.

Para la creación del Demostrador de Conceptos o Red Marconi se aplican metodologías propias del desarrollo de sistema e interfaces, se prueban los subsistemas mediante realización de test y análisis de resultados.

Se desarrolla cada subsistema de forma independiente, se instala, se configura, se prueban las interfaces y se depuran los problemas existentes. Una vez en funcionamiento el modelo básico se desarrolla la aplicación, añadiendo funcionalidad al subsistema.

En concreto se montan dos sistemas completos con IP-Cop y Trixbox en distintos lugares y separados por los routers locales y los firewall de los equipos, donde se prueban las comunicaciones y la funcionalidad de la centralita Trixbox, haciendo uso de proveedores externos de VOIP. Se depuran y analizan los problemas acudiendo a los grupos para la resolución de éstos.

Se prueban los equipos IP y aplicaciones IP sobre la centralita local Trixbox para investigar sus funcionalidades.

De igual forma se investiga y programa el firewall IPCop.

Se monta aplicación local de Teamspeak para depurar problemas de instalación, uso y desarrollo de aplicación.

Se instalan y prueban las centralitas Asterisk, Elastic, Trixbox y 3CX, tanto en máquinas virtuales como físicamente, analizando la funcionalidad de éstas, así como programando la centralita Trixbox y depurando problemas procedentes de la instalación de la tarjeta X100P y la definición de troncales.

Se prueban funcionalidades y programación de las centralitas, recogidas en esta tesis.

Se configuran los PC instalando los sistemas operativos Debian, Ubuntu y XP y se analiza la funcionalidad de las distintas aplicaciones y la interconectabilidad de las plataformas.

Se investiga la instalación y la funcionalidad de GNU- Radio y QtRadio usando máquinas virtuales así como instalaciones limpias de los sistemas operativos, probando cada uno de ellos tanto en Debian como en Ubuntu. Cada aplicación se prueba en instalaciones prototipos, identificando las faltas debidas a dependencias o errores en los scripts de instalación.

Se crea instalaciones de prueba para el software SDR-Radio, definiendo servidor y clientes, investigando en el área de manejo de los equipos usados dentro de esta tesis.

Se impone como línea fundamental de investigación el crear y usar los receptores básicos SDR. Para ello se construye un receptor Softrock y se trabaja con Tony Parks para obtener otro segundo equipo Softrock con una frecuencia central de sintonización variable. Se persigue con ello tener una

base de receptores básicos para el estudio del SDR así como investigar en el manejo del SDR a través de la tarjeta de sonido del PC.

Se definen dos servidores de SDR en los que se instalan los sistemas operativos Windows XP y Ubuntu y se aplican las soluciones Softphone, así como softwares SDR en ambos.

En el equipo definido con la prioridad de servidor XP, se instalan de forma concurrente los receptores Funcube, Softrock, SDR-IQ y RTL-SDR, depurando problemas de configuración de interfaces dentro de Windows XP, recurriendo para la resolución de problemas de integración críticos al grupo SDR-Radio así como a información de las distintas personas involucradas en los desarrollos.

Como punto a destacar, en este servidor se instalan dos tarjetas de red que permiten que sea accedido tanto desde el sistema Marconi como desde la UCA. Además para el acceso remoto se le instala el servidor VNC y el servidor IP-Sound de compartición de escritorio y sonido respectivamente. Todas estas soluciones han requerido de la resolución de problemas tanto a nivel software como hardware.

En el equipo definido con la prioridad de servidor Ubuntu, se definen dos usuarios y se estudian dentro del entorno Ubuntu el manejo de las aplicaciones. A su vez, se prueban tarjetas externas de audio. Se analiza usando las herramientas de Linux el correcto uso de las interfaces. Se programa en este equipo la web Marconi, depurando problemas de interconexión dentro del entorno de la Universidad.

Dentro del demostrador de conceptos Marconi, tenemos los siguientes subsistemas básicos:

- Subsistema de red interna y seguridad (IP-Cop Marconi).
- Subsistema de telefonía y servidor SIP (Trixbox, Videoteléfonos, Teléfonos Soft).
- Subsistema de comunicación y voz sobre IP (Teamspeak).
- Subsistema SDR y ROIP (SDR-Radio, QtRadio, WebSDR, Otros).
- Subsistemas de Vídeo sobre IP (Trixbox, Videoteléfonos, Cámaras IP).
- Subsistemas de ayuda y participación (Servidor Web, Foro S2CN).
- Subsistemas de ayuda a la Enseñanza de las Radiocomunicaciones GNU-Radio.

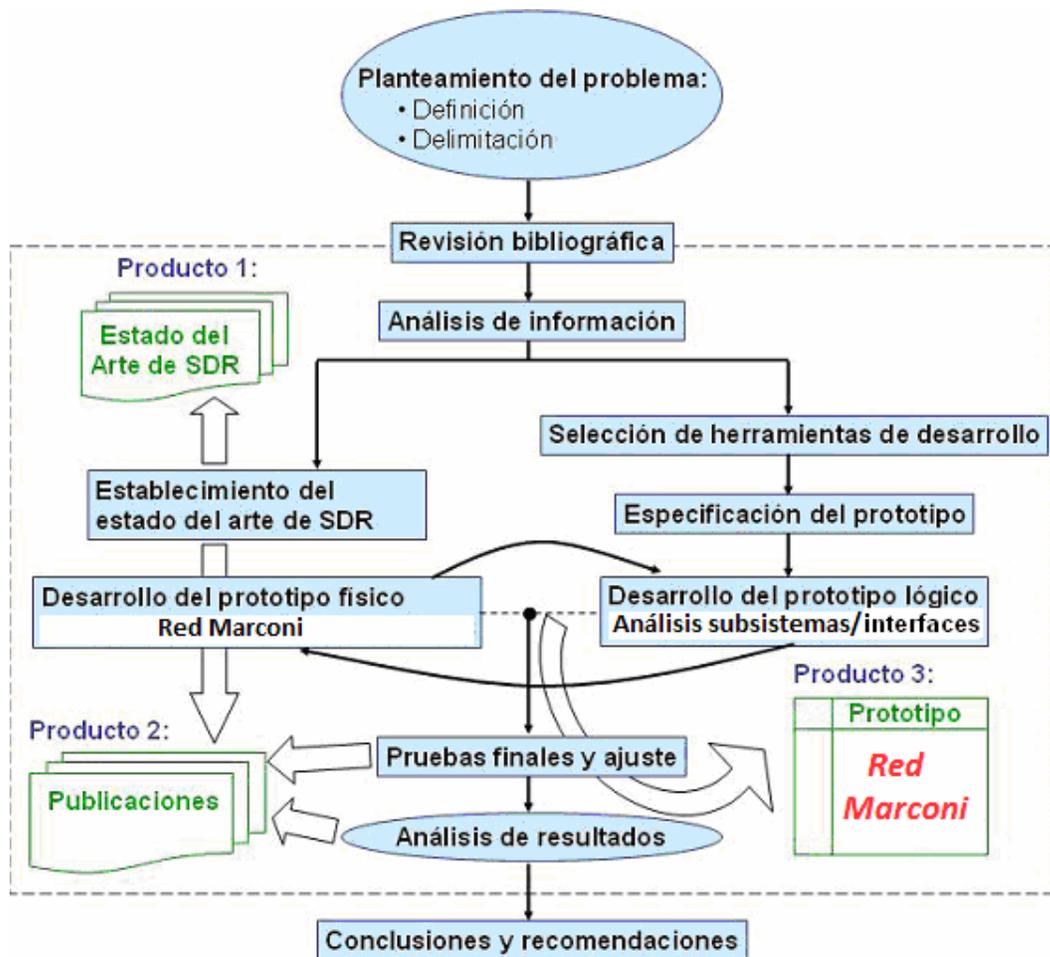


Figura 4.1.1 Descripción de la metodología a utilizar en el desarrollo de prototipos.

4.2 Desarrollo metodológico.

Se realiza una revisión de la bibliografía y de las fuentes de información para llegar a unas sólidas bases conceptuales sobre los distintos subsistemas del demostrador Marconi. (Técnicas: Análisis de contenidos; Instrumentos: Guías para análisis de contenido).

Descripción y modelización funcional del demostrador Marconi. (Técnicas: Análisis descriptivo; Instrumentos: Comparativa de funcionalidades y capacidades de equipos mediante el uso de la internet).

Selección y adquisición de un kit básico para el desarrollo del Demostrador de Conceptos Marconi (Técnicas: Comparativas; Instrumentos: Catálogos de productos).

Análisis prácticos demostrativos del demostrador de Conceptos Marconi. (Técnicas: Construcción

de prototipo; Instrumentos: Kits de desarrollo).

Pruebas y optimización del prototipo. (Técnicas: Análisis de resultados; Instrumentos: Kit de desarrollo).

Análisis de resultados y conclusiones. (Técnicas: Análisis de datos; Instrumentos: Software especializado).

4.3 La Radio Definida por Software en la práctica.

En general los diseños de SDR “prácticos” empiezan por definir el digitalizador disponible y construyen el sistema alrededor de él. 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 así como nivel de señal que pueda manejar el digitalizador disponible.

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

1. 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.
2. 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.
3. 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.

En el Demostrador de Conceptos Marconi se ha implementado este modelo, existiendo los siguientes componentes:

- Equipos receptores que reciben la señal y la transmiten al procesador a través de la entrada Line in estéreo del PC (Softrock).
- Equipos receptores que reciben la señal y la transmiten al procesador a través de la entrada USB del PC (RTL-SDR, Funcube, Perseus, SDR-IQ).
- Equipos receptores con arquitectura cliente-servidor que reciben la señal en el servidor,

normalmente a través de puertos USB, y transmite al cliente las señales recibidas, existiendo normalmente tres flujos de información entre cliente y servidor:

- Audio.
- Datos FTT para representación gráfica de la cascada.
- Canal de comunicación cliente-servidor.

4.4 Tipos de SDR.

La taxonomía de aplicaciones SDR que se utiliza en el sitio web de Christophe F4DAN 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 un PC como digitalizador y software convencional de PC como elemento de procesamiento. Es probablemente el tipo que es más accesible para radioaficionados.

Tipo Ia. El SDR se implementa alimentando a la placa de sonido con la señal de banda base o de salida de audio de un receptor convencional de comunicaciones.

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 coste-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 allí.

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.

En el caso del Demostrador de Conceptos Marconi, se ha implementado el SDR tipo Ic usando los receptores tradicionales Softrock y Funcube, donde se entrega la señal IQ a la placa de sonido del PC, a través de la entrada Line in estéreo de ésta.

En el caso del receptor Funcube, éste se define y se instala como placa de sonido externa en el PC.

Se ha implementado el SDR tipo II usando los receptores SDR-IQ, RTL-SDR, que vienen con sus drivers para el PC, que permiten sintonizar, controlar y mostrar la representación en la escala de frecuencias y tiempo del espectro recibido así como reproducir el audio.

Se ha implementado el SDR tipo V usando los servidores de WebSDR y QtRadio, que a su vez reciben la señal a través de los receptores Perseus, RTL-SDR y Softrock, que procesan la señal recibida de estos receptores y la proveen a sus clientes a través de sus servidores, Dspserver en el caso del QtRadio y WebSDR en el caso del WebSDR.

4.5 Software para SDR.

Un sistema SDR tiene tres componentes: el primero de ellos, es la etapa de procesamiento preliminar de señales por hardware y queda claro que la placa de sonido del PC será en la mayoría de los casos el segundo (dispositivo digitalizador).

Sin embargo el corazón mismo del sistema es el software que procesa digitalmente las señales.

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 descrito 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.

Hay varias formas de clasificar los paquetes de software:

- 1. 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 código fuente disponible.

2. **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.
3. **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.

Los paquetes utilizados dentro del desarrollador de conceptos Marconi, pertenecen a los tipos Cerrados, Semi-Abiertos y Abiertos.

Dentro de los paquetes cerrados, se consideran aquellos desarrollados por la comunidad de radioaficionados para el manejo de los receptores tradicionales.

Se pueden considerar divididos en paquetes standalone (independiente) o clientes-servidor.

Como paquetes standalone destacamos los siguientes:

- HDSDR (Winrad)
- SDR-Sharp
- Rocky
- SDR-Radio Console

Como paquetes cliente-servidor destacamos:

- QtRadio
- WebSDR
- SDR-Radio servidor

Como paquete de tipo semi-abierto / abierto, consideramos QtRadio. Este paquete tiene un grupo de desarrollo que provee los códigos fuente, siguiendo la filosofía de software libre.

Los códigos fuentes se encuentran disponibles en el servidor GIT, existiendo diferentes ramas de desarrollo, así como diferentes líneas para los entornos Web, Android, Linux, Windows.

Dentro del demostrador de conceptos, se ha elegido este grupo de investigación y este programa por la capacidad de análisis de las soluciones así como de la colaboración en el futuro desarrollo de la tecnología SDR.

El paquete QtRadio se ha instalado con la descarga local de sus fuentes y la compilación local de éstos, estando todos estos accesibles para posterior análisis.

A su vez, se ha utilizado los software cerrados de QtRadio para los entornos Windows y Android, de forma que pueda ser fácil la instalación en los PC's individuales de los alumnos de radiocomunicaciones de la UCA, y les permita realizar de forma fácil las practicas requeridas a su especialización.

Como paquete abierto y por su interés para la Enseñanza se ha instalado el paquete GNU-Radio.

Solo recordar en este punto que todos los paquetes utilizados son software libre y por lo tanto su instalación y funcionamiento han requerido la adquisición del conocimiento necesario y la participación en los grupos de desarrollo. Se ha tenido que modificar aquella parte que estos paquetes no cumplían, éste es el caso de uso de receptores determinados, así como su instalación en entornos de sistemas operativos donde por motivos obvios se ha de instalar aquellos drivers o librerías que se requiera.

En el Demostrador de Conceptos Marconi se ha pretendido tener en funcionamiento las aplicaciones “State of the Art” por motivos lógicos de investigación y enseñanza, la integración de estas aplicaciones no ha sido fácil, pues se requería un nivel experto en informática y comunicaciones. A su vez, se ha dispuesto de equipos de depuración que permitieran resolver problemas que lógicamente aparecen en aplicaciones beta.

4.6 Aplicaciones SDR basadas en servidor.

El poner un receptor disponible en una red local o en 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 parámetros de recepción tales como modo, frecuencia y otros parámetros.

Más allá de lo interesante que puede ser este tipo de implementaciones (¿quién no los ha 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 ningún 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 puede realizarse en otra.

De esa manera múltiples 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 sino 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 de un buen público donde evaluar las plataformas.

Un ejemplo notable de una implementación es el receptor implementado en la Universidad de Twente (Holanda) donde segmentos de las bandas de VLF, LF, MF, 20, 40, 80 y 160 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.

Dentro del demostrador de conceptos Marconi y en colaboración con la Universidad de Twente se ha instalado la WebSDR Marconi, dando acceso a dos receptores en la banda de 40 metros y FM.

De esta forma, los alumnos de la UCA pueden tener acceso a sintonizar y escuchar las emisiones desde cualquier lugar en el que se encuentre.

También tenemos que destacar como aplicaciones SDR basadas en servidor la mencionada anteriormente QtRadio y SDR-Radio.

SDR-Radio es una aplicación servidor muy extendida en el mundo de los radioaficionados, teniendo mucha aceptación y un desarrollo e interfaz de usuario realmente destacados. El único

inconveniente que se le podría encontrar es que está desarrollado sólo en Windows, siendo estos servidores, desde el punto de vista del proceso, menos eficientes que los servidores Linux.

Dentro del demostrador de conceptos Marconi, se ha implementado este servidor siendo accesible para todos los alumnos desde dentro de la Universidad.

Dentro del uso del software SDR-Radio se han implementado las versiones beta que permiten integrar receptores adicionales como RTL-SDR y a su vez interactuar como en las otras aplicaciones con los grupos de desarrollo.

De igual forma WebSDR se encuentra en proceso de evolución, esperando futuras aplicaciones con desarrollos en HTML5, descartando de esta forma el uso de la máquina Java, que inhabilita su uso en sistemas Android. Dentro del grupo S2CN se pretende evolucionar en esta área acercándonos a los protocolos WebRTC.

4.7 Aplicaciones VoIP y Telefonía sobre IP.

En el estado actual de las comunicaciones y de los sistemas basados en internet, existe claramente el objetivo de realizar sistemas de telefonía y de voz basados en IP.

Los motivos prácticos son evidentes: ahorro de costes, al disponer de una única red para datos y voz, así como el evidente ahorro de costes al convertir, debido a la internet, llamadas internacionales en llamadas locales.

Dentro del demostrador de conceptos Marconi se ha implementado un par de equipos que habilitan para la práctica de estas tecnologías.

El primero de ellos es una centralita software Trixbox. Este sistema ha sido escogido porque su interfaz gráfica permite seguir de forma visual la identificación en ésta de los equipos IP, así como la resolución de las llamadas.

A esta centralita se le ha provisto de una tarjeta X100P que permite el acceso a la red telefónica conmutada por terminal telefónico simple (POTS).

El objetivo de la implementación de esta centralita es poder enseñar los equipos de telefonía que se irán implantando en los futuros sistemas telefónicos existentes en los barcos.

Como acompañamiento a esta centralita telefónica se dispone de dos videoteléfonos y dos cámaras IP que por sus capacidades, estas últimas de detección de movimiento y alerta de incendios, serán

implementadas en el futuro como sistemas integrados dentro del sistema de telefonía.

Asimismo se implementan sistemas VoIP Teamspeak.

Con la utilización del TS3 Marconi, se permiten las prácticas de radiocomunicaciones a nivel mundial, en cualquier idioma y compartiendo recursos informáticos de forma gratuita sin tener que conocer el funcionamiento de los equipos. Gracias a esto, se ahorra el tiempo de enseñanza del manejo del simulador y se va directamente al grano a practicar los Procedimientos Radioeléctricos de rutina: llamadas comerciales, de socorro, de urgencia o de seguridad, siendo posible la transmisión de datos en forma de audio de banda base y la decodificación simultánea y remota mediante los programas de ordenador apropiados, ya sea para Llamada Selectiva Digital, Morse, Radioteletipo ARQ o FEC, facsímil meteorológico o cualquier otro de uso común en la Marina Mercante, por ejemplo el Navtex.

4.8 Sistema de ayuda al uso y mejora del demostrador Marconi.

Como punto importante dentro del desarrollo de cualquier prototipo o aplicación, se ha de disponer de alguna forma de validar lo implementado y mejorar.

Para todo grupo de desarrollo o soporte es importante recoger las opiniones de las personas que usan sus sistemas y aplicaciones, siendo el usuario, pieza clave en el desarrollo de un sistema útil para la consecución de los fines que se persiguen.

Con este objetivo en mente se ha desarrollado un entorno de ayuda e información que pueda recoger la opinión de los usuarios.

La entrada a este sistema se realiza a través de la página web del grupo S2CN, Marconi.uca.es .

Esta página se ejecuta dentro del servidor de WebSDR del demostrador Marconi.

Esta página nos da la descripción de los objetivos del grupo así como de los links a las principales páginas web del grupo y de la universidad y así mismo nos provee de enlace al Foro de opinión del grupo S2CN.

El foro del grupo S2CN ha sido desarrollado dentro de esta tesis doctoral, usando el software libre de desarrollo de foros phpBB, presentando áreas para compartir información y opinión dentro de las

distintas líneas del grupo S2CN.

A su vez se ha preparado e instalado una distribución libre del software GNU-Radio con objeto de practicar los conceptos básicos de las radiocomunicaciones. Esta aplicación se integra dentro del entorno Marconi, lo que hace que se pueda compartir dentro de la UCA.

5. Resultados – Aplicaciones prácticas realizadas en el Laboratorio del Grupo S2CN.

5.1 Introducción.

El objetivo fundamental de esta Tesis Doctoral ha sido el estudio y aplicación del procesado digital de la señal, así como la puesta en marcha de un entorno de trabajo que ayude a avanzar en el campo de la investigación de las nuevas tecnologías aplicadas a las radiocomunicaciones.

Para ello, se ha desarrollado una red interna IP, detrás del Firewall IP-COP.

Se ha provisto de un router conectado a este Firewall que habilita la conexión wifi dentro del laboratorio (red interna Marconi). Esto permite a los interesados acceder bien con portátiles o bien con PC's, Tabletas o Smartphones con wifi a los equipos de la red Marconi.

El router se ha reconfigurado como router nulo, actuando como receptor wifi y como puente al Firewall IP-Cop, encargado éste de entregar las direcciones IP de la red interna.

El firewall IP-COP tiene dos conexiones de red:

Red Roja, conectada a internet pública a través de la red UCA.

Red verde, conectada a la red interna Marconi.

El firewall IPCop se ha descargado como software libre, instalándose en un PC dedicado y configurado para su funcionamiento. Para ello ha sido necesario configurar el software y programarlo, accediendo a la documentación existente en Internet, así como al conjunto de usuarios y grupos que utilizan este software.

Para la puesta en marcha de este Firewall se han utilizado dos PC's situados en distintas localizaciones y se han hecho pruebas de comunicación a través de los equipos situados detrás de ellos. Hay que subrayar que en las pruebas con los equipos pilotos, los firewall estaban conectados a los proveedores ISP locales, mientras que en el sistema Marconi el firewall IPCop se ha situado detrás del firewall de la UCA, por lo que habido que configurar y testear.

El firewall IPCop permite hacer una red VPN así como permite utilizar el protocolo IPSec.

Se han abierto en el Firewall IP-COP y en la red UCA accesos a los sistemas de la red Marconi a los que se puede acceder desde el exterior (domicilios particulares):

- Servidor Teamspeak con aplicación Marconi para la enseñanza práctica de las radiocomunicaciones.
- Servidor Web Marconi, para guiar en los desarrollos existentes.
- Servidor WebSDR, con receptor VHF RTL-SDR, y receptor de 40m Softrock.
- Servidor QtRadio con receptor Perseus.

A su vez, se ha dejado para acceso interno desde cualquier PC de la UCA el servidor SDR-Radio con los receptores SDR-IQ, Softrock HF, Funcube y RTL-SDR.

Para poder dar señal a los equipos receptores, se ha montado el equipo necesario de antenas (todo ello desarrollado dentro del grupo S2CN).

Se han instalado en la red Marconi los siguientes equipos de Voip y Vídeo-ip:

- Centralita telefónica Trixbox (servidor SIP Marconi).
- Dos Videoteléfonos Grandstream.
- 1 Cámara Wifi Grandstream.
- 1 Cámara Grandstream (RJ45)
- Teléfonos Software : ExpressTalk, Zoiper.

Como centro de la red Ethernet se usan dos switch TP Link TL-SF1008D de 8 puertos 10/100 auto configurables.

5.2 Caminos explorados dentro de la Tesis Doctoral.

Se han establecido varias líneas de trabajo que han permitido desarrollar el demostrador de conceptos:

1. Se ha primado el uso de software libre con fines evidentes de ahorro de costes.
2. Se ha trabajado principalmente con los protocolos TCP/UDP sobre IP y SIP.
3. Se ha usado principalmente el sistema operativo Linux (Debian/Ubuntu).
4. Se ha dado preferencia al uso de Linux para los servidores.
5. Se han explorado los entornos de desarrollo más avanzados en SDR:
 - a. Entorno Web, HTML, PHP, Java.
 - b. Entorno Qt, C++, Python.
 - c. Entorno Eclipse, Android.
6. Al estar SDR basado en proceso de señal en banda base por ordenador, se ha

investigado en dos claras capacidades de las redes de ordenadores:

- a. Arquitectura Cliente/Servidor.
 - b. Uso de los dispositivos de los ordenadores: USB y Tarjetas de sonido.
7. Integración del audio y vídeo dentro de la red Marconi.
 8. Investigación de soluciones multiplataformas dentro del entorno Linux, Windows XP y Android.
 9. Utilización de cada uno de los receptores existentes dentro de las aplicaciones SDR actuales, con la construcción de los receptores Softrock, pioneros en las soluciones SDR.
 10. Investigación dentro de los grupos actuales de desarrollo SDR, primando la implementación de versiones “State of the art” y betas de los programas SDR, aún a costa de un esfuerzo adicional de puesta en marcha. El objetivo ha sido la investigación del desarrollo actual y las líneas generales de desarrollo en el futuro.
 11. Investigación de los programas existentes actualmente para el manejo de los receptores SDR, así como la utilización práctica de cada uno de ellos, probándolos con cada uno de los receptores seleccionados dentro de esta tesis, y llegando a depurar qué aplicación se puede usar adecuadamente, con qué receptor, en qué condiciones y con qué configuración.
 12. Análisis de la evolución de futuro, identificando claramente las arquitecturas sobre las que se desarrollan las aplicaciones SDR, arquitectura 386 y ARM.
 13. Estudio de ROIP partiendo de VOIP, instalando centralitas telefónicas como base del acceso mediante ellas a los equipos y antenas de Radio, explorando el puente de unión a través de las redes actuales de telefonía PSTN.
 14. Objetivo principal es compartir con la comunidad universitaria e interesados en general, para lo que se ha desarrollado una página web HTML de acceso al sistema Marconi, así como un foro de participación desarrollado con el software phpBB.
 15. Implementación de herramientas de análisis como GNU-Radio que permiten la creación, por medio de bloques, de sistemas de radiocomunicaciones.

5.3 Arquitectura de red Marconi.

El centro de la red Marconi es el Firewall IPCOP, que asigna las direcciones a cada uno de los servidores que se conectan a él.

Cada servidor de aplicaciones de la red Marconi tiene su dirección IP fija.

El acceso, desde el exterior a los servidores que lo tienen permitido, se hace a través de IP-Cop con direccionamiento de puerto a través de NAT.

El router wifi permite el acceso a través de este medio a IPCop, recibiendo cada cliente conectado a él una dirección IP DHCP del firewall Marconi.

Los teléfonos y las videocámaras, a su vez, reciben su dirección del firewall, teniendo éstos como servidor SIP la centralita telefónica TRIXBOX.

Se puede ver diagrama de conexiones en figura 5.3.1.

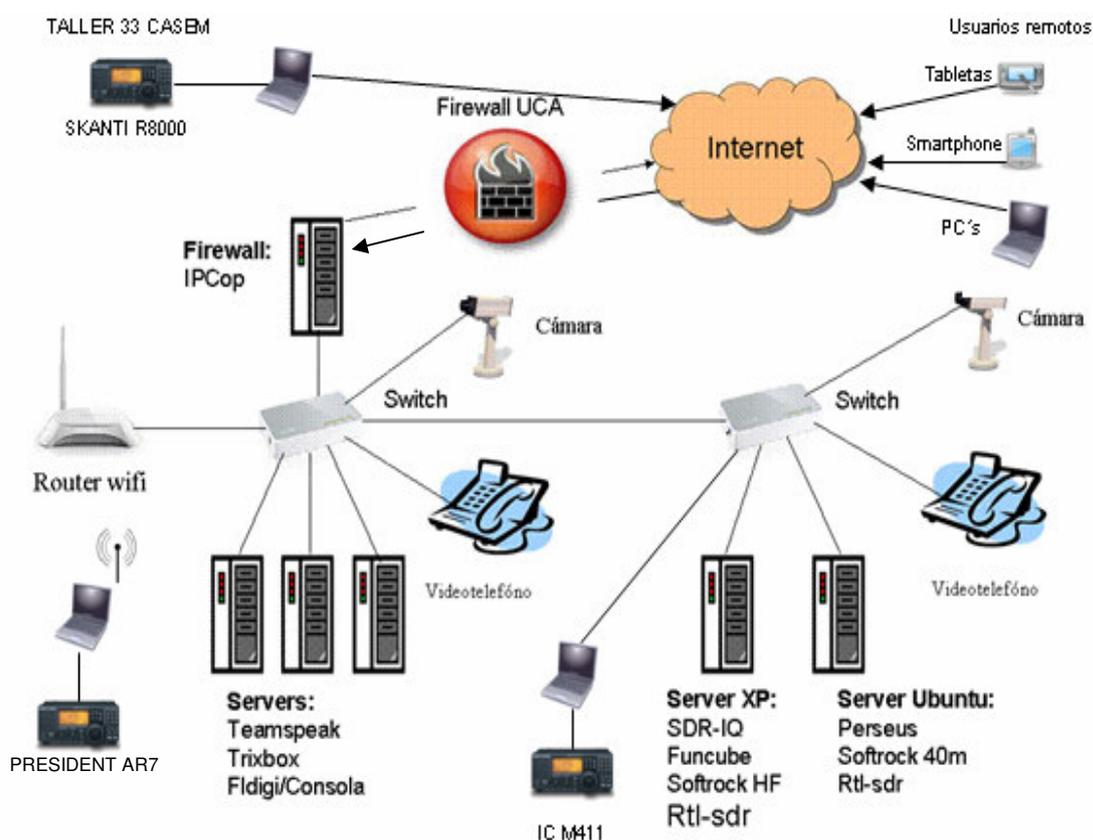


Figura 5.3.1 Arquitectura de red MARCONI.

En las figuras 5.3.2 y 5.3.3 se puede ver la distribución de equipos dentro del área de laboratorio.

Como comentario adicional, aclarar que se ha utilizado un solo monitor, teclado y ratón conectado a un conmutador para acceder a cada uno de los servidores.

En figuras 5.3.4 y 5.3.5 se muestra diagrama de bloques de las antenas, servidores y equipos del sistema Marconi.



Figura 5.3.2 Servidores Teamspeak, Trixbox, IPCop, Fldigi/GNU-Radio y Cámara/Videoteléfono.

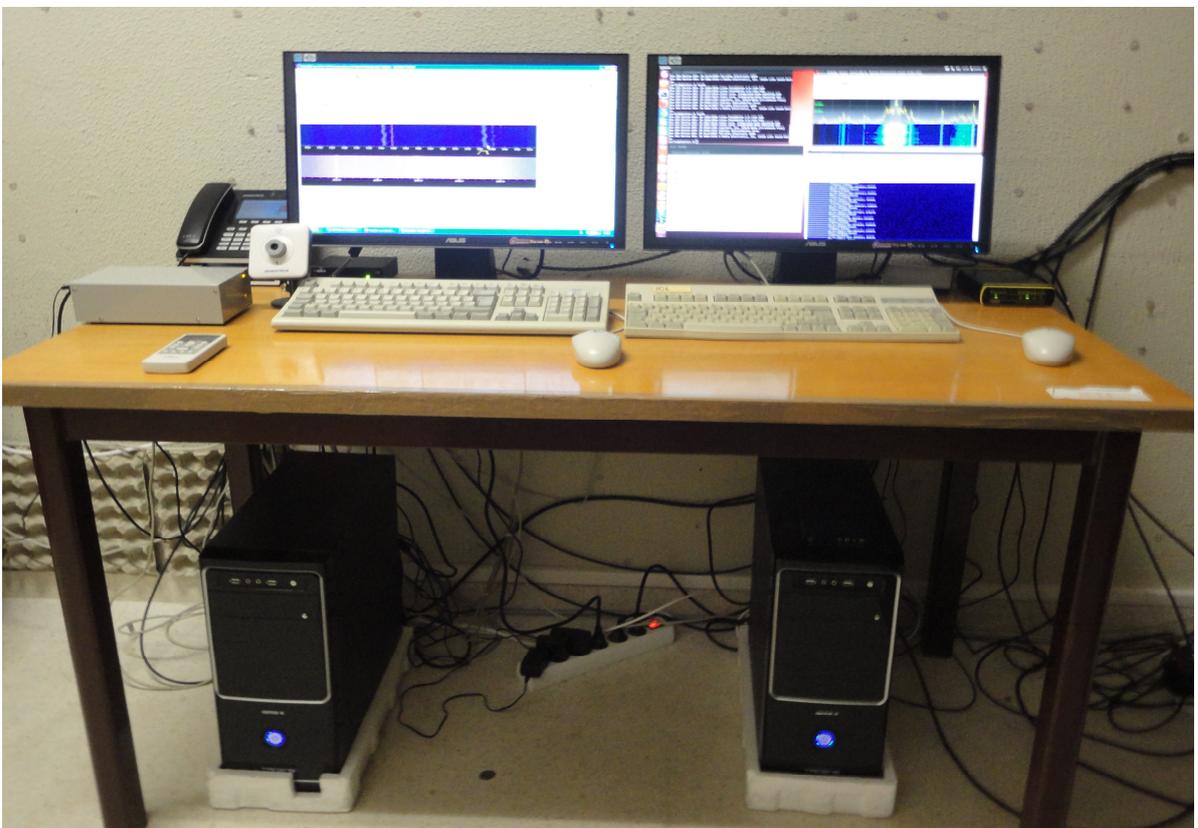


Figura 5.3.3 Servidores XP, Ubuntu equipos de recepción y Cámara/Videoteléfono.

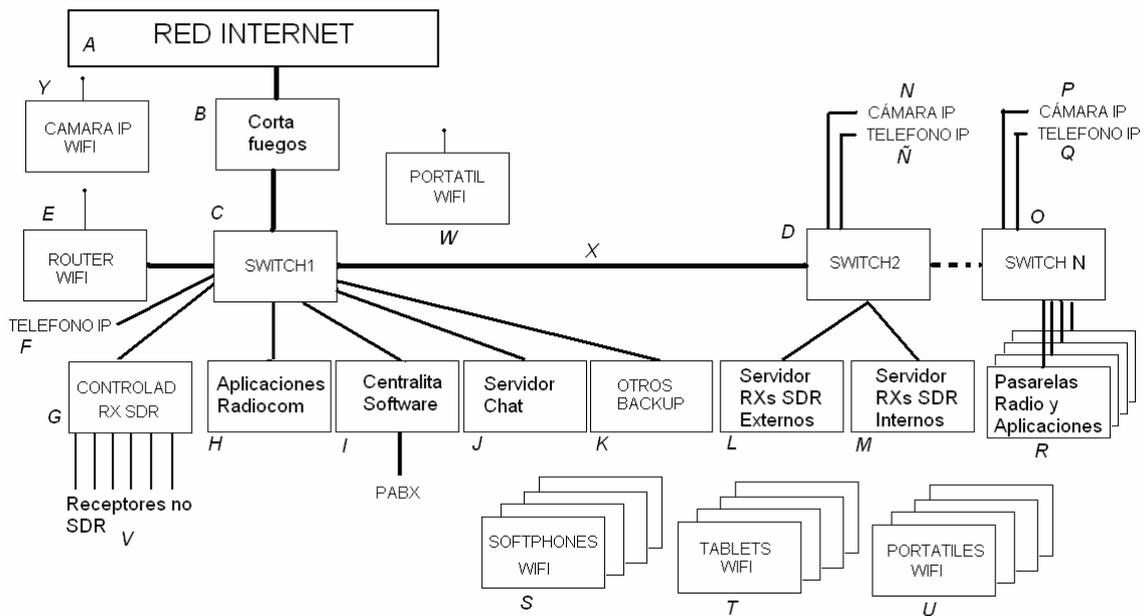


Figura 5.3.4 Sistema de Control y Distribución de señales.

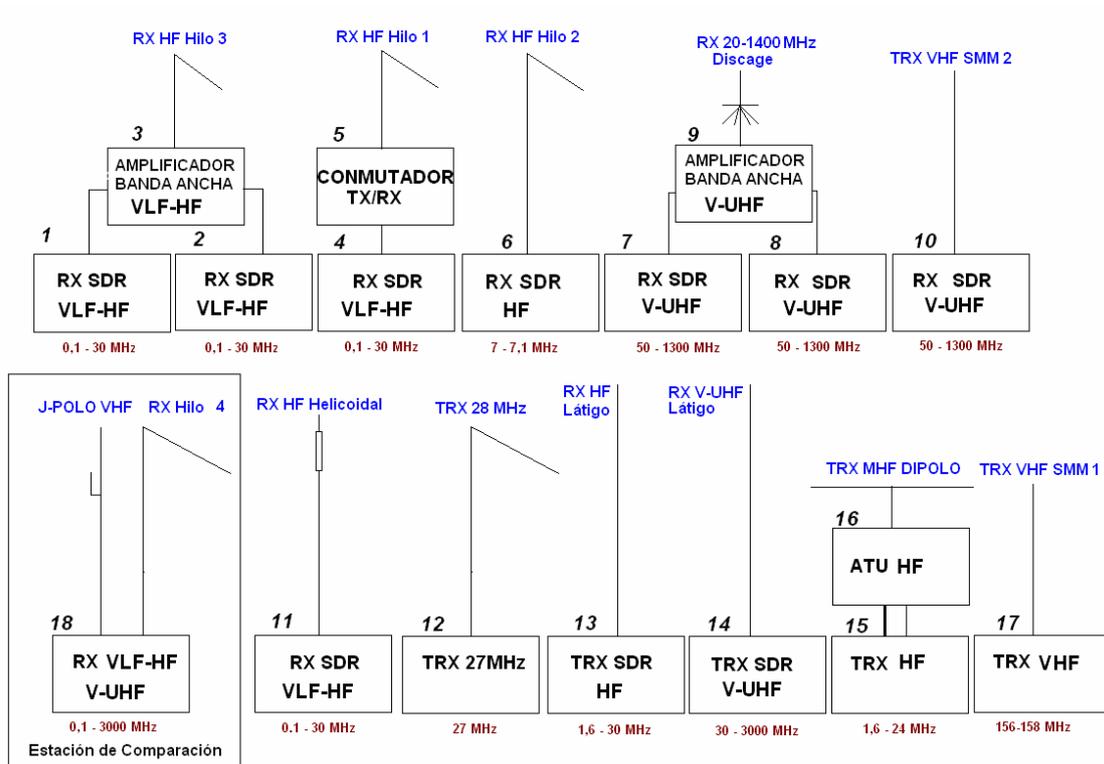


Figura 5.3.5 Cabecera de radiofrecuencia.

5.4 Servidor Teamspeak.

El servidor Teamspeak tiene acceso directo a través del switch al Firewall Marconi.

Para su correcto funcionamiento se han tenido que abrir los siguientes puertos en el Firewall Marconi y el Firewall de la Universidad:

9987	UDP
10011	TCP
30033	TCP
2008	TCP

Estos puertos se han direccionado al servidor Teamspeak en IPCop.

Si al iniciar el servidor éste no arranca adecuadamente, la razón principal es que los puertos no están abiertos.

Para el desarrollo del servidor Teamspeak, se ha pilotado un prototipo local, donde se han resuelto los problemas surgidos durante su puesta en marcha, bien a través de consulta al grupo de servidores del foro Teamspeak, o bien mediante investigación de los scripts de instalación y seguimiento de problemas a través de técnicos expertos. A su vez el servidor piloto de Teamspeak ha servido para investigar y desarrollar la aplicación del servidor Teamspeak Marconi.

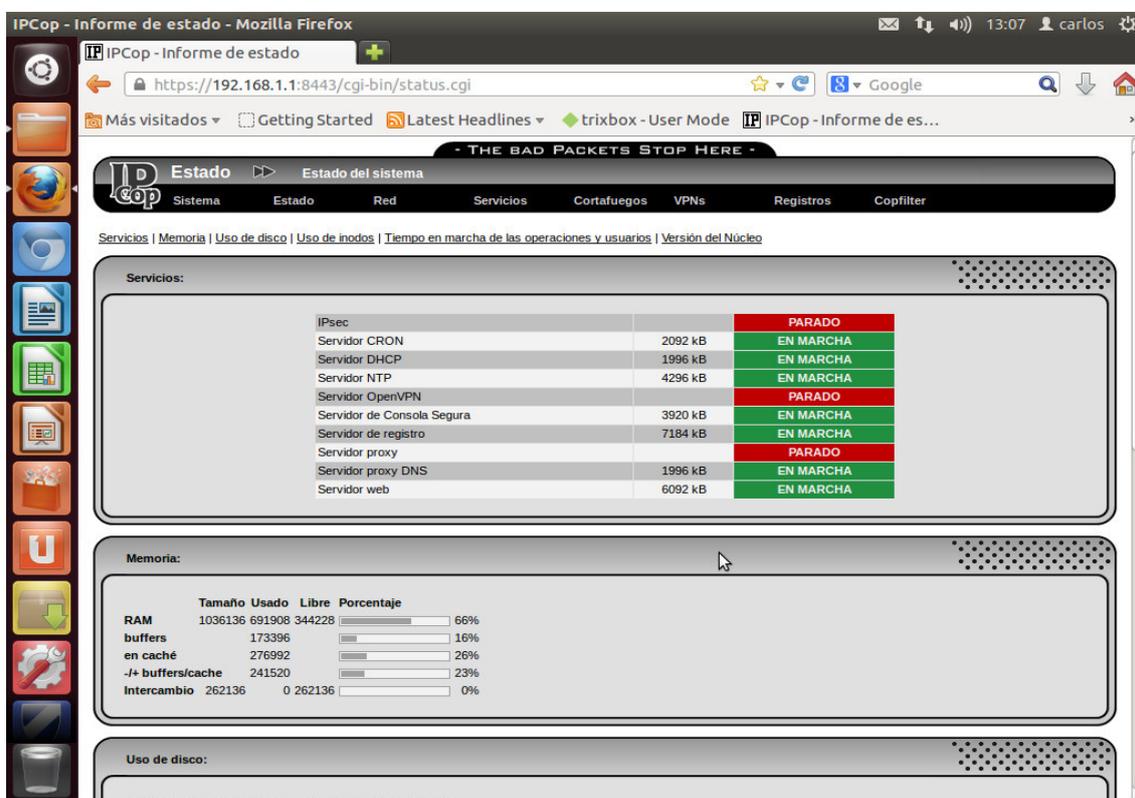


Figura 5.4.1 Firewall IPCop Marconi.

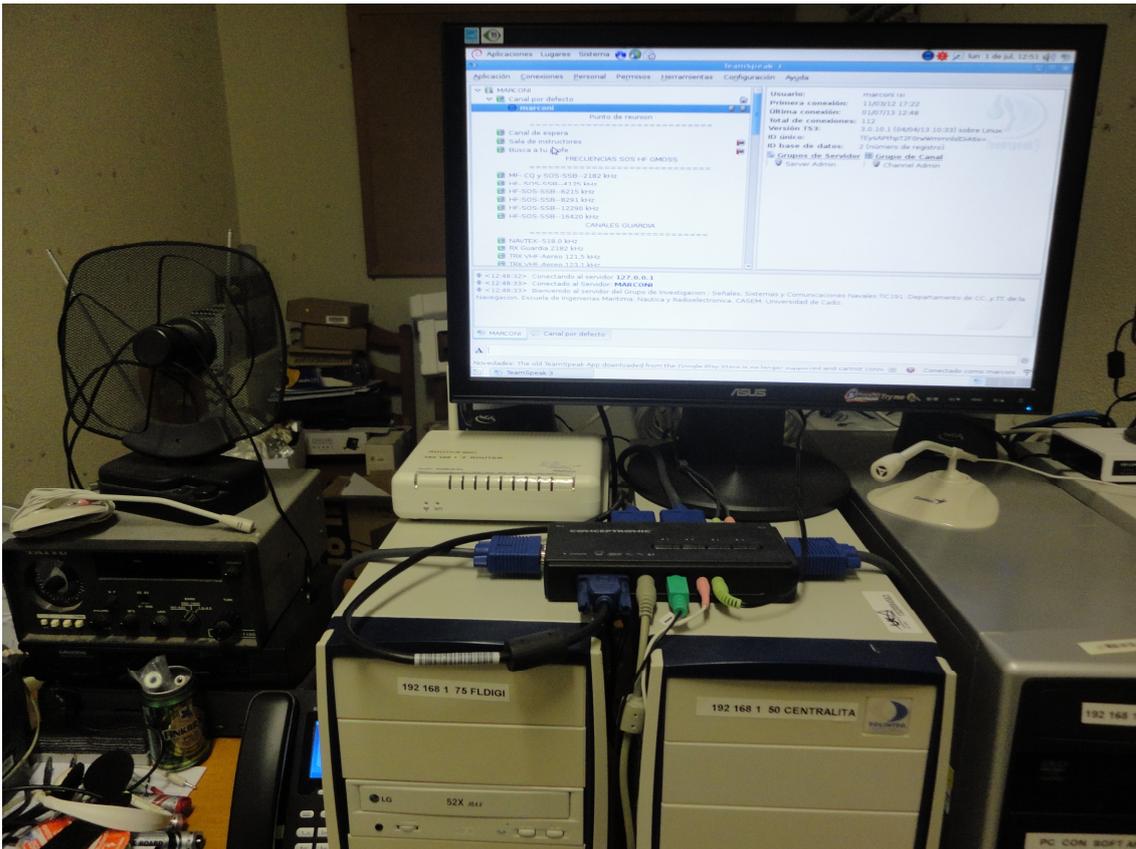


Figura 5.4.2 Pantalla principal del servidor Marconi

5.5 Servidor TRIXBOX,

El servidor TRIXBOX se encuentra directamente conectado a través del switch al IPCop.

De este último recibe su dirección de IP fija dentro del sistema Marconi.

No se ha configurado ningún puerto de acceso a él, ya que su salida está prevista por la tarjeta X100P básica con un acceso FXO a la red PSTN.

La centralita está configurada y probada su salida a través de su conexión FXO.

La otra misión fundamental de la centralita es ser servidor SIP de los equipos conectados a la red Marconi:

- Cámaras de vigilancia
- Videoteléfonos electrónicos.
- Teléfonos Software.

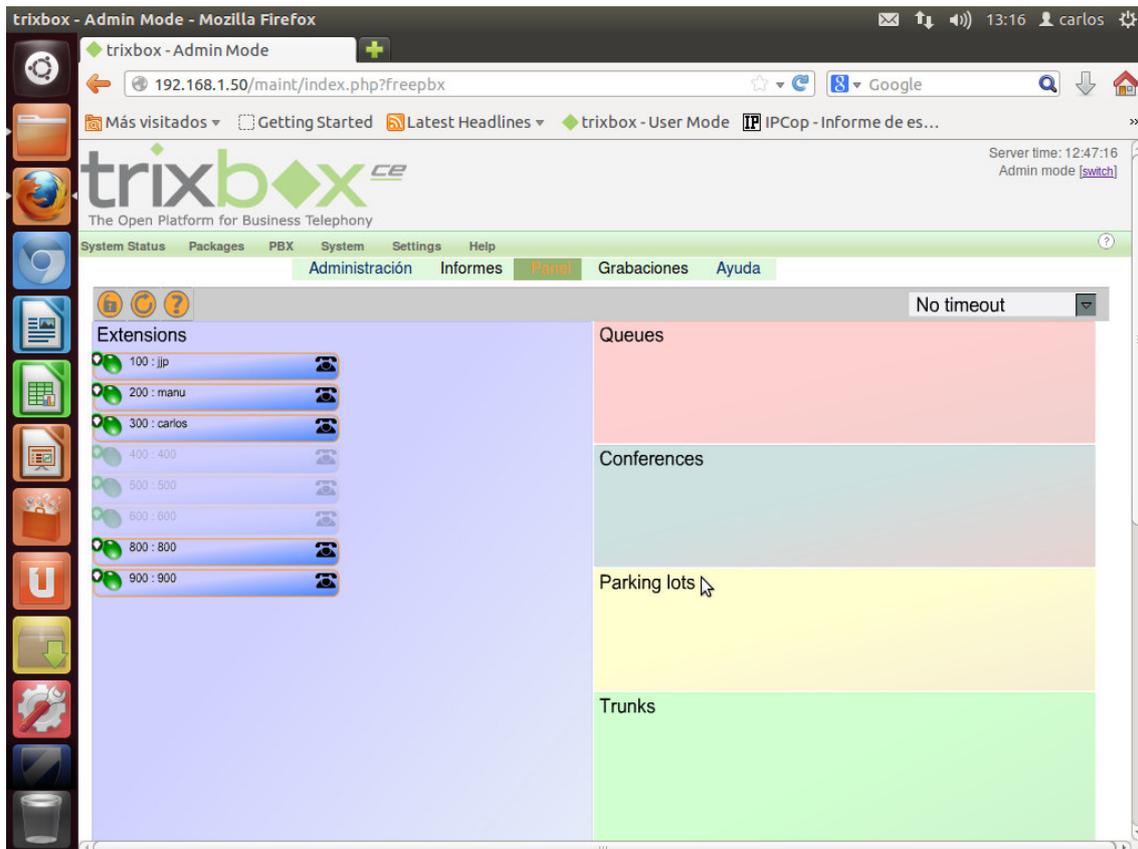


Figura 5.5.1 Panel principal TRIXBOX

En figura 5.5.1 se puede ver el panel principal de la centralita, donde se observan 5 extensiones conectadas.

- Extensión 100, softphone ExpressTalk instalado en Servidor XP.
- Extensión 800, Videoteléfono Grandstream (1)
- Extensión 900, Videoteléfono Grandstream (2)
- Extensión 200. Cámara IP (1)
- Extensión 300. Cámara IP (2)



Figura 5.5.2 Videoteléfono conectado a Trixbox (extensión 800)

En figura 5.5.2 se puede ver el Videoteléfono conectado a la centralita Trixbox, con la extensión número 800 registrada en ésta.

También se puede observar la dirección IP dada por el Firewall IPCop (192.168.1.173).

En figura 5.5.3 se puede ver teléfono software ExpressTalk, registrado en la centralita Trixbox con la extensión 100.

Se observa en figura que la dirección SIP del teléfono ExpressTalk es: 100@192.168.1.50, por lo cual, dentro de las extensiones telefónicas de la centralita marcando 100, comunicaremos con él automáticamente.

Como se ha indicado anteriormente, para poner en marcha la aplicación Trixbox se han instalado dos servidores pilotos con acceso al proveedor de IP “Diamoncard” y con la tarjetas X100P, acudiendo al foro Fonality para solucionar problemas puntuales, así como desarrollando la programación de la centralita sobre estos equipos pilotos.

La aplicación implementada es de libre uso con lo cual la instalación, programación y puesta en marcha ha requerido de la investigación de la soluciones.

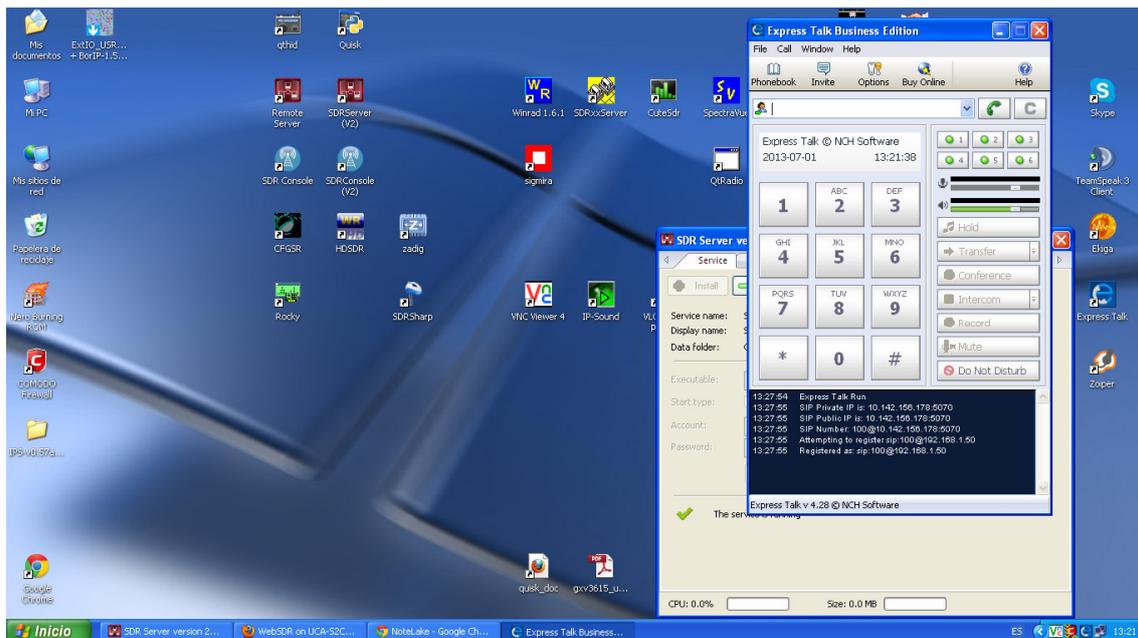


Figura 5.5.3 Teléfono software ExpressTalk conectado a Trixbox (extensión 100)

5.6 Servidores de Receptores de Radio.

Con objeto de poder utilizar los distintos programas SDR existentes actualmente, se implementan dos PC's como servidores de SDR:

El primero con el sistema operativo Windows XP.

El segundo con el sistema operativo Linux Ubuntu (12.04).

Es de destacar que aunque en cada servidor se use o bien software Windows o Linux prioritariamente, ambos tienen los dos sistemas operativos instalados y se instalaron como máquinas idénticas.

Empezaremos por describir el servidor XP.

El servidor XP, del cual vemos la imagen del desktop en figura 5.6.1, se puede considerar dividido en dos áreas:

Área 1: Situada en el lateral derecho del monitor, tiene accesos a los programas VOIP.

- Skype: Cliente Skype configurado para establecer llamadas.
- Teamspeak cliente: Cliente Teamspeak para conectar a Marconi.
- Ekiga: Cliente VOIP configurado para conectar a la centralita Trixbox.

- Express Talk: Teléfono software configurado para conectar a Trixbox.
- Zoiper: Teléfono software configurado para conectar a Trixbox.

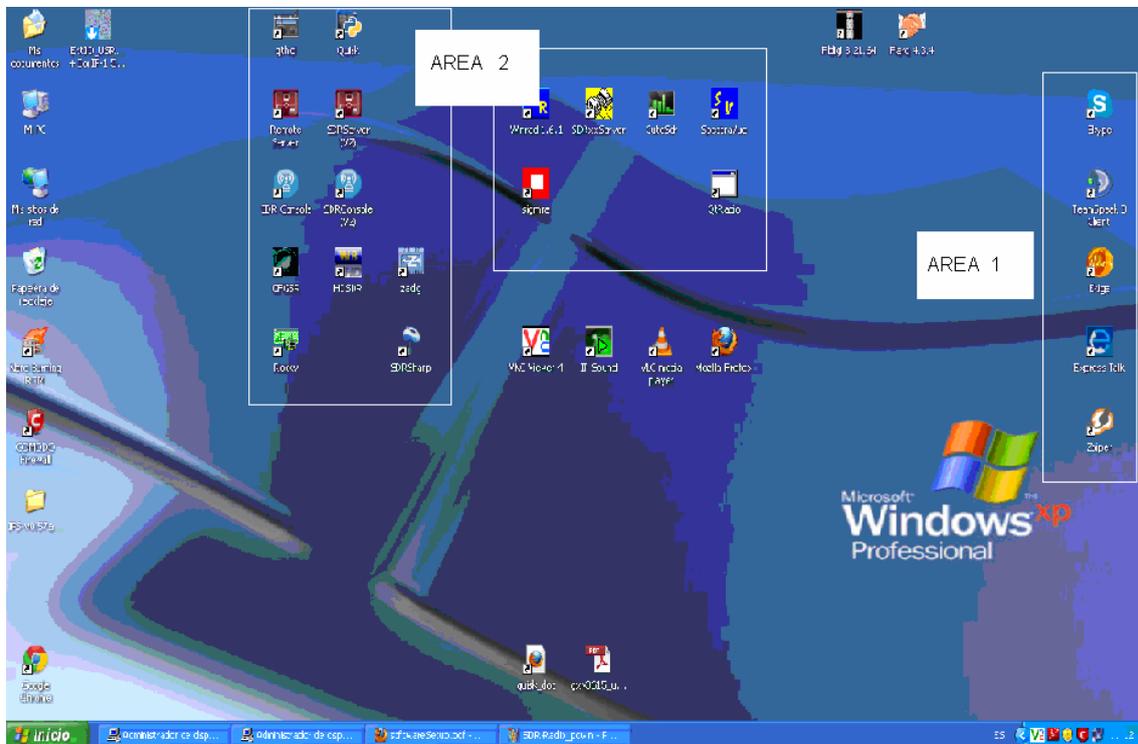


Figura 5.6.1 Desktop servidor de receptores XP.

Área 2: Contiene los programas SDR.

Vamos a describir de una forma un poco más detallada esta área.

Se han implementado en esta área, dos tipos de programas SDR:

El primer tipo es aquel que maneja el equipo instalado donde reside el software:

- Qthid y Quisk para receptor Funcube Dongle
- GFGSR y Rocky para Softrock.
- HDSDR para SoftRock, SDR-IQ, Funcube Dongle, RTL-SDR.
- Winrad para Softrock, SDR-IQ, Funcube Dongle.
- SDRSharp para SDR-IQ, Funcube Dongle, RTL-SDR.
- Spectravue para SDR-IQ.

El segundo tipo es aquel que implementa la arquitectura cliente/servidor:

- SDRxxServer y CuteSDR para SDR-IQ
- QtRadio cliente para conexión a servidores QtRadio.
- Sigmira cliente para conexión a servidores SDR-Radio
- SDR-Radio Server y Console para acceso a Softrock, Funcube, RTL-SDR, SDR-IQ tanto locales como remotos, en este último caso conectado a través de servidor.

Vamos a mostrar a continuación imágenes de uso de cada uno de estos programas en el servidor XP.

Todos los programas indicados anteriormente están configurados y en funcionamiento.

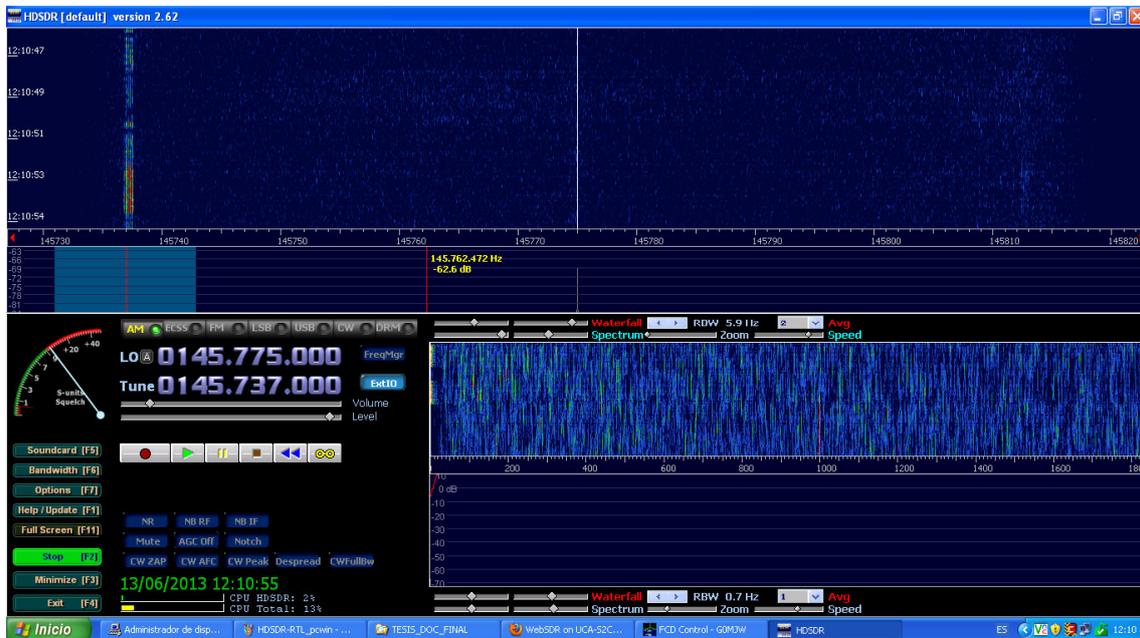


Figura 5.6.2 HSDR y receptor Funcube Dongle

En figura 5.6.2 se ve el programa HSDR conectado al receptor Funcube.

Una vez que presionamos en el icono HSDR del Servidor XP, el programa nos dará opción a elegir la interfaz.

Se elige la que corresponde a Funcube Dongle, ExtIO_FCDPLUS_G0MJW.dll, a continuación podemos empezar a usar el receptor Funcube Dongle con HSDR.

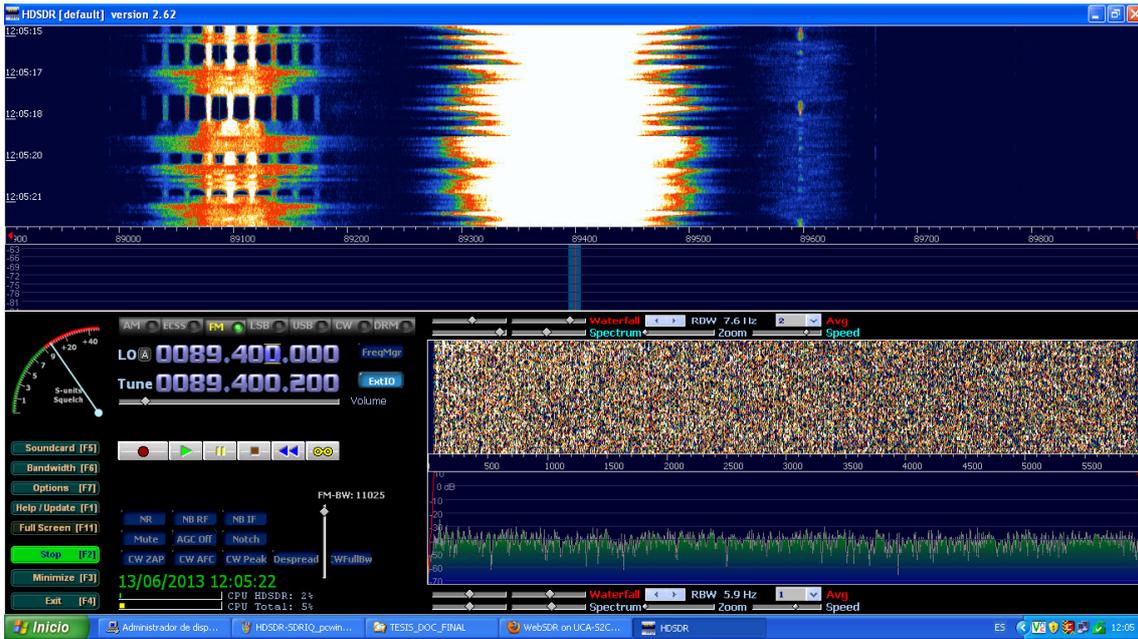


Figura 5.6.3 HSDR y receptor RTL-SDR.

En figura 5.6.3 vemos el programa HSDR con el receptor RTL-SDR.

Hemos de seleccionar el programa HSDR en el Servidor XP y posteriormente escoger la interfaz ExtIO_USRP.dll. A continuación podemos empezar a usar el receptor RTL-SDR.

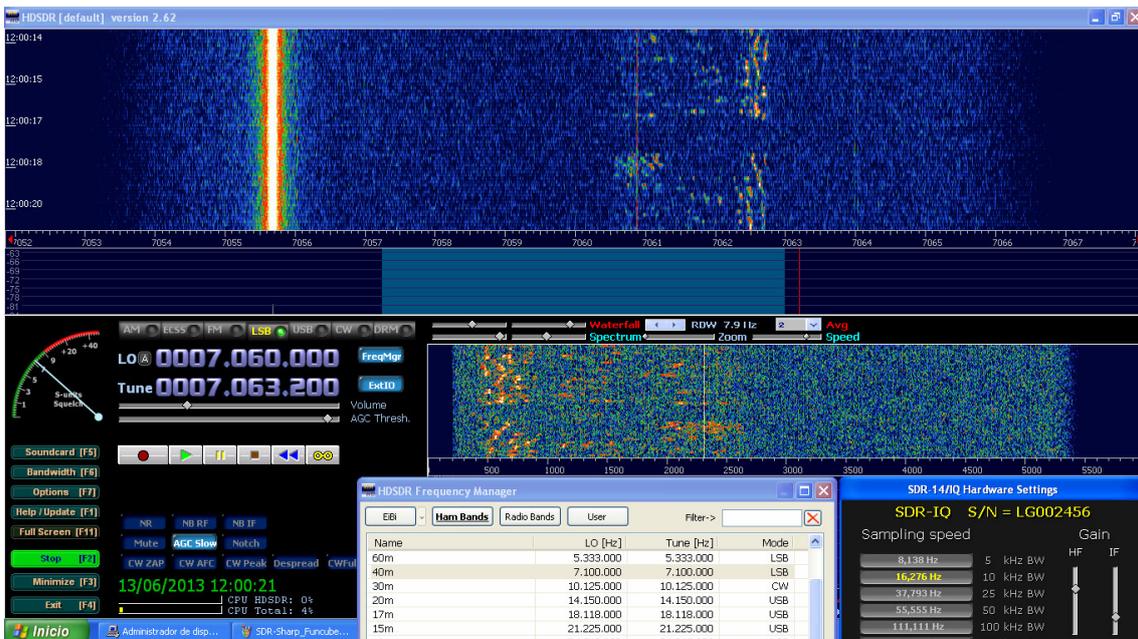


Figura 5.6.4 HSDR y receptor SDR-IQ

En figura 5.6.4 tenemos el software HSDR usando el receptor SDR-IQ.

Hemos de seleccionar el programa HSDR en el Servidor XP, y a continuación seleccionar la interfaz ExtIO_SDR14.dll. Seguidamente podemos manejar el receptor SDR-IQ con el software HSDR.

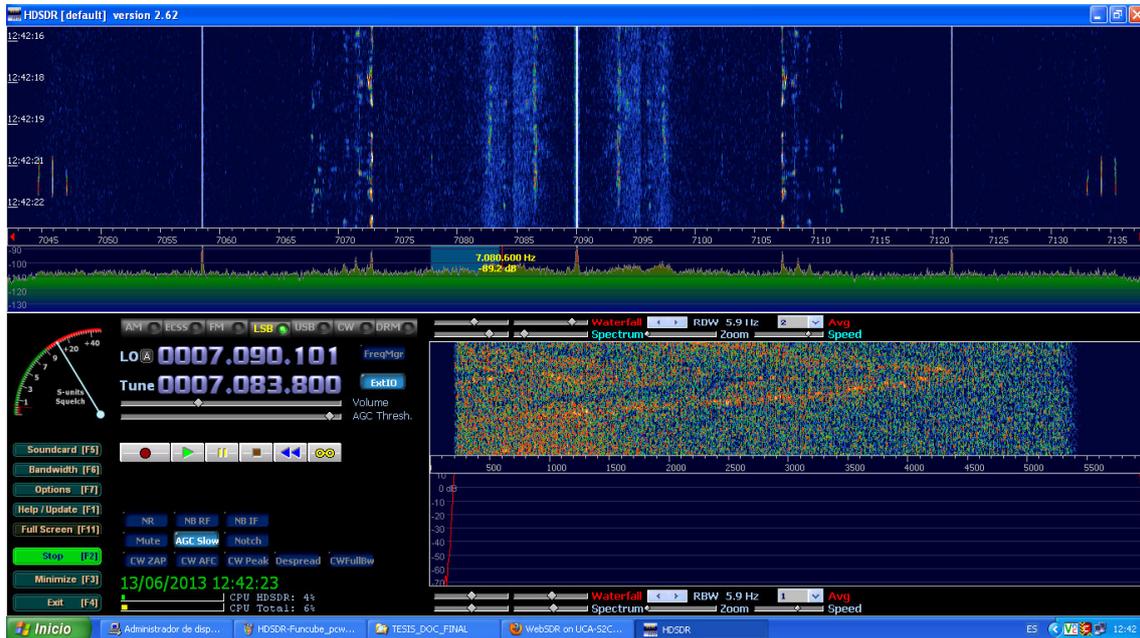


Figura 5.6.5 HSDR y receptor Softrock HF

En la figura 5.6.5 tenemos el software HSDR con el receptor Softrock HF construido por Tony Park para esta Tesis.

Hemos de abrir el programa HSDR presionando en el icono del Servidor XP, y a continuación, seleccionamos el interfaz ExtIO_Si570.dll, seguidamente podemos manejar el receptor Softrock HF con HSDR.

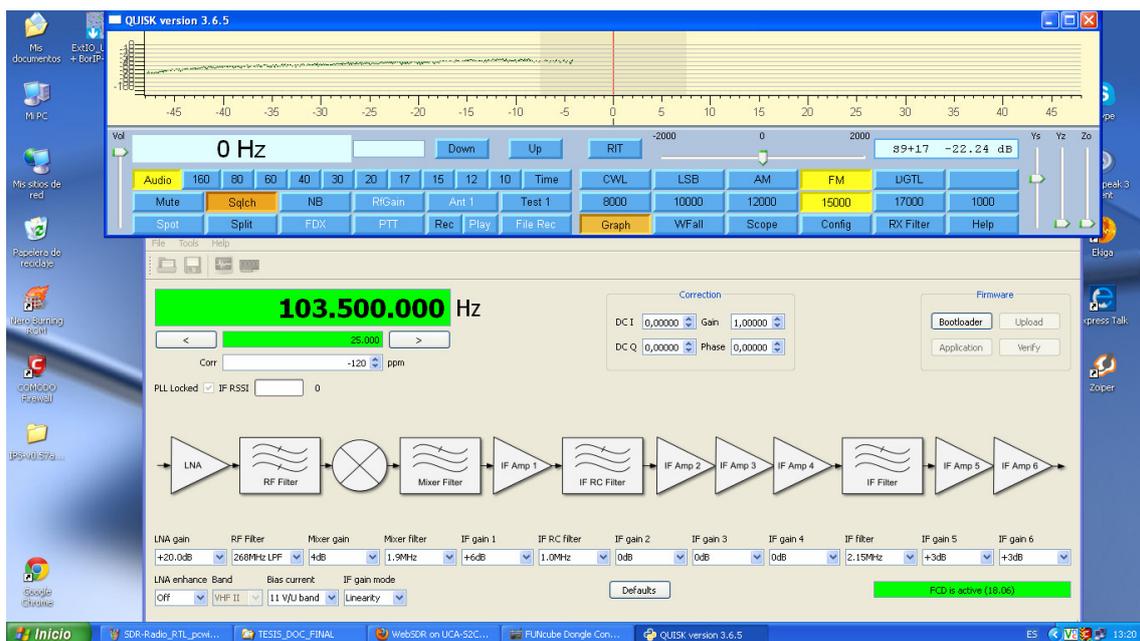


Figura 5.6.6 Qthid Quisk y receptor Funcube Dongle

En figura 5.6.6 tenemos el clásico programa de SDR “Quisk”, para manejo del receptor Funcube Dongle. En este programa se pueden variar los parámetros del receptor Funcube Dongle como si fuera un receptor analógico, lo que permite hacer prácticas de Radiotecnia II a bajo coste.

El programa Qthid permite sintonizar la frecuencia central del receptor Funcube. A su vez, el programa Qthid se puede usar para actualizar el software del propio receptor Funcube.

Hemos de abrir el programa Qthid desde el icono del servidor XP, a continuación abrimos el programa Quisk y seleccionamos la frecuencia deseada.

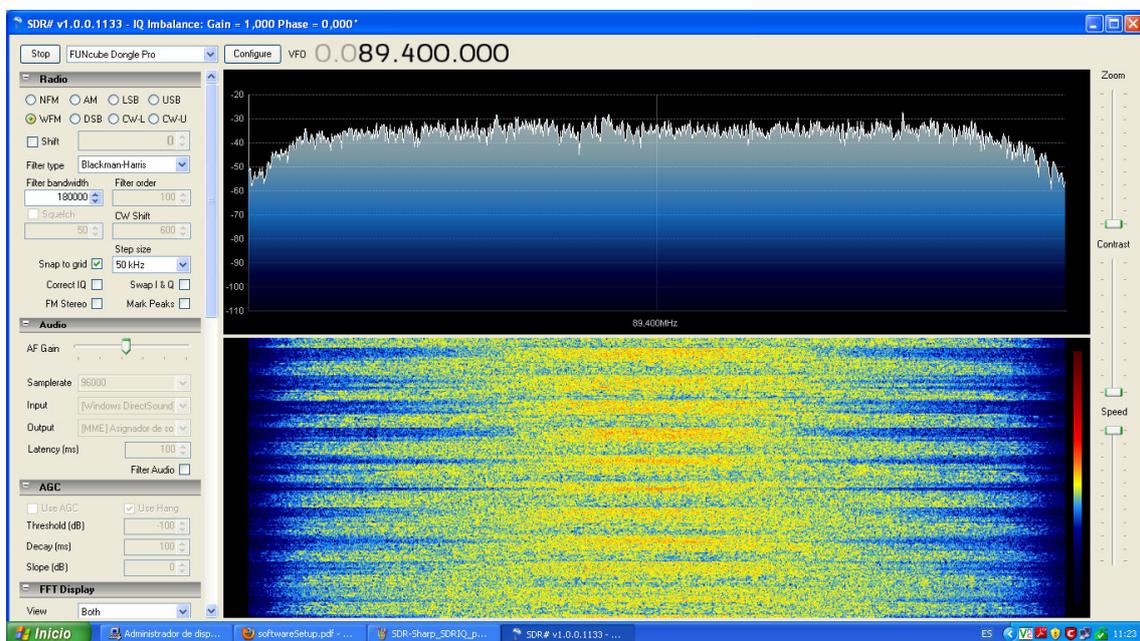


Figura 5.6.7 SdrSharp y receptor Funcube Dongle

En figura 5.6.7 vemos el programa SdrSharp manejando el receptor Funcube Dongle. Para abrirlo hemos de seleccionar el icono del programa SdrSharp en el servidor XP.

Una vez se ha abierto el programa, se selecciona la opción Funcube Dongle pro. A continuación, seleccionamos la banda y la frecuencia deseadas.

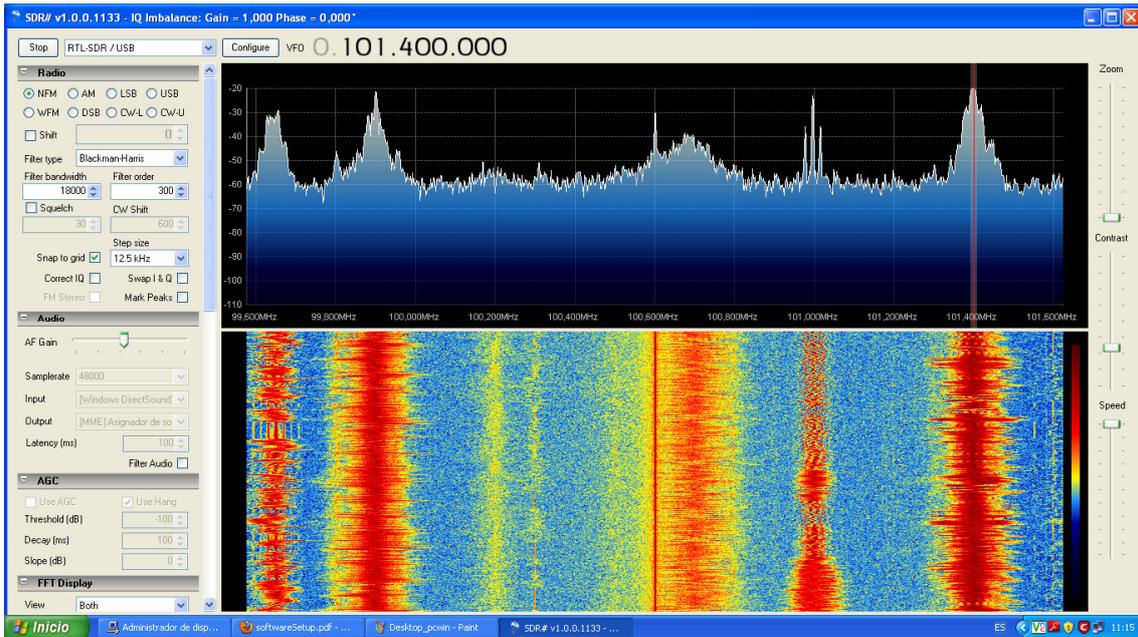


Figura 5.6.8 SdrSharp y receptor RTL-SDR.

En figura 5.6.8 podemos ver el programa SdrSharp usando el receptor RTL-SDR. Para ello abrimos el programa SdrSharp picando en su icono del servidor XP. Dentro del programa SdrSharp seleccionamos RTL-SDR, seguidamente seleccionamos banda y frecuencia.

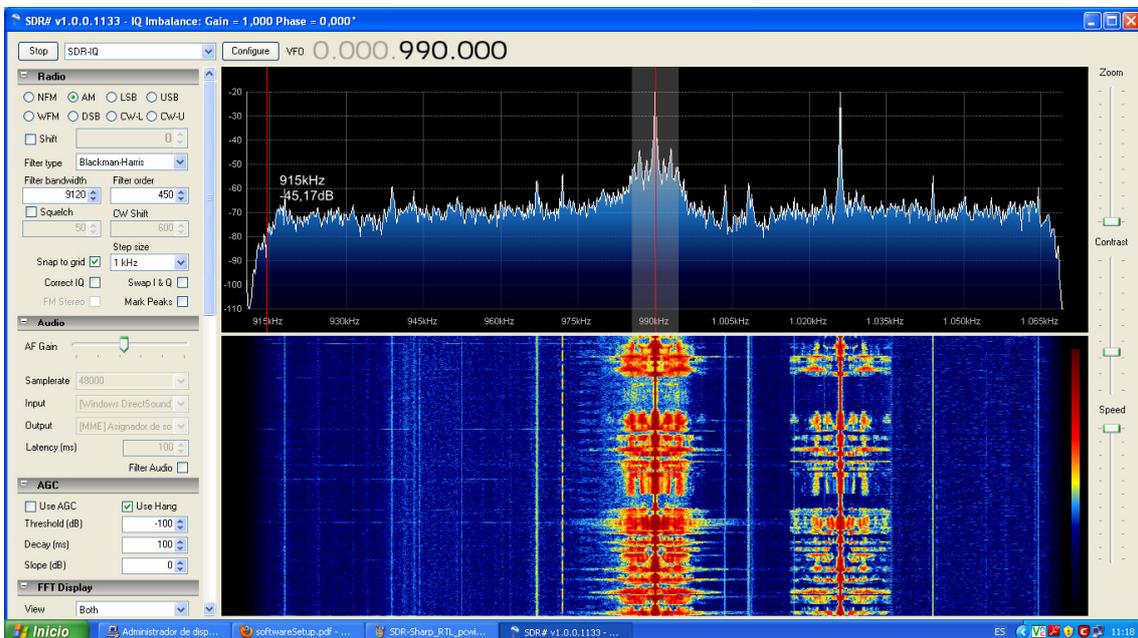


Figura 5.6.9 SdrSharp y receptor SDR-IQ

En figura 5.6.9 vemos el programa SdrSharp usando el receptor SDR-IQ, para ello abrimos el programa como se ha hecho anteriormente y se selecciona el receptor SDR-IQ.

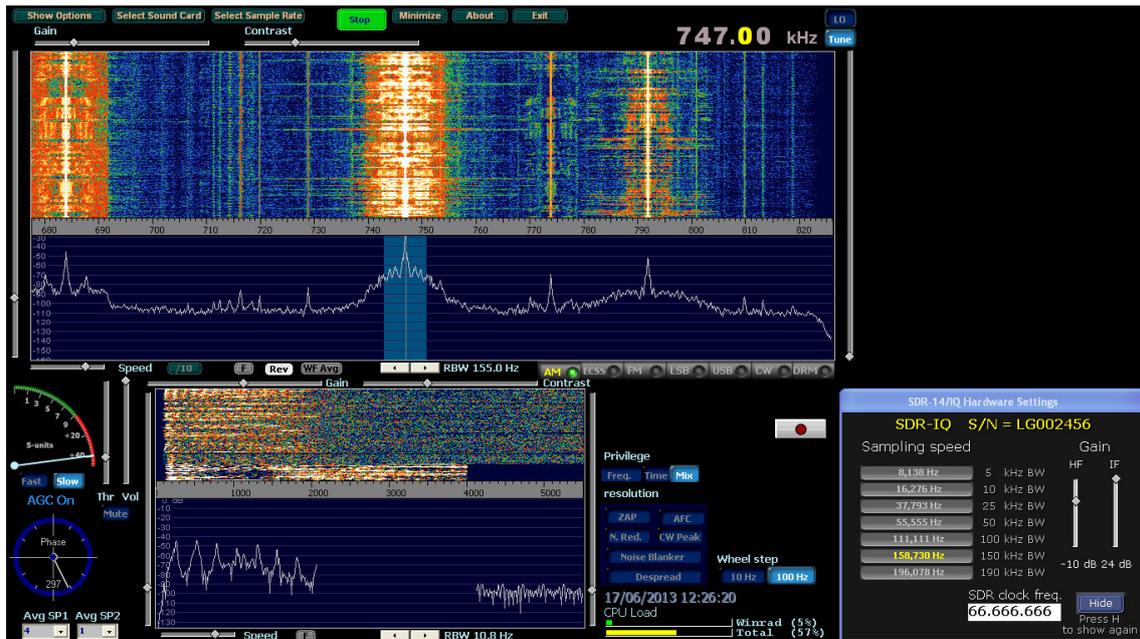


Figura 5.6.10 Winrad y receptor SDR-IQ

En figura 5.6.10 podemos ver el programa Winrad. El programa Winrad es uno de los primeros software SDR desarrollados. Actualmente se ha reprogramado y ha sido sustituido por el software HSDR.

En la figura 5.6.10 se ve el programa Winrad usando el receptor SDR-IQ.

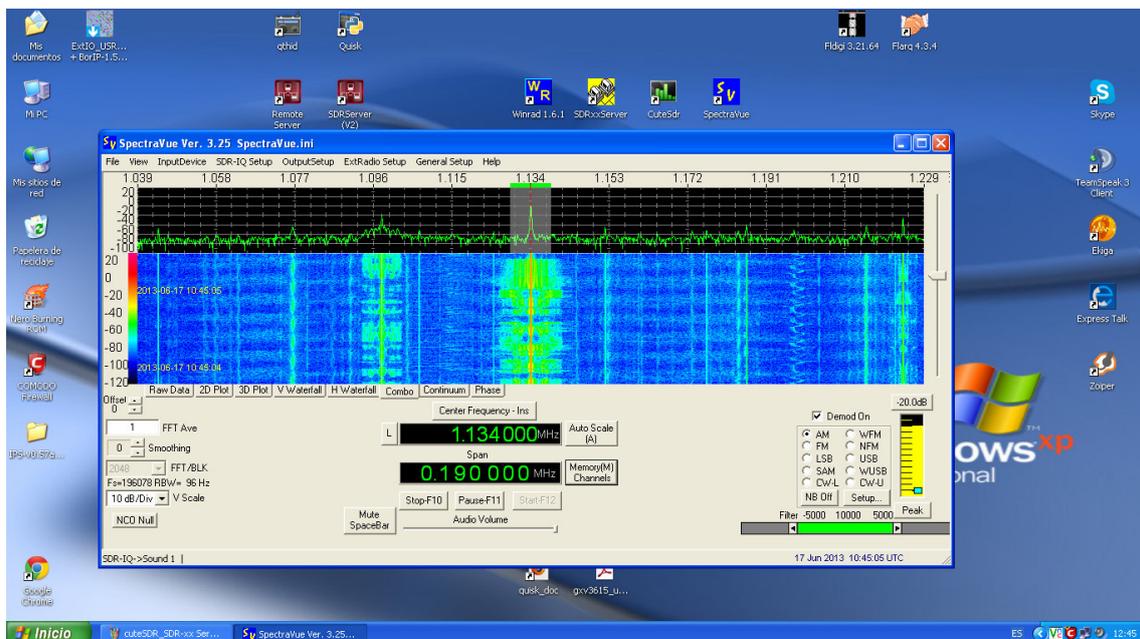


Figura 5.6.11 Spectravue y receptor SDR-IQ

En figura 5.6.11 tenemos el software Spectravue, que viene con los programas de

instalación del receptor SDR-IQ.

Tiene unas opciones muy básicas, pero es suficiente para usar y sintonizar el receptor SDR-IQ. Muestra la clásica cascada, así como opciones intuitivas que nos permiten manejar este receptor.

Hasta ahora, hemos usado los programas SDR para acceder a los receptores y manejarlos, estando éstos conectados a los puertos USB o Ethernet del ordenador donde se encuentra el programa SDR.

La labor fundamental dentro de esta tesis doctoral ha sido el análisis de los programas existentes. Se ha estudiado el manejo adecuado por éstos de los receptores SDR y a su vez se han configurado dentro de Windows los interfaces necesarios para que estos programas funcionen de forma transparente.

Se ha investigado y documentado tal como se ha expuesto anteriormente, qué receptores maneja cada programa y con qué drivers.

La información aquí recogida se encuentra dispersa y desorganizada en Internet.

Existe otra forma de usar los programas SDR.

Algunos de estos programas disponen, por un lado, de una parte servidor, que maneja los receptores y los ponen a disposición de sus clientes, y por otro lado de una parte cliente, que permite al cliente estar situado dentro de cualquier otro ordenador de la red intranet o internet.

El cliente se ha de poder conectar remotamente al servidor, a su vez debe tener concedido acceso dentro del servidor, con usuario y clave de acceso definidos en éste.

En el servidor XP hay tres programas que pueden funcionar con la arquitectura cliente / servidor:

- Sigmira
- SDR-Radio
- SDRxxServer / CuteSDR

Sigmira es un software cliente-servidor creado por Steven A. Harlow [HARL11].

Puede manejar localmente receptores SDR-IQ o SDR-14 y puede acceder en red a servidores SDR-Radio.

Para la configuración e instalación de Sigmira se ha acudido a la web de Steven y se le

ha preguntado a través de correo electrónico aquellos puntos que no funcionaban adecuadamente, bien por falta de descripción o error en su web.

El programa SDR-Radio utilizado es la opción beta que permite el uso del receptor RTL-SDR mediante la instalación de las DLL creadas por Michael Morgan (AA5SH).

Las versiones beta de SDR-Radio las libera su desarrollador Simon Brown, permitiendo la colaboración en su proyecto mediante la web sdr-radio.com.

En figuras 5.6.12, 5.6.13 y 5.6.14 se puede ver el programa Sigmira.

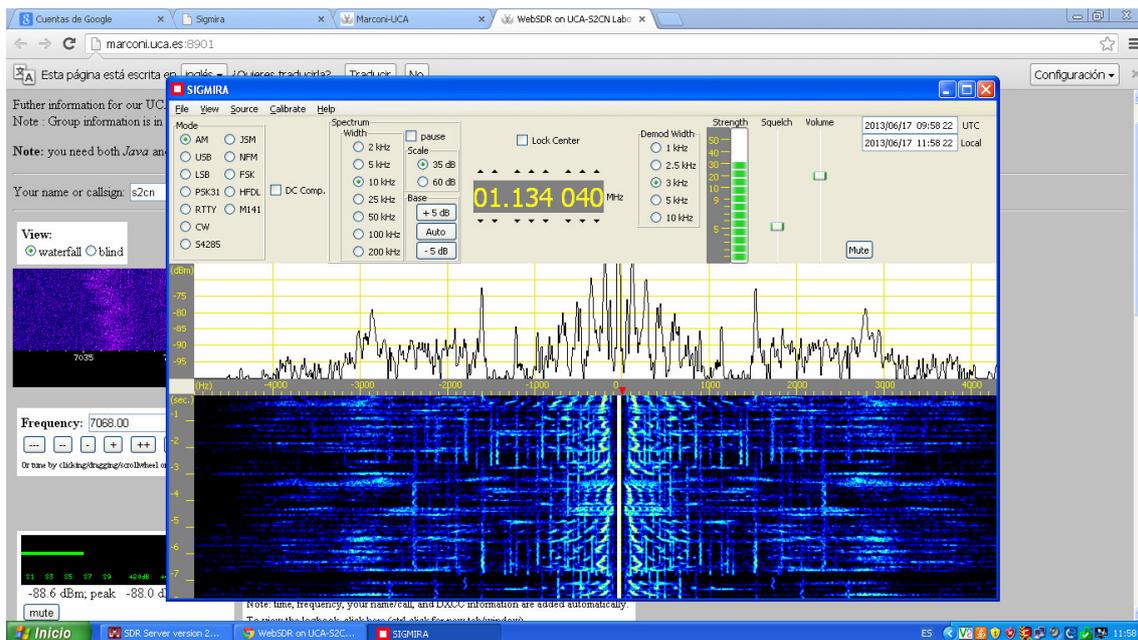


Figura 5.6.12 Sigmira con receptor local SDR-IQ (sobre pantalla WebSDR)

En figura 5.6.12 tenemos imagen del programa Sigmira conectado al servidor local SDR-IQ.

Para abrir el programa hemos de pulsar en su icono situado en el servidor XP. Seguidamente se selecciona el receptor Sigmira.

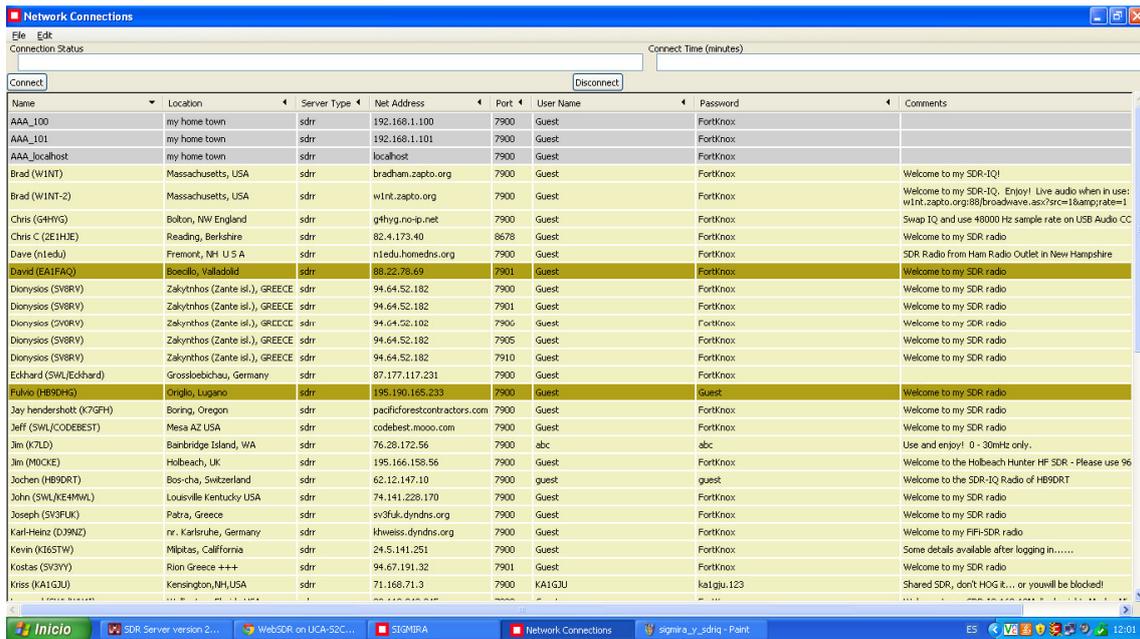


Figura 5.6.13 Sigmira pantalla de servidores.

Para conectar a servidores SDR-Radio se selecciona en la pantalla principal de Sigmira la opción Source.

En la figura 5.6.13 aparece la imagen con la relación de servidores SDR existentes en internet.

Una vez seleccionado uno de los servidores, se pulsa la opción “connect” [HARL11].

Con esto, se recibe la señal del servidor remoto.

Seguidamente se puede manejar este receptor.

En figura 5.6.14 se observa el cliente Sigmira conectado a Servidor remoto.

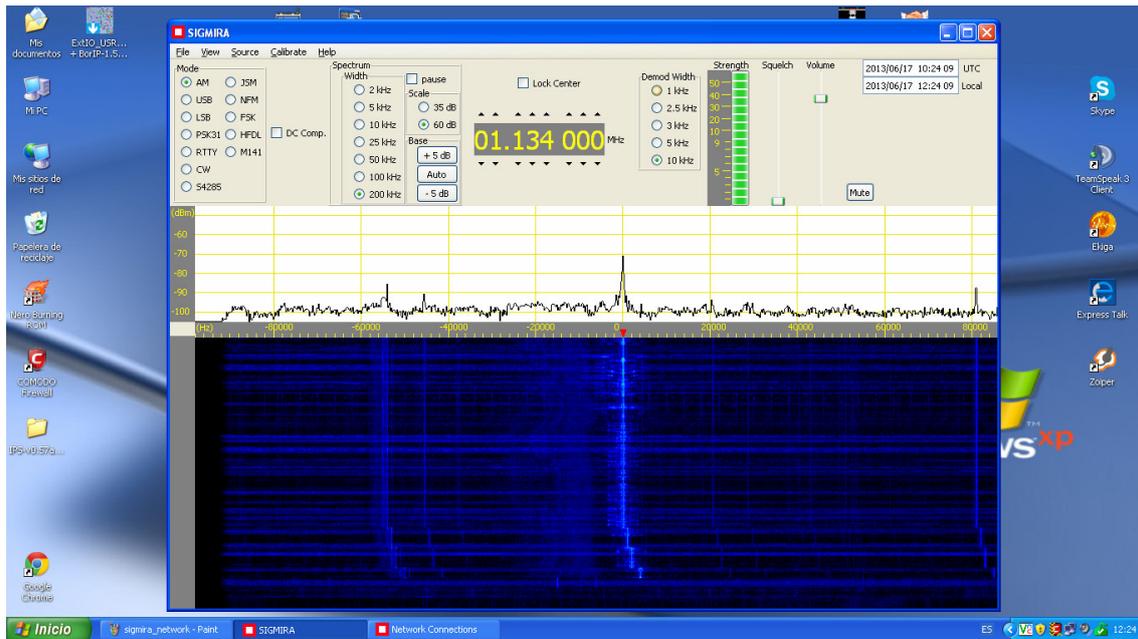


Figura 5.6.14 Sigmira conectado a servidor SDR-Radio con receptor SDR-IQ.

Uno de los mejores software cliente-servidor actuales es SDR-Radio. Este programa ha sido desarrollado por Simon Brown [BROW09] y trabaja con receptores Rfspace como SDR-IQ o SDR-14.

Dentro de esta tesis se ha investigado la opción beta actualmente en desarrollo y que permite el manejo de los receptores SDR-IQ, RTL-SDR, Funcube y Softrock.

SDR-Radio ofrece una interfaz y un software bien elaborados, que permite trabajar de una forma intuitiva y en entorno Windows.

Las instrucciones cliente-servidor se desarrollan en el entorno HTML.

Existe un club de usuarios bastante amplio, así como una web bien documentada que da soporte a las distintas versiones que van saliendo de este programa.

SDR-Radio tiene dos programas:

- Console (Consola, Cliente).
- Servidor

En esta Tesis Doctoral se ha implementado el servidor SDR-Radio en el “Servidor XP”, habilitando este servidor el acceso a los 4 receptores SDR conectados a él:

- Softrock HF
- Funcube Dongle Pro
- RTL-SDR

- SDR-IQ

A su vez, se han creado los usuarios de servidor:

- Carlos
- Juan
- Usuario1 a usuario10

En figura 5.6.16 se puede ver la pantalla principal del servidor, donde se observan los receptores que se ponen a disposición de cada cliente que se conecta a él.

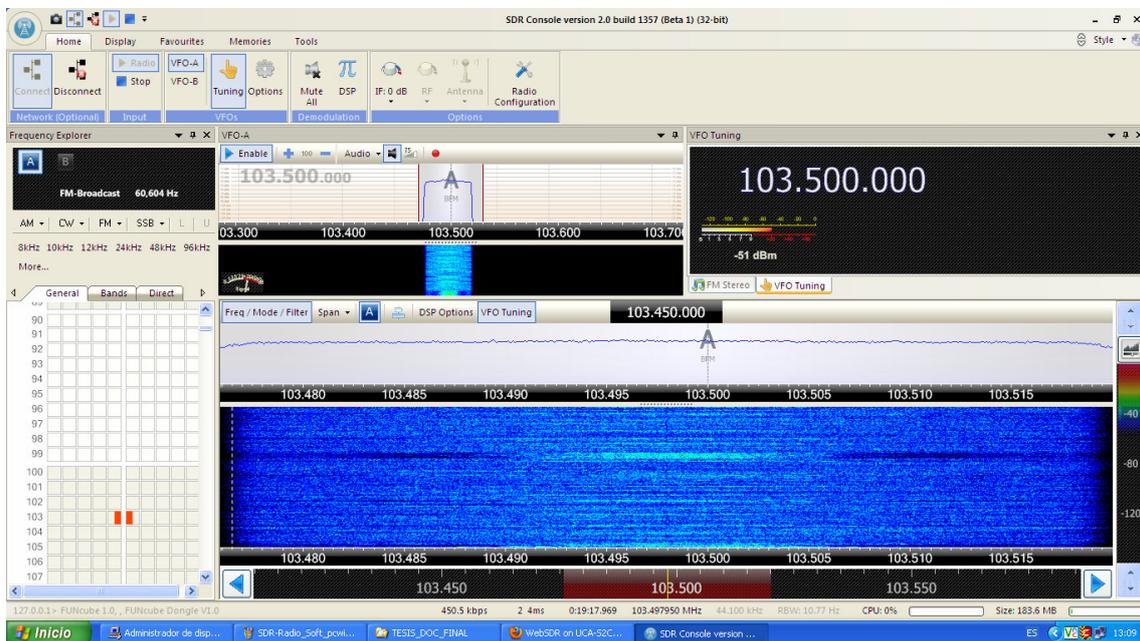


Figura 5.6.15 Cliente SDR-Radio conectado a servidor y usando receptor Funcube

En figura 5.6.15 se observa el Cliente SDR-Radio conectado a un servidor SDR-Radio y compartiendo un receptor Funcube Dongle pro.

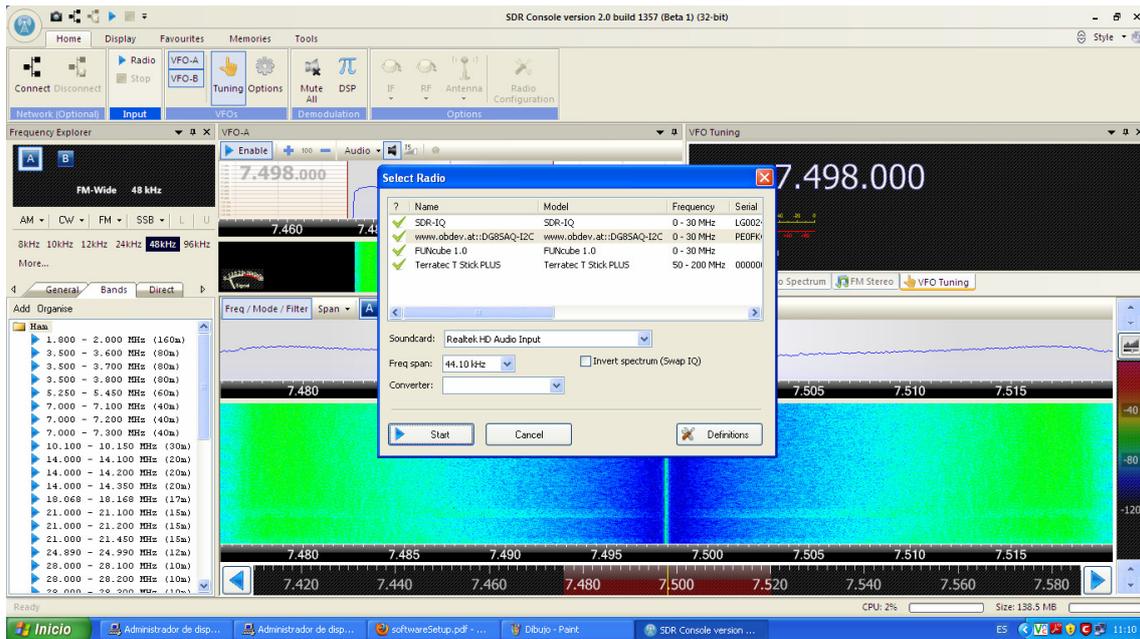


Figura 5.6.16 SDR-Radio Server, pantalla de selección de receptor

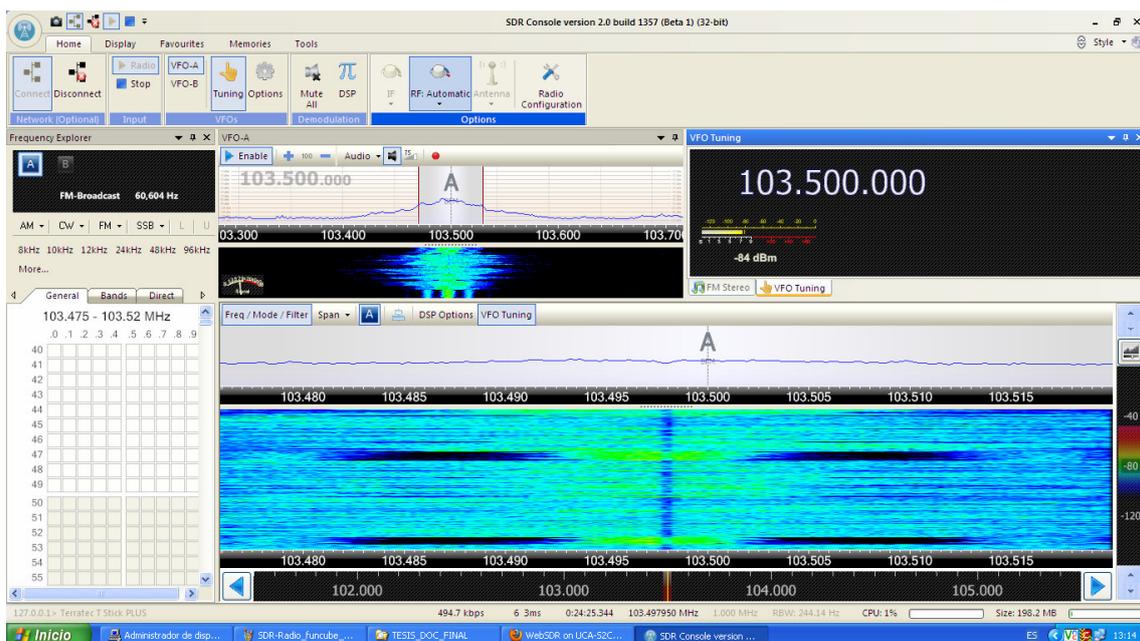


Figura 5.6.17 Cliente SDR-Radio usando receptor RTL-SDR a través de servidor

En figura 5.6.17 se observa cliente SDR-Radio conectado al servidor y compartiendo receptor RTL-SDR.

En figura 5.6.16 se observa detalle del servidor que provee los receptores.

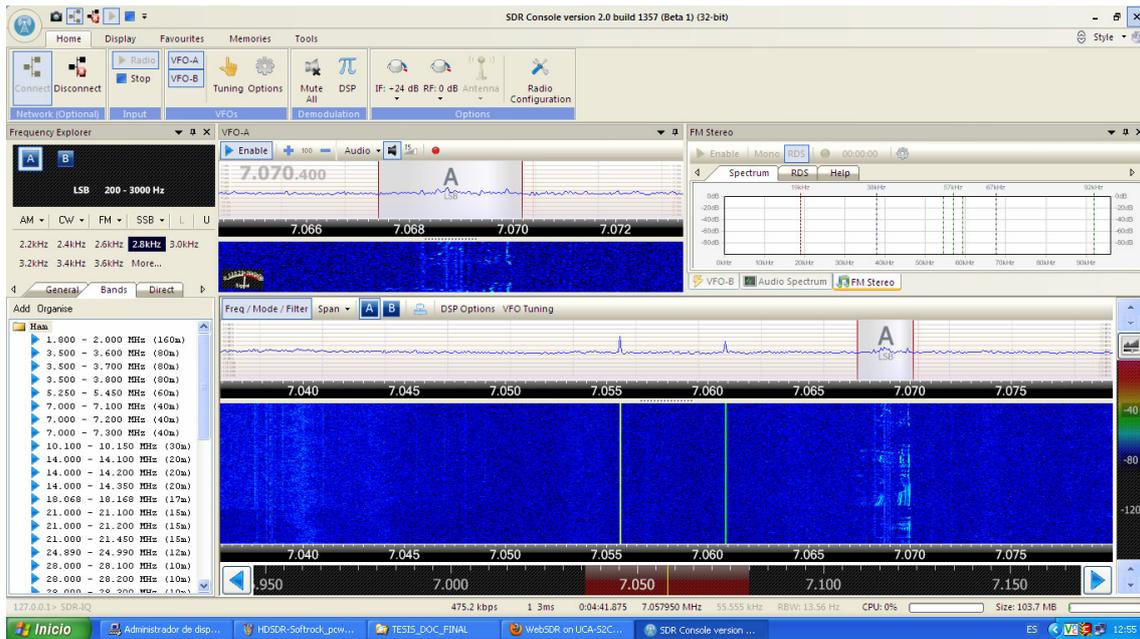


Figura 5.6.18 Cliente SDR-Radio usando receptor SDR-IQ a través de servidor

En figura 5.6.18 se observa detalle de cliente SDR-Radio conectado a servidor, usando receptor SDR-IQ de Rfspace.

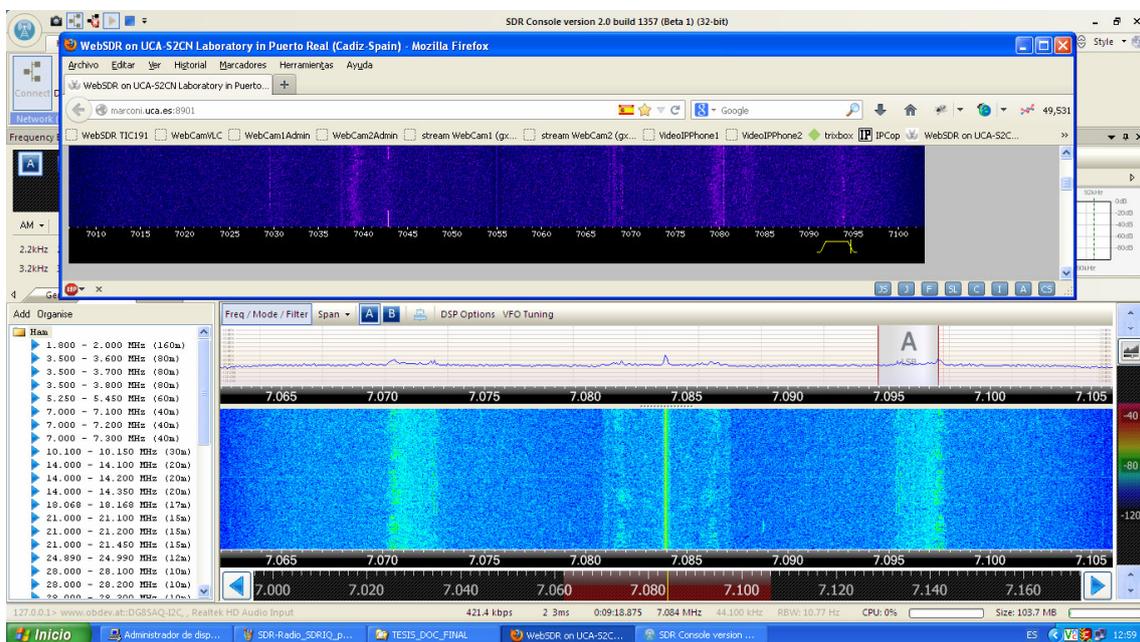


Figura 5.6.19 Cliente SDR-Radio usando receptor Softrock a través de servidor (detalle de WebSDR con Softrock)

En figura 5.6.19 se muestra detalle de cliente SDR-Radio, recibiendo datos de receptor Softrock HF, en la banda de 40 metros.



Figura 5.6.20 Cliente CuteSDR usando SDR-IQ a través de SDRxxServer

Por último vamos a comentar el software cliente-servidor que ofrece RfSpace con su receptor SDR-IQ.

Este programa consta de dos partes:

- CuteSDR (Opción cliente)
- SDRxxServer

Inicialmente se ejecuta el programa SDRxxServer, este programa se conecta al receptor SDR-IQ y a su vez escucha a los clientes que le solicitan el receptor.

El programa CuteSDR, se puede ejecutar en cualquier otro ordenador con conexión IP al servidor. Se le introduce la dirección IP del servidor y a continuación se conecta a éste.

En figura 5.6.20 se pueden ver tanto el cliente como el servidor.

A continuación se describe el servidor Linux (Ubuntu) y sus aplicaciones.

Descripción del servidor de receptores Linux (Ubuntu).

El servidor Linux del sistema Marconi, tiene básicamente las siguientes aplicaciones:

- Servidor Web Marconi.
- Servidor WebSDR Marconi.
- Servidor QtRadio.

Este servidor tiene conectados los siguientes receptores:

- Receptor Softrock Lite II (40 m).
- Receptor Perseus.
- Receptor RTL-SDR.

Los receptores Softrock Lite II y RTL-SDR, se usan en el programa WebSDR

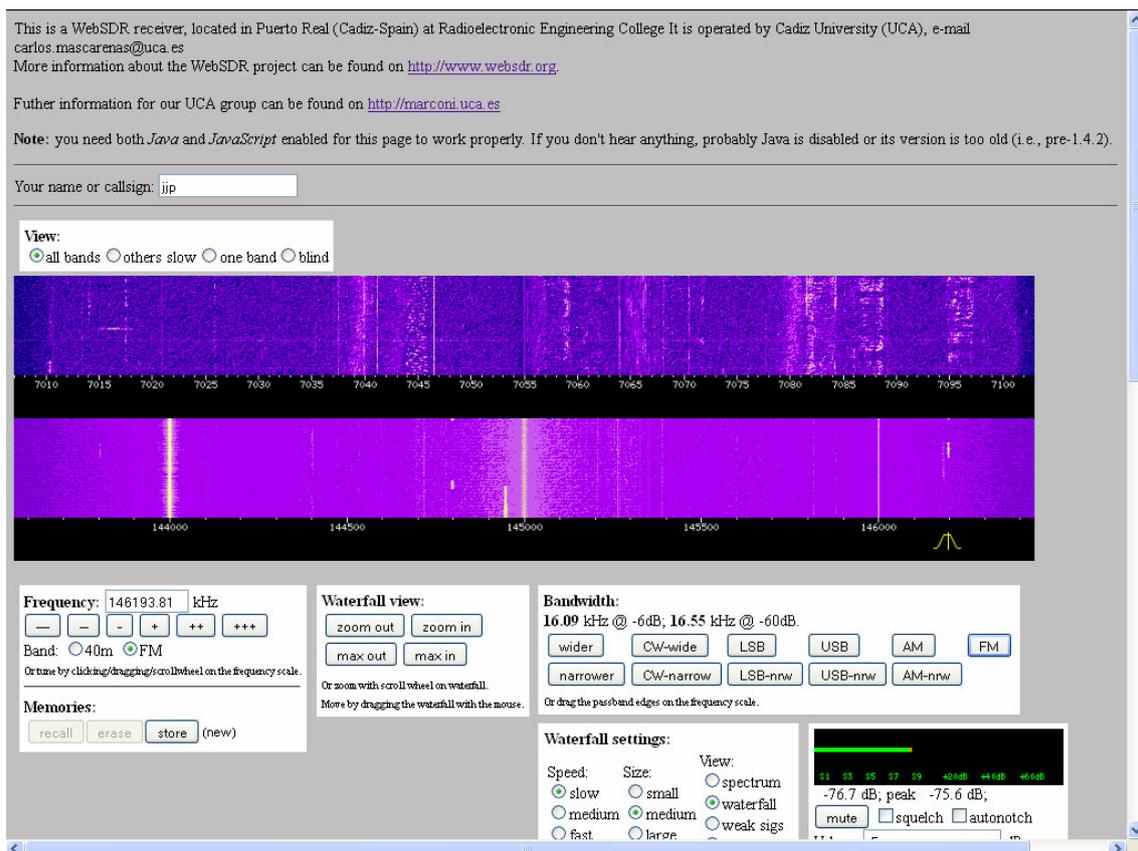


Figura 5.6.21 WebSDR con cascadas 40 metros y FM.

En figura 5.6.21 se puede ver el programa Websdr, donde tenemos las cascadas de 40 metros (Softrock Lite II) y VHF (RTL-SDR).

En figura siguiente (5.6.22), se observa detalle del chat con la participación de los usuarios del WebSDR Marconi.

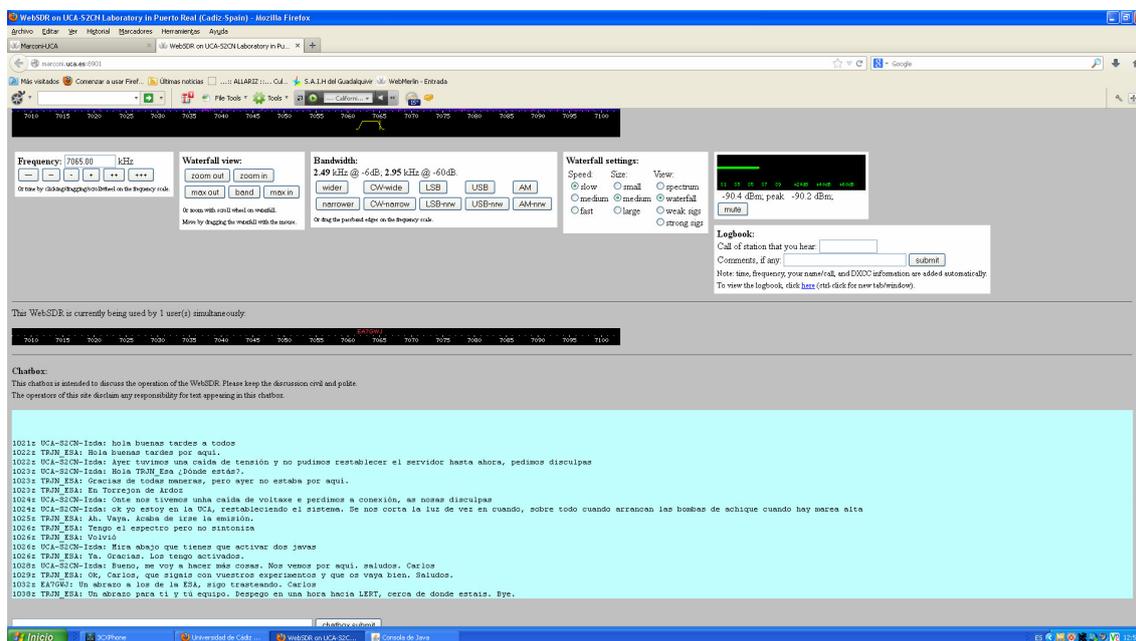


Figura 5.6.22 WebSDR. Detalle de diálogo de usuarios.

El receptor Perseus lo usa el servidor de QtRadio, y está accesible a través de internet en la dirección Marconi.uca.es.

Para acceder a él hay que instalar en los clientes la aplicación Qtradio, y definir en el cliente el servidor Marconi.uca.es.

Como comentario sobre el PC servidor, hay que decir que tiene instalado Windows XP y Ubuntu, siendo el sistema operativo Ubuntu el principal en este PC.

La distribución de Ubuntu es la 12.04.

Se ha instalado y probado el entorno LAMP para poder correr el servidor de Web Marconi.

Se han definido dos usuarios, uno para la aplicación WebSDR y otro para la aplicación QtRadio, ejecutándose estas aplicaciones desde cada uno de estos usuarios.

Se ha colocado una segunda placa de sonido externa estéreo para que pueda funcionar tanto WebSDR como la aplicación cliente de QtRadio.

La aplicación Pulse Audio del sistema operativo Ubuntu es la encargada de controlar el sonido, se ha de configurar de tal forma que el nivel de señal de entrada no sature las gráficas de WebSDR.

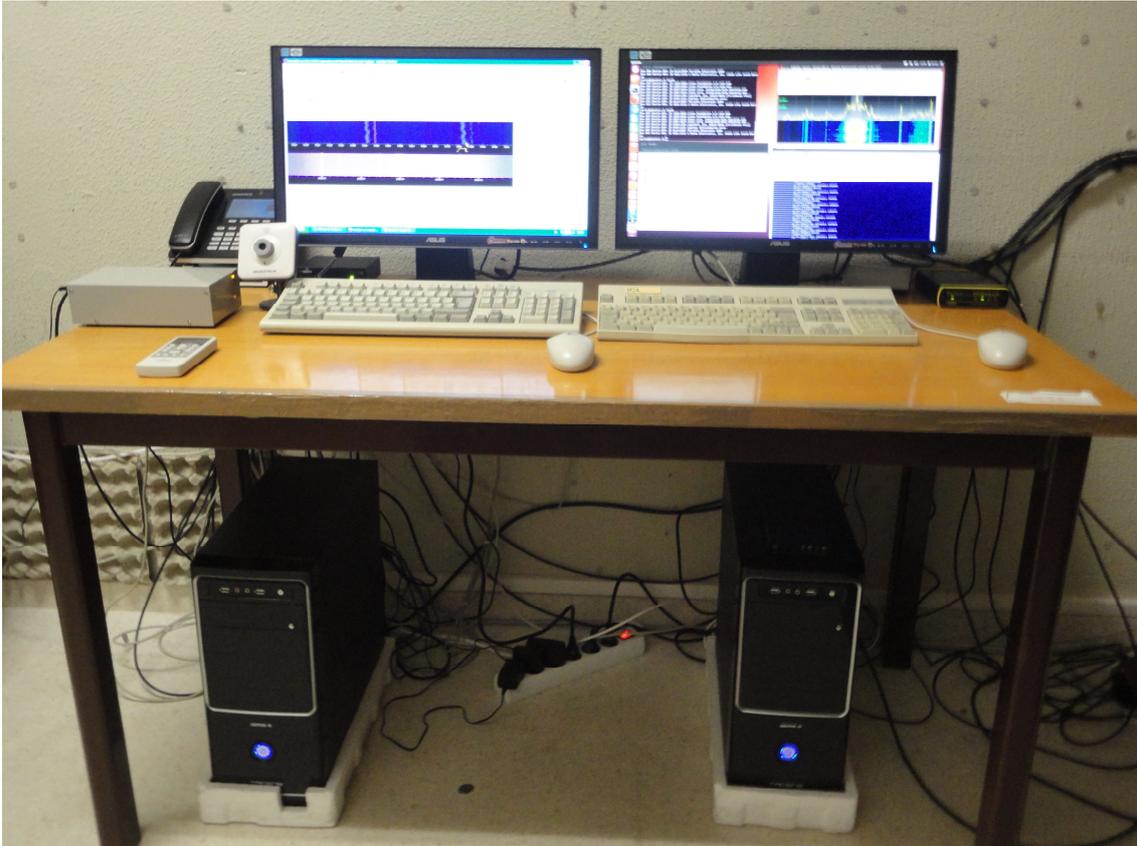


Figura 5.6.23 Servidor XP con programa WebSDR (Cliente) y Servidor Linux con programa QtRadio.

Para tener acceso a estos servidores se han abierto en el firewall de la UCA los siguientes puertos:

- Servidor WebSDR: Puertos 8901 y 80.
- Servidor QtRadio: Puerto 8000.
- Servidor Web: Puerto 80.

A su vez, estos puertos se han abierto en el firewall Marconi y redireccionados a los servidores correspondientes.

Para dar señal a los receptores del sistema Marconi, el grupo S2CN ha instalado un nuevo conjunto de antenas.

A continuación se muestran imágenes de estas figuras 5.6.24, 5.6.25, 5.6.26, 5.6.27, 5.6.28, 5.6.29, 5.6.30, 5.6.31.

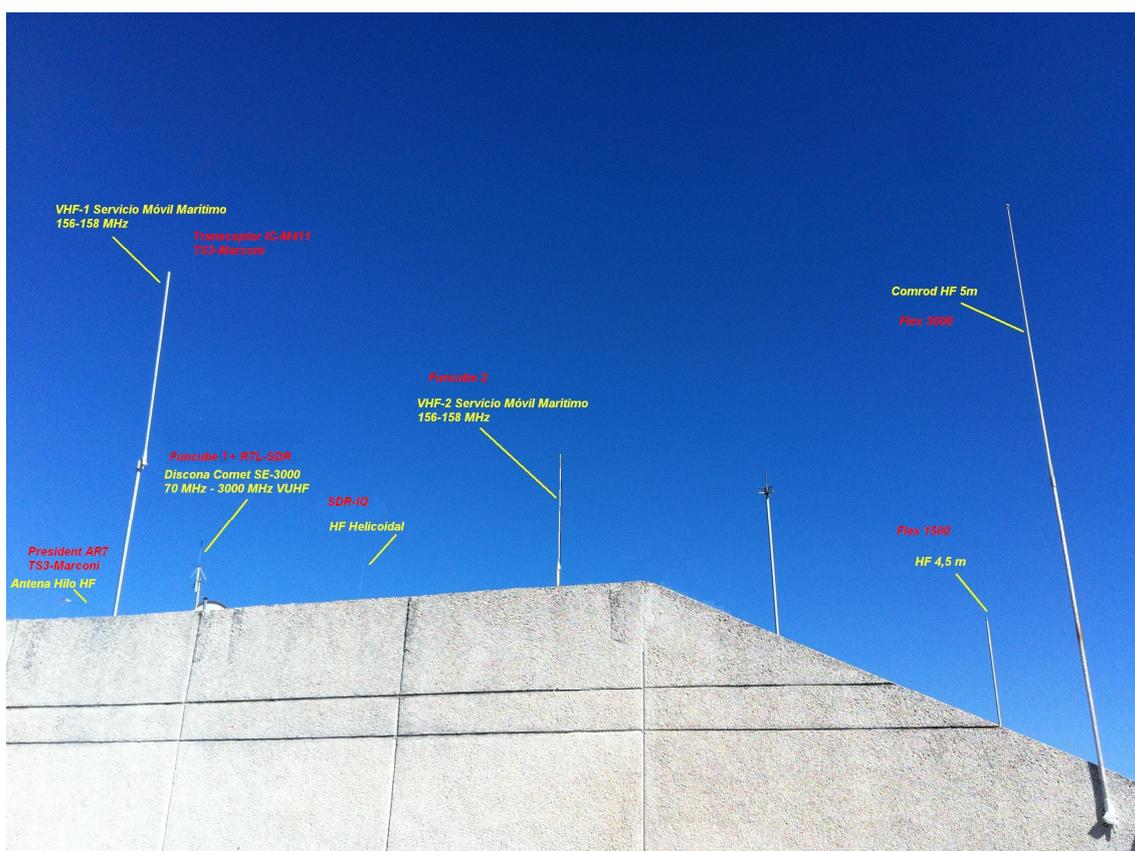


Figura 5.6.24 Antenas usadas dentro del sistema Marconi.



Figura 5.6.25 Detalle de antenas del sistema Marconi para Flex 1500, Funcube, SDR-IQ.



Figura 5.6.26 Detalle de antenas para Softrock, Perseus, Flex 5000 e ICOM ICM-411.



Figura 5.6.27 Detalle de antenas para receptores integrados en Teamspeak.

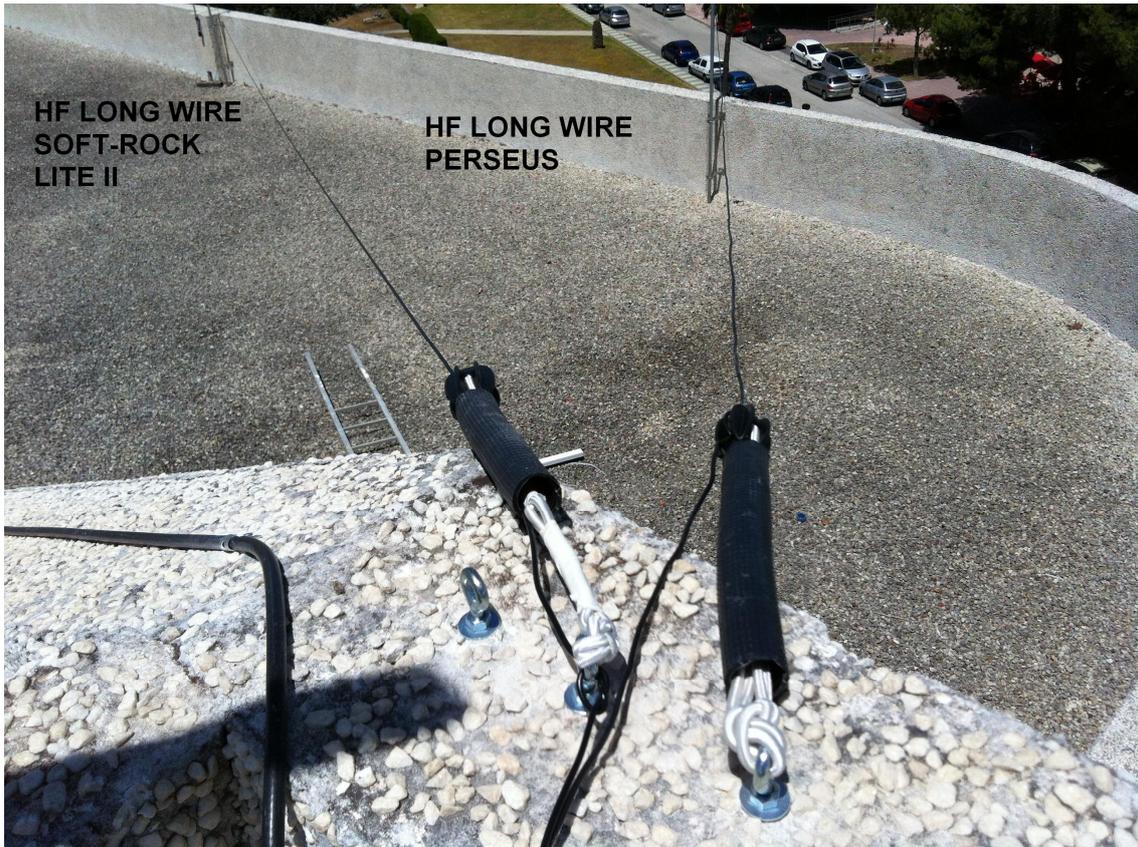


Figura 5.6.28 Detalle de antenas para Softrock y Perseus.

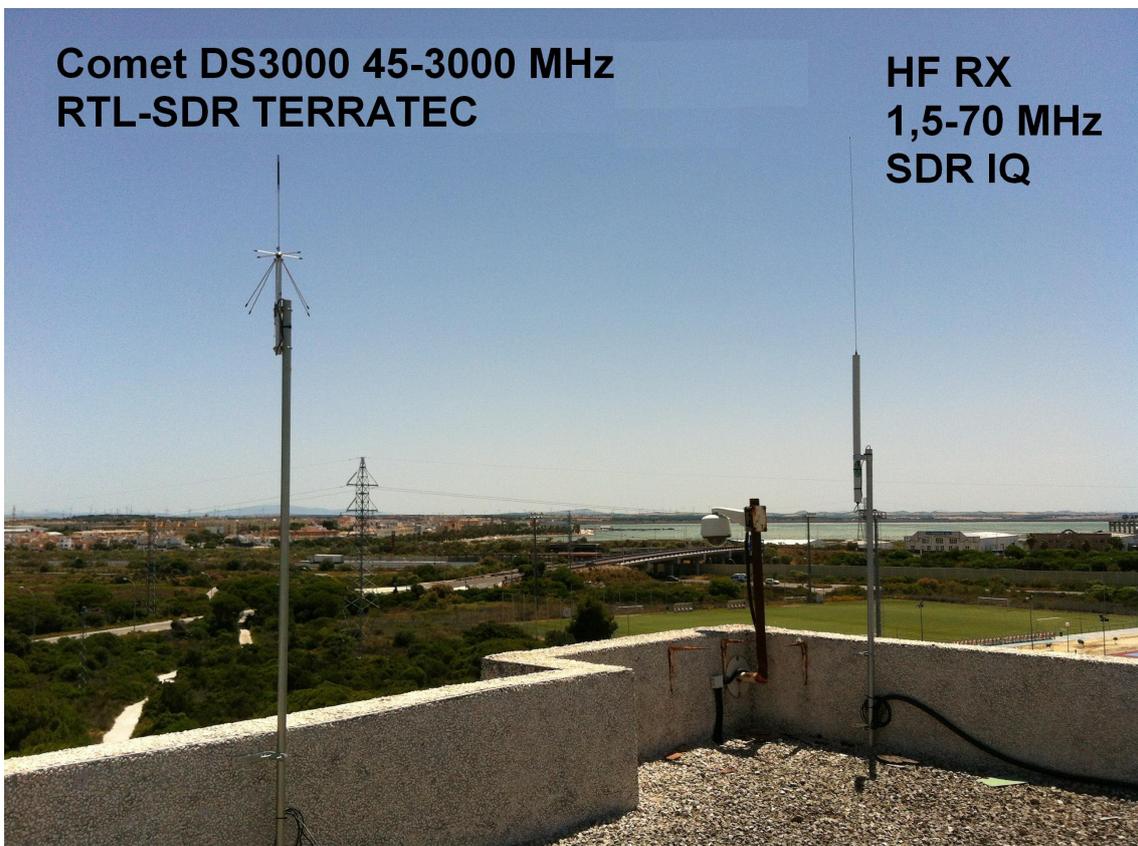


Figura 5.6.29 Detalle de antenas para Terratec (RTL-SDR) y SDR-IQ.



Figura 5.6.30 Detalle de distribución de antenas en el CASEM

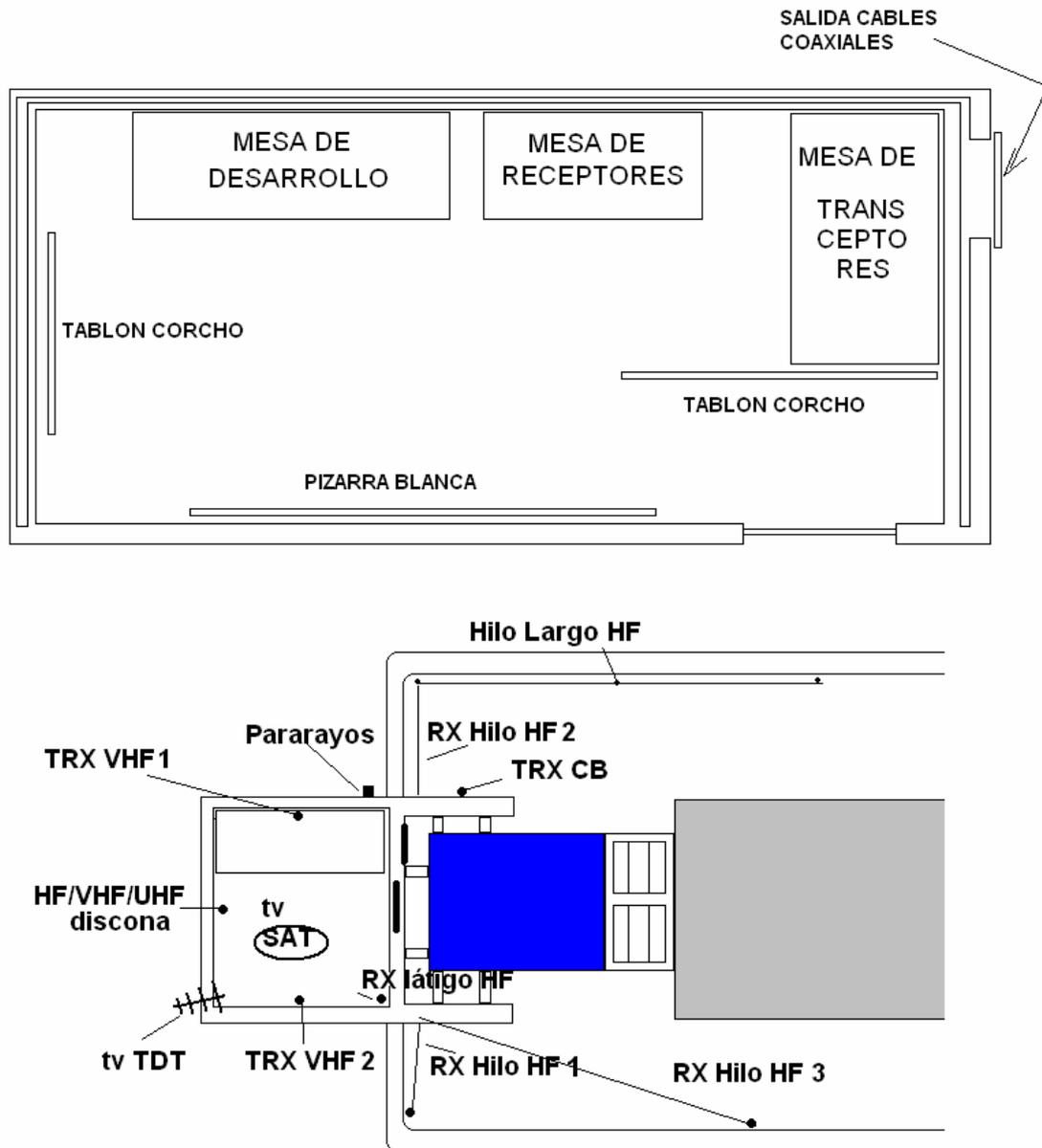


Figura 5.6.31 Detalle de distribución de antenas y equipos en el castillete del CASEM

Relación de Instrumental utilizado en esta Tesis Doctoral y valoración económica del Demostrador de Conceptos.

Equipo	Modelo	Unid.	Precio Unit.	Precio Item
Transceptor SDR Flex 1500 HF+6	Flex 1500	2	752	1504
Transceptor SDR Flex 3000 HF+6	Flex 3000	1	1861	1861
Kit receptor Softrock40	Softrock40	1	30	30
Kit receptor Softrock HF		1	90	90
Receptor SDR VHF-UHF	Funcube Pro	2	160	320
Receptor SDR VHF-UHF RTLSDR	Terratec	2	12	24
Receptor SDR HF-UHF RTL-SDR	TV Dongle	1	15	15
Receptor SDR HF	SDR-IQ	1	590	590
Receptor SDR HF	Perseus	2	700	1400
Rig Expert Tiny CAT USB	Tinicat	2	285	470
Estación meteorológica pantalla táctil	WS8681	1	110	110
Ordenador compatible PC	Genérico	6	reutilizados	0
Ordenadores portátiles PC		3	reutilizados	0
Tarjetas y complementos PC	Genéricos		700	700
Switch LAN	Genéricos	2	12	24
RouterWIFI	Telefónica		reutilizado	0
Teléfonos IP Grandstream		2	150	300
Videocámaras IP Grandstream		2	150	300
Cable coaxial RG213 100m	rg213	1	218	218
Conector PL macho para RG213	Genérico	15	4	60
Carga ficticia d-original tipo PL	AV-DL200M	1	70	70
Carga ficticia d-original tipo N	AV-DL200N	1	70	70
	doriginal			
Conmutadores tipo PL	avsw3m	1	60	60
	doriginal			
Conmutadores tipo N	avsw3n	1	60	60
Transceptor convencional HF 100 W	IC-M700	1	reutilizado	0
Transceptor VHF marítimo	IC-M411	1	reutilizado	0
Transceptor 27 Mhz	President	1	reutilizado	0
Fuente de alimentación 30 A	TBD	1	reutilizado	0
Total de Equipos y accesorios				8276

Instalaciones accesorias obligatorias para el buen funcionamiento del Demostrador.

Instalación de aire acondicionado en el Laboratorio.	1500 Euros
Instalación de red de Área Local Alámbricas	1200 Euros
Instalación de red eléctrica en Laboratorio	800 Euros
Total Instalaciones Accesorias	3500 Euros

Antenas Instaladas en la Azotea de la Pala B del CASEM.

Antena de banda ancha HF		1	80	80
Antena discage 20-1500 MHz		1	170	170
Antena de látigo marina HF	Comrod	1	reutilizada	0
Antena de látigo marina HF		1	reutilizada	0
Antenas de hilo largo HF	Cable eléct	4	20	80
Antena dipolo de hilo HF	Cable eléct	1	40	40
Antenas de látigo VHF marinas		2	reutilizada	0
Antena Jpolo VHF 145 MHz		1	reutilizada	0
Antena látigo 27 MHz		1	reutilizada	0
Instalación antenas + accesorios		20 horas		1200
Total antenas nuevas				1570

Coste total del demostrador de Conceptos

13346 Euros

Tabla 5.1 Costes del demostrador de conceptos Marconi.

5.7. Página Web Marconi.uca.es.

Como página de descripción del sistema Marconi y presentación del grupo S2CN se ha creado la página web Marconi.uca.es, desarrollada en HTML y accesible desde cualquier lugar del mundo.

En esta página se pretende dar una pequeña descripción del sistema Marconi, del grupo S2CN y de la Universidad de Cádiz.

A su vez se pretende agradecer su colaboración a los grupos que han aportado a la realización del sistema Marconi.

Universidad de Cádiz

Este es un servidor de la Universidad de Cádiz

En este servidor se alojan varias aplicaciones para uso de los alumnos de la UCA.

Dirigir preguntas a:
Carlos Mascareñas (carlos.mascarenas@uca.es)
Juan J. Palma (juan.palma@hotmail.com)

Visitas 327

Grupo S2CN - Marconi server

Bienvenido al Servidor Marconi

El Grupo Señales, Sistemas y Comunicaciones Navales (S2CN) nace en 2003 a partir del programa de Doctorado en Tecnologías para la Defensa, del cual surgen sus tres líneas fundamentales de investigación: Señales de Radiocomunicaciones, Compatibilidad Electromagnética y Sistemas de Combate. A partir del Tsunami que asoló el Océano Índico en 2004 se añadió la línea de Protección Civil.

Este Grupo imparte su docencia en la Escuela de Ingeniería Radioelectrónica de la Universidad de Cádiz en Puerto Real (IM66VM) y pone a disposición de sus alumnos las últimas técnicas de enseñanza tanto teórica como práctica con el fin de conseguir el mejor aprovechamiento del tiempo.

Agradecemos la colaboración de las siguientes personas y Grupos:
Dr. Pieter-Tjerk de Boer (WebSDR, Twente University)
Mr. Andrea Mostefusco (Perseus, www.montefusco.com)

Dirección de acceso al servidor WebSDR Marconi

[Websdr Marconi](#)

Direcciones de descarga clientes Teamspeak y QtRadio

[QtRadio cliente \(Dirección del servidor marconi: marconi.uca.es\)](#)
[Teamspeak cliente \(Dirección del servidor marconi: marconi.uca.es\)](#)

Grupo Señales, Sistemas y Comunicaciones Navales

[Presentación del Grupo Señales, Sistemas y Comunicaciones Navales](#)
[Currículum Vitae del Grupo Señales, Sistemas y Comunicaciones Navales](#)

Universidad de Cádiz

[Campus Virtual de la Universidad de Cádiz](#)
[Servicio de Publicaciones de la Universidad de Cádiz](#)

Remer

Figura 5.7.1 Página web del servidor Marconi del grupo S2CN.

Esta página web tiene tres áreas:

- Zona izquierda: Detalles de contactos y control de visitas.
- Zona superior: Objetivos y reconocimientos.
- Zona inferior: Detalles del sistema y enlaces de detalles.



Figura 5.7.2 Detalle página web del servidor Marconi del grupo S2CN (1).



Figura 5.7.3 Detalle página web del servidor Marconi del grupo S2CN (2).

5.8 Página Foro Marconi.uca.es.

Dentro de la página Web principal del sistema Marconi, se han colocado enlaces a la página Web del foro del grupo S2CN.

Este foro se ha creado como una aplicación más dentro del sistema Marconi, con objeto de recoger la opinión sobre él, así como de establecer un diálogo dentro entre aficionados a las radiocomunicaciones, profesionales, alumnado y profesorado, está desarrollado usando el software phpBB.

The screenshot shows the forum interface for 'Comentarios sobre WebSDR Marconi'. It includes a navigation bar with 'Índice general' and 'VARIOS', a search bar, and a list of topics. The topics table is as follows:

Temas	Autor	Respuestas	Vistas	Último mensaje
Hola a todos los red-escuchas	Carlos Mascareñas	0	6	Lun Jul 22, 2013 12:29 pm Carlos Mascareñas →
Hola a todos.	jpalma1	0	19	Sab Jul 13, 2013 7:36 am jpalma1 →

Figura 5.8.1 Detalle página Web del foro S2CN. Información sobre la aplicación WebSDR.

The screenshot shows the main forum page with a table of forum categories. The table is as follows:

Foro	Temas	Mensajes	Último mensaje
USO GENERAL DEL FORO			
Como usar el foro Reglas generales de uso del foro.	1	1	Lun Jul 15, 2013 2:50 pm jpalma1 →
Preséntate Preséntate en esta sección para que te conozcamos / Please introduce yourself.	2	7	Mié Jul 17, 2013 10:16 am Larrey →
Ayúdanos a mejorar, danos tu opinión o sugerencia Tu opinión nos importa.	1	6	Mar Jul 16, 2013 11:55 am jimf53 →
GRUPO S2CN			
Líneas de Investigación Líneas de Investigación del grupo S2CN.	4	4	Lun Jul 29, 2013 7:18 am jpalma1 →
Software Software de procesado y decodificación de señales.	0	0	No hay mensajes
Red radio de emergencia (no oficial) Remer	2	3	Mié Jul 17, 2013 10:13 am Carlos Mascareñas →
Radioelectrónica Naval	0	0	No hay mensajes

Figura 5.8.2 Detalle página Web del foro S2CN. Página principal del foro.

5.9 Servidor GNU Radio.

GNU Radio es una aplicación de desarrollo de código libre y abierto que proporciona bloques de procesamiento de señales para implementar software radio.

GNU Radio se puede crear con hardware de RF externo de bajo costo para crear radios definidos por software o sin hardware utilizando el entorno de simulación.

En el punto 2.3.6 de esta Tesis se ha descrito esta aplicación, si bien usando su receptor de radio (Tarjeta USRP de Ettus Research), a finales del año pasado el grupo GNU Radio ha integrado en su plataforma el receptor de bajo coste RTL-SDR motivo por el que se estudia este software y sus requerimientos de instalación, haciendo como siempre en esta tesis una instalación inicial donde se depuran dependencias y scripts de instalación.

Más detalles se pueden ver en la página inicial de GNU Radio.

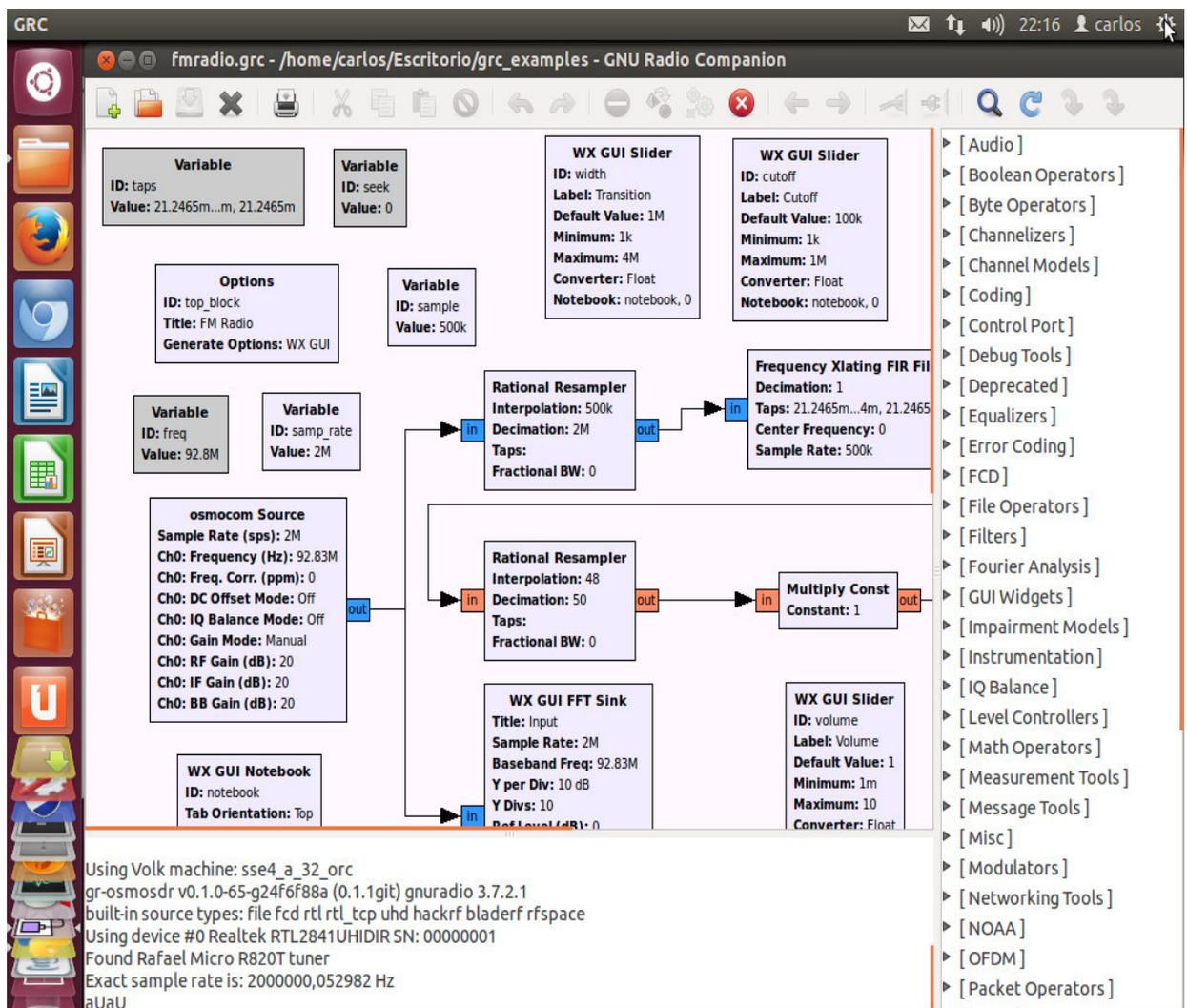


Figura 5.9.1 Pantalla principal de GNU Radio

En figura 5.9.1 podemos observar la pantalla principal de GNU Radio Companion, interfaz gráfica de GNU Radio.

Se observan las siguientes áreas:

- Relación de bloques constructivos de GNU Radio (Lado derecho)
- Líneas de comandos ejecutadas por la aplicación para definir el receptor RTL-SDR (Lado inferior).
- Panel central con diagramas de bloques del ejemplo en ejecución.

En figura 5.9.2 vemos el ejemplo de radio fm en ejecución.

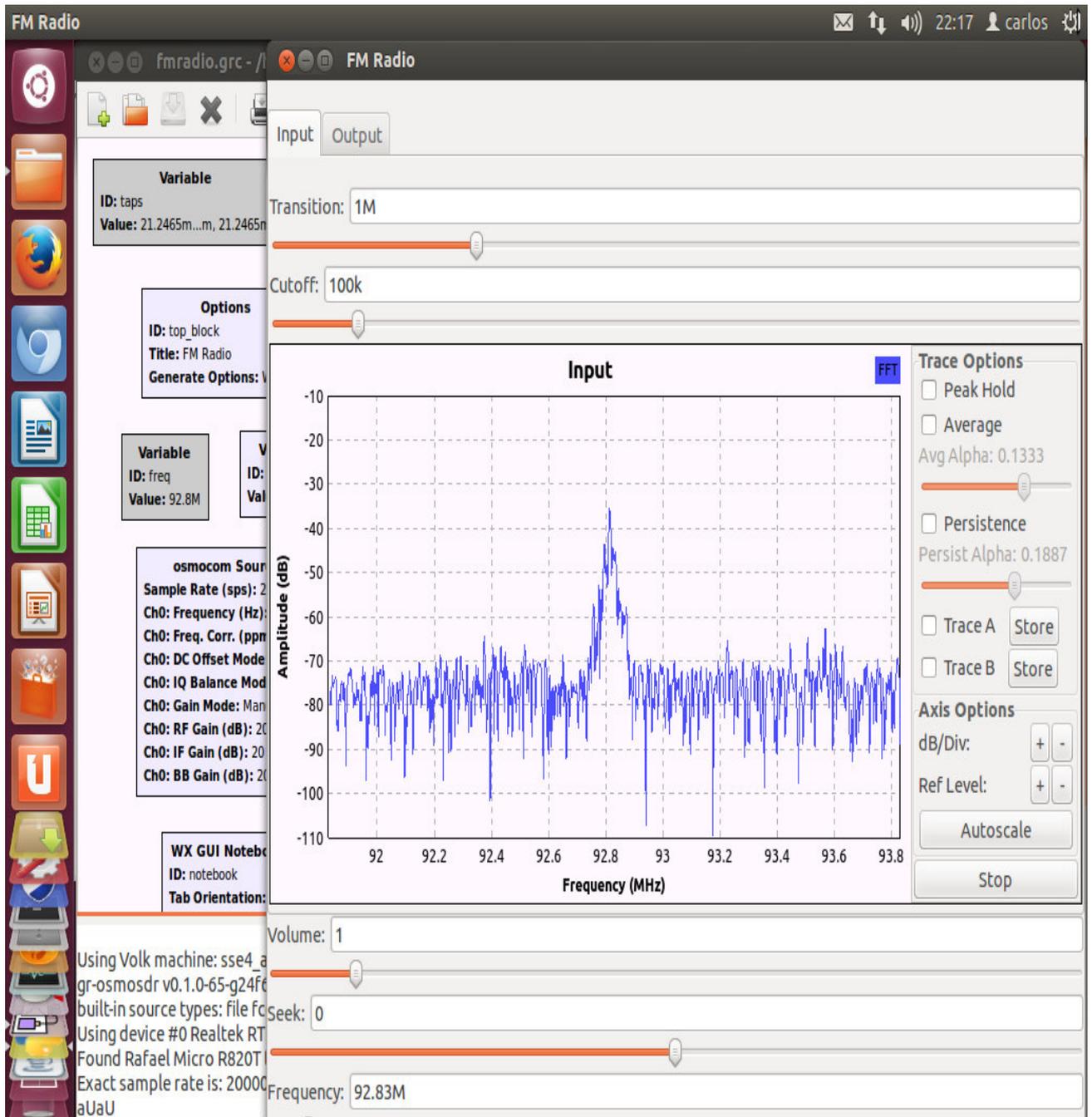


Figura 5.9.2 Ejemplo de radio FM usando GNU Radio

5.10 Propuestas de Patentes.

5.10.1 Demostrador de Conceptos de Radiofrecuencia diseñado para su aplicación a la Enseñanza Superior y con aplicaciones, de forma local o remota, al Control y Análisis de Radiocomunicaciones en cualquier banda de Radio.

SECTOR DE LA TÉCNICA.

Enseñanza de la Física, la Informática, la Matemática y la Tecnología de los Campos Electromagnéticos y las Radiocomunicaciones.

ESTADO DE LA TÉCNICA.

Actualmente se enseñan los fundamentos de los campos electromagnéticos mediante los conceptos de la Física Clásica, la formulación matemática y a veces la simulación mediante programas de computador, pero se basa fundamentalmente en ejercicios matemáticos que acaban siendo un mero resultado numérico o gráfico, sin ningún significado concreto para el alumno.

La invención propuesta supone una herramienta que permite la captación y el análisis de los campos electromagnéticos y su medición, así como la realización de prácticas individuales o grupales con resultados prácticos y que mejoran la asimilación de los conceptos de la teoría aprendida en clase con anterioridad o la generación de dichos conceptos cuando se practica el método de la enseñanza por descubrimiento.

Se diferencia de los sistemas clásicos de enseñanza en que utiliza la técnica “en línea” en tiempo real hacia cualquier parte del Mundo con conexión a Internet, lo que permite compartir recursos y abaratar costes de enseñanza, al mismo tiempo que se comparten experiencias entre alumnos y docentes, ya sea en tiempo real o a través de la página web y foro del Demostrador de Conceptos.

La invención permite la demostración de los conceptos de, entre otros:

- Amplificación
- Amplificadores de línea
- Amplitud
- Análisis de Espectros
- Análisis de Señales Temporales
- Anchura de Banda
- Antenas

- Arquitectura Cliente-Servidor
- Bandas Laterales
- Cables coaxiales
- Cámara IP
- Cascada de señales
- Centralita analógica
- Centralita software
- Conector Coaxial
- Conmutador
- Control automático
- Cortafuegos
- Derivación
- Distorsión
- Distribuidor de Radiofrecuencia.
- Elevación
- Equipos de radio definidos por software.
- Error.
- Espectro de frecuencias.
- Federación virtual de circuitos.
- Filtros.
- Frecuencia
- Frecuencia de portadora
- Frecuencia Intermedia
- Ganancia
- Horizonte Radio
- Horizonte Terrestre
- Impedancia
- Índice de Modulación.
- Interferencia
- Intranet.
- Leyes de Maxwell
- Longitud de Onda
- Mantenimiento
- Mezcla
- Modulaciones analógicas y digitales.
- Muestreo.
- Órbita
- Oscilador Local
- Pasarela de Radiocomunicaciones.
- Pérdidas en los cables coaxiales
- Polarización
- Procesado Digital de la Señal.
- Radio direction finding
- Radio Sobre Protocolo Internet
- Radioastronomía
- Radioteletipo.
- Realimentación.
- Receptores de radiofrecuencia
- Retardo de propagación
- Rotación terrestre

- Router wifi
- Ruidos térmicos, Johnson y Flicker.
- Satélite
- Seguimiento de Satélites
- Señales IQ.
- Servidor de Chat.
- Switch
- Teléfono IP
- Teléfono software.
- Tiempo de Respuesta
- Tiempo de Retardo.
- Tiempo de Subida
- Velocidad Binaria
- Velocidad de Transmisión.
- Video sobre IP
- Videoteléfono.
- Voz sobre IP
- WIFI

EXPLICACIÓN DE LA INVENCION

La invención es un sistema completo de enseñanza y consiste en un Conjunto de equipos electrónicos de radiofrecuencia, cómputo, control y análisis que permiten llevar las señales contenidas en los campos electromagnéticos circundantes a las antenas hasta cualquier parte del Mundo que cuente con una conexión Internet o Intranet. Esta última opción es la que permite al alumnado la realización de prácticas de Radiotecnía tanto en el Centro de Enseñanza como en su domicilio o fuera de él, ahorrando costes económicos de aprendizaje al compartir el equipamiento didáctico. Al mismo tiempo se generan una serie de Servicios hacia la Sociedad al inyectarse en la nube de Internet algunos tipos de radiocomunicaciones generales de interés para ella.

La invención se divide en dos grandes bloques integrados entre sí. Uno es la Cabecera de Radiofrecuencia o Sensores de Radio y el otro el Sistema de Control y Distribución de las Señales recibidas mediante distintos procedimientos.

CABECERA DE RADIOFRECUENCIA

La Cabecera de Radiofrecuencia se muestra en la Figura 5.10.1 en forma de diagrama de bloques.

Está compuesta por antenas, cables de bajada, amplificadores y distribuidores de señal de radio y equipos receptores y/o transceptores. Los equipos de radiocomunicaciones pueden ser equipos convencionales, basados en componentes físicos o pueden ser equipos de radio definidos por software (SDR) los cuales disponen de una menor infraestructura electrónica y una mayor capacidad de adaptación a la situación operativa por la que se atraviesa.

Las antenas. Su misión es captar las señales electromagnéticas que viajan por el éter, provocando una inducción de tipo eléctrico o magnético (según sea la antena) y convertirla en una débil corriente eléctrica que podrá ser amplificada, o no, por amplificadores de banda ancha (3) y (9) de frecuencia central y anchura de banda adecuada a las frecuencias a recibir y equipos a demodular.

Las antenas pueden ser de diverso tipo, por lo general son cables metálicos lineales protegidos mediante una cubierta aislante ante el calor y el contacto, de tamaño similar a la cuarta parte de la Longitud de Onda de la Frecuencia que se quiere recibir. Dicha longitud de Onda se calcula como $300/F$, siendo F la frecuencia en Megahercios.

La disposición física del cable lineal suele depender de su tamaño físico y de la plataforma que lo porte, pudiendo instalarse de forma vertical, horizontal, inclinada, en “V”, “V invertida” o en zigzag utilizando diversos puntos de apoyo o mástiles.

Por lo general las antenas de frecuencias por debajo de 7 MHz suelen estar fabricadas en hilo multi- o unifilar y se les dota de posición o polarización horizontal o inclinada, mientras que las antenas de menos de 12 metros de longitud ya se instalan con posición o polarización vertical y se embute el cable de la antena dentro de una cápsula de fibra de vidrio que las autoporta, si bien también se pueden fabricar mediante tubo de aluminio de sección adecuada.

Dado que el mayor alcance de las Ondas de Radio se obtiene en Onda Corta (3-30 MHz), la invención dispone de distintos receptores definidos por software que cubren estas bandas, además de las de Onda Larga (30 a 300 kHz) y Onda Media (0,3 a 3 MHz).

Estos receptores pueden compartir un amplificador de banda ancha de VLF hasta HF con una antena de hilo (o de otro tipo), como son los receptores (1) y (2) que comparten el amplificador (3), o puede disponer de su propia antena aunque sea de otro tipo, por ejemplo helicoidal, como el receptor (11).

Con el fin de proteger las etapas de radiofrecuencia de los receptores de banda ancha cuando están transmitiendo los transceptores con elevada potencia se le puede añadir un Conmutador de Transmisión/Recepción (5) que deriva a tierra la señal recibida por la antena (en este caso RX HF Hilo1) y la aísla del receptor (4) protegiéndolo contra corrientes elevadas emitidas desde la propia invención.

Otro tipo de receptores de HF pueden ser los de banda estrecha, que son aquellos que sólo toman una pequeña sección de radiofrecuencia y que no son resintonizables a distancia, como el receptor (6), el cual dispondrá de una antena de longitud y polarización acorde con su frecuencia central de trabajo. La anchura de banda del receptor de banda estrecha (6) dependerá del sistema de tratamiento matemático que utilice el receptor definido por software.

Pasando a las bandas de VHF (30 a 300 MHz) y UHF (300 a 3000 MHz) se aplican los mismos criterios que en las bandas inferiores, pudiendo existir equipos receptores (7) y (8) que comparten amplificadores de banda ancha (9) ajustados a su rango de frecuencias y receptores con una antena directamente acoplada a ellos (10). Los cables coaxiales serán los idóneos para no producir pérdidas de transmisión significativas en los cables y dependiendo de la distancia entre antena y receptor puede ser necesario el uso de amplificadores de línea cerca del extremo de la antena.

Los transceptores SDR disponen cada uno de su propia antena, aunque podrían compartir varios la misma antena si se utilizan filtros combinadores de radiofrecuencia y dicha antena resuena a las frecuencias de trabajo. En la invención se han incluido un transceptor de MF/HF (13) y otro de VHF/UHF (14).

En cuanto a los transceptores convencionales se han incluido, siempre cada uno con su antena, un equipo de banda ciudadana (27 MHz) (12), un transceptor de MF/HF (15) dotado de acoplador de antena (16) y antena dipolo y un transceptor de VHF (17) integrado con su propia antena de látigo.

Con fines de intercomparación y calibración en modo local se incluye un receptor de 0,1 a 3000 MHz (18), que bien puede ser un receptor de radio multimodo y multibanda o un instrumento de laboratorio como un receptor de cuasi-pico o un analizador de espectros o ingenio similar. Dicho instrumento está dotado de dos o más antenas seleccionables con el fin de recibir mejor las señales y adaptarse a las mismas condiciones que el resto de los receptores.

SISTEMA DE CONTROL Y DISTRIBUCIÓN DE SEÑALES.

El Sistema de Control y Distribución de Señales (SCDS) consiste en un conjunto local de ordenadores conectados en Red Interna (Intranet) con capacidad de análisis de señales y conversión y decodificación de datos con acceso desde el Exterior del mismo (Internet) y desde lugares Internos pero remotos al mismo (Intranet). Puede verse su diagrama de bloques en la figura 5.10.2.

El corazón del Sistema está formado por un ordenador que alberga un cortafuegos (B) que aísla la red mundial Internet (A) del resto del SCDS y que recibe toda la información desde el exterior, ya sean datos a procesar o peticiones de datos digitales. Este cortafuegos actúa como antivirus y antitroyanos, protegiendo la seguridad informática del SCDS en la red de alta velocidad de transmisión de datos.

Para poder compartir la defensa del Cortafuegos se utiliza un Switch (C) o distribuidor de datos digitales de “n” enchufes o puertas, por las que se conectarán el resto de los dispositivos y las ampliaciones a otros distribuidores de datos o Switch (D) (O), tantos como sean necesarios en la red (X).

Una de las puertas se conectará directamente a un Router inalámbrico (E) con fines de acceso local y ensayo didáctico desde dispositivos portátiles tipo Smartphones (S), Tablet (T), Ordenadores portátiles (U) e incluso Cámaras IP-WIFI (Y), lo que permite el acceso a múltiples usuarios de forma simultánea e independiente del dispositivo inalámbrico de acceso.

Otro de los componentes del sistema reside en la Centralita Software (I), que consiste en un ordenador dotado de una tarjeta telefónica digital y el software adecuado para su control y administración. Esta Centralita integra al Demostrador con las redes telefónicas convencionales, permitiendo que el audio de cualquier dispositivo de radio pueda transmitirse por una red telefónica nacional o internacional, y viceversa, que una transmisión analógica de voz pueda acceder, mediante línea telefónica externa a uno de los transceptores (12), (13), (14), (15), (16) o (17) de la cabecera de radiofrecuencia.

El Servidor de Chat (J) realiza la función de control y administración de comunicaciones de audio en banda base realizadas sobre las redes con protocolo internet (VOIP), seleccionando los usuarios que pueden acceder a cada canal o sala del sistema y su nivel de acceso a las posibilidades de transmisión y/o recepción por radio de canales específicos.

El Servidor de Chat (J) está configurado en salas o canales que simulan los canales radioeléctricos en los que los alumnos podrán realizar prácticas de radiocomunicaciones en cualquier idioma y siguiendo los procedimientos específicos de cada servicio, así mismo hay canales específicos para que el profesorado de Procedimientos Radioeléctricos intercambie impresiones antes o después de las prácticas del alumnado o para que el profesorado de idiomas extranjeros realice lo mismo.

Este mismo servidor (J) dispone de canales especiales para ser utilizados por otros colectivos que estén interesados y cuyo acceso se realiza mediante contraseña.

Finalmente, existen canales de transmisión y/o recepción de estaciones de radio reales, pudiéndose acceder a las mismas por autorización del Administrador, mediante contraseña o de forma libre, según sea el caso. La señal se inyecta en el sistema mediante equipos de radiocomunicaciones conectados a ordenadores portátiles dedicados (W), controladores de equipos receptores SDR (V), Servidores de Receptores SDR de acceso externo permitido (L), Servidores de Receptores SDR de Acceso externo prohibido (M) o pasarelas de radio (R) integradoras de equipo de radio convencionales y SCDS. En todos estos casos sólo se accederá a la difusión de la señal de Audio o VOIP, pero no al control de los equipos de radio.

Los receptores (V) (1) (2) (4) (6) (7) (8) (10) (11) (12) (13) (14) (15) (17) proporcionarán el tipo de acceso que les permita el estado de su técnica de fabricación, pudiendo realizarse directamente en Banda Base de Audio, en una Frecuencia Intermedia analógica o digital, de forma codificada o bruta, en forma digital totalmente compatible con el puerto de entrada de la pasarela de radio o en otra forma, no incluida en las opciones anteriores, que pueda posteriormente ser tratada de forma digital para su uso.

De forma local es posible acceder al uso y Control completo del Demostrador de Conceptos mediante las aplicaciones de radiocomunicaciones contenidas en un ordenador (H) conectado a la Intranet y que tiene acceso a tanto a los transceptores y receptores de todo tipo como a los teléfonos IP (F) (Ñ) (Q), como a las cámaras IP (Y) (N) (P), a la red Internet (A), a la red telefónica convencional a través de la centralita software (I), al Chat, a través del servidor de Chat (J) y a los Softphones (S), Tablet (T) y Ordenadores Portátiles (U) y (W) a través del enrutador WIFI (E).

Con el fin de mantener la configuración y la gestión de los datos recibidos y transmitidos por cada aplicación, así como la captura de datos de uso y administración se dispone de un ordenador con alta capacidad de disco duro de reserva (K) en el que se guardan las últimas configuraciones eficaces del Demostrador.

En este ordenador (K) también se aloja la página web del Demostrador de Conceptos, así como la página del foro de debate diseñado exprofeso para el mismo y que permite la retroalimentación por parte de los usuarios hacia los diseñadores de la Invención así como su posterior mejora.

En cuanto a los servidores de tipo (L) son aquellos que están habilitados para ser controlados y configurados desde el exterior de la entidad que dispone del Demostrador de Conceptos y por lo tanto del SCDS. A ellos se accede desde la nube de la Internet y los niveles de acceso pueden ser gestionados por el Administrador del Demostrador de Conceptos, aunque no puede seleccionar quién va a poder conectarse a ellos.

Los servidores de tipo (M) son aquellos que sólo se puede acceder desde la Intranet de la entidad que dispone del Demostrador de Conceptos, siempre a través del Cortafuegos, los niveles de acceso pueden ser gestionados por el Administrador del Demostrador de Conceptos y sí puede seleccionar quién va a poder conectarse a ellos.

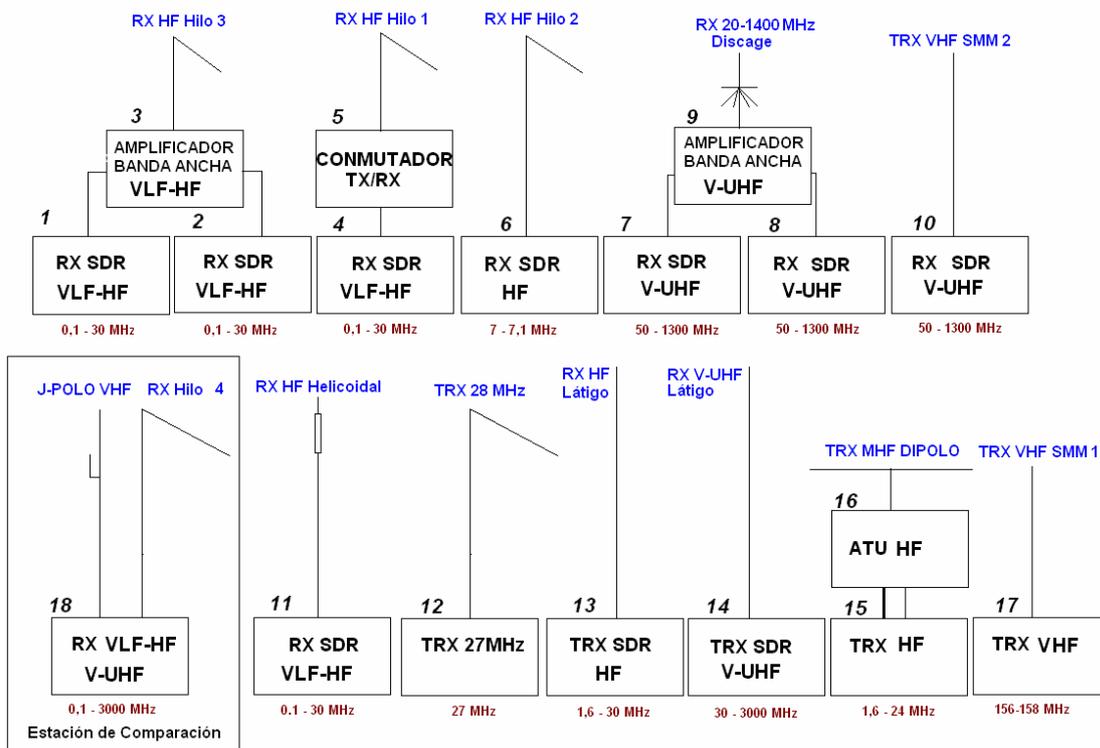
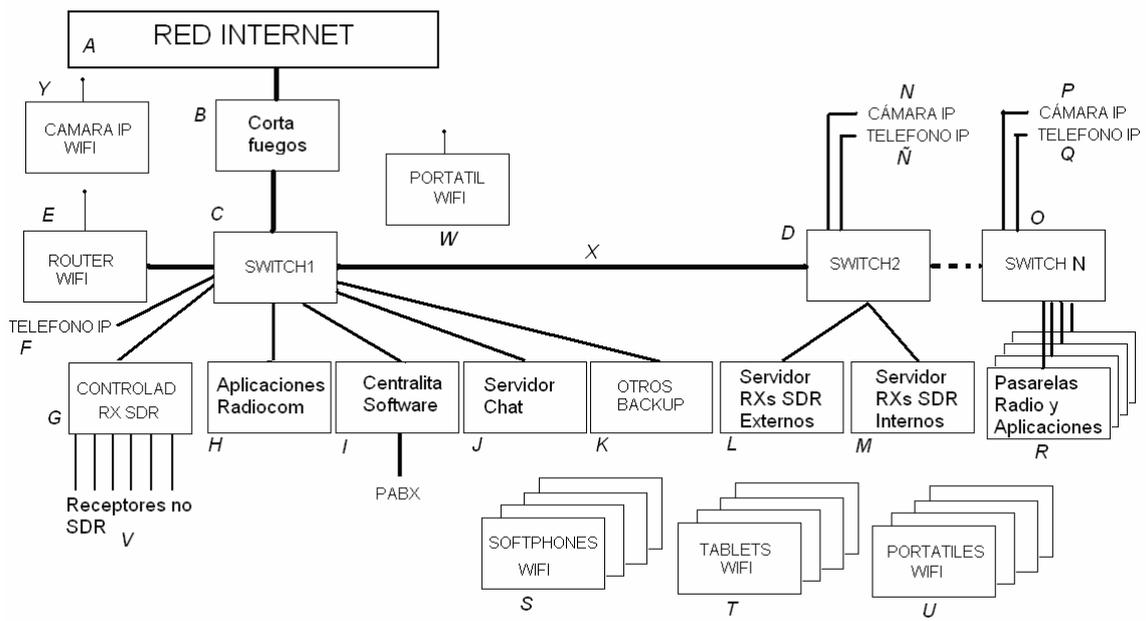


Figura 5.10.1 Cabecera de Radiofrecuencia



Sistema de Control y Distribución de Señales

Figura 5.10.2 Sistema de Control y Distribución de Señales.



MINISTERIO DE INDUSTRIA, ENERGÍA Y TURISMO



Oficina Española de Patentes y Marcas

INSTANCIA DE SOLICITUD



Nº SOLICITUD: P 2 0 1 3 0 1 2 0 0
FECHA Y HORA DE ENTRADA EN OEPM:
FECHA Y HORA DE ENTRADA EN LUGAR DISTINTO A LA OEPM: <i>Cadix 24-12-13</i>
LUGAR DE PRESENTACIÓN <i>Cadix</i> CODIGO

1. IDENTIFICACIÓN DE LA SOLICITUD

(1) MODALIDAD: <input checked="" type="checkbox"/> PATENTE DE INVENCION <input type="checkbox"/> MODELO DE UTILIDAD		(3) EXPEDIENTE PRINCIPAL O DE ORIGEN: UNIVERSIDAD DE CÁDIZ	
(2) TIPO DE SOLICITUD: <input type="checkbox"/> ADICIÓN A LA PATENTE <input type="checkbox"/> SOLICITUD DIVISIONAL <input type="checkbox"/> CAMBIO DE MODALIDAD <input type="checkbox"/> TRANSFORMACIÓN SOLICITUD PATENTE EUROPEA <input type="checkbox"/> ENTRADA EN FASE NACIONAL DE SOLICITUD PCT		MODALIDAD: Entrada	Nº SOLICITUD: 001 Nº. 201400100000239
		FECHA PRESENTACIÓN: 17/01/2014 16:29:26	

2. TÍTULO DE LA INVENCION (4)

DEMOSTRADOR DE CONCEPTOS DE RADIOFRECUENCIA DISEÑADO PARA SU APLICACIÓN A LA ENSEÑANZA SUPERIOR Y CON APLICACIONES, DE FORMA LOCAL O REMOTA, AL CONTROL Y ANÁLISIS DE RADIOCOMUNICACIONES EN CUALQUIER BANDA DE RADIO.

3. IDENTIFICACIÓN DEL SOLICITANTE

(5) APELLIDOS Y NOMBRE /DENOMINACIÓN SOCIAL UNIVERSIDAD DE CÁDIZ				NIF/PASAPORTE Q1132001G	
DIRECCIÓN POSTAL C/ ANCHA, 16		CÓDIGO POSTAL Y LOCALIDAD 11001	PROVINCIA CÁDIZ	PAIS RESIDENCIA ESPAÑA	CODIGO PAIS RESIDENCIA ES
PAIS DE NACIONALIDAD ESPAÑA		CÓDIGO PAIS NACIONALIDAD ES	CNAE (6) 083	PYME (7) 5	
DIRECCIÓN CORREO ELECTRÓNICO otri@uca.es	Nº TELÉFONO FIJO 956015173	Nº TELÉFONO MÓVIL	(8) INDICACIÓN DEL MEDIO DE NOTIFICACIÓN PREFERENTE <input type="checkbox"/> CORREO POSTAL <input checked="" type="checkbox"/> CORREO ELECTRÓNICO		
PORCENTAJE DE TITULARIDAD (9): 100 %		NOTA: DE NO ESPECIFICARSE DICHO PORCENTAJE, LA OEPM PRESUMIRÁ IGUALES LAS CUOTAS DE LOS SOLICITANTES.			
EL SOLICITANTE TAMBIÉN (10) <input type="checkbox"/> SI ES INVENTOR:		<input checked="" type="checkbox"/> NO		MODO DE OBTENCIÓN DEL DERECHO: (11) <input checked="" type="checkbox"/> INVENCION LABORAL <input type="checkbox"/> CONTRATO <input type="checkbox"/> SUCESIÓN <input type="checkbox"/> OTROS (Especificar):	
(12) EL SOLICITANTE ES UNA UNIVERSIDAD PÚBLICA ESPAÑOLA		<input checked="" type="checkbox"/> SI			

Ejemplar para el solicitante

4. OTROS SOLICITANTES Y/O INVENTORES (13)

LOS DEMAS SOLICITANTES Y/O INVENTORES SE INDICAN EN HOJA COMPLEMENTARIA

P 2 0 1 3 0 1 2 0 0

5. IDENTIFICACIÓN DEL REPRESENTANTE

(14) REPRESENTACIÓN		(15) Nº PODER GENERAL	
<input checked="" type="checkbox"/> EL SOLICITANTE NO ESTÁ REPRESENTADO	EL SOLICITANTE ESTÁ REPRESENTADO POR: <input type="checkbox"/> AGENTE DE LA PROPIEDAD INDUSTRIAL <input type="checkbox"/> OTRO REPRESENTANTE		
(16) ACTUACIÓN POR MEDIO DE AGENTE	NOMBRE	CÓDIGO DE AGENTE	
(17) ACTUACIÓN POR MEDIO DE OTRO REPRESENTANTE	NOMBRE	DIRECCIÓN POSTAL	N.I.F.
DIRECCIÓN CORREO ELECTRÓNICO	Nº TELÉFONO	(18) INDICACIÓN DEL MEDIO DE NOTIFICACIÓN PREFERENTE	
		<input type="checkbox"/> CORREO POSTAL <input type="checkbox"/> CORREO ELECTRÓNICO	

6. OTROS DATOS

(19) REIVINDICACIÓN DE PRIORIDAD	PAIS ORIGEN	CODIGO PAIS	FECHA	NÚMERO
<input type="checkbox"/> SI <input checked="" type="checkbox"/> NO				
(20) EXPOSICIONES OFICIALES	NOMBRE		FECHA	LUGAR
<input type="checkbox"/> SI <input checked="" type="checkbox"/> NO				
(21) EFECTUADO DEPÓSITO DE MATERIAL BIOLÓGICO	AUTORIDAD DE DEPÓSITO	CODIGO PAIS	FECHA	NÚMERO
<input type="checkbox"/> SI <input checked="" type="checkbox"/> NO				
LISTAS DE SECUENCIAS DE AMINOÁCIDOS Y ÁCIDOS NUCLEICOS				
La descripción contiene un listado de secuencias biológicas en concordancia con la norma ST.25 OMPI <input type="checkbox"/> SI <input checked="" type="checkbox"/> NO				
Se adjunta un soporte de datos legible por ordenador que incluye el listado de secuencias biológicas en concordancia con la norma ST.25 OMPI <input type="checkbox"/> SI <input checked="" type="checkbox"/> NO				
El solicitante declara por medio de esta instancia, que la información registrada en el soporte de datos legible por ordenador es idéntica a la contenida en el listado de secuencias biológicas incluido en la descripción de la versión escrita de esta solicitud <input checked="" type="checkbox"/>				
(22) EL SOLICITANTE SE ACOGE AL APLAZAMIENTO DE TASAS PREVISTO EN EL ART. 162 DE LA LEY 11/1986 DE PATENTES				<input type="checkbox"/> SI

7. ÍNDICE DE DOCUMENTOS QUE SE ACOMPAÑAN / FECHA Y FIRMA (23)

<input checked="" type="checkbox"/> DESCRIPCIÓN. Nº PÁGINAS: <input checked="" type="checkbox"/> Nº DE REIVINDICACIONES: <input checked="" type="checkbox"/> DIBUJOS. Nº PÁGINAS: <input type="checkbox"/> LISTA DE SECUENCIAS. Nº PÁGINAS: <input type="checkbox"/> SOPORTE LEGIBLE POR ORDENADOR DE LISTA DE SECUENCIAS <input checked="" type="checkbox"/> RESUMEN <input type="checkbox"/> FIGURA A PUBLICAR EN BOPI Nº: ____ <input type="checkbox"/> DOCUMENTO DE PRIORIDAD <input type="checkbox"/> TRADUCCIÓN DEL DOCUMENTO DE PRIORIDAD	<input type="checkbox"/> DOCUMENTO DE REPRESENTACIÓN <input type="checkbox"/> JUSTIFICANTE DEL PAGO DE TASA DE SOLICITUD <input type="checkbox"/> HOJA DE INFORMACIÓN COMPLEMENTARIA <input type="checkbox"/> PRUEBAS DE LOS DIBUJOS <input type="checkbox"/> SOLICITUD CAP <input type="checkbox"/> OTROS:	FIRMA DEL SOLICITANTE O REPRESENTANTE  Manuel Bethencourt Núñez-Vicerecotor Investigación-Transferencia FIRMA DEL FUNCIONARIO 
--	--	--

Ejemplar para el solicitante

ADVERTENCIA: POR DISPOSICIÓN LEGAL LOS DATOS CONTENIDOS EN ESTA SOLICITUD PODRÁN SER PUBLICADOS EN EL BOLETÍN OFICIAL DE LA PROPIEDAD INDUSTRIAL E INSCRITOS EN EL REGISTRO DE PATENTES DE LA OEPM, SIENDO AMBAS BASES DE DATOS DE CARÁCTER PÚBLICO Y ACCESIBLES VÍA REDES MUNDIALES DE INFORMÁTICA

OFICINA ESPAÑOLA DE PATENTES Y MARCAS (OEPM)
 Paseo de la Castellana, 75 - 28071 Madrid (España)

3101 (06.13)

Pág. 2 de 2



MINISTERIO DE INDUSTRIA, ENERGÍA Y TURISMO



Oficina Española de Patentes y Marcas

Nº SOLICITUD:
P 2 0 1 3 0 1 2 0 0

FECHA Y HORA DE ENTRADA EN OEPM:

FECHA Y HORA DE ENTRADA EN LUGAR DISTINTO A LA OEPM:
Cadix 24-12-13

HOJA DE INFORMACIÓN COMPLEMENTARIA

1. IDENTIFICACIÓN DE LA SOLICITUD

(1) MODALIDAD:
 PATENTE DE INVENCION MODELO DE UTILIDAD

2. IDENTIFICACIÓN DE LOS SOLICITANTES (2)

(3) APELLIDOS Y NOMBRE / DENOMINACIÓN SOCIAL / INFORMACIÓN RELATIVA A UNIVERSIDAD PÚBLICA ESPAÑOLA	PAÍS NAC.	CÓD. NAC.	NIF/PASAPORTE	(4) ESTA PERSONA TAMBIÉN ES INVENTOR: <input type="checkbox"/> SI <input type="checkbox"/> NO	(5) MODO DE OBTENCIÓN DEL DERECHO: <input type="checkbox"/> INVENCION LABORAL <input type="checkbox"/> CONTRATO <input type="checkbox"/> SUCESIÓN <input type="checkbox"/> OTROS (Especificar): _____ PORCENTAJE DE TITULARIDAD: %
ESTE SOLICITANTE ES UNIVERSIDAD PÚBLICA: <input type="checkbox"/> SI				<input type="checkbox"/> SI <input type="checkbox"/> NO	<input type="checkbox"/> INVENCION LABORAL <input type="checkbox"/> CONTRATO <input type="checkbox"/> SUCESIÓN <input type="checkbox"/> OTROS (Especificar): _____ PORCENTAJE DE TITULARIDAD: %
ESTE SOLICITANTE ES UNIVERSIDAD PÚBLICA: <input type="checkbox"/> SI				<input type="checkbox"/> SI <input type="checkbox"/> NO	<input type="checkbox"/> INVENCION LABORAL <input type="checkbox"/> CONTRATO <input type="checkbox"/> SUCESIÓN <input type="checkbox"/> OTROS (Especificar): _____ PORCENTAJE DE TITULARIDAD: %
ESTE SOLICITANTE ES UNIVERSIDAD PÚBLICA: <input type="checkbox"/> SI				<input type="checkbox"/> SI <input type="checkbox"/> NO	<input type="checkbox"/> INVENCION LABORAL <input type="checkbox"/> CONTRATO <input type="checkbox"/> SUCESIÓN <input type="checkbox"/> OTROS (Especificar): _____ PORCENTAJE DE TITULARIDAD: %
ESTE SOLICITANTE ES UNIVERSIDAD PÚBLICA: <input type="checkbox"/> SI				<input type="checkbox"/> SI <input type="checkbox"/> NO	<input type="checkbox"/> INVENCION LABORAL <input type="checkbox"/> CONTRATO <input type="checkbox"/> SUCESIÓN <input type="checkbox"/> OTROS (Especificar): _____ PORCENTAJE DE TITULARIDAD: %
ESTE SOLICITANTE ES UNIVERSIDAD PÚBLICA: <input type="checkbox"/> SI				<input type="checkbox"/> SI <input type="checkbox"/> NO	<input type="checkbox"/> INVENCION LABORAL <input type="checkbox"/> CONTRATO <input type="checkbox"/> SUCESIÓN <input type="checkbox"/> OTROS (Especificar): _____ PORCENTAJE DE TITULARIDAD: %

NOTA: DE NO ESPECIFICARSE EL PORCENTAJE DE TITULARIDAD, LA OEPM PRESUMIRÁ IGUALES LAS CUOTAS DE LOS SOLICITANTES.

3. IDENTIFICACIÓN DE LOS INVENTORES (Incluir sólo aquellos inventores que no figuren como solicitantes) (6)

Ejemplar para el solicitante	APELLIDOS Y NOMBRE	PAÍS NACIONALIDAD	CODIGO NAC.	NIF/PASAPORTE
	MASCAREÑAS PEREZ-IÑIGO, CARLOS	ESPAÑOLA	ES	00796438V
PALMA GUERRERO, JUAN JOSE	ESPAÑOLA	ES	31218149L	

P.201301200

4. OTROS DATOS

(7) EXPOSICIONES OFICIALES			
NOMBRE	FECHA		LUGAR

(8) REIVINDICACIÓN DE PRIORIDAD			
PAIS ORIGEN	CÓDIGO PAIS	FECHA	NÚMERO

Ejemplar para el solicitante

OFICINA ESPAÑOLA DE PATENTES Y MARCAS (OEPM)
Paseo de la Castellana, 75 - 28071 Madrid (España)

3102 (06.13)

2 de 2

UNIVERSIDAD DE CÁDIZ
C/ Ancha, 16
11001 Cádiz
Cádiz

Madrid, a 3 de enero de 2014

Admisión a trámite de la solicitud de Patente Nacional 201301200

La Oficina Española de Patentes y Marcas (OEPM) le comunica que su solicitud de patente 201301200 ha sido admitida a trámite con asignación de fecha de presentación correspondiente al día 24/12/2013 y que no se encuentra afectada por lo previsto en el Título XII de la Ley 11/1986, de 20 de marzo, de Patentes, relativo a patentes secretas.

De acuerdo con el artículo 59.2 de la Ley 11/1986 de Patentes, a partir de la fecha antes mencionada usted podría gozar de una protección provisional frente a cualquier tercero que hubiera llevado a cabo una utilización de la invención siempre y cuando notifique a dicho tercero la presentación y el contenido de esta solicitud. Esta protección implicaría el derecho a exigir una indemnización razonable si dicho tercero prosiguiera utilizando su invención entre la fecha de la notificación y la fecha de publicación de la mención en el Boletín Oficial de la Propiedad Industrial (BOPI) de que la patente ha sido concedida. El citado derecho existiría a partir de la fecha de notificación fehaciente y se podría ejercer una vez se publicara la mención de la concesión en el BOPI. El BOPI puede consultarse en la web de la OEPM.

La OEPM le remitirá cumplida información de las diferentes etapas del procedimiento de concesión con indicación de los actos a llevar a cabo, los plazos para cumplimentarlos y las fechas a partir de las cuales dichos plazos comienzan a contar.

Para cualquier consulta en relación con su expediente, puede ponerse en contacto con la OEPM llamando al teléfono de información 902 157 530 (en horario de 9:00 a 14:30, de lunes a viernes).

Atentamente,



Fdo.: Leopoldo Belda Soriano
Jefe de Área de Examen de Patentes

5.10.2 Demostrador de conceptos de radiofrecuencia con aplicaciones múltiples en el campo de la Enseñanza Superior.

SECTOR DE LA TÉCNICA.

Enseñanza de la Física y la Tecnología de los Campos Electromagnéticos y las Radiocomunicaciones.

ESTADO DE LA TÉCNICA.

Actualmente se enseñan los fundamentos de los campos electromagnéticos mediante los conceptos de la Física Clásica, la formulación matemática y a veces la simulación mediante programas de computador, pero se basa fundamentalmente en ejercicios matemáticos que acaban siendo un mero resultado numérico o gráfico, sin ningún significado concreto para el alumno.

La invención propuesta supone una herramienta que permite el análisis de los campos electromagnéticos y su medición, así como la realización de prácticas individuales o grupales con resultados prácticos y que mejoran la asimilación de los conceptos de la teoría aprendida en clase con anterioridad o la generación de dichos conceptos cuando se practica el método de la enseñanza por descubrimiento.

La invención permite la demostración de los conceptos de, entre otros:

- Aceleración
- Acelerómetro
- Amplificación
- Amplificadores de línea
- Amplitud
- Análisis de Espectros
- Análisis de Señales Temporales
- Ángulo Horario
- Antenas
- Ascensión Recta
- Azimut
- Bandas Laterales
- Cables coaxiales
- Cascada de señales
- Compás Giroscópico
- Compás Magnético
- Control automático
- Declinación
- Demora
- Derivación

- Dirección de tiro
- Distorsión
- Efeméride
- Elevación
- Equipos de radio definidos por software.
- Error.
- Espectro de frecuencias.
- Foco
- Frecuencia
- Frecuencia de portadora
- Frecuencia Intermedia
- Ganancia
- Horizonte Radio
- Impedancia
- Índice de Modulación.
- Interferencia
- Leyes de Maxwell
- Longitud de Onda
- Marcación
- Mezcla
- Modulaciones analógicas y digitales.
- Nadir
- Norte Magnético
- Norte Verdadero
- Órbita
- Oscilador Local
- Pérdidas en los cables coaxiales
- Polarización
- Procesado Digital de la Señal.
- Radio detection and ranking
- Radio direction finding
- Radio Sobre Protocolo Internet
- Radioastronomía
- Radiodemora
- Radiotelescopio
- Realimentación.
- Receptores de radiofrecuencia
- Retardo de propagación
- Rotación terrestre
- Ruidos térmicos, Johnson y Flicker.
- Satélite
- Seguimiento de Satélites
- Señales IQ.
- Tiempo de Respuesta
- Tiempo de Retardo.
- Tiempo de Subida
- Zenit

EXPLICACIÓN DE LA INVENCION

La invención consiste en un sistema completo de enseñanza compuesto por:

a) Una plataforma metálica autoportada que está formada por un mástil metálico vertical de diámetro adecuado para instalar sobre su cúspide un rotor de antenas y en la parte inferior una base lo suficientemente consistente como para poder soportar el peso del rotor más sus antenas y el tubo del mástil. Este mástil puede ser intercambiado por un trípode, polín o base adecuada.

b) Un rotor de antenas que disponga de giros para permitir la elevación vertical y rotación horizontal de las antenas de radio que se instalarán sobre una cruceta, verga o soporte adecuado, en el enganche fabricado a propósito. Este rotor será controlado mediante cables eléctricos y dispositivo controlador del rotor, ya sea de forma manual o por ordenador.

c) Antenas en las principales bandas de frecuencias a estudiar desde el punto de vista de la enseñanza de la física, las radiocomunicaciones y la radioastronomía, esto es:

Banda Ancha en 45-900 MHz

Banda Ancha de 900 a 2700 MHz.

Banda Ancha de 3 a 4 GHz, Banda C

Banda Ancha de 10,9 a 12,7 GHz, Banda Ku

Dipolo cerrado de 406 MHz

Yagi para 1420 MHz

Dipolo Cerrado en 1420 MHz

d) Amplificadores de línea para todas las antenas.

e) Receptores de radio definidos por software para las frecuencias inferiores a 900 MHz, con integración a ordenador mediante conexión USB.

f) Un receptor de televisión analógica vía satélite y otro receptor de televisión digital vía satélite por cada antena de Banda Ancha de 3 a 4 GHz, Banda C

g) Un receptor de televisión analógica vía satélite y otro receptor de televisión digital vía satélite por cada antena de Banda Ancha de 10,9 a 12,7 GHz, Banda Ku

h) Un receptor de radio definido por software para las frecuencias inferiores a 900 MHz, con integración a ordenador mediante conexión USB por cada receptor de televisión por satélite.

- i) Un receptor de televisión analógica/digital/vídeo/audio por cada receptor de televisión vía satélite.
- j) Un sistema informático de proceso de señales y datos.
- k) Un sistema informático de distribución de datos.
- l) Instrumental de laboratorio de radiotecnica y electrónica, como osciloscopio, frecuencímetro, analizador de espectros, generador de radiofrecuencia.
- m) Software apropiado para el análisis de señales en los dominios del tiempo y la frecuencia.

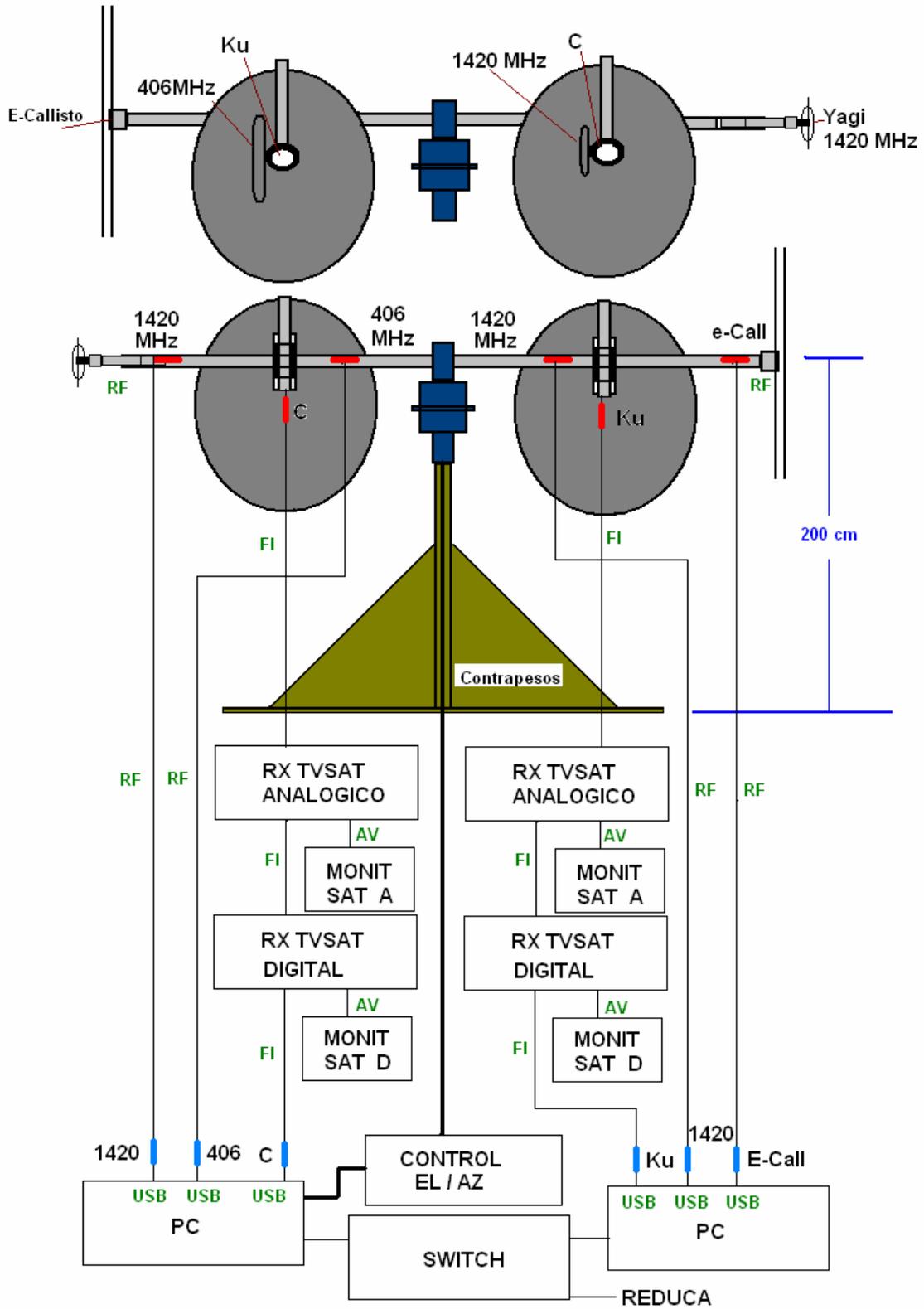


Figura 5.10.3 Diagrama de Bloques Simplificado de la Invención

5.10.3 Demostrador de conceptos de radiocomunicaciones vía satélites ecuatoriales con aplicaciones múltiples en el campo de la enseñanza superior.

SECTOR DE LA TÉCNICA.

Enseñanza de la Física y la Tecnología de los Campos Electromagnéticos y las Radiocomunicaciones.

ESTADO DE LA TÉCNICA.

Actualmente se enseñan los fundamentos de los campos electromagnéticos mediante los conceptos de la Física Clásica, la formulación matemática y a veces la simulación mediante programas de computador, pero se basa fundamentalmente en ejercicios matemáticos que acaban siendo un mero resultado numérico o gráfico, sin ningún significado concreto para el alumno.

La invención propuesta supone una herramienta que permite el análisis de los campos electromagnéticos y su medición, así como la realización de prácticas individuales o grupales con resultados prácticos y que mejoran la asimilación de los conceptos de la teoría aprendida en clase con anterioridad o la generación de dichos conceptos cuando se practica el método de la enseñanza por descubrimiento.

La invención permite la demostración de los conceptos de, entre otros:

- Aceleración
- Acelerómetro
- Amplificación
- Amplificadores de línea
- Amplitud
- Análisis de Espectros
- Análisis de Señales Temporales
- Ángulo Horario
- Antenas
- Ascensión Recta
- Azimut
- Bandas Laterales
- Cables coaxiales
- Cascada de señales
- Compás Giroscópico
- Compás Magnético
- Control automático
- Declinación
- Demora

- Derivación
- Dirección de tiro
- Distorsión
- Efeméride
- Elevación
- Equipos de radio definidos por software.
- Error.
- Espectro de frecuencias.
- Foco
- Frecuencia
- Frecuencia de portadora
- Frecuencia Intermedia
- Ganancia
- Horizonte Celeste
- Horizonte Radio
- Horizonte Terrestre
- Impedancia
- Índice de Modulación.
- Interferencia
- Leyes de Maxwell
- Longitud de Onda
- Marcación
- Mezcla
- Modulaciones analógicas y digitales.
- Nadir
- Norte Magnético
- Norte Verdadero
- Órbita
- Oscilador Local
- Pérdidas en los cables coaxiales
- Polarización
- Procesado Digital de la Señal.
- Radio detection and ranking
- Radio direction finding
- Radio Sobre Protocolo Internet
- Radioastronomía
- Radiodemora
- Radiotelescopio
- Realimentación.
- Receptores de radiofrecuencia
- Retardo de propagación
- Rotación terrestre
- Ruidos térmico, Johnson y Flicker.
- Satélite
- Seguimiento de Satélites
- Señales IQ.
- Tiempo de Respuesta

- Tiempo de Retardo.
- Tiempo de Subida
- Zenit

EXPLICACIÓN DE LA INVENCION

La invención consiste en un sistema completo de enseñanza compuesto por:

a)Una plataforma metálica autosoportada que está formada por un soporte de antenas en forma de L, U u otra forma física y sujeto a una superficie vertical o bien un mástil metálico vertical de diámetro adecuado para instalar sobre su cúspide un rotor de antenas y en la parte inferior una base lo suficientemente consistente como para poder soportar el peso del rotor más sus antenas y el tubo del mástil. Este mástil puede ser intercambiado por un trípode, polín o base adecuada.

b)Un rotor de antenas que disponga de giro para permitir la rotación horizontal de las antenas parabólicas o semicirculares de radio que se instalarán sobre el enganche fabricado a propósito. Este rotor será controlado mediante uno o varios cables eléctricos y dispositivo controlador del rotor, ya sea de forma manual o por ordenador.

c)Una o más antenas en las principales bandas de frecuencias a estudiar desde el punto de vista de la enseñanza de la física, las radiocomunicaciones y la radioastronomía, esto es:

Banda Ancha de 900 a 2700 MHz.

Banda Ancha en 1500 a 1700 MHz

Banda Ancha de 3 a 4 GHz, Banda C

Banda Ancha de 10,9 a 12,7 GHz, Banda Ku

Dipolo cerrado de 406 MHz

Dipolo cerrado de 435 MHz

Dipolo cerrado de 1240 MHz

Dipolo Cerrado en 1420 MHz

Las antenas podrán instalarse de forma individual o grupal, utilizando cavidades resonantes si fuera necesario y una o más polaridades.

d)Un amplificador común o Amplificadores de línea para todas las antenas.

e)Receptores de radio definidos por software para las frecuencias inferiores a 900 MHz, con integración a ordenador mediante conexión USB.

- f)Un receptor de televisión analógica vía satélite y otro receptor de televisión digital vía satélite por cada antena de Banda Ancha de 3 a 4 GHz, Banda C
- g)Un receptor de televisión analógica vía satélite y otro receptor de televisión digital vía satélite por cada antena de Banda Ancha de 10,9 a 12,7 GHz, Banda Ku
- h)Un receptor de radio definido por software para las frecuencias inferiores a 1700 MHz, con integración a ordenador mediante conexión USB por cada receptor de televisión por satélite.
- i)Un receptor de televisión analógica/digital/vídeo/audio/VGA por cada receptor de televisión vía satélite.
- j)Un sistema informático de proceso de señales y datos.
- k)Un sistema informático de distribución de datos.
- l)Instrumental de laboratorio de radiotecnica y electrónica, como osciloscopio, frecuencímetro, analizador de espectros, generador de radiofrecuencia.
- m)Software apropiado para el análisis de señales en los dominios del tiempo y la frecuencia.
- n)Un sistema electrónico de referenciación geográfica.

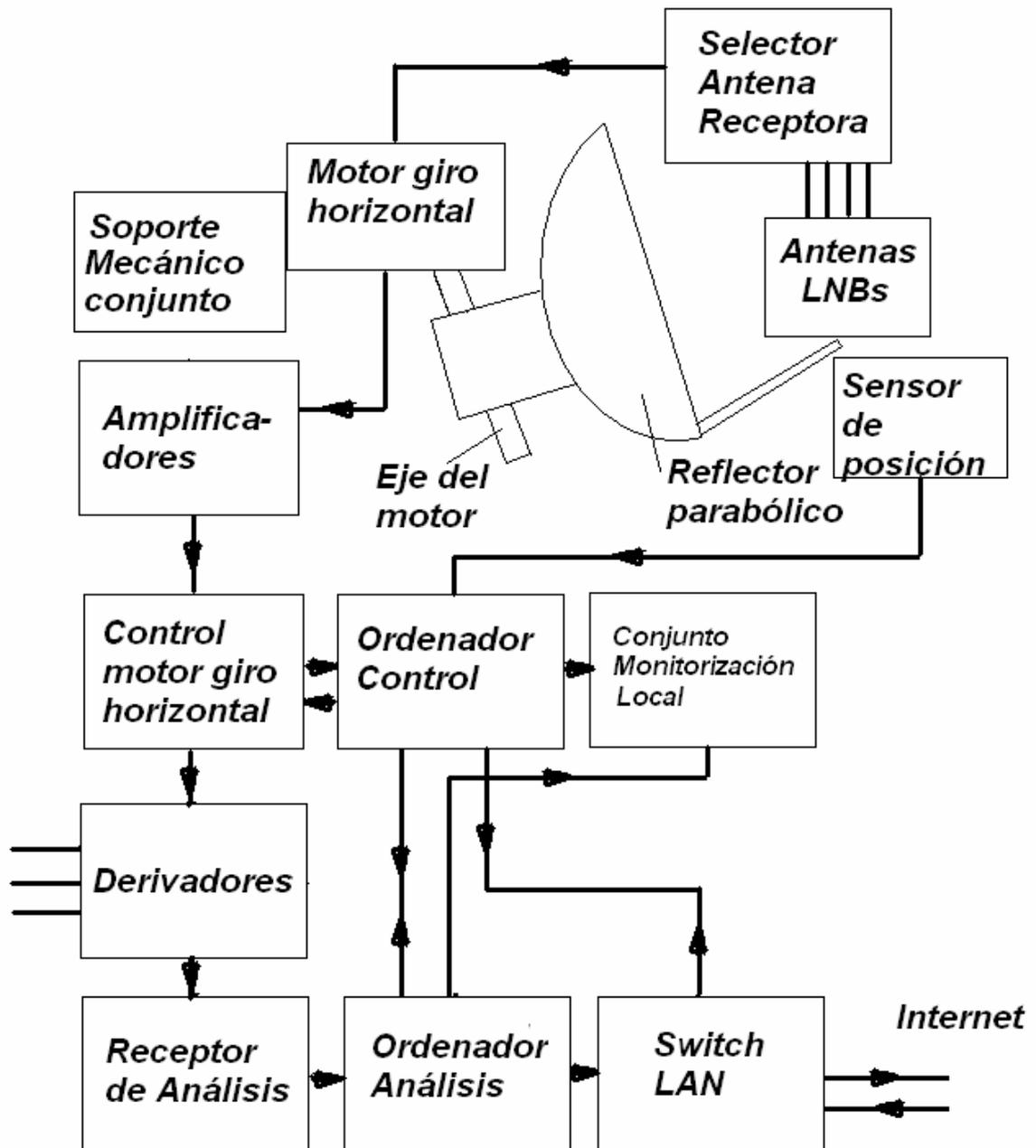


Figura 5.10.4 Diagrama de Bloques Simplificado de la Invención



Nº SOLICITUD: P 201400105
FECHA Y HORA DE ENTRADA EN DEPA:
FECHA Y HORA DE ENTRADA EN LUGAR DESTINADO A LA DEPA: <i>Cádiz 7-2-14</i>
LUGAR DE PRESENTACIÓN: <i>Cádiz</i> CÓDIGO:

INSTANCIA DE SOLICITUD



1. IDENTIFICACIÓN DE LA SOLICITUD

(1) MODALIDAD: <input checked="" type="checkbox"/> PATENTE DE INVENCION <input type="checkbox"/> MODELO DE UTILIDAD		(3) EXPEDIENTE PRINCIPAL O DE ORIGEN:
(2) TIPO DE SOLICITUD: <input type="checkbox"/> ADICIÓN A LA PATENTE <input type="checkbox"/> SOLICITUD DIVISIONAL <input type="checkbox"/> CAMBIO DE MODALIDAD <input type="checkbox"/> TRANSFORMACIÓN SOLICITUD PATENTE EUROPEA <input type="checkbox"/> ENTRADA EN FASE NACIONAL DE SOLICITUD PCT		MODALIDAD: Nº SOLICITUD: FECHA PRESENTACIÓN:

2. TÍTULO DE LA INVENCION (4)

DEMONSTRADOR DE CONCEPTOS DE RADIOCOMUNICACIONES VÍA SATÉLITES ECUATORIALES CON APLICACIONES MÚLTIPLES EN EL CAMPO DE LA ENSEÑANZA SUPERIOR.

3. IDENTIFICACIÓN DEL SOLICITANTE

(5) APELLIDOS Y NOMBRE (DENOMINACIÓN SOCIAL)				NIF/REGISTRO	
UNIVERSIDAD DE CÁDIZ				Q11328810	
DIRECCIÓN POSTAL		CÓDIGO POSTAL Y LOCALIDAD	PROVINCIA	PAÍS RESIDENCIA	CÓDIGO PAÍS RESIDENCIA
C/ ANCHA, 16		11001	CÁDIZ	ESPAÑA	ES
PAÍS DE NACIONALIDAD		CÓDIGO PAÍS NACIONALIDAD	CIUDAD (6)	PYME (7)	
ESPAÑA		ES	CADIZ	S	
DIRECCIÓN CORREO ELECTRÓNICO	Nº TELÉFONO FIJO	Nº TELÉFONO MÓVIL	(8) INDICACIÓN DEL MEDIO DE NOTIFICACIÓN PREFERENTE		
omr@uca.es	958015173		<input type="checkbox"/> CORREO POSTAL <input checked="" type="checkbox"/> CORREO ELECTRÓNICO		
PORCENTAJE DE TITULARIDAD (9): 100 %	NOTA: DE NO ESPECIFICARSE DICHO PORCENTAJE, LA DEPA PRESUMIRÁ IGUALES LAS CUOTAS DE LOS SOLICITANTES.				
EL SOLICITANTE TAMBIÉN (10) <input type="checkbox"/> SI <input checked="" type="checkbox"/> NO COINVENTOR:	MODO DE OBTENCIÓN DEL DERECHO (11) <input checked="" type="checkbox"/> INVENCIÓN LABORAL <input type="checkbox"/> CONTRATO <input type="checkbox"/> SUCESIÓN <input type="checkbox"/> OTROS (Especificar):				
(12) EL SOLICITANTE ES UNA UNIVERSIDAD PÚBLICA ESPAÑOLA	<input checked="" type="checkbox"/> SI				

Ejemplar para el solicitante

4. OTROS SOLICITANTES Y/O INVENTORES (13)

LOS DEMÁS SOLICITANTES Y/O INVENTORES SE INDICAN EN HOJA COMPLEMENTARIA.

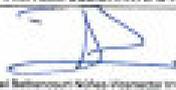
5. IDENTIFICACIÓN DEL REPRESENTANTE

(13) REPRESENTACIÓN		(16) Nº PODER JURISD.
<input checked="" type="checkbox"/> EL SOLICITANTE NO ESTÁ REPRESENTADO	EL SOLICITANTE ESTÁ REPRESENTADO POR: <input type="checkbox"/> AGENTE DE LA PROPIEDAD INDUSTRIAL <input type="checkbox"/> OTRO REPRESENTANTE	
(14) ACTUACIÓN POR MEDIO DE AGENTE	NOMBRE	CÓDIGO DE AGENTE
(15) ACTUACIÓN POR MEDIO DE OTRO REPRESENTANTE	NOMBRE	DIRECCIÓN POSTAL
		R.U.F.
DIRECCIÓN CORREO ELECTRÓNICO	Nº TELÉFONO	INDICACIÓN DEL MEDIO DE NOTIFICACIÓN PREFERENTE
		<input type="checkbox"/> CORREO POSTAL <input type="checkbox"/> CORREO ELECTRÓNICO

6. OTROS DATOS

(18) REIVINDICACIÓN DE PRIORIDAD	PAÍS ORIGEN	CÓDIGO PAÍS	FECHA	NOMBRE
<input type="checkbox"/> SI <input checked="" type="checkbox"/> NO				
(19) REIVINDICACIÓN DE OTORIDAD	NOMBRE		FECHA	LUGAR
<input type="checkbox"/> SI <input checked="" type="checkbox"/> NO				
(21) EFECTUADO DEPÓSITO DE MATERIAL BIOLÓGICO	AUTORIDAD DE DEPÓSITO	CÓDIGO PAÍS	FECHA	NÚMERO
<input type="checkbox"/> SI <input checked="" type="checkbox"/> NO				
LISTA DE SECUENCIAS DE AMINOÁCIDOS Y ÁCIDOS NUCLEICOS				
La descripción contiene un listado de secuencias biológicas en concordancia con la norma ST 25 OMP1 <input type="checkbox"/> SI <input checked="" type="checkbox"/> NO				
Se adjunta un soporte de datos legible por ordenador que incluye el listado de secuencias biológicas en concordancia con la norma ST 25 OMP1 <input type="checkbox"/> SI <input checked="" type="checkbox"/> NO				
El solicitante declara por medio de esta instancia, que la información registrada en el soporte de datos legible por ordenador es idéntica a la contenida en el listado de secuencias biológicas incluido en la descripción de la versión escrita de esta solicitud <input checked="" type="checkbox"/>				
(22) EL SOLICITANTE SE ACCEDE AL APLAZAMIENTO DE TASAS PREVISTO EN EL ART. 162 DE LA LEY 1/1986 DE PATENTES				<input type="checkbox"/> SI

7. ÍNDICE DE DOCUMENTOS QUE SE ACOMPAÑAN | FECHA Y FIRMA (23)

<input checked="" type="checkbox"/> DESCRIPCIÓN Nº PÁGINAS <input checked="" type="checkbox"/> Nº DE REIVINDICACIONES <input checked="" type="checkbox"/> DIBUJOS Nº PÁGINAS <input checked="" type="checkbox"/> LISTA DE SECUENCIAS Nº PÁGINAS <input type="checkbox"/> SOPORTE LEGIBLE POR ORDENADOR DE LISTA DE SECUENCIAS <input checked="" type="checkbox"/> RESUMEN <input type="checkbox"/> FOLIA A PUBLICAR EN BOPI Nº: ____ <input type="checkbox"/> DOCUMENTO DE PRIORIDAD <input type="checkbox"/> TRADUCCIÓN DEL DOCUMENTO DE PRIORIDAD	<input type="checkbox"/> DOCUMENTO DE REPRESENTACIÓN <input type="checkbox"/> JUSTIFICANTE DEL PAGO DE TASA DE SOLICITUD <input type="checkbox"/> FICHA DE INFORMACIÓN COMPLEMENTARIA <input type="checkbox"/> PRUEBAS DE LOS DIBUJOS <input type="checkbox"/> SOLICITUD CAP <input type="checkbox"/> OTROS:	FIRMA DEL SOLICITANTE O REPRESENTANTE  Manuel Batemanou / Oficina Asesora de Investigación Científica FIRMA DEL FUNCIONARIO 
---	---	---

Ejemplar para el solicitante

ADVERTENCIA: POR DISPOSICIÓN LEGAL LOS DATOS CONTENIDOS EN ESTA SOLICITUD PODRÁN SER PUBLICADOS EN EL BOLETÍN OFICIAL DE LA PROPIEDAD INDUSTRIAL E INSCRITOS EN EL REGISTRO DE PATENTES DE LA DEPM, SIENDO AMBAS BASES DE DATOS DE CARÁCTER PÚBLICO Y ACCESIBLES VÍA REDES MUNDIALES DE INFORMÁTICA.

OFICINA ESPAÑOLA DE PATENTES Y MARCAS (OEPM)
 Paseo de la Castellana, 141 - 28071 Madrid (España)

3101 (05.13)

Pág. 2 de 2



MINISTERIO DE INDUSTRIA, ENERGÍA Y TURISMO

Oficina Española de Patentes y Marcas

Nº SOLICITO:
P 201400105

FECHA Y HORA DE ENTRADA EN DEPÓSITO:

FECHA Y HORA DE ENTRADA EN JUEGOS DISTINTO A LA OEPM:
Cádiz 7-7-14

HOJA DE INFORMACIÓN COMPLEMENTARIA



1. IDENTIFICACIÓN DE LA SOLICITUD

(1) MODALIDAD:
 PATENTE DE INVENCION MODELO DE UTILIDAD Cádiz

2. IDENTIFICACIÓN DE LOS SOLICITANTES (3)

(3) APELLIDOS Y NOMBRE / DENOMINACIÓN SOCIAL / INFORMACIÓN RELATIVA A UNIVERSIDAD PÚBLICA ESPAÑOLA	PAS. NAC.	CÓD. NAC.	IMP. PASAPORTE	(4) ESTA PERSONA TAMBIÉN ES INVENTOR:	(5) MODO DE OBTENCIÓN DEL DERECHO:
ESTE SOLICITANTE ES UNIVERSIDAD PÚBLICA: <input type="checkbox"/> SI				<input type="checkbox"/> SI <input type="checkbox"/> NO	<input type="checkbox"/> INVENCION LABORAL <input type="checkbox"/> CONTRATO <input type="checkbox"/> SUCESIÓN <input type="checkbox"/> OTROS (Especificar): _____ PORCENTAJE DE TITULARIDAD: %
ESTE SOLICITANTE ES UNIVERSIDAD PÚBLICA: <input type="checkbox"/> SI				<input type="checkbox"/> SI <input type="checkbox"/> NO	<input type="checkbox"/> INVENCION LABORAL <input type="checkbox"/> CONTRATO <input type="checkbox"/> SUCESIÓN <input type="checkbox"/> OTROS (Especificar): _____ PORCENTAJE DE TITULARIDAD: %
ESTE SOLICITANTE ES UNIVERSIDAD PÚBLICA: <input type="checkbox"/> SI				<input type="checkbox"/> SI <input type="checkbox"/> NO	<input type="checkbox"/> INVENCION LABORAL <input type="checkbox"/> CONTRATO <input type="checkbox"/> SUCESIÓN <input type="checkbox"/> OTROS (Especificar): _____ PORCENTAJE DE TITULARIDAD: %
ESTE SOLICITANTE ES UNIVERSIDAD PÚBLICA: <input type="checkbox"/> SI				<input type="checkbox"/> SI <input type="checkbox"/> NO	<input type="checkbox"/> INVENCION LABORAL <input type="checkbox"/> CONTRATO <input type="checkbox"/> SUCESIÓN <input type="checkbox"/> OTROS (Especificar): _____ PORCENTAJE DE TITULARIDAD: %
ESTE SOLICITANTE ES UNIVERSIDAD PÚBLICA: <input type="checkbox"/> SI				<input type="checkbox"/> SI <input type="checkbox"/> NO	<input type="checkbox"/> INVENCION LABORAL <input type="checkbox"/> CONTRATO <input type="checkbox"/> SUCESIÓN <input type="checkbox"/> OTROS (Especificar): _____ PORCENTAJE DE TITULARIDAD: %

NOTA: DE NO ESPECIFICARSE EL PORCENTAJE DE TITULARIDAD, LA OEPM PRESUMIRÁ IGUALES LAS CUOTAS DE LOS SOLICITANTES.

3. IDENTIFICACIÓN DE LOS INVENTORES (Incluir sólo aquellos inventores que no figuren como solicitantes) (6)

APELLIDOS Y NOMBRE	NAC. NACIONALIDAD	CÓDIGO NAC.	IMP. PASAPORTE
MASCAREÑAS PEREZ-IÑIGO, CARLOS	ESPAÑOLA	ES	00796438V
PALMA GUERRERO, JUAN JOSE	ESPAÑOLA	ES	31218149L

Ejemplar para el solicitante

4 OTROS DATOS

EXPOSICIONES OFICIALES			
LOCALIDAD	FECHA	LUGAR	
REGISTRACIÓN DE PROYECTO			
REG. DISEÑO	CÓDIGO PAIS	FECHA	NÚMERO

Ejemplar para el solicitante

OFICINA ESPAÑOLA DE PATENTES Y MARCAS (OEPM)
Paseo de la Castellana, 75 - 28071 Madrid (España)

3100 (94.10)

2 de 2

UNIVERSIDAD DE CÁDIZ
C/ Ancha, 16
11001 Cádiz
Cádiz

Madrid, a 17 de febrero de 2014

Admisión a trámite de la solicitud de Patente Nacional 201400105

La Oficina Española de Patentes y Marcas (OEPM) le comunica que su solicitud de patente 201400105 ha sido admitida a trámite con asignación de fecha de presentación correspondiente al día 07/02/2014 y que no se encuentra afectada por lo previsto en el Título XII de la Ley 11/1986, de 20 de marzo, de Patentes, relativo a patentes secretas.

De acuerdo con el artículo 59.2 de la Ley 11/1986 de Patentes, a partir de la fecha antes mencionada usted podría gozar de una protección provisional frente a cualquier tercero que hubiera llevado a cabo una utilización de la invención siempre y cuando notifique a dicho tercero la presentación y el contenido de esta solicitud. Esta protección implicaría el derecho a exigir una indemnización razonable si dicho tercero prosiguiera utilizando su invención entre la fecha de la notificación y la fecha de publicación de la mención en el Boletín Oficial de la Propiedad Industrial (BOPI) de que la patente ha sido concedida. El citado derecho existiría a partir de la fecha de notificación fehaciente y se podría ejercer una vez se publicara la mención de la concesión en el BOPI. El BOPI puede consultarse en la web de la OEPM.

La OEPM le remitirá cumplida información de las diferentes etapas del procedimiento de concesión con indicación de los actos a llevar a cabo, los plazos para cumplimentarlos y las fechas a partir de las cuales dichos plazos comienzan a contar.

Para cualquier consulta en relación con su expediente, puede ponerse en contacto con la OEPM llamando al teléfono de información 902 157 530 (en horario de 9:00 a 14:30, de lunes a viernes).

Atentamente,



Fdo.: Eduardo Martín Pérez
Jefe de Área de Examen de Patentes

5.11. Artículos derivados de esta Tesis Doctoral.

5.11.1 Artículo publicado en revista Radioaficionados (URE).

RADIOAFICIONADOS es una publicación mensual de carácter social, que reciben todos los socios de la URE en sus domicilios. En ella se informa de los temas de interés para el radioaficionado, como son:

Divulgación técnica. Los artículos que se publican en el área de divulgación técnica son, en su mayor parte, originales enviados por los propios asociados. También se seleccionan artículos escogidos de las principales revistas de asociaciones miembros de la IARU que, una vez traducidos, son publicados con sus créditos correspondientes. Los artículos técnicos tratan en su mayoría de antenas ya que, en el estado actual de la técnica, es éste el campo donde principalmente pueden experimentar los radioaficionados, sin que por ello queden fuera del ámbito de la revista otro tipo de artículos y experimentos relacionados con los diversos aparatos y componentes de una estación de radioaficionado. El nivel de los artículos técnicos es variable: unas veces van dirigidos al radioaficionado novel y otras, al que ya conoce y practica más en profundidad este "hobby".

HF. El aficionado al DX encontrará en esta sección de la revista cumplida información de las expediciones que se realizan cada año a distintos lugares de nuestro planeta, de las estaciones de países más buscados, de managers de QSL, además de noticias y reportajes de actividades a nivel nacional (expediciones a islas, castillos, etc.).

VHF. Las bandas altas (VHF, UHF, Microondas) también tienen cabida en estas páginas, con información de concursos, actividades especiales de experimentación, modos de propagación, etc.

Satélites. La actividad de los radioaficionados a través de sus propios satélites queda reflejada en la revista mediante la publicación periódica de las frecuencias, condiciones y modos de operación de los satélites, además de otros aspectos técnicos.

Concursos y Diplomas. Casi todos los fines de semana se celebran concursos nacionales e internacionales. Las bases completas de cada uno de ellos se publican en la revista del mes anterior a su celebración. Al radioaficionado que le gusten los diplomas tiene donde elegir entre más de 2.400 diplomas que promueven las sociedades de radioaficionados de todo el mundo. En la revista *Radioaficionados* se publican las novedades que se dan en esta faceta y, periódicamente, las bases de los diplomas más prestigiosos.

Actividades de la URE y sus Secciones. Dado el carácter social de la revista, ésta también acoge la información de las actividades propias de la URE y de los actos sociales de las distintas Secciones, como son las celebraciones del llamado Día del Radioaficionado, entrega de premios de concursos, etc.

Demostrador de conceptos de radiofrecuencia

Para ejercicios de las asignaturas del grado de ingeniería radioelectrónica

Autores: Juan José Palma Guerrero,
Carlos Mascareñas y Pérez-Iñigo (EA7GWJ)
Departamento de Ciencias y Técnicas de la Navegación
Escuela de Ingeniería Radioelectrónica. Universidad de Cádiz.
carlos.mascarenas@uca.es



Resumen: El rápido avance de la informática y la microelectrónica permiten revolucionar los sistemas tradicionales de enseñanza de la radioelectrónica civil, utilizando tecnologías que eran impensables hace diez años, sobre todo por su alto coste y contenido militar.

El alumno del Grado de Ingeniería Radioelectrónica tiene la posibilidad de "ver", escuchar y manipular la radiofrecuencia recibida en el C.A.S.E.M. desde un ordenador ubicado en el taller de su escuela, el aula de teoría o su propio domicilio gracias a receptores de banda ancha que le envían datos procesados en tiempo real a través de Internet, posibilitando la realización de prácticas académicas programadas fuera del horario lectivo, cosa impensable hasta ahora.

En este artículo se analiza un proyecto de Demostrador de Conceptos que puede ser compartido por todas las universidades del mundo con el fin de mejorar el rendimiento del tiempo dedicado por el alumnado al aprendizaje del análisis de radiocomunicaciones.

Palabras-clave: Enseñanza; Entrenamiento; Radiocomunicaciones; Radios Definidas por Software, Redes.

1. Introducción.

El estado actual de la tecnología hace que por muy poco dinero, comparado con hace 10 o 20 años, se pueda disponer de una serie de receptores instalados en la Escuela de Ingeniería y Radioelectrónica y al mismo tiempo incrementar la formación de los ingenieros radioelectrónicos con técnicas similares a las de los equipos de Guerra Electrónica (EW), los cuales permiten la perfecta comprensión de conceptos a los que antes sólo se llegaba mediante la imaginación del alumno y la pericia del profesor simulándolos en la pizarra o en un ordenador.

Estos receptores EW (Lackey, R. I. y Upmal, D. W., 1995) están vetados para el público civil y son de amplio uso en las Armadas de todo el mundo, siendo el puesto de Oficial de Comunicaciones uno de los más codiciados en cuanto a las especialidades de la Armada se refiere, ya que asegura un puesto de trabajo como asesor militar de aquellas empresas civiles que proveen de tecnología a los Ministerios de Defensa de todo el mundo.

Los receptores SDR constan de una parte hardware que reciben la señal radio y proveen la señal audio IQ al software implementado en un equipo PC u otros equipos capaces de correr el software SDR. La conexión de los receptores de radio hardware, se puede hacer bien por IP, USB o RS232, permitiendo el software SDR la arquitectura cliente servidor, así como la teledescarga de soluciones.

La diferencia fundamental de los receptores SDR con los receptores normales es que son de banda ancha, es decir no disponen de filtro de frecuencia intermedia y su operación se realiza mediante la sintonización software de los mismos en el puesto re-

ceptor, lo que permite que las mismas señales lleguen a todos los operadores, siendo éstos los que seleccionan el canal a escuchar en cada momento. (Mitola, J. 2000. Burns, P. 2002. Youngblood, G. 2002-2003. Kenington, P.B, 2005. Galvis, A., Ceballos, C.A., y De Sanctis Gil, L. 2007.)

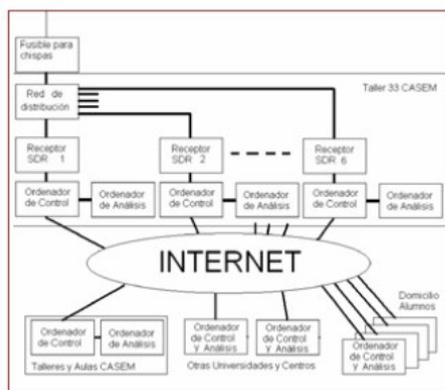


Figura 1. Diagrama parcial conceptual del Demostrador de Conceptos.

La red que estamos utilizando se compone de un conjunto de 6 estaciones receptoras SDR para cubrir las bandas de 0 a 1700 MHz, instalados en el Taller del Grupo Señales, Sistemas y Comunicaciones Navales y se pretende que pueda ser utilizada desde cualquier punto habilitado de la geografía nacional mediante la red Internet con acceso a través de la página web del sistema Marconi (<http://marconi.uca.es>), dando nuestra Universidad un servicio de valor añadido al público.

Los receptores SDR adquiridos e instalados por el Grupo disponen de distintos niveles de acceso según sea su aplicación, dependiendo de las bandas de frecuencias atribuidas a los servicios de radiocomunicaciones estudiados en cada momento.

Por ejemplo, mientras que hay un receptor Softrock en la banda de 40 m a disposición de todo el público dentro del proyecto Websdr de la Universidad de Twente (Twente 2012) y se ha terminado la instalación de un receptor de la banda de 2m para el mismo proyecto, el receptor HF Perseus se encuentra accesible a través del desarrollo del grupo QtRadio cuyo ejecutable se encuentra accesible en la dirección marconi.uca.es.

Asimismo, están disponibles receptores Softrock, Funcube, SDR-IQ y RTL-SDR bajo una arquitectura cliente-servidor usando el conocido programa SDR-Radio, pudiéndose acceder a ellos en modo local, o bien de forma remota con los cliente SDR-Radio, ya sea desde aulas de informática o mediante los medios audiovisuales de las aulas.

Además, y tanto para la docencia de los procedimientos radioeléctricos como para la intercomunicación de los alumnos durante las prácticas, disponen de un servidor Team Speak 3 en el que es posible mantener conversaciones en tiempo real, tanto en castellano como en inglés, pulsando un "PTT" y utilizando micrófonos y auriculares.

Para publicar estos medios, el grupo S2CN de la Universidad de Cádiz ha creado la página web marconi.uca.es que va describiendo los equipos y sistema a los que se puede ir accediendo mediante el desarrollo MARCONI creado para tal fin por este grupo.

Como una pequeña descripción del desarrollo Marconi apuntar que éste consta como acceso al sistema de un Firewall IPCOP detrás del que se encuentra los servidores del grupo:

- ▶ Servidor de comunicación VOIP Teamspeak.
- ▶ Servidor SIP-PBX Trixbox.
- ▶ Consola y servidor fldigi.
- ▶ Servidor SDR – Linux (QtRadio, Web server, Websdr server).
- ▶ Servidor SDR – Windows (SDR-Radio, HSDR, SDR-Sharp).

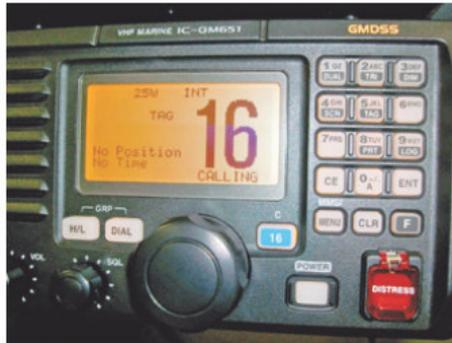


Figura 2. Transceptor de VHF del Servicio Móvil Marítimo

2. Estado de la Cuestión

Los receptores utilizados en el servicio móvil marítimo sólo permiten trabajar en banda superestrecha, es decir 3 kHz para fonía H3E o J3E y 500 Hz para radiotelelipo y DSC F1B/J2B en HF o también 16K0F3EJN (Figura 2) en el caso de VHF (Mascareñas, 2011). Además hacen imprescindible la instalación de una infraestructura de control remoto que podemos considerar como obsoleta, cuando existe la red Internet con una distribución en malla que asegura en un 95% el control de los receptores aunque se corten las líneas de transmisión de datos.

Con el proyecto Websdr (Twente 2012), figura 3, la filosofía que se plantea es completamente distinta. La capacidad de seleccionar una entre múltiples frecuencias disponibles de manera remota permite abaratar costes de instalación, ya que la selección de la frecuencia exacta, en banda estrecha, la hace el usuario, por lo que el alumno puede utilizar su instalación de Internet en su domicilio para realizar aquellas prácticas que le seleccione el profesor como "tarea para casa".

Los receptores SDR disponibles en el mercado permiten la recepción de señales de hasta 2 MHz de anchura de banda desde 100 kHz hasta 40 MHz de modo directo (Perseus) y de 40 a 1700 MHz (RTL-SDR), o con una anchura de banda dependiente de la tarjeta de sonido del ordenador que realiza la decodificación (Funcube).

Los receptores VUHF más populares han sido los Funcube durante los tres últimos años, pero están siendo desbancados por los receptores RTL-SDR, vulgarmente utilizados como decodificadores TDT, ya que éstos últimos proporcionan un ancho de banda de análisis de 2 MHz (figura 4), mientras que el Funcube se queda constreñido a 96 kHz, es decir, a la tarjeta de sonido.

The screenshot shows the WebSDR interface in a browser. At the top, it identifies the location as 'UCA S2CN Laboratory in Puerto Real (Cádiz, Spain)'. Below this, there is a text area with information about the WebSDR project and a note about Java and JavaScript requirements. A search field contains 'EA7GWJ'. The main part of the interface features a waterfall plot showing frequency activity between 7000 and 7100 kHz. Below the plot are several control panels: 'Frequency' set to 7088.09 kHz, 'Bandwidth' set to 2.49 kHz, and 'Waterfall settings' with options for speed, size, and view. There is also a 'Logbook' section with a 'submit' button and a 'Chatbox' at the bottom.

Figura 3. Selección de una frecuencia entre 7,00 MHz y 7,10 MHz

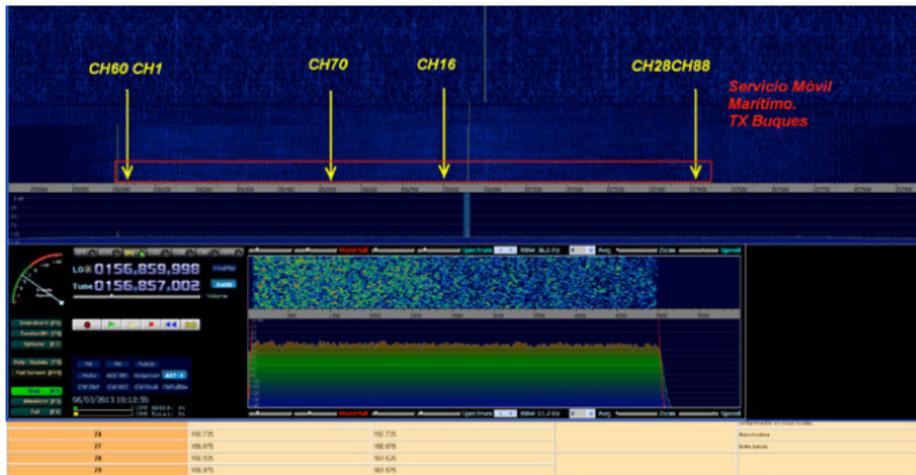


Figura 4. Pantalla del Programa WinRad controlando un receptor RTL-SDR en la Banda de VHF del Servicio Móvil Marítimo



Figura 5. Receptor Perseus integrado con arquitectura cliente-servidor. Receptor Funcube a la izquierda de la Pantalla.

En el caso del receptor SDR PERSEUS que se basa en un chip capaz de digitalizar todo lo que entra por antena a una velocidad de muestreo de 80 MHz permitiendo trabajar con frecuencias de hasta 40 MHz y proporcionar una señal digitalizada con 14 bits en la conversión analógica a digital, de modo que se convierte en un receptor de HF capaz de digitalizar todo el espectro simultáneamente desde 10 kHz hasta 30 MHz, la arquitectura cliente-servidor no permite el control simultáneo de los parámetros de configuración por más de un Operador, por lo que suele ser el profesor, o el alumno que realiza el ejercicio, el que sintoniza y ajusta sus parámetros, mientras el resto de alumnos observan en sus pantallas la operación del compañero y en sus auriculares/ordenadores los resultados del ejercicio.

PERSEUS es compatible con el programa Winrad/HSDR y otros programas de control SDR y es posible también utilizarlo como analizador de espectros en el margen de 10 kHz a 40 MHz con un margen de 100db y una resolución de 10 kHz, para esta función se suministra el software HFSPAN junto con el receptor.

Tiene entre otras opciones la posibilidad de grabar todas las señales dentro del espectro de recepción 400, 200 o 100 kHz con el fin de realizar un análisis posterior o servir de prueba legal ante un incidente marítimo.

Para conseguir enlazar cada receptor con la Red Internet es necesario utilizar un ordenador Pentium de no muy alta velocidad de procesamiento, aunque sería importante utilizar uno de última generación con, al menos, un (1) Terabyte de disco duro y 3 GHz de reloj con el fin de permitir la multitarea sin alterar las capacidades de muestreo del programa, existen también otras opciones que permiten usar ordenadores mas pequeños, tipo Raspberry Pi o Xdroid , pero

estas obligan a usar sistemas operativos Linux que permiten optimizar el uso de los recursos hardware.

Obviamente el equipo que soporta el servidor SDR ha de tener tarjetas de red y puertos USB, así como conexión a Internet en caso de querer ser accedidos desde cualquier parte del mundo, o intranet en caso de querer que sólo accedan personas de la organización propietaria del servidor.

Como es obvio la dirección internet asociada a los equipos debe ser fija o bien usar direcciones virtuales. (Figura 5).

En el caso de la Estación Remota del Grupo de Investigación, cada uno de los receptores de HF se integran con una antena de hilo de distinta longitud (8 a 20m) para recibir Onda Corta.

Para poder mantener el sincronismo en la recepción de las señales, los ordenadores toman su señal de reloj del servicio de hora del Real Instituto y Observatorio de la Armada en San Fernando, Cádiz (ROA, 2012). (Figura 6)

Desde el puesto del alumno se puede seleccionar cualquiera de las bandas de radiofrecuencia y proceder a grabar la señal que desee y volver a escuchar y analizar cuantas veces quiera, ver la figura 7.

Para el tratamiento de las señales de banda base, recibidas en el ordenador del alumno, es necesario que el mismo ordenador esté dotado de una tarjeta de audio de buena calidad, una gran capacidad de disco duro y los programas de análisis de señales y datos que sean idóneos para cada momento. Estos programas suelen ser generados por radioaficionados y son de todos conocidos, como el JVcom32, HFFax, DSCdecoder, Shipplotter y otros que son los perfectos complementos del HSDR, QTRadio, Goldwave, Audacity, Spectravue, etc.

En la figura 6 puede verse una imagen del interface hombre-máquina del Perseus, cualquier operador entrenado, como pueda ser un ingeniero radioelectrónico, un ingeniero de telecomunicaciones o un oficial de comunicaciones o de guerra electrónica de la Armada es capaz de interpretar lo que está viendo y escuchando, mejorando la señal recibida mediante la adopción de los filtros adecuados o mediante la adecuación de los filtros de paso de banda.

3. Beneficios

El espíritu de Bolonia es "aprender a aprender" y además que el crédito europeo suponga de 25 a 30 horas de dedicación del alumno a la asignatura.



Figura 6.- Análisis del espectro de una modulación en amplitud de una estación de radiodifusión de onda corta recibida en Cádiz

El alumno del grado en Ingeniería Radioelectrónica encuentra el problema añadido del acceso a herramientas de análisis muy caras como equipos individuales adquiridos a las grandes marcas de instrumentación, véase HP, Rohde & Swarz y similares, mientras que la utilización de los equipos SDR con software disponible por la red, unidos a la sabia elección de software de control y de análisis de señales y procesado de datos procedentes de los mismos permite:

- La recepción local en banda ancha de las señales y la selección, con filtros programables, de la señal a analizar.
- La grabación local de dichas señales en radiofrecuencia para su posterior análisis pormenorizado.
- El procesado local de datos en tiempo real o en tiempo diferido, con la posibilidad de transferirse señales y datos mediante periféricos tipo DVD o Pendrive para realizar análisis sobre los registros de otros compañeros o entregados por el profesor.
- La decodificación local de los datos.
- Todo lo anterior de (a) a (d) pero desde su domicilio o desde otra aula o taller de la Universidad de Cádiz.
- El desarrollo de programas sobre tratamiento de señales mediante procedimientos de demodulación de las señales I/Q.
- El acceso, mediante convenio o libre, de otras Universidades y usuarios a la RED mediante la página web de la Universidad de Cádiz. (Figura 7)

4. Software de Análisis de Radiocomunicaciones en Banda Base.

Si bien una de las funciones de la Red es distribuir por la red Internet, hasta los usuarios, la información digitalizada recibida por los receptores, posteriormente hay que procesarla, de ahí la inclusión de los ordenadores de análisis de la figura 1. (Figuras 8 y 9)

Como siempre, es necesario utilizar un computador dotado de una buena tarjeta de audio que permita analizar en tiempo real o diferido las bandas bases extraídas de los receptores en modo remoto (o local) con el fin de procesarlas en caso de necesidad.

Si bien las conversaciones de voz sólo es necesario grabarlas, cuando se transmiten en claro, y quitarles el ruido, no es tan simple cuando vienen cifradas y puede ser necesario un análisis posterior en tiempo real o en tiempo acelerado.

El caso de las transmisiones de datos es muy diferente, ya que los datos hay que descifrarlos, bien con máquinas dedicadas (mé-

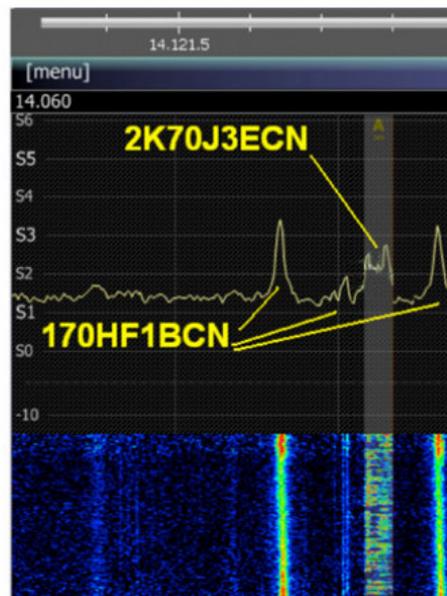
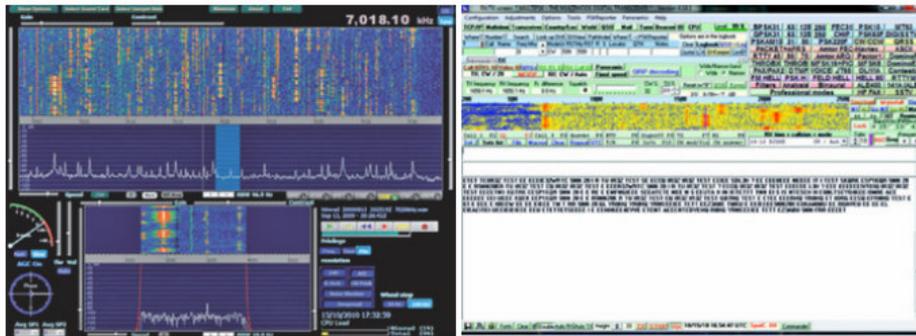


Figura 7. Pantalla del alumno para la identificación de emisiones, en remoto

do obsoleto) o con máquinas multipropósito o computadores dotados con el software adecuado, el cual ya hemos mencionado.

Entendemos que el software adecuado es aquel que debe recibir los siguientes tipos de radiocomunicaciones:

- ▶ Telegrafía ARQ
- ▶ Telegrafía FEC
- ▶ Telegrafía Baudot
- ▶ Telegrafía Clover.
- ▶ Llamada selectiva digital MF/HF.
- ▶ Llamada selectiva digital VHF.
- ▶ Radiofacímil meteorológico LF y HF.



Figs. 8 y 9. Pantalla de control y pantalla de análisis de radiocomunicaciones



Figura 10. Radioaficionado escuchando su transmisión telegráfica en remoto

- ▶ Sistema universal de identificación de buques.
- ▶ Otros sistemas telegráficos de interés para el alumno.

5. El Servidor TS3 Marconi

El servidor TS3 Marconi nace a través de una idea obtenida después de conocer y usar uno de los autores el servidor TS3-PTT, de amplio uso por los colaboradores de la REMER.

El planteamiento fue crear un simulador de radiocomunicaciones marítimas a nivel global y que reprodujera el espacio radioeléctrico español en banda marina, con el fin de que los alumnos fueran capaces de practicar los procedimientos radioeléctricos en fonía o en grafía que exige la Unión Internacional de Telecomunicaciones en los barcos acogidos al Sistema Mundial de Socorro y Seguridad Marítima.



Figura 11. Servidor TS3 Marconi con análisis de espectros de audio del programa Spectralab de Sound Technologies (Demo).

Después de un año de pruebas se ha ido configurando un potente simulador con acceso desde el exterior de la UCA que permite la conexión simultánea de más de 48 usuarios, lo que supone doce alumnos utilizando cuatro equipos de radiocomunicaciones (dos VHF con DSC, un HF con DSC y un Inmarsat C/RxFax/TrxRTTY) simultáneamente, preferiblemente en inglés, durante sus ejercicios de clase.

Al trabajar en banda base, es posible utilizar programas que transmitan señales Navtex o Teletipos ARQ y FEC o Fax meteorológico de manera simultánea o selectiva en recepción y trabajar por pares o en grupos en los más de 24 canales de radio disponibles.

El acceso a los canales simulados del Servicio Móvil Marítimo o de la Red de Estaciones Costeras de Correspondencia Pública está

restringido a los alumnos de Náutica y Radioelectrónica por motivos didácticos. (Figura 11)

Agradecimientos

Los autores desean agradecer su colaboración al Dr. Pieter-Tjerk de Boer del Departamento de Matemáticas de la Universidad de Twente por todas las facilidades de acceso al Proyecto Websdr (websdr.org), al Sr. Andrea Montefusco por la ayuda otorgada en la programación y explotación del receptor Perseus, al Sr. Alex Lee leader del grupo de desarrollo de QtRadio, al Sr. Tony Parks (KB9YIG) por el soporte dado con los equipos Softrock, Al grupo y foro de Teamspeak por la ayuda en la resolución de problemas, así como a los grupos y foros fonality (trixbox) y SDR-radio y a la Universidad de Cádiz por la cesión del local donde se encuentra instalado el Laboratorio del Grupo S2CN, así como por la colaboración magnífica del Centro Integrado de Tecnologías de la Información en cuanto al acceso a las infraestructuras desde el interior y el exterior de la UCA.

10. Bibliografía

- ▶ Burns, P. 2002. *Software Defined Radio for 3G*. Artech House Publishers. Nueva York. 2002. 300p.
- ▶ Galvis, A., Ceballos, C.A., y De Sanctis Gill. 2007. *SDR: La alternativa para la evolución inalámbrica a nivel físico*. Semillero de Tecnologías Inalámbricas (STI) – GIDATI. Universidad Pontificia Bolivariana. Medellín. 2007
- ▶ Kenington, P.B, 2005. *RF and Baseband Techniques for Software Defined Radio*. Artech House Publishers. Londres. 2005. 352p.
- ▶ Lackey, R. I. y Upmal, D. W., 1995. "SpeakEasy: The military Software Radio". IEEE Communications Magazine, Vol.33 No.5. Nueva York, mayo de 1995. p56-61.
- ▶ Mascareñas, C. Manual Básico de Sistemas de Comunicaciones Marítimas, 2ª Ed. Servicio de Publicaciones de la Universidad de Cádiz. Cádiz. 2011
- ▶ Mitola, J. 2000. *Software Radio Architecture: Object oriented approaches to wireless systems engineering*. John Wiley & Sons, Inc. Nueva York. 2000. 561p.
- ▶ R. Schiphorst. 2000. *Demonstration of the Software-Radio Concept*. Tesis de Maestría. Universidad de Twente, Departamento de Ingeniería Eléctrica. Twente. 128p.
- ▶ ROA 2012. <http://goo.gl/8pgTm>
- ▶ Twente 2012. <http://websdr.ewi.utwente.nl:8901/>
- ▶ UCA2010. <http://goo.gl/CmeCH>
- ▶ Youngblood, G. 2002-2003. *A Software-Defined Radio for the Masses – Part I, II, III and IV*. Sixth Market, Inc. Austin. 2002 a 2003. Total 40p. ●

5.11.2 Artículo publicado en CISTI'2014 - 9ª Conferencia Ibérica de Sistemas y Tecnologías de Información.

La CISTI es un evento técnico-científico anual, que tiene como objetivo presentar y discutir los conocimientos, nuevas perspectivas, experiencias e innovaciones en el campo de los sistemas y tecnologías de información.

El siguiente artículo sobre el sistema Marconi, ha sido presentado y aceptado.

A continuación se muestra artículo y nota de aceptación.

Marconi, Investigación sobre ROIP y SDR para la enseñanza de las Radiocomunicaciones

Marconi, SDR & ROIP Research for Radiocommunications Teaching

Carlos Mascareñas Pérez-Iñigo y Juan José Palma Guerrero
Grupo Señales, Sistemas y Comunicaciones Navales, Departamento Ciencias y Técnicas de la Navegación.
Universidad de Cádiz
Cádiz, España.
carlos.mascarenas@uca.es juan.palma@hotmail.com

Resumen: El sistema Marconi consigue resolver el problema de las prácticas de Radio en el espacio y en el tiempo, permitiendo a los alumnos acceder gratuitamente y de forma remota a equipos receptores de radiocomunicaciones y practicar en ellos desde el aula de teoría, el taller informático, o incluso, desde sus domicilios.

Palabras clave: Entrenamiento, acceso a plataformas distribuidas, Radiocomunicaciones, Radio definida por software, Redes.

Abstract: The " Marconi " system arises from the lack of space and time available for Radio practices and to allow students online free access to radio receiving equipment and remotely practice Radiocommunications grade subjects at the classroom, computers room or at the own home.

Key-words: Training, Distributed and cross-platform access, Radiocommunications, Software defined Radio, Networking.

I. INTRODUCCIÓN

De todos es sabido que el usuario de informática busca la movilidad. Históricamente se empezó usando ordenadores centralizados con acceso por Terminal, pasando después al PC y posteriormente al portátil y actualmente a Tabletas y Móviles. Por esto se ha de perseguir el acceso de estos equipos a los receptores de radio.

De igual forma las antenas de radio son equipos que requieren de unas determinadas dimensiones y un lugar específico de ubicación.

La actual ROIP y la SDR tienen su base en la informática y las telecomunicaciones. Por tanto se han de plantear las siguientes cuestiones: Cómo accedemos a los equipos de forma segura y con qué medios.

Se definen dos objetivos fundamentales de los que surgirán las líneas de desarrollo:

-La investigación sobre SDR y ROIP (Radio sobre internet)

-La creación de aplicaciones reales que puedan ser utilizadas por los alumnos en las prácticas de las asignaturas.

Las anteriores premisas llevan a utilizar en el sistema "Marconi" la arquitectura cliente-servidor, controlando el servidor los receptores y antenas y accediendo los usuarios a través de sus PC's, Tabletas o Móviles a este servidor de forma remota.

A su vez, se ha de considerar, tanto el acercamiento de los servidores a los receptores y antenas por causas evidentes de pérdidas de señal, así como el bajo consumo que deben tener los servidores al hallarse a veces en instalaciones poco accesibles.

Por la orientación hacia la Enseñanza y prácticas de las Radiocomunicaciones del sistema "Marconi", se ha de usar una amplia gama de receptores, controlándolos con el menor número posible de servidores, a fin de reducir costes de mantenimiento (Backup, HW). Esta es una de las dificultades principales en el desarrollo de este sistema, todos sabemos que, una a una, las aplicaciones y desarrollos funcionan adecuadamente, sin embargo su complejidad aumenta de forma considerable al integrarlas en un único equipo.

El estado actual de la tecnología posibilita adquirir receptores SDR de bajo coste que permiten llevar a cabo la recepción de las señales de radiofrecuencia. Estos receptores de conversión directa realizan la conversión de radio recibida a una frecuencia muy inferior, en la banda de audio. Esta señal sin demodular se pasa a la tarjeta del ordenador, realizando éste por software el resto de etapas [1]. Esta tecnología y la evolución de los ordenadores, tanto en proceso como en volumen, permiten el avance de la SDR. Es de destacar la aparición de nuevos ordenadores de tamaño reducido como el Raspberry Pi con 512 Megs de memoria, CPU ARM de 700 MHz y tarjeta SD y el Odroid de 2 Gigas de memoria, CPU ARM de 1.6 GHz.

Como se observa, por el tamaño de alguno de los equipos, los sistemas operativos a usar son críticos. Se requieren sistemas operativos de poco peso como pueden ser: distribuciones de Linux como Linaro, Raspbian (distribución derivada de Debian), Ubuntu o Android.

Esto nos lleva a una de las claves en la integración de las aplicaciones y de los equipos SDR: la diversidad de plataformas, sistemas operativos y lenguajes de desarrollo.

Es por ello, que existen actualmente una gran cantidad de grupos de trabajo que desarrollan e investigan en el área de SDR. A su vez, existen iniciativas ya consolidadas en empresas que producen productos SDR como pueden ser FlexRadio y Ettus Research así como algunas otras iniciativas en constante evolución.

Dentro de este entorno, el sistema Marconi, empieza con unos supuestos básicos:

- Primar el uso de líneas de investigación GNU Linux, por motivos de coste (Distribuciones libres) y por soluciones de futuro.
- Aplicar soluciones cliente servidor por motivos obvios de acceso y concentrar los receptores en unos pocos servidores, por motivos de mantenimiento y backup.

A su vez, se definen otros objetivos por su orientación a la Enseñanza:

- Utilizar una amplia variedad de receptores SDR.
- Investigar soluciones actuales de VOIP y extrapolar a la ROIP.
- Proveer acceso a los receptores tanto desde la Universidad de Cádiz como desde Internet pública.
- Dar al alumno de la Universidad de Cádiz una forma de investigar, bien con recursos para compartir preguntas, iniciativas, etc, como recursos para avanzar sobre las materias que se imparten [2][3].

II. ESTADO DE LA CUESTION

En el estado actual del SDR existe una gran cantidad de soluciones y equipos, destacando los de la marca Flexradio y Ettus Research. Los entornos de desarrollo y aplicaciones son muy diversos, si bien cabe distinguir dos grandes áreas:

- Desarrollos en Qt, C++ y Python.
- Desarrollos en entorno Java, HTML, PHP.

Como se ha indicado anteriormente, los desarrollos se ejecutan sobre Linux (Principalmente servidores), Windows, Android, Mac e iPhone.

Existen receptores de bajo precio como los clásicos Softrock desarrollados y distribuidos por Tony Parks [4], precursor en los desarrollos de SDR. El coste de estos receptores es de 25 a 100 Euros, si bien los más baratos se distribuyen en forma de kits que hay que montar.

Los receptores RTL-SDR se pueden obtener por unos 10 euros, y están soportados en internet por el grupo Osmocon [5].

También tenemos los clásicos receptores Funcube Dongle, que actúan como una tarjeta de sonido adicional dentro de los PC's y son reconocidos por prácticamente todas las aplicaciones existentes.

Además existen unos receptores SDR de prestaciones más amplias y que rondan el margen de precio de 500 a 1000 euros como son los Perseus y SDR-IQ.

Todos estos equipos receptores suelen venir con su software, si bien éste no integra otros equipos, siendo soluciones dedicadas, y por tanto no integrables en entornos donde un solo servidor pueda manejar varios receptores.

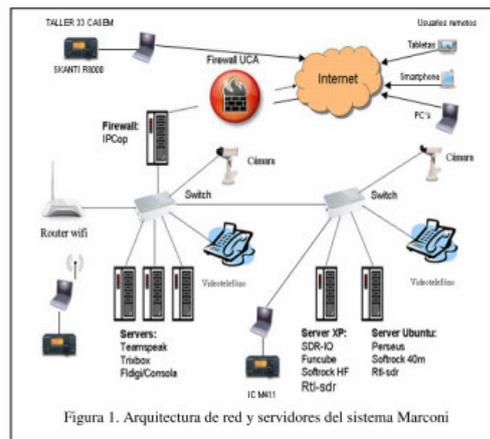


Figura 1. Arquitectura de red y servidores del sistema Marconi

Del análisis del mercado actual se concluye que no existen soluciones comerciales baratas que permitan una solución con arquitectura cliente-servidor y clientes multiplataforma, siendo además la integración de diversos receptores en un servidor un plus de dificultad.

Es por ello que se toma la línea de investigar e integrarse dentro de los diversos grupos de desarrollo existentes. A lo largo de la duración de este estudio, las soluciones han ido avanzando, permitiendo obtener prototipos y soluciones beta, que se han ido utilizando tanto para proporcionar las funcionalidades requeridas en un desarrollo dirigido a la Enseñanza, como las líneas de futuro que permitan avanzar al grupo de investigación, en este caso el grupo Señales, Sistemas y Comunicaciones Navales de la Universidad de Cádiz [6].

El mundo actual de los desarrollos SDR, acoge a investigadores de áreas muy diversas, siendo la mayoría de ellos radioaficionados, así como Profesores, Universitarios y Profesionales de las áreas de Informática, Ingeniería y Telecomunicación que, a través de internet y grupos de Yahoo y Google, o en su caso, sitios webs dedicados como GNU-Radio [7] y otros, comparten sus avances, problemas y desarrollos.

Hay que destacar aquellos grupos en los que el grupo de investigación Señales, Sistemas y Comunicaciones Navales de la Universidad de Cádiz se ha involucrado para avanzar en el conocimiento de las últimas tecnologías SDR, éstos son:

- Sdr-widget [8], grupo liderado por el Señor Alex Lee, y en el que participan el Señor Andrea Mostefusco (Perseus), el Profesor Rob Frohne (Universidad de Walla Walla), el Señor John Tucker (N8MDP).

- Ultra Cheap SDR [9], grupo liderado por el señor Adam Nielsen, dedicado al uso del receptor RTL-SDR sobre Linux con GNU Radio y sobre Windows con HSDR, en el también publica el señor Balint Seeber (Ettus Research).

- SDR-RADIO.com [10], grupo liderado por el señor Simon Brown, dedicado al uso de los software SDR-Radio para los receptores RF Space, así como softrock y desarrollos futuros.

- Perseus_SDR [11], grupo liderado por el señor Nico Palermo (Jefe de diseño y dueño de Microtelecom), dedicado a las aplicaciones del receptor Perseus, y en el que colabora el señor Andrea Montefusco.

- Airspy [12], grupo liderado por el señor Youssef Touil, desarrollador de SDR Sharp y a su vez impulsor de la nueva iniciativa de receptor SDR, VHF/UHF/SHF Airspy.

- Sdr-users [13], grupo liderado por "The SDR Institute Project".

- BasicDSP [14], grupo liderado por el señor Niels Moseley, investigador en la Universidad de Twente.

- Osmocom [5], grupo de colaboración en el desarrollo del software abierto para las comunicaciones móviles.

- GNU Radio [7], grupo liderado por el señor Johnathan Corgan, desarrollador de GNU Radio.

III. ESTADO ACTUAL DEL DEMOSTRADOR DE CONCEPTOS MARCONI

A. Equipos y aplicaciones

Tal y como se menciona en el resumen inicial, la primera inquietud es dar acceso a la información de una forma segura.

Para ello había que dotar al sistema Marconi de un firewall que permitiera el uso de redes privadas virtuales (VPN) y protocolos de IP seguros IPsec. Se vieron opciones de software libres y por motivos de funcionalidad y costes se eligió IPCop [17]. Éste se sitúa detrás del firewall general de la Universidad de Cádiz, y permite definir 3 niveles de seguridad: red roja o red exterior al sistema, red verde o sistemas seguros, y red naranja para el acceso wifi a la red Marconi (véase fig. 1). Como punto a añadir en esta primera parte del demostrador recalcar que el usar software libre conlleva la necesidad de probar las soluciones, así como de depurar más en profundidad la funcionalidad. Se recurre al uso de sistemas virtuales que

permitan probar las aplicaciones y solucionar problemas de instalación y funcionamiento.

Considerando la necesidad de acceso de los alumnos con sus equipos portátiles, se colocó un router wifi nulo, con la única misión de habilitar el camino hacia la red de estos equipos.

La red interna se completa con dos switches 10/100 de coste reducido (ver fig. 1).

Una vez establecidas las comunicaciones básicas, se pasó a colocar dentro de la red, tanto por motivos de enseñanza como de investigación en ROIP/VOIP, un servidor SIP: en este caso una centralita telefónica (ver fig. 1).

Después de analizar aplicaciones e investigar las PBX existentes actualmente en el mercado y que tuvieran una distribución libre para uso en la investigación, se pasó a seleccionar aquellas que usaban software libre [16][17]. Se depuró su funcionamiento mediante la definición de dos pilotos formados por el conjunto de Firewall/PBX separados físicamente, y unidos a Internet pública por los proveedores de ISP locales. Recalcar que para la depuración de las llamadas se usaron hardphone y softphones. Una vez en marcha los prototipos se integraron en el sistema Marconi.

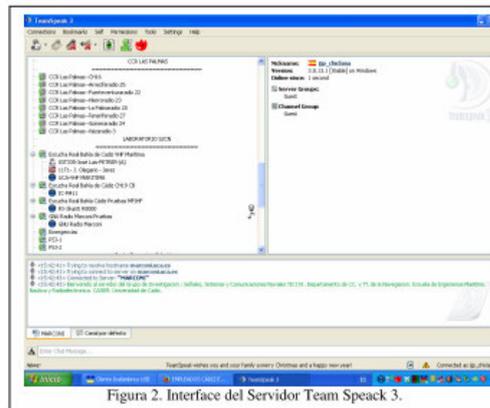


Figura 2. Interface del Servidor Team Speack 3.

Un punto importante a considerar es la integración de la aplicación Teamspeak dentro del sistema Marconi, como clave para practicar la escucha y la transmisión de las radiocomunicaciones navales en las clases prácticas, sin la necesidad de emitir y recibir realmente, y permitiendo el acceso remoto de los estudiantes de la Universidad de Cádiz (ver fig. 2).

Tenemos que agradecer en este punto la cortesía de la compañía TeamSpeak [18] al proveernos de licencias sin coste para su uso en la Enseñanza.

Una vez puesta en marcha la infraestructura básica de radiocomunicaciones, se siguió con los equipos de recepción SDR.

Tal y como hemos enunciado en el punto de introducción, se pretende utilizar equipos SDR de coste moderado, y por motivos de enseñanza, implementar tanto los equipos clásicos Softrock y Funcube como los nuevos equipos Perseus de Microtelecom, SDR-IQ de RFSpace o los RTL-SDR de Terratec.

Las principales características de estos equipos son:

- Receptor Softrock Lite II sintonizado en la banda de 40 metros con frecuencia central de 7,056 MHz.
- Receptor softrock RX ensemble II HF Receiver con recepción en la banda 1,8 a 30 MHz.
- Receptor Funcube Dongle con recepción en la banda de 64MHz a 1700 MHz.
- Receptor RTL-SDR Terratec con recepción en la banda de 52 MHz a 2200 MHz.
- Receptor Perseus de Microtelecom con recepción en la banda de 10KHz a 30MHz.
- Receptor SDR-IQ de RFSpace con recepción en la banda de 500Hz a 30MHz.

TABLA I. COMPARATIVA ENTRE EQUIPOS RECEPTORES INTEGRADOS EN EL SISTEMA MARCONI

Receptor	Tabla comparativa			
	Frecuencias	Nivel Integración	Ancho de Banda	Precio equipo
Softrock LII	HF Monobanda	Tarjeta sonido	96 kHz	80€ (kit)
Softrock E-II	HF Multibanda	Tarjeta sonido	96 kHz	200 € (kit)
Perseus	ELF-HF Multibanda	Puerto USB	2 MHz	750 €
SDR IQ	ELF-HF Multibanda	Puerto USB	2 MHz	700 €
President AR7	27 MHz Monocanal	Pasarela Radio	3 kHz	150 €
ICOM M411	VHF Marino Scanner	Pasarela Radio	3 kHz	150 €
Funcube Don.	VHF-UHF	Puerto USB	40 kHz	80 €
RTL-SDR	VHF-UHF	Puerto USB	3 MHz	10 €

Una vez seleccionados los receptores, hemos de distribuir su uso por los distintos servidores, con objeto de explorar el estado actual del SDR y las soluciones futuras, como indicamos en el punto 1. Se establecen dos servidores básicos, uno basado en Linux (Ubuntu) y otro basado en Windows XP.

Por motivos de espacio no se detallan todas las pruebas hechas, indicando solo la solución implementada. Apuntar como recomendación el uso de máquinas virtuales para depurar soluciones así como la formación en Linux, Windows, evidentemente hardware y conocimiento de entornos HTML, Java, Android, C++, Python y algunos más, ya que como enunciamos en el primer punto la investigación en SDR tiene grupos y líneas diversas.

El servidor de receptores Ubuntu aloja las aplicaciones servidor con acceso al exterior de la Universidad de Cádiz [6].

Se han elegido las aplicaciones en investigación dentro de los grupos indicados anteriormente, en concreto QtRadio (grupo sdr-widget) [19][20] y WebSDR [21]. Este último en pruebas en la Universidad de Twente (Lidera el profesor Dr. Pieter de Boer).

Las condiciones de compilación, y puesta en marcha escapan a este artículo. Sin embargo, apuntar que se habilita en el PC una tarjeta audio por aplicación, y se definen dos usuarios, uno para cada aplicación por evidentes motivos de seguridad.

La aplicación WebSDR maneja los receptores Softrock y RTL-SDR (Terratec). Para este último hay que obtener sus drivers y compilarlos [5]. La aplicación QtRadio controla el receptor Perseus, para el que hay que descargar y compilar los drivers provistos por Microtelecom [22].

Adicionalmente en el servidor Ubuntu se desarrolla la aplicación HTML de acceso al sitio Web marconi.uca.es, por lo que este servidor tiene instalado el entorno LAMP.

En las figs. 3 y 4 se muestran las imágenes que cualquiera puede obtener accediendo a la página marconi.uca.es de la Universidad de Cádiz y usando las aplicaciones indicadas en esta página.

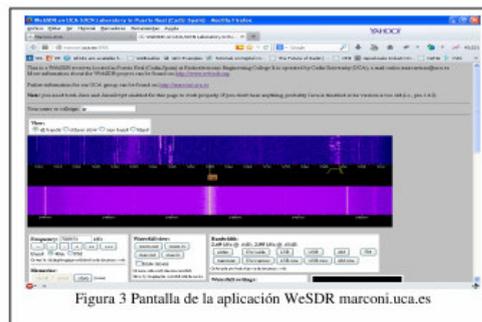


Figura 3 Pantalla de la aplicación WeSDR marconi.uca.es

El segundo servidor de receptores del sistema Marconi, es un servidor XP, al que se le ha dotado de dos tarjetas de red, con objeto de acceder a través de él al sistema Marconi desde cualquier PC de la Universidad de Cádiz.

De esta forma, se pueden compartir los recursos Marconi para la Enseñanza. Estos recursos son PBX Trixbox, teléfonos Softphone y cámaras de vídeo (SIP) a las que se puede acceder con los navegadores mediante streaming de voz y datos y próximamente PBX FreePBX y WebRTC softphones.

Dentro de este servidor se encuentra instalada la versión beta del software SDR-Radio [23] (ver fig. 5), que maneja los siguientes receptores de radio: Softrock RX ensemble II HF, Funcube Dongle, RTL-SDR y SDR-IQ. Se requieren interfaces adicionales para el uso del receptor RTL-SDR [24]

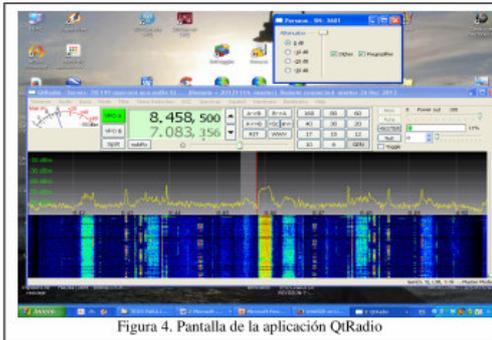


Figura 4. Pantalla de la aplicación QtRadio

A su vez, entre otros programas SDR, están instalados los programas HSDSDR y SDRSharp para las prácticas de radiocomunicaciones de los alumnos.

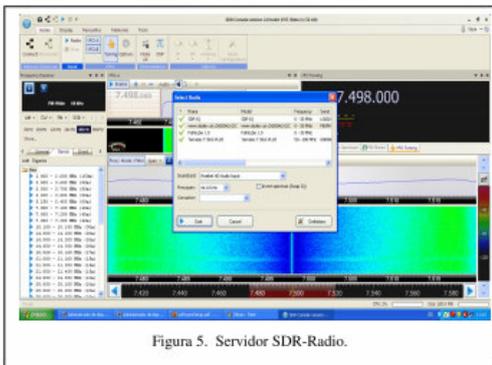


Figura 5. Servidor SDR-Radio.

B. Instalaciones y Antenas

Con objeto de llevar la señal a los receptores controlados por el sistema Marconi, se han instalado las siguientes antenas:

- Antena TAGRA 27 MHz 5/8 para el receptor President AR7 conectado al servidor Teamspeak.
- Antena VHF SMM para el receptor ICOM IC M-411 conectado al servidor Teamspeak.
- Antena Comet DS3000 45-3000 MHz para los receptores RTL-SDR (Terratec).
- Antena HF Long wire para los receptores Softrock.
- Antena HF Long wire para el receptor Perseus.
- Antena VHF SMM para el receptor Funcube.
- Antena HF RX 1,5 a 70 MHz para el receptor SDR-IQ

C. Sistemas de ayuda a la Enseñanza

Como hemos apuntado al comienzo de este artículo, el sistema Marconi es un sistema orientado a la Enseñanza y a la Investigación. Es por ello interesante disponer de aplicaciones que permitan debatir y recoger información del grupo sobre temas de interés.



Figura 6. Página principal y foro Marconi

Por esto, además de programar la página de entrada al sistema Marconi en HTML, se ha programado una página de foro de debate, usando la herramienta phpBB. Imágenes de esta página principal como de la herramienta de debate se pueden ver en figs 6 y 7.

Como aplicación de enseñanza se ha integrado la aplicación de software libre GNU-Radio [7], cuyas pruebas de uso se pueden oír a través del servidor Teamspeak marconi.uca.es, y ver a través del servidor XP en el resto de la Universidad de Cádiz.

IV. BENEFICIOS ACADÉMICOS DEL SISTEMA MARCONI

El alumno del Grado en Ingeniería Radioelectrónica encuentra el problema añadido del acceso a herramientas de análisis muy caras como son los equipos individuales adquiridos a las grandes marcas de instrumentación.



Figura 7. Secciones del Foro Marconi

Mientras que la utilización de los equipos SDR con software disponible por la red, unidos a la sabia selección de software de control y de análisis de señales y procesado datos procedentes de los mismos permite:

- a) La recepción local en banda ancha de las señales y la selección, con filtros programables, de la señal a analizar.
- b) La grabación local de dichas señales en radiofrecuencia para su posterior análisis pormenorizado.
- c) El procesado local de datos en tiempo real o en tiempo diferido, con la posibilidad de transferirse señales y datos mediante periféricos tipo DVD o Pendrive para realizar análisis sobre los registros de otros compañeros o entregados por el profesor.
- d) La decodificación local de los datos.
- e) Todo lo anterior (de a á d) pero desde su domicilio o desde otra aula o taller de la Universidad de Cádiz.
- f) El desarrollo de programas sobre tratamiento de señales mediante procedimientos de demodulación de las señales I/Q.
- g) El acceso, mediante convenio o libre, de otras Universidades y usuarios a la RED mediante la página web de la Universidad de Cádiz.
- h) Lo que es fundamental y prioritario “ver y tocar la radio”, así como “la onda de radio” de forma gratuita, manipularla, decodificar los mensajes y ver las formas espectrales de las radiocomunicaciones en el aula de teoría, en el taller o en su domicilio.

V. CONCLUSIONES

La evolución actual de los ordenadores y la informática, permite acceder de forma distribuida a los equipos receptores y antenas de radio, permitiendo compartir de forma remota las señales de radio a través de internet.

El progresivo aumento del ancho de banda, evita problemas en la transmisión de gran cantidad de datos por la red. Se plantea el reto de hacer cada vez más seguros los datos así transmitidos.

El acceso a las tecnologías SDR de Banda Ancha permite reducir el coste de instalaciones de investigación y/o enseñanza desde el momento en el que los recursos son compartidos por múltiples grupos.

Del análisis del mercado actual se concluye que no existen soluciones comerciales baratas que permitan una solución con arquitectura cliente-servidor y clientes multiplataforma, excepto los sistemas militares y de guerra electrónica (lo cuales no son accesibles al público) siendo además la integración de diversos receptores en un servidor un gran problema de dificultad técnica y organizativa.

El alumnado dispone de herramientas para la asimilación de conceptos que antes no tenía, por lo que ahora puede aprender por sí solo, y realizar varias veces las prácticas desde su domicilio hasta poder alcanzar el nivel de competencia exigido por el Convenio de Formación, Titulación y Guardia de la Gente de Mar de la Organización Marítima Internacional que recoge los acuerdos internacionales de la Unión Internacional de Telecomunicaciones sobre formación de Radiooperadores y Certificados de Radioelectrónico de Primera y Segunda Clase.

AGRADECIMIENTOS

Los autores quieren agradecer al Grupo de Investigación Señales, Sistemas y Comunicaciones Navales de la Universidad de Cádiz la financiación del Proyecto Marconi y el uso de sus instalaciones e infraestructuras.

REFERENCIAS BIBLIOGRÁFICAS

- [1] P.B. Kenington, RF and Baseband Techniques for Software Defined Radio. Artech House Publishers, Londres, 2005.
- [2] C. Mascareñas, Manual Básico de Sistemas de Comunicaciones Marítimas, 2ª Ed. Servicio de Publicaciones de la Universidad de Cádiz, Cádiz, 2011
- [3] UCA2010. <http://www.uca.es/es/tratarAplicacionAsignaturasPlanEstudios.do?idTitulacion=41415&acceso=t>
- [4] T. Park, “Softrock receivers documentation”, <http://fivedash.com/>
- [5] OsmocomSDR. “RTL-SDR drivers”, <http://sdr.osmocom.org/trac/>
- [6] S2CN Marconi. Página web Marconi. <http://marconi.uca.es>
- [7] J. Corgan, GNU Radio web site. <http://gnuradio.org/redmine/projects/gnuradio/wiki>
- [8] QtRadio. Grupo sdr-widget. <https://groups.google.com/forum/?hl=es#forum/sdr-widget>
- [9] RTL-SDR. Grupo Ultra Cheap SDR. <https://groups.google.com/forum/?hl=es#forum/ultra-cheap-sdr>
- [10] SDRRadio. Grupo sdr-radio-com. <https://groups.yahoo.com/neo/groups/sdr-radio-com/info>
- [11] Perseus. Grupo perseus_SDR. https://groups.yahoo.com/neo/groups/perseus_SDR/info
- [12] Airspy. Grupo airspy. <https://uk.groups.yahoo.com/neo/groups/airspy/info>
- [13] SDR Radio Users. Grupo sdr-users. <https://groups.yahoo.com/neo/groups/sdr-users/info>
- [14] BasicDSP. Grupo basicDSP. <https://groups.yahoo.com/neo/groups/basicdsp/info>
- [15] IPCop. IPCop support forum. <http://www.ipcop.com/phpbb3/>
- [16] Trixbox. Trixbox web page. <http://www.trixbox.com/>
- [17] FreePBX. FreePBX forum. <http://www.freepbx.org/forums>
- [18] Teamspeak. Teamspeak community forum. <http://forum.teamspeak.com/forum.php>
- [19] J. Melton, JMonitor Software. <http://g0orx.blogspot.com/es/>
- [20] A. Lee, “The ghpsdr3-alex Project”. http://napan.ca/ghpsdr3/index.php/Main_Page
- [21] P. De Boer, “Websdr Software”, Twente 2012. <http://websdr.ewi.utwente.nl:8901/>
- [22] A. Montefusco, Palermo, “N. Perseus library”. <http://www.montefusco.com/perseus/>
- [23] S. Brown, “SDR-Radio Software”. <http://v2.sdr-radio.com/Home.aspx>
- [24] M.Morgan, “RTL-SDR/SDR-Radio drivers”. <http://www.aa5sh.com/>

[Inicio](#)

[Cerrar](#)

Fwd: [CISTT2014] Your submission has been accepted!

De: Prof. Dr. Carlos Mascareñas y Perez-Higo (carlos.mascarenas@uca.es)
Enviado: domingo, 30 de marzo de 2014 19:50:09
Para: juan.palau@hotmail.com

----- Mensaje recibido -----

From: CISTT2014 [cistt2014@gmail.com]
Enviado: 30 mar 2014 11:58
Destinatario: carlos.mascarenas@uca.es
Asunto: [CISTT2014] Your submission has been accepted!

Dear author,

On behalf of the CISTT2014 - 9^a Conferencia Ibero de Sistemas y Tecnologías de Información, I am pleased to inform you that your submission 44, titled "MARCONI INVESTIGACIÓN SOBRE ROP Y ROP PARA LA ENSEÑANZA DE LAS RADIOCOMUNICACIONES". We have included the reviewers' comments at the end of this message.

Please consider reviewers' comments and the rules of edition (<http://www.aitis.org/aitis2014> <http://www.aitis.org/aitis2014/oc14openconf.php> "super" link) to prepare the current ready version of the paper, with the current number of pages. Then, save the file in doc or docx format and upload it until April 12 at <http://www.aitis.org/aitis2014/oc14openconf.php> to prepare the current ready version of the paper, with the current number of pages. Then, save the file in doc or docx format and upload it until April 12 at <http://www.aitis.org/aitis2014/oc14openconf.php>

Additionally, you also need to do your conference registration until April 12 for the paper can be published and presented. The registration system is available at <http://aitis.org/aitis2014>.

To present your paper during the conference you should prepare and print a poster with A1 or A2 dimension, vertical orientation, and delivery it to the conference secretariat for exhibition.

Congratulations,
Chair, CISTT2014
<http://www.aitis.org/aitis2014>
cistt2014@gmail.com

Las referencias bibliográficas web deben servir como soporte al artículo, no como elemento fundamental de la bibliografía

El autor presenta su investigación sobre SDR y ROP para la enseñanza de las radiocomunicaciones. El objetivo del artículo es relevante. El resumen del artículo debe ser ampliado.

En mi opinión el autor no explica como arriba a la conclusión de que no existen soluciones comerciales con arquitectura cliente-servidor. Sería interesante que el autor presente un cuadro comparativo de receptores (párrafos 3 al 7) indicando precio, arquitectura y gr

La sección IV ha sido descrito de manera insuficiente: se debe incluir aspectos relacionados al impacto en la emulación, dificultades encontradas y como han sido solucionados. Sería interesante incluir estadísticas de uso en las prácticas y asimilación de conceptos.

Las referencias web se han incluido páginas web. Las referencias

El artículo debe ser mejorado. La figura 6 es redundante, la relación de antenas instaladas están descritas en el texto.

Ampliar y mejorar la sección Conclusiones.

El artículo no sigue las normas de presentación de artículos a la conferencia <http://www.aitis.org/aitis2014/index.php/callforpapers-articles>

El trabajo está profetamente escrito. La redacción no es correcta, tiene errores de ortografía.

El Abstract es extremadamente corto. El título de cada sección tiene punto final y no deber ser así. Al menos un párrafo no tiene el espaciado con el párrafo anterior y hace falta la sangría al inicio del párrafo. La nota al pie de la página no deberían estar incluida.

Es notorio que esta versión es un borrador, quizá el primero, porque durante todo el trabajo escriben "La universidad CCCCCC" y "El grupo SSSSS".

Las citas están en desorden, hacen falta algunas de ellas. En la lista de referencia dejan en blanco los espacios para las citas [2], [3] y [24].

La calidad de las figuras es mala.

Inmediatamente antes de la sección V. Conclusiones, escriben "La página web de la UUU".

En los agradecimientos dice "Los autores quieren agradecer al Grupo de Investigación XXXXXXXXXXXXXXXXXXXXXXXXXXXX de la Universidad CCCCC...", obviamente esto es incorrecto.

Todas las referencias son a páginas de Internet (o eso parece), y está mal el formato de las referencias.



Carlos Mascareñas y Perez-Higo
Profesor Titular de Universidad
Grupo de Gestión y Teoría de la Navegación
Universidad de Cádiz
CAREM
Campus del Río San Pedro
11019 Puerto Real, Cádiz, España
carlos.mascarenas@uca.es
Tel: +34 956 016 543 Fax: +34 956 016 487
Móvil: +34 628 707 705
carlos.mascarenas@uca.es
Página Web: <http://marconilug.es>

Reserva de Confidencialidad
This e-mail and any files transmitted with it are confidential. If you have received this e-mail in error, please notify the system manager. This e-mail and any files transmitted with it are confidential. If you have received this e-mail in error, please notify the system manager. If you have received this e-mail in error, please notify the system manager. If you have received this e-mail in error, please notify the system manager.

5.12 Ejemplo de práctica de Radiotecnia ejecutada por alumno del Grado de Ingeniería Radioelectrónica

A continuación se anexa ejercicio realizado por el alumno Juan José Núñez de esta escuela usando los medios explorados en esta tesis.

2014

PRÁCTICAS RADIOTECNIA II RECEPTOR SDR



JUAN JOSÉ NÚÑEZ MUÑOZ
INGENIERÍA RADIOELECTRÓNICA
21/01/2014

INDICE

- 1.- INTRODUCCIÓN.
- 2.- RECEPTOR UTILIZADO Y CARACTERISTICAS.
- 3.- INSTALACIÓN Y SOFTWARE UTILIZADO.
- 4.- TABLA DE FRECUENCIAS.
- 5.- SEÑALES RECIBIDAS.
- 6.- CONCLUSIONES.

1.- INTRODUCCIÓN.

El objetivo de esta práctica es el conocimiento y aprendizaje de una forma de recibir las señales de radio mediante un USB con un software, y la ayuda de un ordenador, es una forma barata y sencilla de poder recibir señales de radio y TV.

2.- RECEPTOR UTILIZADO Y CARACTERISTICAS.

El receptor que hemos utilizado es el siguiente:
"FM+DAB USB DVB-T RTL2832U+R820T y w/ MCX antenna"



Sus características técnicas son:

- Sintonizador Rafael Micro R820T, que recibe desde 24 a 1766MHz.
- Microcontrolador RTL2832U.

3.- INSTALACIÓN Y SOFTWARE UTILIZADO.

El programa utilizado es ZADIG, que instala unos drivers específicos para el chip RTL2832U. A través de este enlace se puede seguir la instalación fácilmente paso a paso:

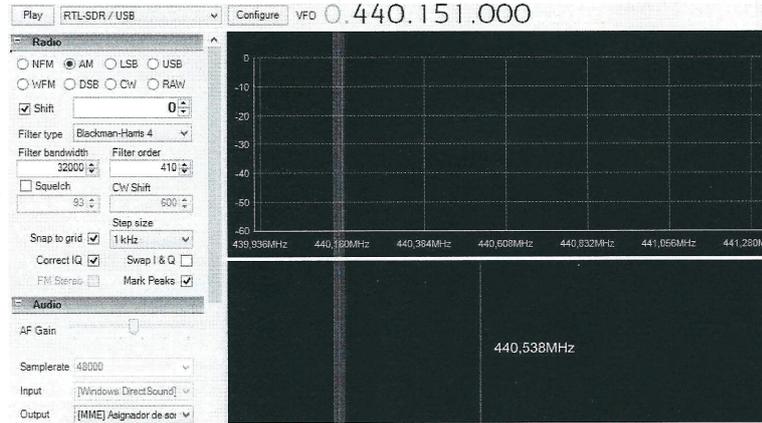
<http://www.rtl-sdr.com/rtl-sdr-quick-start-guide/>

También nos descargaremos el programa SDR, con el cual controlaremos nuestro receptor, de una manera gráfica.

<http://sdrsharp.com/downloads/sdr-install.zip>

Se descomprime, se ejecuta el archivo install.bat y se creará una carpeta con el SDR# y el ZADIG.

En la carpeta que ha creado, ejecutáis el SDRSharp.exe y os aparecerá una ventana como esta:



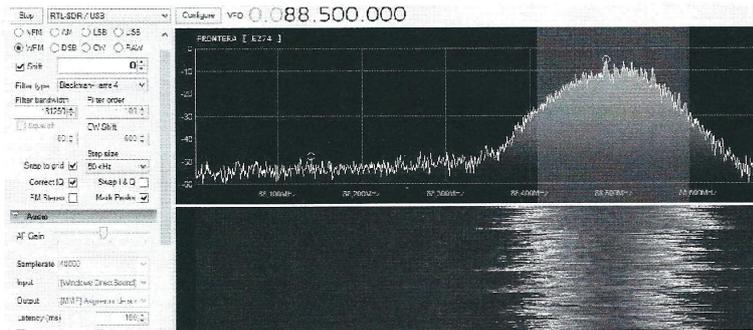
Arriba, entre los botones de Stop y Configure, se selecciona RTL por USB, se le da al play y listo, a movernos a ver que encontramos.

4.- TABLA DE FRECUENCIAS.

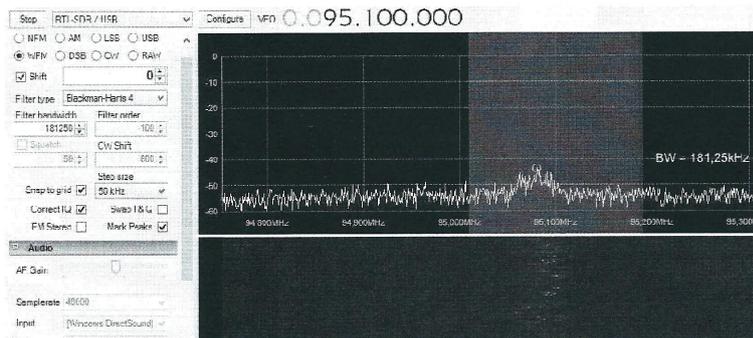
FRECUENCIAS	MODO EMISIÓN	SERVICIO
156.025 a 158.200 Mhz		Servicio Móvil Marítimo
160.625 a 162.800 Mhz		Servicio Móvil Marítimo costeras
88 a 108 Mhz	180kF3EJN	Radiodifusión FM
27 a 28 Mhz	6k00A3EJN/2k70J3EJN/16k0F3EJN	CB (Banda Ciudadana)
28 a 29,7 Mhz	100kA1ACN/2k70J3EJN	HAM (Aficionados)
108 a 136 Mhz	6k00A3EJN	SMA (Servicio Móvil Aéreo)
136 a 138	16k0F3EJN	NOAA
144 a 146	16k0F3EJN	HAM (Aficionados)
128.5	FM	Aproximación Sevilla
118.5	AM	Torre control aeropuerto Jerez
85 a 87 Mhz	FM	Guardia civil

5.- SEÑALES RECIBIDAS.

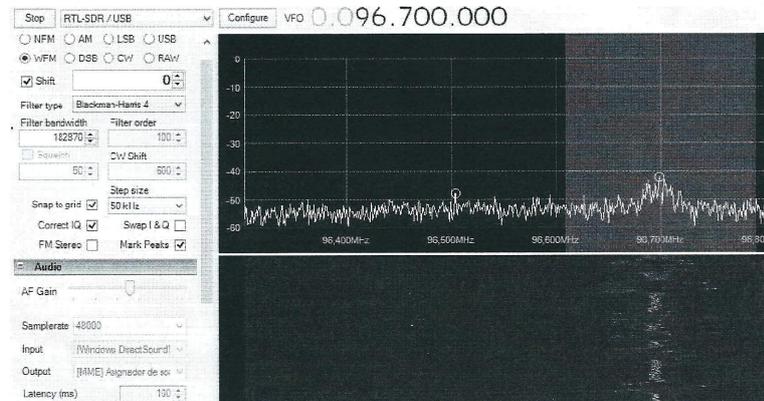
Lo primero que encuentro son las cadenas de radiodifusión en WFM con un Ancho de banda de 180 KHz, desde 88 a 108 Mhz. Aquí empiezan las cadenas de Radio, la primera que encuentro es Cadena Dial de la Frontera.



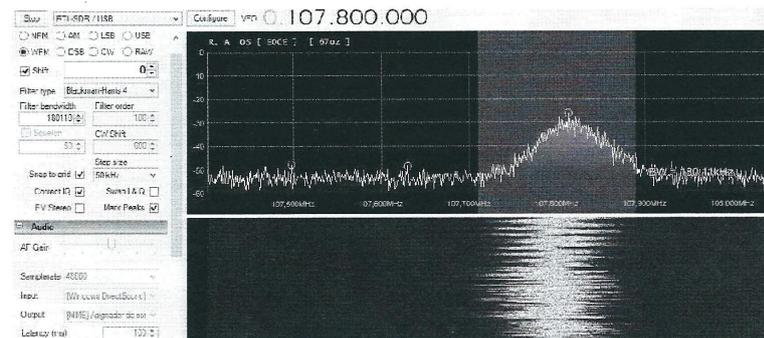
Los cuarenta principales



Otra cadena que recibo con poca señal:

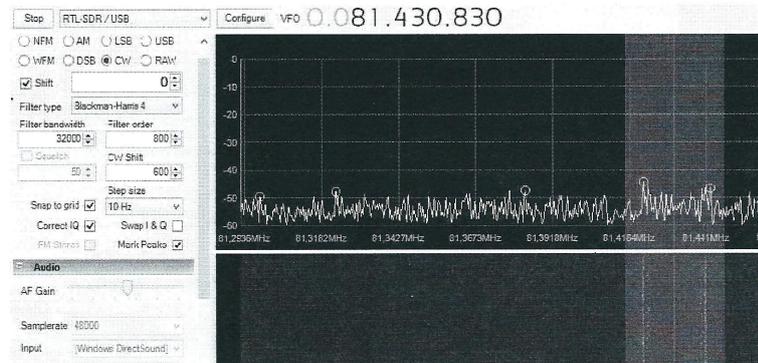


En la frecuencia de 107,8 Mhz, encuentro la emisora de radio local de Arcos. Que supongo que por cercanía entra con tanta fuerza y tan claramente con respecto al resto de emisiones, junto con cadena dial son las dos que mas fuerte llegan.



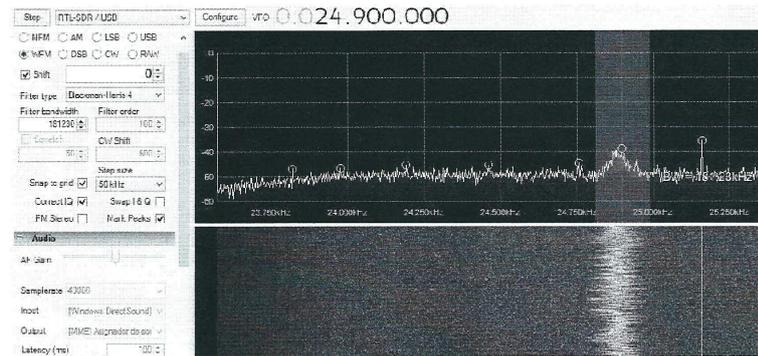
El ancho de banda para los canales con emisión en FM es de 180 Khz, Esta es la última señal que cojo.

En esta frecuencia encuentro una señal, que resulta ser la interferencia que produce mi ratón wifi sobre el receptor.



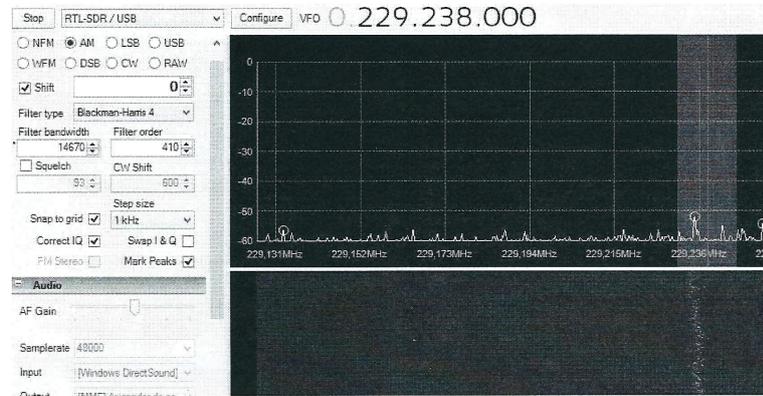
Sale señal cuando el ratón esta en movimiento.

En 24 Mhz recibo esta señal, que es una emision de radiodifusión en FM, lo que es extraño es que sea en esta frecuencia.



Más debajo de 24 Mhz no se recibe nada.

Por la frecuencia de 229 Mhz se recibe una comunicación, pero muy debil y con mucha interferencia no se escucha bien lo que se dice.



6.- CONCLUSIONES.

Se trata de una manera económica y sencilla de tener nuestro propio receptor desde 24 Mhz a 1766 Mhz, esto abarca desde la HF, hasta la UHF, pasado por la VHF.

Se necesita una antena mayor para poder captar más y mejor las señales, ya que con la que trae apenas puedo escuchar las cadenas de radio por radiodifusión en FM.

Además por la falta de tiempo, no he podido profundizar todo lo deseado en el receptor para poder hacer un análisis más detallado de las señales recibidas, ya que se necesita un poco de práctica en el software para ir comprendiendo para que funcionan cada una de las funciones que trae. Pero lo más importante, como es el ancho de banda, la utilización básica y el modo de emisión de cada una de las señales que se reciben, han quedado claros con el desempeño de esta práctica.

5.13 Colaboración con la WebSDR de Chile.

A continuación se muestra el mensaje de agradecimiento y la imagen de la WebSDR de Chile como ejemplo de lo que se ha pretendido en esta tesis doctoral, mantener relaciones con las Universidades y grupos para compartir experiencias y avanzar en la enseñanza e investigación.

Re: websdr

De: **RiM ADA** (websdrchile@gmail.com) Has movido este mensaje a su ubicación actual.
Enviado: martes, 10 de diciembre de 2013 13:26:56
Para: Juan J. Palma (juan.palma@hotmail.com)
CC: Carlos Mascareñas (carlos.mascarenas@uca.es)

muchas gracias don juan palma. la información entregada a sido vital para el desarrollo del proyecto y la defensa de este.

Daniel Quinteros.

2013/12/10 Juan J. Palma <juan.palma@hotmail.com>

Hola Daniel, encantado de saber de ti.

Te contesto a tu pregunta.

Puedes poner mas bandas. Si la entrada es audio (softrock) para cada banda has de disponer de una tarjeta de sonido en tu ordenador.

Tienes que duplicar para cada nuevo dispositivo los comandos del websdr.cfg siguientes:

band 40m (la que corresponda).

device \$hw:0,0,0 (el que corresponda).

samplerate 96000 (el que corresponda).

centerfreq 7055 (el que corresponda).

antenna funny dipole (la que corresponda).

Son las mismas líneas que has puesto para el dispositivo que ya tienes.

Toda la configuración del websdr se hace en el fichero websdr.cfg, lo demás es cosa de jugar con el ordenador y tener claro la instrucción que has de poner en la línea device.

Un saludo, Juan J. Palma

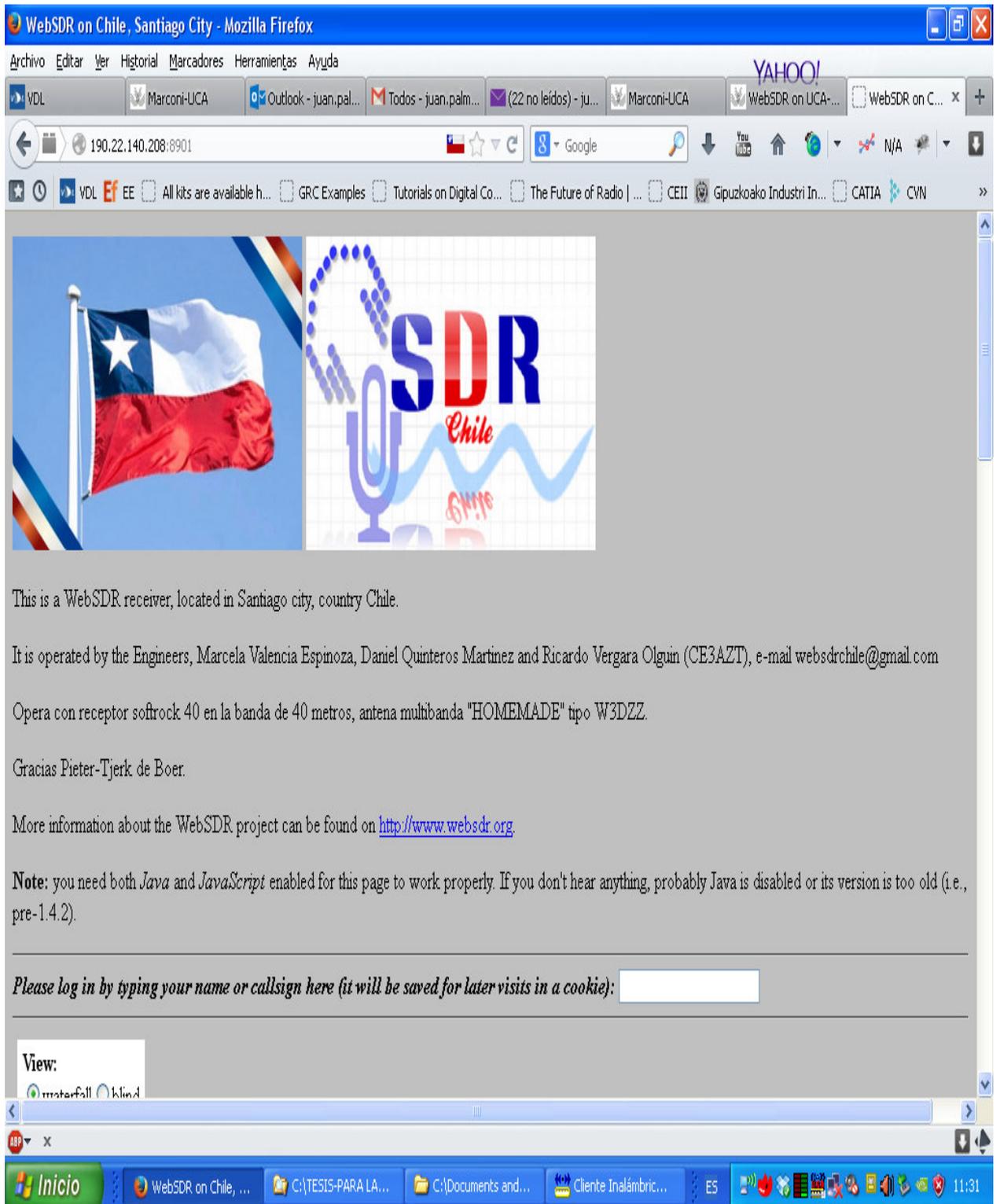


Figura 5.13.1 WebSDR de Chile.

6. Conclusiones.

Durante el desarrollo de esta Tesis Doctoral se han logrado integrar diferentes redes de comunicaciones con distintos protocolos y diversos grados de tecnificación pero aunque esto no sea nuevo, sólo ha sido posible cuando la tecnología electrónica ha permitido un alto grado de reconfiguración y adaptabilidad a los requisitos necesarios para desarrollar esta aplicación.

En el caso de utilizar mezcladores y amplificadores de bajo ruido, la aplicación de los equipos de radio configurados por software se vuelve insospechadamente flexible, pasando a poderse utilizar en sistemas de radionavegación, anticolidión, radioastronomía de ondas milimétricas, etc.

La utilización de Plataformas de Chat, configuradas como se ha hecho con la plataforma TS3-Marconi, puede ser utilizada como un Simulador de Radiocomunicaciones, como una Red de Estaciones Costeras si se integra con equipos fijos del Servicio Móvil Marítimo, como Integrador de equipos de radio con usuarios si se instala en un buque, como red de comunicaciones interiores si se instala en un buque o en una empresa multinacional. Mientras la imaginación humana no tenga límites, las plataformas de Chat seguirán siendo herramientas poderosas de intercomunicación y sobre todo de enseñanza a distancia.

En tiempos de Crisis económica, si se quiere llegar a la Excelencia en el campo de la enseñanza-aprendizaje, es necesario compartir recursos. No todas las Escuelas tienen por qué disponer de los mismos equipos y laboratorios de coste moderado o bajo, sino que se pueden especializar los laboratorios y talleres de las Escuelas o Centros, utilizando a distancia los equipos de los demás, siempre que todos y cada uno aporte un equipo, laboratorio o taller de la RED. Esta filosofía está muy bien entendida por los Radioaficionados y las Universidades extranjeras que investigan en Teoría de la Señal y Comunicaciones. QUID PRO QVO. Te cambio conocimientos a cambio de conocimientos, servicios a cambio de servicios. No te pido dinero, te pido que colabores en mi (nuestro) proyecto.

Hemos de decir que, gracias al Demostrador de Conceptos desarrollado en esta Tesis Doctoral, los Alumnos del Grado de Ingeniería Radioelectrónica de la Universidad de Cádiz pueden hacer las prácticas de multitud de asignaturas de su carrera desde su domicilio, lo mismo que desde su ordenador portátil en el Aula, Taller o Sala de Estudio, lo mismo que los Alumnos del Grado de Ingeniería Radioelectrónica de la Universidad de la Laguna, la Escuela Técnica de Electricidad y Electrónica de la Armada de El Ferrol, las Escuelas de Ingeniería de Telecomunicaciones, Informática, Electrónica, las Facultades de Física de toda España y cualquier Centro de enseñanza del Mundo que disponga de una conexión a Internet.

Si todos los Centros citados, pusieran a la disposición del resto el mismo tipo de equipamiento, conseguiríamos una mejor formación de nuestro alumnado, por lo que estimamos que es necesario crear una RED internacional de enseñanza compartida.

Como **corolario** a la investigación en el área de la ROIP se concluye que el estado actual en cuanto a líneas de investigación está un poco disperso, sin embargo se empiezan a aplicar los avances en el área de las telecomunicaciones y las informáticas.

Se distinguen dos claras líneas de trabajo en líneas de arquitectura de ordenador: la arquitectura 386 y la arquitectura ARM actualmente en boga en los dispositivos móviles y tabletas, esta última arquitectura evidentemente en uso debido a su consumo reducido. En el afán de acercar los equipos procesadores a las antenas receptoras, se avanza en el uso de procesadores cada vez más pequeños como el Raspberry Pi o el Odroid, todos ellos de precio reducido y con arquitectura ARM.

Se comienza a aplicar con cada vez mayor fuerza el sistema operativo Linaro (Distribución Linux) para ARM, con el objetivo de que pueda ser utilizado en procesadores pequeños.

Al igual que en la informática tradicional se pasa de los ordenadores de gran tamaño, a los personales, posteriormente a los portátiles y sucesivamente a los móviles y Tablet, buscando siempre el objetivo de comunicación y portabilidad. La introducción de la informática en el área de la ROIP va en el mismo sentido compartiendo de esa forma lo único que por ahora parece difícil reducir de tamaño, las antenas. Para ello se acercan

los procesadores a los equipos receptores y antenas, usando procesadores de bajo consumo y reducidos y a ser posible alimentados por USB, y de igual forma se distribuyen los accesos a las antenas usando la ya clásica Internet.

Este último punto es un punto a tener en cuenta ya que la infraestructura de comunicaciones no es barata y cada tecnología tiene su sitio y hueco de mercado. Es por ello, que los sistemas de radiocomunicaciones, actualizados con las nuevas tecnologías y con la ayuda de la SDR, pueden encontrar su lugar en competencia directa con sistemas en principio más costosos en aquellos sitios donde no exista una infraestructura que lo soporte. Por experiencia se sabe que los cambios en tecnologías de infraestructuras son caros y difíciles de llevar a cabo, no siendo éste el caso en la SDR y la ROIP.

Por último apuntar que este Demostrador de Conceptos se empezó a desarrollar como un prototipo orientado a la participación y a la investigación en las nuevas tecnologías de la SDR motivo por el cual ha sido objetivo tener equipos en funcionamientos y no hacer solo un estudio teórico.

Teniendo en cuenta esto, y dado el entorno en desarrollo en el que nos encontramos, las líneas generales han llevado a investigar en las nuevas iniciativas dentro de esta área, desarrollándose algunas en el momento en el que se realizaba esta Tesis. Es por ello que se ha ido participando en los grupos de trabajo de cada iniciativa intentando probar las soluciones (obviamente inicialmente ninguna funcionaba) y añadiendo la solución al Demostrador de Conceptos Marconi cuando funcionaba. Ninguna de las aplicaciones usadas en el Demostrador Marconi es comercial, y actualmente es una línea de desarrollo en evolución. Si bien aquellas aplicaciones que se han usado para desarrollo se pueden encontrar en un mercado profesional (Teamspeak, phpBB, HTML, Debian, Ubuntu, LAMP, JAVA, Android, Compiladores C y C++, Qt4 y Qt5, Eclipse...) y su uso requiere un conocimiento en estas áreas.

Los programas probados son también de uso libre, habiendo probado muchos más de los indicados en esta Tesis, pero recogiendo en ésta los que funcionaban, después de haberlos probados de forma exhaustiva con cada uno de los receptores seleccionados y

depurando los evidentes problemas de funcionamiento en las distintas plataformas (DLL, SO,...).

Una última aportación que no pensábamos incluir pero que el avance del grupo de desarrollo así nos lo ha exigido es GNU Radio.

GNU Radio es una plataforma perfecta para la simulación y la creación de radios usando diagramas de bloque.

Dentro del entorno de WebSDR (aplicación montada en esta Tesis), ha aparecido a principios de este año (2014) un servidor WebSDR que usa como receptor un servidor de GNU Radio virtualizado. En diciembre de 2013 GNU Radio ha sacado una distribución donde integra el receptor RTL-SDR usando drivers del grupo Osmocom.

Evidentemente dado su carácter didáctico y su precio asequible (Precio del receptor 10 euros, frente a los 3000 euros de coste de la tarjeta USRP de Ettus Research) se investiga, prueba, depura e instala un servidor con GNU Radio.

Esto demuestra cuál ha sido el interés durante esta Tesis Doctoral: la investigación en las áreas de ROIP y SDR con participación en los grupos de desarrollo y la instalación de prototipos como base fundamental para participar, mantener las relaciones y las bases de la investigación sobre esta área apasionantemente despierta.

7. Bibliografía.

[ANDR11]

GOOGLE, *Android™ mobile technology platform 2.3.4*, Copyright © 2010 Google Inc. All rights reserved.

[BATE02]

BATES; R.J., *Broadband Telecommunications Handbook, Second Edition*, Copyright © 2002 by The McGraw–Hill Companies, Inc. ISBN 0–07–139851–1

[BOE77]

MINISTERIO DE EDUCACIÓN Y CIENCIA, *Orden por la que se aprueba el plan de estudios de la Carrera de Náutica, Sección de Puente, Máquinas y Radioelectrónica*, BOE 214, 24 octubre 1977, Pag 233-242

[BOE79]

MINISTERIO DE TRANSPORTES Y COMUNICACIONES, *Orden de 27 de septiembre de 1979, determinando las Escuelas que otorgarán los títulos profesionales para el Servicio Radioelectrónico de la Marina*, NR 251, 19 octubre 1979, pag 243-283

[BOER13]

Información accesible en <http://websdr.org>

[BLOS04]

BLOSSOM; E., *GNU Radio: Tools for Exploring the Radio Frequency Spectrum; Listening to FM Radio in Software, Step by Step*, Linux Journal Jun/Sep 2004.

[BONS98]

BONSER; W., *Speakeasy Military Software Defined Radio*, 1998 International Symposium in advanced Radio technologies.

[BORO et al. 09]

BORONCZYK; T., NARAMORE; E., GERNER; J., LE SECOUAMEC; Y., STOLZ; J., GLASS; M.K., *Beginning PHP6, Apache, MySQL Web Development*, Copyright © 2009 by Wiley Publishing, Inc., Indianapolis, Indiana, ISBN: 978-0-470-39114-3

[BRIC07]

BRICLE; F., *The FSM Virtual Radio Kernel*, Copyright © 2007 by Frank Brickle Creative Commons License ATTR/NC/ND

[BROW09]

BROWN; S., *Client-Server Protocol Interfacing to the SDR–Radio Server*. SDR-Radio 2009.

[BURN02]

BURNS; P., *Software Defined Radio for 3G*. Artech House Publishers. Nueva York. 2002.

[BURN09]

BURNETTE; E., *Hello, Android Introducing Google's Mobile Development Platform, 2nd Edition*. Copyright © 2009 Ed Burnette. ISBN-10: 1-934356-49-2

[CABE00]

CABEZA GALAN; A., *Fundamentos básicos de las telecomunicaciones, Servicios de formación de Telefónica de España S.A.U.*

[CHAV05]

CHÁVEZ REYES; J.A., *Software Radio y el USRP*. Santa Catarina Mártir, Cholula, Puebla. 2005.

[CONÇ07]

GONÇALVEZ; F., *Asterisk PBX Guía de configuración*, Copyright 2005© por Flavio E. Gonçalves.

[CLAN et al. 09]

CLANCEY; C, GOLDSCHMITT; H, KASTNER; J, OBERLANDER; E, WALKER; P, *IPCop v1.4.21 Administration Manual*, Copyright © 2002-2009 Chris Clancey, Harry Goldschmitt, John Kastner, Eric Oberlander, Peter Walker

[CUMM et al. 99]

CUMMINGS; M., HARUYAMA; S., “*FPGA in the Software Defined radio*”. *IEEE Communications Magazine*, Vol.37 No.2. Nueva York, febrero de 1999.

[DARI et al.08]

DARIE; C, BALANESCU; E, *Beginning PHP and MySQL E-Commerce From Novice to Professional, Second Edition*. Copyright © 2008 by Cristian Darie and Emilian Balanescu, ISBN-13 (pbk): 978-1-59059-864-1

[DEMP et al 06]

DEMPSTER; B, EATON-LEE; J, *Configuring IPCop Firewalls Closing Borders with Open Source*, Copyright © 2006 Packt Publishing, ISBN 1-904811-36-1

[DEMP et al. 06]

DEMPSTER; B., GARRISON; K., *Trixbox Made Easy*, Copyright © 2006 Packt Publishing, ISBN 1-904811-93-0.

[DESA12a]

DESARROLLOWEB; *Manual de Flash*, www.desarrolloweb.com

[DESA12b]

DESARROLLOWEB; *Programación en PHP*, www.desarrolloweb.com

[DESA12c]

DESARROLLOWEB; *Manual de jQuery*, www.desarrolloweb.com

[DIMA08]

DIMARZIO; J.F. , *Android™ A Programmer's Guide*, Copyright © 2008 by The McGraw-Hill Companies, DOI: 10.1036/0071599886

[EB1A13]

EB1AGG; *Apuntes de radio Configuración HDSDR y SDRSharp en Windows con RTL2832U*.

[ECK11]

ECK; D.J, *Introduction to Programming Using Java, Version 6.0*, June 2011, Hobart and William Smith Colleges.

[EGUI08]

EGUÍLUZ PÉREZ; J, *Introducción a AJAX*, www.librosweb.es

[ESCU et al. 06]

ESCUADERO-PASCUAL; A., BERTHILSON; L., *VoIP para el desarrollo. Una guía para crear una infraestructura de voz en regiones en desarrollo*, Internacional Development Research Centre, Diciembre 2006.

[FERR12]

FERRER; J., GARCÍA; V., GARCÍA; R., *Curso completo de HTML*. Copyright Jorge Ferrer, Rodrigo Garcia y Victor García.

[FONG et al. 02]

FONG; P.J. , KNIPP; E. , GREY; D, *Configuring Cisco Voice Over IP*, Copyright © 2002 by Syngress Publishing, Inc, ISBN: 1-931836-64-7

[GUER et al. 11]

GUERRERO MARTÍNEZ; J.F., FRANCÉS VILLORA; J.V., *Sistemas electrónicos para el tratamiento de la información*, Universidad de Valencia-Escuela Técnica Superior de Ingeniería 2010-2011.

[GOME99]

GÓMEZ HERNÁNDEZ; J.E., FLORES SANTACRUZ; D., AHIJADO MARTÍN-NAVARRO; F.J., *Curso de Linux Versión 2.0*, 1999 Gómez, Ahijado, Flores.

[GUST08]

GUSTAVO VEGA; S, *Asterisk – The Open Source PBX, Telefonía IP mediante Asterisk PBX*, Universidad Nacional de Luján, Int. Ruta 5 y 7, 6700 Luján, Buenos Aires, Republica Argentina, Año 2008.

[HARA et al. 02]

HARADA; H., PRASAD; R., *Simulation and Software radio for mobile communications*. Artech House Publishers. Nueva York. 2002.

[HARL11]

HARLOW; S.A., *Sigmira 1r7 Operations Manual*, Copyright 2008, 2011, Steven A. Harlow

[HART04]

HART; C., “*Software Defined Radio (SDR): North American and European market demand analysis*”. Venture Development Corporation. Miami. 2004.

[HASE08]

HASEMAN; C, *Android Essentials*, 2008 by Chris Aceman, ISBN-13 (electronic): 978-1-4302-1063-4

[IRON06a]

IRONTEC; ## *Astirectory: Usuarios SIP en Directorio LDAP* ## .

[IRON06b]

IRONTEC; CORROTXATEGI; G, BAZ; I, *Voz sobre IP y Asterisk*.

[IRON07]

IRONTEC; CORROTXATEGI; G, BAZ; I, BONILLA; J, *Curso Asterisk*.

[JOHN06]

JOHNSON; L., *The high performance Software Defined Radio Project*, Disponible en www.tapr.org/pdf/dec2006-HPSDRproject-KK7P.pdf

[KELL05]

KELLY; T, *VOIP para Dummies*, Copyright © 2005 by Wiley Publishing, Inc., Indianapolis, Indiana, ISBN: 0-7645-9948-8

[KENI05]

KENINGTON; P.B., *RF and Baseband Techniques for Software Defined Radio*. Artech House Publishers. Londres. 2005.

[LACK et al. 95]

LACKEY; R.I.,UPMAL; D.W., “*SpeakEasy: The military Software Radio*”. *IEEE Communications Magazine*, Vol.33 No.5. Nueva York, mayo de 1995.

[LAND09a]

LANDIVAR; E., *Comunicaciones Unificadas con Elastix Volumen 1*, Copyright (c) 2008-2009 Edgar Landívar.

[LAND09b]

LANDIVAR; E., *Comunicaciones Unificadas con Elastix Volumen 2*, Copyright (c) 2008-2009 Edgar Landívar.

[LATH86]

LATHI; B.P., *Sistemas de Comunicación*, Nueva Editorial Interamericana, 1986.

[MAIL12a]

MAILXMAIL.COM, *Cómo crear un podcast y/o tu propio canal de audio*.

[MAIL12b]

MAILXMAIL.COM, CAMPOS;J.A., *Cómo grabar archivos de audio de voz*.

[MART et al. 07]

MARTÍNEZ; L.F., TERAN; W., *Instalación y configuración de un servidor Asterisk*, junio 2007.

[MASC00]

MASCAREÑAS, C. *Medios de enseñanza de la Radioelectrónica en la Universidad de Cádiz*. Revista de la Unión de Radioaficionados Españoles. Madrid. pp. 58-62. Febrero. 2000.

[MASC11]

MASCAREÑAS, C. *Manual Básico de Sistemas de Comunicaciones Marítimas. 2ª Ed.* Servicio de Publicaciones de la Universidad de Cádiz. Cádiz. 2011.

[MEGG et al. 07]

MEGGELEN; J.V., MADSEN; L., SMITH; J., *Asterisk™: The Future of Telephony, Second Edition*, Copyright © 2007, 2005 O'Reilly Media, Inc. All rights reserved, ISBN-10: 0-596-51048-9

[MITO00]

MITOLA; J., *Software Radio Architecture: Object oriented approaches to wireless systems engineering*. John Wiley & Sons, Inc. Nueva York. 2000.

[MITO92]

MITOLA; J., *Software radio-survey, critical evolution and future directions*, Telesystems Conference 1992 NTC-92, vol 13 pages 15-23, 1992

[NICH10]

NICHOLS; K., *Softrocks SDR Kits Custom Enclosures and Anodizing*, Kevin Nichols KA7OFR, Dic 2010.

[NOVA08]

NOVAVOX, *X100P SE PCI Card Installation Guide*, Novavox Limited

www.novavox.co.uk

[NYQU28]

NYQUIST; H., *Certain topics in telegraph transmission theory*, Trans. AIEE, vol 47, pag 617-644, Apr. 1928.

[QUIN et al.12]

QUINTERO; G., CEBALLOS BETANCOUR; C., DE SANCTIS GIL; L., *SDR: La alternativa para la evolución inalámbrica a nivel físico. Semillero de Tecnologías Inalámbricas (STI) – GIDATI*

[REME13]

Información accesible en <http://www.proteccioncivil.org/remer>

[ROGE et al. 09]

ROGERS; R., LOMBARDO; J., MEDNIEKS; Z., MEIKE; B., *Android Application Development*, Copyright © 2009 Rick Rogers, John Lombardo, Zigurd Mednieks, and Blake Meike. ISBN: 978-0-596-52147-9

[SANC04]

SÁNCHEZ; J., *Java2 incluye Swing, Threads, programación en red, JavaBeans, JDBC y JSP / Servlets*, J. Sánchez 2004.

[SAWY12]

SAWYER MCFARLAND; D., *JavaScript & jQuery: The Missing Manual, Second Edition*, Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472., ISBN: 978-1-449-3-9902-3

[SCH100]

SCHIPHORST; R., *Demonstration of the Software-Radio Concept*. Tesis de Maestría. Universidad de Twente, Departamento de Ingeniería Eléctrica. Twente. 2000.

[SDRR10]

SDR-RADIO; *Client-Server User Guide Using the Client-Server support in the SDR-RADIO.com console*, January 22nd, 2010

[SDRR11a]

SDR-RADIO; *IQ Data Files Recording, Playback, Analysis*. SDR-Radio 2011.

[SDRR11b]

SDR-RADIO; *Remote Server "Over the Hills and Far Away"*, SDR-Radio 2011.

[SDRR11c]

SDR-RADIO; *Funcube Dongle SDR-RADIO.com Console*, SDR-Radio 2011.

[SDRR12]

SDR-RADIO; *Soundcard-Based SDRs. SDR-RADIO.com Console*. SDR-Radio 2012.

[SELL09a]

SELLÉS ROSA; F., *Manual de Asterisk y otras hierbas*, Copyright ©2009 Fabian Sellés Rosa, UCA.

[SELL09b]

SELLÉS ROSA; F., *Introducción a la telefonía IP utilizando estándares*, Copyright ©2009 Fabian Sellés Rosa, UCA.

[SMIT97]

SMITH; S.W., *The Scientist and Engineer's Guide to Digital Signal Processing*, Copyright 1997 by California Technical Publishing.

[TRAN et al. 98]

TRANter; W.B., et al. *An overview of the VirginiaTech Program in Software Radios implemented with Reconfigurable Computing*. Presentación, Instituto Politécnico de Virginia y Universidad Estatal. Blacksburg. 1998.

[TUTT02]

TUTTLEBEE; W., *Software Defined Radio: Origins, drivers and international perspectives*. John Wiley & Sons, Inc. Nueva York. 2002.

[UIT12]

UNIÓN INTERNACIONAL DE TELECOMUNICACIONES,

<http://www.itu.int/es/pages/default.aspx>

[YOUN03]

YOUNGBLOOD; G., *A Software-Defined Radio for the Masses – Part I, II, III and IV*. Sixth Market, Inc. Austin. 2002 a 2003.

Enlaces WEB:

American QRP Club “SoftRock-40”

<http://www.amqrp.org/kits/softrock40/index.html>

Andrea Mostefusco Web

<http://www.montefusco.com/>

Building HF SoftRock RX Ensemble II Receiver Kit :

http://www.wb5rvz.org/ensemble_rx_ii/index?projectId=16

Building softrock lite II :

http://www.wb5rvz.org/softrock_lite_ii/index/projectId=8

Building softrock VHF :

http://www.wb5rvz.com/sdr/ensemble_rx_ii_vhf/

Building softrock RXTX :

http://www.wb5rvz.org/ensemble_rxtx/index?projectId=14

Comunidad de usuarios Asterisk-ES

<http://comunidad.asterisk-es.org/index.php?title=Portada>

Control remoto por VNC

<http://www.aitoroliveira.com/hazlotumismo/vnc.php>

Creando una radio en internet

<http://creandounaradio.blogdiario.com/>

FlexRadio Systems

<http://www.flex-radio.com/>

Funcube Dongle con Quisk SDR

<http://www.oz9aec.net/index.php/component/content/article/63-sdr/419-funcube-dongle-with-quisk-sdr>

G8JCF's Software Defined Radio

<http://www.g8jcf.dyndns.org/>

ghpsdr3-alex Installing QtRadio, dspserver, and an SDR driver

<http://www.n8mdp.com/ghpsdr3-alex.php>

ghpsdr3-alex / QtRadio / HPSDR Page

<http://www.nitehawk.com/w3sz/ghpsdr3-alex.htm>

Github A.Montefusco, GNU Radio

<https://github.com/amontefusco/gnuradio-amontefusco/tree/perseus>

Github n8mdp/jmonitorWIDE

<https://github.com/n8mdp/jmonitorWIDE>

GNU Radio

<http://gnuradio.org/redmine/projects/gnuradio/wiki>

GNU Radio Hardware

<http://gnuradio.org/redmine/projects/gnuradio/wiki/Hardware?version=9>

Ham Radio Science, RTL2832U / R820T vs RTL2832U / E4000

<http://www.hamradioscience.com/rtl2832u-r820t-vs-rtl2832u-e4000/>

Ham Radio Science, RTL1090 ADS-B Decoder for Windows

<http://www.hamradioscience.com/rtl1090-ads-b-decoder-for-windows/>

High Performance Software Defined Radio, An Open Source Design

<http://openhpsdr.org/index.php>

How to install Linrad under Linux and PC-BSD

<http://www.sm5bsz.com/linuxdsp/install/compile/lincompile.htm>

HTML Web

<http://www.psycobyte.com/html/index.html>

Instalación de RTL-SDR

http://inst.eecs.berkeley.edu/~ee123/fa12/rtl_sdr.html

Introduction to Programming Using Java, Sixth Edition

<http://math.hws.edu/javanotes/>

Jmonitor

<http://www.n8mdp.com/jmonitorWIDE.php>

Jmonitor Listen and Control my Softrock Ensemble II SDR

<http://www.n8mdp.com/jmonitor.php>

John Melton Blogspot

<http://g0orx.blogspot.com.es/>

KD5TFD - Round Rock, Texas
<http://ewjt.com/kd5tfd/index.html>

Linux Software Repositories
<http://www.google.com/linuxrepositories/>

N8MDP web
<http://www.n8mdp.com>

OK1HRA remote HF station
<http://ok1hra.nagano.cz/>

Perseus on Internet, Andrea Motefusco
<http://www.montefusco.com/perseuscs/>
Planeplotter – RTL1090
<http://planeplotter.pbworks.com/w/page/62409382/RTL1090>

PM-SDR
<http://www.iw3aut.altervista.org/>

QtRadio Listen to Linux-based SDR's
<http://www.n8mdp.com/qtradio.php>

QtRadioManager
<http://www.oliver-goldenstein.de/qtradiomanager/>

Radioafición.com, SDR Perseus, como configurar V4.0
<http://www.radioaficion.com/HamNews/articles/3234-sdr-perseus-como-configurar-v40.html>

Remote Control Your Station Over The Internet
<http://www.remoterig.com/wp/>

RTL y borIP
http://www.youtube.com/watch?v=_2PaSeR-4Ck

RTL-HDSDR
<http://www.youtube.com/watch?v=WZijmI3s4Ec>

RTL-SDR
<http://sdr.osmocom.org/trac/>

SDR-Radio con RTL-SDR
<http://www.hamradioscience.com/sdr-radio-com-v2-0-now-supports-rtl2832u-sdr/>

SDR Sharp
<http://sdrsharp.com/>

SDR-Shell for IF Applications
<http://people.wallawalla.edu/~rob.frohne/SDR/SDR-Shell/>

SDR & Linux a Beginner's Guide to Installation
http://www.n8mdp.com/sdr_quisk.html

SDR on Linux (e.g. Ubuntu)
<http://www.n8mdp.com/sdrLinux.php>

Servidores de QtRadio
<http://qtradio.napan.ca/qtradio/qtradio.pl>

Sitio de descarga sdr-radio v2 (soporta rtl-sdr)
<http://v2.sdr-radio.com/Download.aspx>

TAPR organization
<http://www.tapr.org/>

The Acorn-SDR Project
<http://www.g3ukb.co.uk>

Tutorial HTML
<http://html.conclase.net/tutorial/html/>

Tutoriales de Programación
<http://tutorialesdeprogramacion.blogspot.com/2009/03/aptana-123-ajax-javascript.html>

Unión Radioaficionados Españoles
<http://www.ure.es/>

WB5RVZ.ORG - Documentation Projects
<http://www.wb5rvz.org/>

WB5RVZ Software Defined Radio Homepage
<http://www.wb5rvz.com/sdr/>

WebSDR organization
<http://websdr.org/>

“www.KD5QLM.com” An Amateur Radio Blog
<http://www.kd5qlm.com/>

8. Apéndices.

Apéndice 1. Instalación de las fuentes de GNU Radio.

Para instalar las fuentes de GNU Radio, primero se descargan los siguientes paquetes de GNU:

- gnuradio-core: el núcleo de las librerías.
- gnuradio-examples: ejemplos de GNU Radio, donde encontraremos los programas para TX y RX de FM.
- gr-audio-oss: soporte para tarjetas de sonido que utilizan OSS (Open Sound System).
- gr-audio-alsa: soporte para tarjetas de sonido que utilizan ALSA (Advanced Linux Sound Architecture).
- gr-usrp: librerías que unen a GNU Radio con el Universal Software Radio Peripheral.
- gr-wxgui: wxPython basado en herramientas GNU, incluye un osciloscopio y FFT's.
- gr-howto-write-a-block: incluye ejemplos de cómo escribir un bloque.
- usrp: soporte para la tarjeta USRP.

Antes de instalar gnuradio-core es necesario tener algunos paquetes pre-instalados. Estos paquetes son los siguientes:

FFTW (Fastest Fourier Transform in the West): Ésta es una subrutina en C que es capaz de manejar transformadas discretas de Fourier en una o más

dimensiones, el tamaño de la entrada es arbitrario y maneja datos complejos y reales. FFTW es software libre por lo que se convierte en una librería de FFT para cualquier aplicación.

(<http://www.fftw.org/download.html>)

Cppunit: Es una unidad de prueba para chequear bloques en C++.

(https://sourceforge.net/project/showfiles.php?group_id=11795)

SWIG (Simplified Wrapper and Interface Generator): Son herramientas para el desarrollo de software que conectan programas escritos en C o C++ con una gran variedad de lenguajes de programación de alto nivel. SWIG es utilizado con diferentes tipos de lenguaje como Python.

(<http://sourceforge.net/projects/swig/>)

Numarray y Numpy: Son dos módulos de Python necesarios para el funcionamiento de GNU Radio. Éstos se utilizan para cálculos numéricos y los podemos encontrar en la dirección siguiente.

(<http://sourceforge.net/projects/numpy>)

wxPython: Son herramientas para el lenguaje de programación Python, y permite crear programas con funciones e interfaces gráficas de forma simple. Está implementado como un módulo de Python.

(https://sourceforge.net/project/showfiles.php?group_id=10718)

Hay que instalar también las siguientes librerías: automake1.8, gcc-3.4 y g++-3.4.

Las únicas librerías que no se encuentran en los repositorios son las de SWIG por lo que es necesario instalarlas manualmente.

Una vez tenemos todas las librerías, se ejecutan los comandos:

```
# ./configure  
# make  
# make check
```

```
# make install
```

Para obtener los paquetes de GNU Radio se utiliza CVS que es una forma de descargar los paquetes en su última versión sin necesidad de hacerlo desde una página de Internet.

Seguidamente se deben obtener las actualizaciones para el USRP, para ello lo único que se debe hacer es descargar este archivo de www.gnu.org/software/gnuradio.

Apéndice 2. Configuración de Asterisk.

Asterisk se controla a través de archivos de configuración localizados en el directorio `/etc/asterisk`.

La gramática de comandos del lenguaje Asterisk es:

Grupos simples:

El formato es básico, los objetos se declaran con todas las opciones en la misma entrada, ejemplo de esto son los archivos `extensions.conf`, `meetme.conf`, `voicemail.conf`.

Ejemplo :

```
[sección]
```

```
objeto1 => op1,op2,op3
```

```
objeto2 => op1b,op2b,op3b
```

Objeto 1 se crea con las opciones `op1`, `op2`, `op3` y el objeto 2 con las opciones `op1b`, `op2b`, `op3b`.

Formato de objeto con herencia de opciones:

Se usa en fichero de interfaces `zapata.conf` y otros. El archivo de configuración tiene varias secciones, que contienen declaraciones de uno o más canales u objetos.

Ejemplo:

```
[sección]
```

```
op1 = bas
```

```
op2 = adv
```

```
objeto=>1
```

```
op1 = int
```

```
objeto => 2
```

Se crea el objeto 1 con las opciones op1 = bas y op2 =adv.

El objeto 2 tiene la op1 = int, y la op2 se queda igual a adv, ya que no se redefine.

Objeto entidad compleja:

Se usa en ficheros de configuración complicados, como iax.conf y sip.conf.

Cada entidad recibe su propio contexto, pudiendo existir un contexto general para las configuraciones globales.

```
[entidad1]
```

```
op1=valor1
```

```
op2=valor2
```

```
[entidad2]
```

```
op1=valor3
```

```
op2=valor4
```

La entidad 1 tiene valores valor1 y valor2 para las opciones op1 y op2. La entidad 2 tiene valor3 y valor4 para las opciones op1 y op2.

Plan de llamadas:

Plan de marcado básico:

```
[entrada]
```

```
exten=>s,1,answer()
```

```
exten=>s,2,playback(goodbye)
```

```
exten=>s,3,hangup()
```

Una llamada que entre por la FX0 se envía al contexto [entrada], configurada en el fichero zapata.conf para el canal FX0.

La llamada se envía a la extensión s (inicio), y con prioridad 1, 2, 3 va ejecutando las aplicaciones .

Otro ejemplo:

[entrada]

exten=>s,1,Answer()

exten=>s,2,Background(saludos)

exten=>s,3,hangup()

exten=>1,1,playback(soporte)

exten=>1,2,goto(soporte,s,1)

exten=>2,1,playback(entrenamiento)

exten=>2,2,goto(entrenamiento,s,1)

exten=>3,1,playback(ventas,)

exten=>3,2,goto(ventas,s,1)

Al llamar por la interfaz FX0, la comunicación se pasa a la extensión s, que atiende la llamada usando el comando Background, que hace sonar un saludo, y aguarda el marcado de un dígito.

Al pulsar el 1, va a la extensión 1, prioridad 1, toca un mensaje, y el comando goto() lo manda al contexto soporte .

[entrada]

exten=>s,1,Answer()

exten=>s,2,Background(saludo)

exten=>s,3,hangup()

exten=>1,1,playback(soporte)

exten=>1,2,Dial(SIP/8000)

exten=>2,1,playback(entrenamiento)

exten=>2,2,Dial(ZAP/1)

exten=>3,1,playback(ventas)

exten=>3,2,Dial(IAX/8002)

En este ejemplo, en vez de mandar a un contexto, se manda a los canales.

Apéndice 3. Estimación del tráfico

Una forma de estimar el tráfico en una centralita telefónica, es conociendo el número de llamadas por usuarios.

En PBX de empresas se estima 1 llamada cada 5 líneas.

En PBX de áreas residenciales se estima 1 llamada cada 16 líneas.

Veamos un ejemplo:

Sede central: 120 líneas, 24 llamadas simultáneas

Filial1: 30 líneas, 6 llamadas simultáneas

Filial2: 15 líneas, 3 llamadas simultáneas

Usando el CÓDEC g.729 tendríamos:

Ancho banda filial 1 (Frame-relay): $26,8 * 6 = 160,8$ Kbps

Ancho banda filial 2 (Frame-relay): 80,4 Kbps

Método Erlang B :

Erlang es una medida de tráfico en telecomunicaciones. Se usa para definir el volumen de tráfico de una hora.

Supongamos que tenemos 20 llamadas de 5 minutos en una hora.

Los minutos totales de tráfico en esa hora son: $20 * 5 = 100$ minutos.

Las horas de tráfico en esa hora son = $100 / 60 = 1,66$ Erlangs (horas de tráfico por hora).

Para usar el método Erlang debemos saber los minutos de llamadas en la hora más ocupada BHT (Busy Hour Traffic), esto último se puede hacer de dos formas, la primera tarificación, la segunda simplificación (BHT=17% del número de minutos llamados durante el día) .

Otra variable importante es el GoS (grade of service). Define la probabilidad de que las llamadas se bloqueen por falta de líneas.

Pueden ser 5 de 100 (0,05), 1 de 100 (0,01).

Seguimos con el ejemplo:

En filial 1 suponemos en la BHT 100 llamadas de 3 minutos y GoS 0,01.

$BHT=100*3=300 \text{ min} / 60 = 5 \text{ Erlangs}$.

Entrando en la calculadora erlangs <http://www.erlang.com/calculator/erlb/>

The screenshot shows the 'Erlang B Calculator' web page. The header includes the 'Westbay Engineers Limited' logo and navigation links: Home | Free calculators | Products | Tech. papers | Forum | About us. The main content area displays the calculator interface with the following fields and values:

BHT (Erl.)	Blocking	Lines
<input type="radio"/> Unknown 5.000	<input type="radio"/> Unknown 0.010	<input checked="" type="radio"/> Unknown 11

Buttons: Calc., Results, Help.

A windows version is available for immediate download at just 95 US Dollars. [Click here for more information](#)

Brief instructions: Here is a brief introduction to the calculator. For more detailed information, press the *Help* Button. For a running record of the results calculated, press the *Results* button. Both these functions open new browser windows on your desktop.

Figura A3.1 Calculadora Erlang para el ancho de banda

Para este ejemplo son necesarios 11 enlaces.

Códec Seleccionado: g.729

Tipo de canal: PPP

Banda estimada para filial 1: $26.4 \times 11 = 290.4 \text{ Kbps}$

En la filial 2, tenemos que el día más ocupado hubo 1200 llamadas, de 2 minutos y

veinte segundos.

En este caso para la hora más ocupada estimamos el 17% del total del día, con un GoS de 0,01 .

Volumen diario total: $1200 * 140 \text{ s} = 168000$ segundos por día = 46,66 horas.

Hora más ocupada = $17\% * 46,66 = 7,93$ erlangs.

The screenshot shows the Westbay Engineers Limited website. The main content area displays the 'Erlang B Calculator' tool. The calculator window has three input fields: 'BHT (Erl.)' with a value of 7.930, 'Blocking' with a value of 0.010, and 'Lines' with a value of 15. Below the inputs are buttons for 'Calc.', 'Results', and 'Help'. A promotional message below the calculator states: 'A windows version is available for immediate download at just 95 US Dollars. Click here for more information'. The website header includes the company logo and navigation links: Home | Free calculators | Products | Tech. papers | Forum | About us. A sidebar on the left contains a 'Free calculators' section with a 'Calculator index' listing: Erlang B, VoIP bandwidth, Call center, Call minutes, Erlangs to VoIP, Ext. Erlang B, Erlang C, and Engset. Below this is a 'Site highlights' section with a 'Home page' link.

Figura A3.2 Estimación de líneas necesarias .

Son necesarias 15 líneas.

Banda estimada para filial 2: $26.4 * 15 = 396$ Kbps

Apéndice 4. Configurar un cliente SIP.

Los pasos para configurar un cliente SIP son:

- 1 Configurar una entrada en el archivo sip.conf para cada teléfono.
- 2 Configurar el teléfono.
- 3 Configurar la extensión en el plan de marcado.

Paso 1, Configurando sip.conf :

Teléfono normal:

```
[4101]
type=friend
context=default
callerid=Manuel Palma<8550>
host=dynamic
canreinvite=yes
dtmfmode=info
mailbox=1234@default
disallow=all
allow=ulaw
allow=g729
secret=contraseña
deny=0.0.0.0/0.0.0.0
permit=192.168.1.0/255.255.255.0
musicclass=default
```

Paso 2, Configuración del teléfono:

Se deben seguir las instrucciones del teléfono.

Paso 3, definir la extensión en el archivo sip.conf:

Ejemplo:

```
exten=> _41XX,1, Dial(SIP/${EXTEN},10,t)
```

Llamando a una extensión que comienza con 41 y dos dígitos más, el usuario conectado recibe la llamada.

Aparte de esto, hay que configurar las extensiones en el fichero extensions.conf.

Opciones de configuración para los teléfonos:

type=friend

Se usa para teléfonos que hacen y reciben llamadas.

context=default

Contexto usado para recibir llamadas de este teléfono. Este teléfono solo puede llamar a números incluidos en este contexto del plan de marcado.

callerid=Manuel Palma <8550>

Se puede definir un identificador de llamadas para cada teléfono, con esto se puede ver el que llama en los displays de los teléfonos.

host=dynamic

Esto permite que el usuario se registre dando su IP actual.

canreinvite=yes

El protocolo SIP intenta conectar los puntos finales directamente.

dtmfmode=info

Sirve para que el teléfono pase el DTMF necesario a operaciones como transferencia, captura de llamada. Sus valores pueden ser:

dtmfmode=info . Las informaciones DTMF se pasan por la señalización SIP.

dtmfmode=inband . En esta forma el DTMF pasa como señal de audio normal .

dtmfmode=rfc2833. Permite pasar los tonos DTMF usando el protocolo RTP.

mailbox=1234@default . Permite asociar una cuenta de buzón al teléfono.

disallow=all. Deshabilita todos los códecs.

secret=contraseña. Permite poner contraseña para el usuario.

deny=0.0.0.0/0.0.0.0

permit=192.168.1.0/255.255.255.0. Permite sólo el registro de teléfonos de la red 192.168.1.0

musicclass=default. Música en espera.

videosupport=yes/no. Permite llamadas entre dos usuarios con capacidad de vídeo.

language=br. Es el código de lenguaje definido en indications.conf – Define el lenguaje para los prompts y señales locales de telefonía.

rtptimeout=60. Termina la llamada si no hay actividad RTP x segundos, cuando no estuviera en espera (hold).

Rtpholdtimeout=120. Termina la llamada si no hay actividad RTP (cuando está en espera).

qualify=yes/no o tiempo en milisegundos. Envía periódicamente un mensaje para medir el tiempo de respuesta de peer. Cuando es mayor de 2000 ms considera el peer fuera de alcance.

Conexión a un proveedor SIP :

Asterisk es cliente de un servidor SIP, recibe y coloca llamadas para este servidor.

La recepción de llamadas se dirige a una extensión Asterisk.

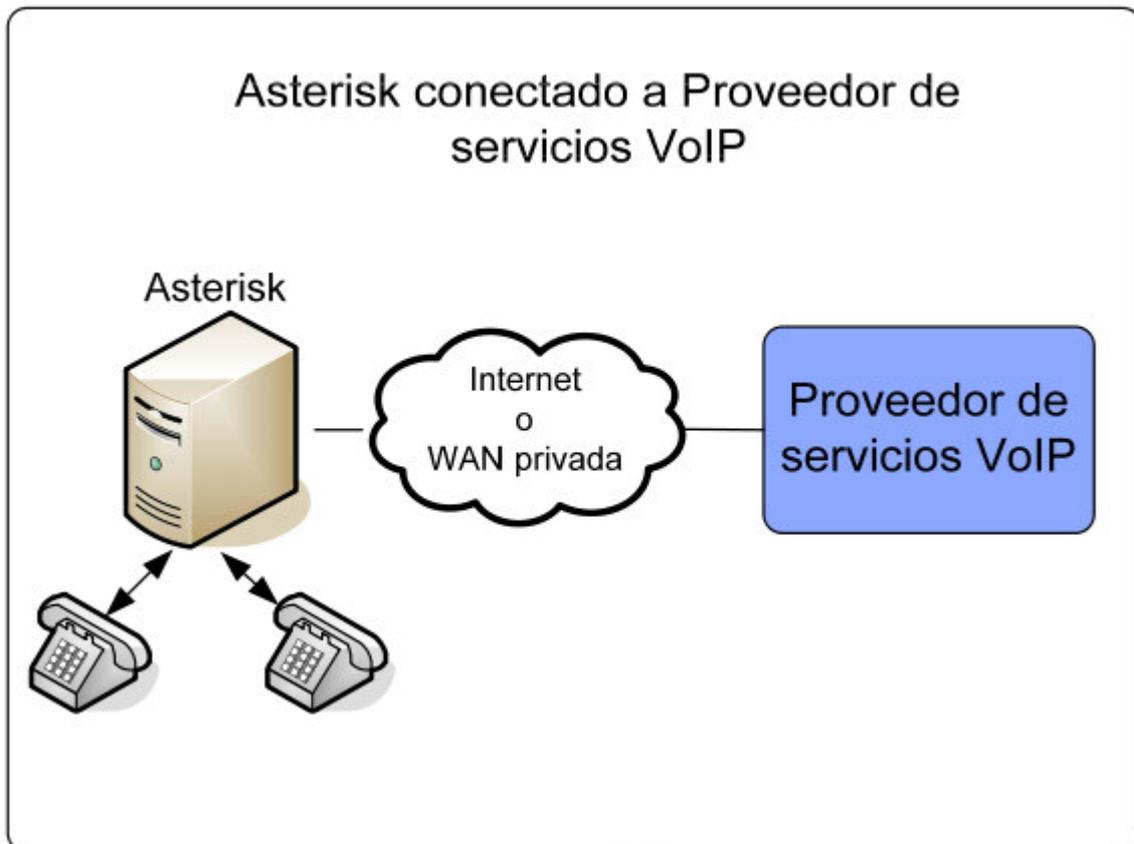


Figura A4.1 Asterisk conectado a servidor VoIP

Los pasos para configurar un proveedor SIP son:

Paso 1- Dentro de archivo sip.conf, en la sección [general] colocar una línea de registro en proveedor SIP.

Paso 2. Crear una entrada de tipo [peer] para el proveedor para simplificar el marcado.

Paso 3- Colocar una ruta de salida en el plan de marcado.

Paso 1: Registrar el proveedor (sip.conf).

En la sesión [general] colocamos:

```
register=>621538:password@fwd.fwdnet.net/4100
```

```
register=>ip1140623535:password@gvt.com.br/8573
```

Primera instrucción, queremos recibir cualquier llamada del primer proveedor en la extensión 4100 y del segundo proveedor en la extensión 8573.

Paso 2: Configurar el [peer] (sip.conf).

Crear una entrada del tipo [peer] para el proveedor en el sip.conf . Observar la línea “insecure=very” , que es necesaria si se quiere recibir llamadas del FWD . Si no se coloca esa línea, Asterisk le pide al proveedor la contraseña, como el proveedor no tiene cuenta en Asterisk, la llamada es rechazada. Esto sucede, por ejemplo, con el proveedor GVT mostrado en el ejemplo siguiente:

```
[gvt]
context=entrada
type=friend
callerid="ip1140623535" <1140623535>
dtmfmode=inband
canreinvite=no
username=ip1140623535
secret=[omitido por seguridad]
host=gvt.com.br
fromuser=ip1140623535
fromdomain=gvt.com.br
insecure=invite
```

Paso 3, Crear ruta de salida en el plan de marcado.

En este ejemplo, escogemos el dígito 010 como ruta de salida para el FWD. Para llamar al 610000, hemos de marcar 010610000.

```
exten=>_010.,1,SetCIDNum(621538})
exten=>_010.,2,SetCIDName(Flavio Goncalves)
```

```
exten=>_010.,3,Dial(SIP/${EXTEN:3}@gvt)
```

```
exten=>_010.,4,Playback(invalid)
```

```
exten=>_010.,5,Hangup
```

Opciones de configuración específica para proveedores:

```
register=>621538:password@fwd.fwdnet.net/4100
```

El comando register, en la sesión [general] del archivo sip.conf se usa para registrar el proveedor.

La extensión 4100 es donde esperamos recibir las llamadas de este proveedor .

Los parámetros de registro se pueden alterar en la sesión global .

```
Registertimeout=20
```

Intenta registrarse cada 20 segundos .

```
Registerattienepts=10
```

Intenta registrarse un máximo de 10 veces . (si fuera cero , reintenta continuamente)

```
CLI>sip show registry
```

Permite ver si Asterisk se registró correctamente .

```
username=ip1140623535
```

Es el usuario usado.

```
host=gvt.com.br
```

Es el Host a contactar.

```
fromuser=ip1140623535
```

```
fromdomain=gvt.com.br
```

Son parámetros a veces necesarios para autenticación en proveedores de voz sobre IP. Se colocan en el campo “From” del paquete SIP .

```
insecure=very
```

Cuando nos conectamos a un proveedor de voz sobre IP, y hacemos una llamada, el proveedor pide los datos identificativos. En el caso inverso, cuando el proveedor

encamina una llamada para Asterisk, éste pide autenticación del proveedor , enviando un mensaje con código 407 (Proxy Authentication Required). Evidentemente, un proveedor no tiene contraseña de acceso para todos sus clientes Asterisk. Este comando se usa para evitar que Asterisk envíe el paquete (407 Proxy Authentication Required).

Si se usa insecure=invite, Asterisk no pedirá autenticación en las llamadas que llegan del proveedor. Si se usa insecure=port , Asterisk encontrará el [peer] por la dirección IP no importando el puerto. Se pueden combinar los dos en “insecure=invite;port” .

Autenticación de las llamadas SIP entrantes:

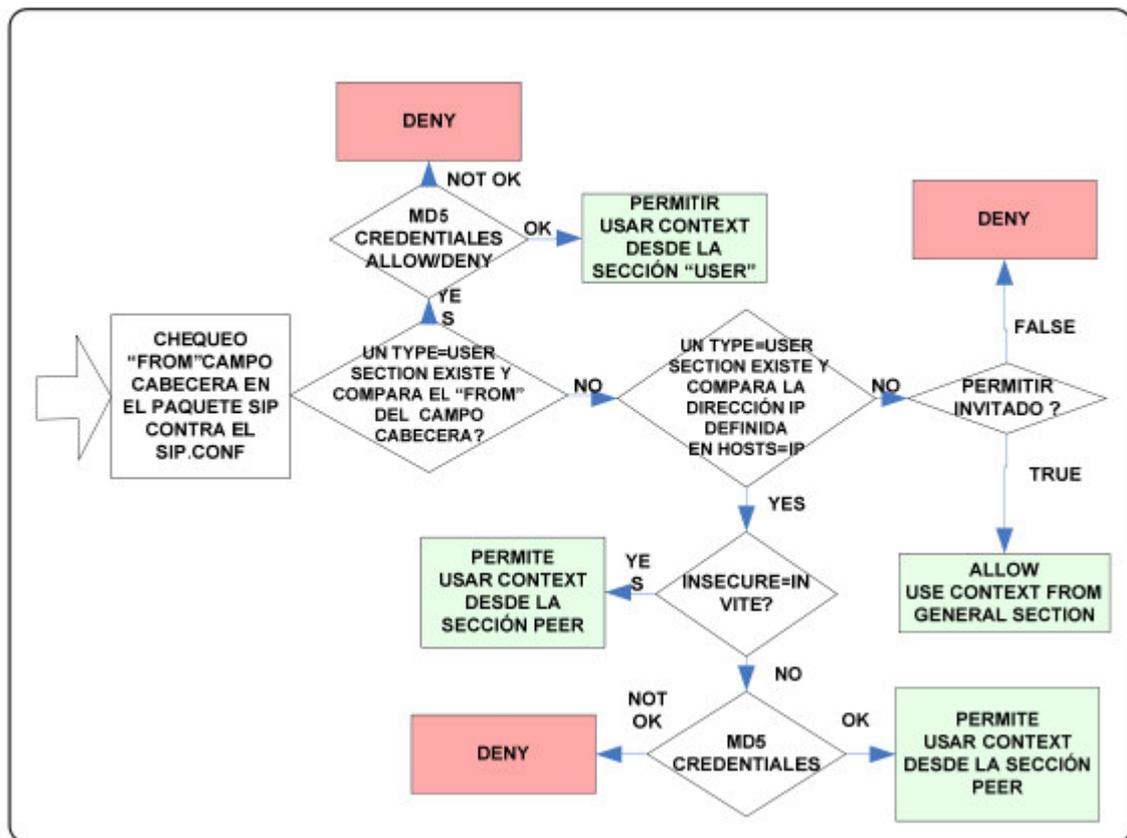


Figura A4.2 Diagrama de flujo de autenticación llamadas SIP.

Cuando Asterisk recibe una llamada SIP, se sigue el flujo indicado arriba.

Los parámetros de la sección [general] que regula la autenticación son:

allowguest=yes/no

Controla si un usuario sin un “peer” correspondiente, se puede autenticar sin un

nombre y contraseña.

insecure=invite;port

Impide que Asterisk genere un mensaje “407 Proxy Authentication Required” .

autocreatepeer=yes/no

Si esta opción es yes, cualquier usuario SIP se puede registrar en Asterisk como peer.

Apéndice 5. SIP NAT Transversal.

Existen cuatro tipos de NAT:

Full Cone

Restricted Cone

Port Restricted Cone

Symmetric

Full Cone:

Ejemplo, ordenador detrás de NAT con IP 10.0.0.1 enviando y recibiendo en el puerto 8000 se mapea a la puerta externa 200.180.4.168:1234.

Cualquiera desde Internet puede enviar paquetes a la dirección y puerto 200.180.4.168:1234 y estos paquetes pasan a 10.0.0.1:8000.

Restricted Cone:

El cliente interno envía un paquete para un ordenador externo, el NAT mapea al cliente 10.0.0.1:8000 para 200.180.4.168:1234. El ordenador externo puede enviar paquetes de vuelta, pero el NAT bloqueará los paquetes de otros ordenadores. El ordenador externo puede enviar paquetes independientemente del puerto.

Port Restricted Cone:

Si el cliente envía a un ordenador externo por el puerto 1010, el NAT no permitirá paquetes de vuelta por distinto puerto.

Simétrico:

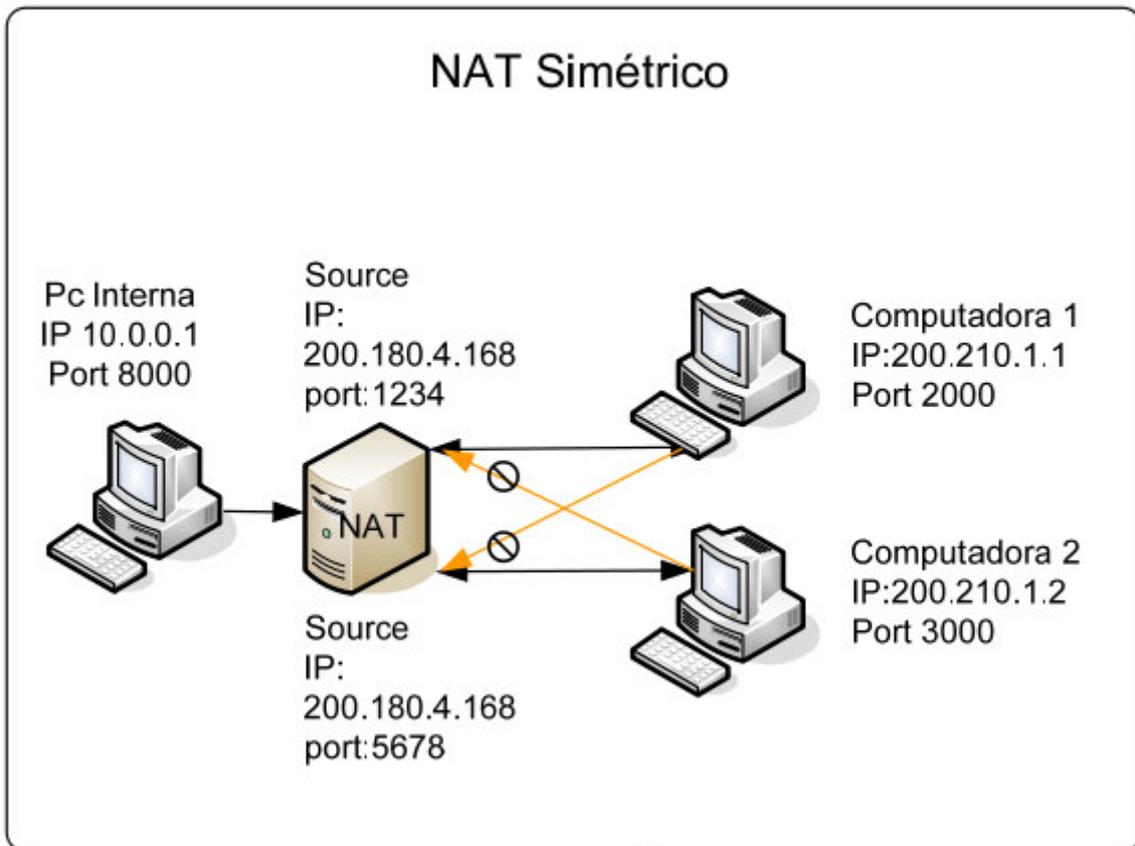


Figura A5.1 NAT simétrico.

Si un cliente envía de 10.0.0.1:8000 para el ordenador B, este puede ser mapeado como 200.180.4.168:1234. Si envía desde el mismo puerto para una dirección IP distinta, éste se mapea como 200.180.4.168:5678, El ordenador A y B debe responder a estos puertos.

	Precisa enviar datos para abrir a entrada	Puerto IP bien determinado para retorno	Restringe la entrada a la dirección IP:Puerto de Destino
Full Cone	No	Si	No
Restricted Cone	Si	Si	Solo la IP
Port Restricted Cone	Si	Si	Si
Simétrico	Si	No	Si

Figura A5.2 Consideraciones protocolos NAT.

STUN (Simple Transversal UDP NAT): Permite retornar el par IP / Puerto público, y determina el tipo de NAT usado.

Las peticiones STUN tienen los siguientes parámetros:

RESPONSE-ADDRESS: El servidor STUN envía su respuesta al par IP:Puerto especificado en este atributo . Si no existe este parámetro, el servidor STUN envía sus respuestas al par IP:Puerto de donde recibió el pedido .

Las respuesta del servidor STUN contiene las siguientes informaciones :

MAPPED-ADDRESS – El par IP:Puerto del cliente .

CHANGED-ADDRESS – La dirección IP fuente de la respuesta, si la petición se hizo con el flag Change IP .

SOURCE-ADDRESS : El par IP:Puerto de donde se envió la respuesta STUN.

Usando una combinación de peticiones diferentes al servidor STUN, se puede identificar si se está detrás de un firewall o de un NAT y de que tipo.

Apéndice 6. Instalación de QtRadio (versión SSL).

Para Linux Debian o Ubuntu.

Si queremos tener la última versión de QtRadio, hemos de descargarlo de Git. El PPA descargado no tiene más de unos días de diferencia con el máster.

Para compilar las fuentes descargadas, debemos descargar los compiladores:

Ubuntu/Debian

```
sudo apt-get install make gcc g++
```

```
sudo apt-get install autoconf automake autotools-dev libtool git subversion
```

Hay que instalar los paquetes de pre-requisitos (Ubuntu 12.04 y posteriores):

Instalando portaudio19-dev con jackd2

Si está jackd1 y portaudio19-dev ya instalados

Usando Synaptic

Desinstalar portaudio19-dev

Desinstalar jackd1

Instalar Jackd2

Instalar libjack-jackd2-dev

Instalar portaudio19-dev

Si estamos empezando de cero

Usando Synaptic

Instalar Jackd2

Instalar libjack-jackd2-dev

Instalar portaudio19-dev

```
sudo apt-get install jackd2 libjack-jackd2-dev portaudio19-dev
```

```
sudo apt-get install make gcc g++ autoconf automake autotools-  
dev libtool git subversion
```

```
sudo apt-get install libusb-0.1-4 libusb-dev libfftw3-dev libpulse-  
dev libsamplerate0-dev
```

```
sudo apt-get install libusb-1.0-0-dev libconfig8-dev xdg-utils
libortp-dev
sudo apt-get install libevent-dev
```

Si la máquina no tiene una tarjeta de vídeo rápida OpenGL, se debe instalar también:

```
sudo apt-get install mesa-utils
sudo apt-get install mesa-common-dev
sudo apt-get install libgl1-mesa-dev
sudo apt-get install x11proto-xext-dev
sudo apt-get install libxrender-dev
```

Instalando códec2 codificador de baja velocidad y decodificador de audio:

Ver http://www.rowetel.com/blog/?page_id=452

Seguidamente obtenemos códec2 usando subversión:

```
cd
svn co https://freetel.svn.sourceforge.net/svnroot/freetel/códec2-dev
cd códec2-dev
autoreconf -i
./configure
make
sudo make install
sudo ldconfig
```

Instalando Qt5:

<http://qt-project.org/downloads>

Se debe descargar el fichero de librería Qt5.0.0 64 bit para x86_64 y 32 bit para PC x86.

Después de descargar ir al directorio descargado y hacer:

```
$ chmod +x the-name-of-the-downloaded-file
```

```
$ sudo ./the-name-of-the-downloaded-file
```

Después de esto deberemos tener instalado algo como /opt/Qt5.0.0

Colocando el path para qt5:

Desde terminal:

```
$export
```

```
PATH=/opt/Qt5.0.2/5.0.2/gcc/bin:/opt/Qt5.0.2/Tools/QtCreator/bin:$PATH
```

Para testearlo:

```
qmake -v
```

```
QMake versión 3.0
```

```
Using Qt versión 5.0.2 in /opt/Qt5.0.2/5.0.2/gcc/lib
```

Instalar libssl-dev usando Synaptic.

Construyendo las extensiones Perseus (usar máster branch):

Instalar las librerías de prerequisites y paquete wget::

```
sudo apt-get install libboost-all-dev wget
```

Descargar, compilar e instalar la librería libperseus-sdr:

```
cd
```

```
wget http://libperseus-sdr.googlecode.com/files/libperseus_sdr-0.6.tar.gz
```

```
tar -zxvf libperseus_sdr-0.6.tar.gz
```

```
cd libperseus_sdr-0.6/
```

```
./configure && make
```

```
sudo make install
```

```
sudo ldconfig
```

Instalar LibQt4-opengl-dev:

Synaptic

```
sudo apt-get install libqt4-opengl-dev
```

```
sudo apt-get install mesa-utils  
sudo apt-get install libxrender-dev
```

Compilar QtRadio:

```
cd  
git clone https://github.com/alexlee188/ghpsdr3-alex.git  
cd ghpsdr3-alex  
git checkout master
```

```
sh ./cleanup.sh  
autoreconf -i
```

```
sh ./alex-conf.sh
```

Finalmente se construye cada parte:

```
make  
sudo make install
```

En versión SSL de QtRadio, hacer:

```
openssl genrsa -out pkey 2048
```

```
openssl req -new -key pkey -out cert.req
```

```
openssl x509 -req -days 365 -in cert.req -signkey pkey -out cert
```

Apéndice 7. Detalle de instalación del Servidor Teamspeak 3.

A continuación se describe el procedimiento paso a paso de instalación de Teamspeak:

- 1.) Login a la consola del servidor linux
- 2.) Crear usuario “ts3srv” para correr TeamSpeakServer 3 .

Comando: adduser ts3srv

```
debian:~# adduser ts3srv
```

```
Adding user 'ts3srv' ...
```

```
Adding new group 'ts3srv' (1001) ...
```

```
Adding new user 'ts3srv' (1001) with group 'ts3srv' ...
```

```
Creating home directory '/home/ts3srv' ...
```

```
Copying files from '/etc/skel' ...
```

```
Enter new UNIX password:
```

```
Retype new UNIX password:
```

```
passwd: password updated successfully
```

- 3.) Cambiar al nuevo usuario “ts3srv” y abrir el directorio del usuario “/home/ts3srv”.

Comando 1: su ts3srv

Comando 2: cd /home/ts3srv

Salida Consola:

Code:

```
debian:~# su ts3srv
```

```
ts3srv@debian:/root$ cd /home/ts3srv
```

- 4.) Descargar binario de TeamSpeakServer 3 .

Comando: wget http://ftp.4players.de/pub/hosted/ts...0-beta6.tar.gz

Salida Consola:

Code:

```
ts3srv@debian:~$
```

```
wget http://ftp.4players.de/pub/hosted/ts3/releases/beta-6/teamspeak3-server_linux-x86-3.0.0-beta6.tar.gz
```

```
06:11:21--
```

```
http://ftp.4players.de/pub/hosted/ts3/releases/beta-6/teamspeak3-server_linux-x86-3.0.0-beta6.tar.gz  
'teamspeak3-server_linux-x86-3.0.0-beta6.tar.gz'
```

```
Resolving ftp.4players.de... 194.97.2.74, 194.97.2.75,  
194.97.2.76, ...
```

```
Connecting to ftp.4players.de|194.97.2.74|:80... connected.
```

```
HTTP request sent, awaiting response... 200 OK
```

```
Length: 2,655,753 (2.5M) [application/x-tar]
```

```
100%[=====>]
```

```
2,655,753
```

```
223.36K/s
```

```
ETA 00:00
```

```
06:11:34 (208.44 KB/s) - 'teamspeak3-server_linux-x86-3.0.0-beta6.tar.gz' saved [2655753/2655753]
```

5.) Desempaquetar el fichero descargado de *.tar.gz a *.tar .

```
Comando gunzip teamspeak3-server_linux-x86-3.0.0-beta6.tar.gz
```

```
Salida Consola:
```

```
Code:
```

```
ts3srv@debian:~$ gunzip teamspeak3-server_linux-x86-3.0.0-beta6.tar.gz
```

6.) Descomprimir archivo tar

```
Comando: tar xvf teamspeak3-server_linux-x86-3.0.0-beta6.tar
```

```
Salida Consola:
```

```
Code:
```

```
ts3srv@debian:~$ tar xvf teamspeak3-server_linux-x86-3.0.0-beta6.tar
```

7.) Listar directorio "/home/ts3srv" .

Comando: ls

Salida Consola:

Code:

```
ts3srv@debian:~$ ls
teamspeak3-server_linux-x86
teamspeak3-server_linux-x86-3.0.0-beta6.tar
```

8.) Cambiar a sub-directorio “teamspeak3-server_linux-x86”.

Comando: cd teamspeak3-server_linux-x86

Salida Consola:

Code:

```
ts3srv@debian:~$ cd teamspeak3-server_linux-x86
```

9.) Listar directorio “/home/ts3srv/teamspeak3-server_linux-x86” .

Comando: ls

Salida Consola:

Code:

```
ts3srv@debian:~/teamspeak3-server_linux-x86$ ls
```

10.) Para crear un “ts3server.ini” comenzar TeamSpeakServer 3 con parámetro createinfile=1 y parar de nuevo después.

(Strg+C).

Comando: ./ts3server_linux_x86 createinfile=1

Salida Consola:

Code:

```
ts3srv@debian:~/teamspeak3-server_linux-x86$
./ts3server_linux_x86 createinfile=1
```

Logging started

2009-12-22 11:02:34.533515|INFO

|ServerLibPriv |

| Server Version: 3.0.0-beta3 [Build: 9419]

2009-12-22 11:02:34.537384|INFO

|DatabaseQuery |

| dbPlugin name:

SQLite3 plugin, ©TeamSpeak Systems GmbH

2009-12-22 11:02:34.540304|INFO

|DatabaseQuery |

| dbPlugin version: 3.6.4

2009-12-22 11:02:34.631452|INFO

|SQL

|

| db_CreateTables() tables created

I M P O R T A N T

Server Query Admin Account created

loginname= "serveradmin", password= "w+XdDkES"

11:02:35.459696|WARNING

11:02:35.538614|INFO

11:02:36.259302|INFO

11:02:36.264417|INFO

11:02:36.267560|WARNING

11:02:36.270712|WARNING

11:02:36.274492|WARNING

11:02:36.277475|WARNING

11:02:36.280851|INFO

11:02:36.310000|INFO

|Accounting

|FileManager

|VirtualServer

|VirtualServer

|VirtualServer

|VirtualServer

|VirtualServer

|VirtualServer

|VirtualSvrMgr

|Query

Unable to find valid license key, falling back to limited
functionality

listening on 0.0.0.0:30033

listening on 0.0.0.0:9987

client 'server'(id:0) added token for servergroup 'Server
Admin'(id:6)

ServerAdmin token created, please use the line below
token=QrNPmLxuv41iB92L9TU2iFUQc7kkREMHC3m
McSU5

dbLoadVirtualServers() VirtualServer(1) started (Default)

listening on 0.0.0.0:10011

11.) Anotar loginname y password del Server Query Admin Account para

referencias futuras. Ésta sólo se mostrará la primera vez que comience TeamSpeakServer 3 .

Code:

```
-----  
I M P O R T A N T  
-----
```

```
Server Query Admin Account created
```

```
loginname= "serveradmin", password= "w+XdDkES"  
-----
```

12.) Anotar Serveradmin token – también se encontrará el token en el primer fichero log creado "logfile in /home/ts3srv"

/teamspeak3-server_linux-x86/logs". Ese token se necesita para obtener acceso de administrador con TeamSpeakClient 3 .

Code:

```
11:02:36.267560|WARNING
```

```
11:02:36.270712|WARNING
```

```
11:02:36.274492|WARNING
```

```
11:02:36.277475|WARNING
```

```
|VirtualServer
```

```
|VirtualServer
```

```
|VirtualServer
```

```
|VirtualServer  
-----
```

```
ServerAdmin token created, please use the line below
```

```
token=QrNPmLxuv41iB92L9TU2iFUQc7kkREMHC3mMcSU5  
-----
```

13.) Modificar “ts3server.ini” en “/home/ts3srv/teamspeak3-server_linux-x86” si se necesita. Normalmente no se necesita un cambio.

Excepto si se quiere especificar una cierta ip o puerto o si quieres usar una base de datos mysql .

Code:

```
machine_id=  
default_voice_port=9987  
voice_ip=0.0.0.0  
licensepath=  
filetransfer_port=30033  
filetransfer_ip=0.0.0.0  
query_port=10011  
query_ip=0.0.0.0  
dbplugin=ts3db_sqlite3  
dbpluginparameter=  
dbsqlpath=sql/  
dbsqlcreatepath=create_sqlite/  
logpath=logs  
logquerycommands=0
```

14.) Finalmente podemos lanzar nuestro creado TeamSpeakServer 3 en una ventana, de forma que siga corriendo en background.

Comando: `sudo -u ts3srv screen -dmS TSS3 ./ts3server_linux_x86`

Salida Consola:

Code:

```
ts3srv@debian:~/teamspeak3-server_linux-x86$ sudo -u ts3srv  
screen -dmS TSS3 ./ts3server_linux_x86
```

ts3srv@debian:~/teamspeak3-server_linux-x86\$

15.) Chequear si TeamSpeakServer 3 está corriendo .

Comando: ps aux

Salida Consola:

Code:

ts3srv@debian:~/teamspeak3-server_linux-x86\$ ps aux

USER

PID %CPU %MEM

VSZ

RSS TTY

STAT START

root

1 0.0 0.1

1944

636 ?

Ss

05:21

.

.

.

ts3srv

2957 0.0 0.1

3956

928 ?

Ss

07:14

ts3srv

2958 0.1 0.8 34856 4192 pts/1

Ssl+ 07:14

TIME COMMAND

0:01 init [2]

0:00 SCREEN -dmS TSS3 ./ts3server_linux_x86

0:00 ./ts3server_linux_x86

16.) Ahora nos podemos conectar a TeamSpeakServer 3 usando TeamSpeakClient 3 .

Conexión con el servidor:

- 1.) Comenzar TeamSpeakClient 3 .
- 2.) Abrir Bookmarks .
- 3.) Picar en Add Bookmark .
- 4.) Introducir datos de conexión (En caso de conectarse localmente al servidor 127.0.0.1, en otro caso identificar la dirección a la que nos conectemos).
- 5.) Picar OK.
- 6.) Conectar a TeamSpeakServer 3.
- 7.) Successfully connected.
- 8.) Picar en Permissions->Use Token .
- 9.) Introducir Token anotado en paso 12 de la instalación de TeamSpeakServer 3 y picar OK.
- 10.) Ahora somos serveradmin . Ver icono al lado del nick.
- 11.) Picar en Tools->ServerQuery.
- 12.) Ejecutar el siguiente comando para ganar acceso completo a todos los setting. Usar logindetails anotados en paso 11 de la instalación de TeamSpeakServer 3 y presionar la flecha verde para correr la query.

Code:

```
login serveradmin w+XdDkES
```

```
use sid=1
```

```
servergroupaddclient sgid=2 cldbld=2
```

13.) Resultado correcto de la consulta.

Una vez realizadas estas tareas ya está Teamspeak instalado y se ha accedido como administrador para empezar a definir la aplicación.

Apéndice 8. Descripción detallada de comandos de conexión SDR_Radio.

Descripción comando Login :

Todos los comandos se hacen usando XML.

1. El servidor envía un comando “StartLogin” al cliente después de aceptar la petición de conexión TCP/IP .

```
<SDR-RADIO-COMMS xml:lang="EN" Command="StartLogin">
</SDR-RADIO-COMMS>
```

2. El cliente envía un comando “ConnectRequest“.

```
<SDR-RADIO-COMMS xml:lang="EN" Command="ConnectRequest">
</SDR-RADIO-COMMS>
```

3. El server envía un comando “ConnectUseKey“ que contiene la clave usada por el cliente para encriptar el password.

```
<SDR-RADIO-COMMS xml:lang="EN" Command="ConnectUseKey"
Key="88B7-9AA6-20BB-EC9E-6929-3D26-F4FF-51F2">
</SDR-RADIO-COMMS>
```

Atributo	Descripción
Key	Clave usada para encriptar el password .

4. El cliente envía un comando “ConnectLogin“ que contiene el username, password y key. El username y password deben coincidir con los definidos en el servidor.

```
<SDR-RADIO-COMMS xml:lang="EN" Command="ConnectLogin"
Key="88B7-9AA6-20BB-EC9E-6929-3D26-F4FF-51F2" Username="Guest"
Password="E087-8926-EEA1-F608-F254-8BB0-E5F8-2D62"
WindowsUser="Simon" ComputerName="BLACK-BEAUTY">
</SDR-RADIO-COMMS>
```

Atributo	Descripción
Key	Clave usada para encriptar el password .
Username	Login Username, definidas en el apartado de cuentas

	del servidor.
Password	Valor del password encriptado.
WindowsUser	El nombre de la ventana de cuenta usada en el ordenador cliente.
ComputerName	El nombre del ordenador cliente.

5. El servidor manda el comando "ConnectLoginStatus" con el estado, normalmente OK , normalmente a menos que se haya enviado un usuario o password incorrectos.

```
<SDR-RADIO-COMMS xml:lang="EN" Command="ConnectLoginStatus"
Status="OK" Login="1">
</SDR-RADIO-COMMS>
```

Petición de dispositivo :

El cliente ahora pide la relación de dispositivos libres.

1. El cliente manda el comando "GetInputDevices".

```
<SDR-RADIO-COMMS xml:lang="EN" Command="GetInputDevices">
</SDR-RADIO-COMMS>
```

2. El servidor replica con el comando "ReturnInputDevices" y una lista de Radios SDR y tarjetas de sonido. Las tarjetas de sonido se usan para los receptores Softrock y Funcube.

```
<SDR-RADIO-COMMS xml:lang="EN" Command="ReturnInputDevices"
RFSpaceNames="SDR-IQ #IV001292" RFSpaceDevices="SDR-IQ"
SoundcardNames="Microphone (Logitech Wireless Headset)"
SoundcardDevices="0">
</SDR-RADIO-COMMS>
```

Atributos	Descripción
RFSpaceName	Las radios SDR de RFSpace, cada entrada contiene el tipo y número de serie.
RFSpaceDevices	Las radios SDR de RFSpace, cada entrada

SoundCardNames	contiene el tipo de dispositivo. El nombre de la tarjeta de sonido que soporta la grabación desde una fuente de entrada (usada por radios tipo SoftRock).
SoundcardDevices	Ids del dispositivo tarjeta de sonido que corresponde a los nombres.

Comenzar Radio :

El cliente selecciona “comenzar SDR radio“.

1. El comando Start contiene el aparato radio a arrancar y algunos parámetros que se deban usar.

```
<SDR-RADIO-COMMS xml:lang="EN" Command="Start">
  <Open InputName="SDR-IQ" DeviceID="0" DeviceType="2" BufferSize="0"
  SampleRate="158730" Frequency="7075000" FilterWidth="5" Location="0"
  Loop="0" Invert="0" IFGain="2" RFGain="-10">
  </Open>
  <Params>
  <VFO Index="0" Enable="1" Offset="-5304" Mode="1" FilterLow="200"
  FilterHigh="2800">
  </VFO>
  <VFO Index="1" Enable="0" Offset="0" Mode="1" FilterLow="200"
  FilterHigh="2800">
  </VFO>
  <VFO Index="2" Enable="0" Offset="10000" Mode="1" FilterLow="200"
  FilterHigh="2800">
  </VFO>
  <Active AGCAAttack="10" AGCDecay="1000" AGCHang="0" AFGain="0"
  ActiveVFO="0" IFGain="2" NB="0" NB1Threshold="0" NB2Threshold="0"
  NR="0" NR1AdaptiveSize="0" NR2AdaptiveSize="0" RFGain="1"
  RadioFrequency="7075000" SampleRate="158730" SampleSize="0"
  WaterfallSpeed="1">
  </Active>
```

```

<Output ID="0" Name="Speakers (High Definition Audio Device)"> </Output>
</Params>
</SDR-RADIO-COMMS>

```

Nodo	Descripción
Open	El nombre del dispositivo a abrir y cualquier parámetro asociado.
Params	Parámetros para la demodulación

El nodo Open contiene el nombre del dispositivo a abrir y cualquier parámetro asociado. En el ejemplo se abre una radio SDR.

Parámetro	Descripción
InputName	Un string XML formateado conteniendo el nombre del dispositivo a abrir y cualquier parámetro asociado.
DeviceID	Reservado , poner a 0
DeviceType	0 = Soundcard, 1 = Wave file, 2 = RFSpace SDR radio
BufferSize	Reservado , poner a 0
SampleRate	El sample rate a usar, t.b.d.
Frequency	La frecuencia en Hercios
FilterWidth	Reservado , poner a 0
Location	Reservado , poner a 0
Loop	Reservado , poner a 0
Invert	Reservado , poner a 0
IFGain	IF ganancia, t.b.d.
RFGain	RF ganancia, t.b.d.

El Parámetro nodo contiene la información necesaria para la demodulación.

Nodo hijo	Parámetro	Descripción
VFO	Índice	Índice zero-based VFO, normalmente se soporta un máximo de tres VFOs.
Habilitado		1 Habilitado, otro 0
Offset		Offset en Hercios desde la radio frecuencia.
Modo		El modo normal para el demodulador: LSB = 0, USB = 1, DSB = 2, WUSB = 3, CW-L = 4, CW-U = 5, AM = 6, FM = 7, WFM = 8, NFM = 9
Filtro-bajo (FilterLow)		Frecuencia baja del filtro en Hercios .
Filtro-alto (FilterHigh)		Frecuencia alta del filtro en Hercios .

2. El servidor devuelve el estado en el comando "ServerStatus".

```
<SDR-RADIO-COMMS xml:lang="EN" Command="ServerStatus" Rate="0.0
KB/s" Open="1" Name="SDR-IQ">
</SDR-RADIO-COMMS>
```

El estado del servidor la envía el servidor en intervalos regulares para actualizar al cliente con el ratio de envío y el estado de la conexión radio.

Formato de los mensajes TCP/IP:

Todos los valores numéricos son 16-bit UNICODE.

El formato general de cada mensaje enviado entre cliente y servidor es :

Campo	Bytes	Descripción
Longitud	4	Longitud total del mensaje.

Tipo	4	Tipo de mensaje, uno de: XML (1), Audio (2), FFT (3)
Dato	Variable	El dato enviado

XML

XML se usa para todos los comandos y reportes de estado enviados entre el cliente y el servidor .

Campo	Bytes	Descripción
Longitud	4	Longitud total del mensaje .
Tipo	4	Mensaje tipo, uno de: XML (1), Audio (2), FFT (3)
Dato	Variable	El dato enviado, ésto incluye NULL XML que se envía como string 16-bit UNICODE.tring

Audio

El audio se envía como dato 8 KHz monoaural con 16-bits por muestra. En este momento, los códecs no se usan para ahorrar ancho de banda. En el futuro se usarán FLAC y Ogg Vorbis .

Campo	Bytes	Descripción
Longitud	4	Longitud total del mensaje.
Tipo	4	Low word = Audio (2), High word = fags, opcional .
Flag	Valor	Descripción
FLAC	x02	FLAC códec en uso
Vorbis	x04	Ogg Vorbis códec en uso
8kHz	x20	Sample rate 8kHz, 16 bits por muestra .
48kHz	X40	Sample rate 48kHz, 16 bits por muestra .
Dato	Variable	Dato enviado

Dato FTT :

El dato FTT se comprime usando algoritmo ZIP para salvar ancho de banda.

Campo	Bytes	Descripción
Longitud	4	Longitud total mensaje
Tipo	4	Low word = FFT (3), High word = flags opcional flags
Flag	Valor	Descripción
ZIP	x01	FLAC códec en uso
Dato	Variable	El dato enviado

Apéndice 9. Instalación de GNU Radio Marconi.

A continuación, se muestra el resultado de la compilación del servidor GNU Radio Marconi.

```
jjp@jjp-VirtualBox:~$ ./gnuradio_build.sh
```

This script will install Gnu Radio from current GIT sources

You will require Internet access from the computer on which this script runs. You will also require SUDO access. You will require approximately 500MB of free disk space to perform the build.

This script will, as a side-effect, remove any existing Gnu Radio installation that was installed from your Linux distribution packages.

It must do this to prevent problems due to interference between a linux-distribution-installed Gnu Radio/UHD and one installed from GIT source.

The whole process may take up to two hours to complete, depending on the capabilities of your system.

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

NOTE: if you run into problems while running this script, you can re-run it with the --verbose option to produce lots of diagnostic output to help debug problems.

This script has been written to anticipate some of the more common problems one might

encounter building ANY large, complex software package. But it is not perfect, and there are certainly some situations it could encounter that it cannot deal with gracefully. Altering the system configuration from something reasonably standard, removing parts of the filesystem, moving system libraries around arbitrarily, etc, it likely cannot cope with. It is just a script. It isn't intuitive or artificially

intelligent. It tries to make life a little easier for you, but at the end of the day if it runs into trouble, a certain amount of knowledge on your part about system configuration and idiosyncrasies will inevitably be necessary.

Proceed?y

Starting all functions at: mar ene 21 18:45:03 CET 2014

SUDO privileges are required

Do you have SUDO privileges?y

Continuing with script

[sudo] password for jjp:

Installing prerequisites.

====> THIS MAY TAKE QUITE SOME TIME <=====

Checking for package libfontconfig1-dev

Checking for package libxrender-dev

Checking for package libpulse-dev

Checking for package swig

Checking for package g++

Checking for package automake

Checking for package autoconf

Checking for package libtool

Checking for package python-dev

Checking for package libfftw3-dev

Checking for package libcppunit-dev

Checking for package libboost1.48-all-dev

Checking for package libusb-dev

Checking for package libusb-1.0-0-dev

Checking for package fort77

Checking for package libssl1.2-dev

Checking for package python-wxgtk2.8

Checking for package git-core

Checking for package libqt4-dev

Checking for package python-numpy

Checking for package ccache

Checking for package python-opengl

Checking for package libgsl0-dev

Checking for package python-cheetah

Checking for package python-lxml

Checking for package doxygen

Checking for package qt4-dev-tools

Checking for package libusb-1.0-0-dev

Checking for package libqwt5-qt4-dev

Checking for package libqwtplot3d-qt4-dev

Checking for package pyqt4-dev-tools

Checking for package python-qwt5-qt4

Checking for package cmake

Checking for package git-core

Checking for package wget

Checking for package libxi-dev

Checking for package python-docutils

Checking for package gtk2-engines-pixbuf

Checking for package r-base-dev

Checking for package python-tk

Checking for package liborc-0.4-0

Checking for package liborc-0.4-dev

Checking for package libasound2-dev

Checking for package python-gtk2

Done checking packages

Checking for library libusb ...Found library libusb

Checking for library libboost ...Found library libboost

Checking for library libcppunit ...Found library libcppunit

Checking for library libfftw ...Found library libfftw

Checking for library libgsl ...Found library libgsl

Done

This script will fetch Gnu Radio version 3.7/maint from the repositories, along with compatible

extras.

Is this OK?y

Fetching various packages (Gnu Radio, UHD, gr-osmosdr, gr-iqbal, etc)

via the Internet

=====> THIS MAY TAKE QUITE SOME TIME <=====

Fetching Gnu Radio via GIT...Done

Fetching UHD via GIT...Fetching rtl-sdr (rtl-sdr, gr-osmosdr, gr-iqbal, hackrf and bladeRF) via GIT

Done

Starting function uhd_build at: mar ene 21 20:52:03 CET 2014

Building UHD...

=====> THIS WILL TAKE SOME TIME <=====

[sudo] password for jjp:

Done building/installing UHD

Done function uhd_build at: mar ene 21 22:48:32 CET 2014

Starting function firmware at: mar ene 21 22:48:33 CET 2014

Images destination: /usr/local/share/uhd/images

Downloading images from: http://files.ettus.com/binaries/maint_images/archive/uhd-images_003.006.002-release.zip

Downloading images to: /tmp/tmpm0RPMp/uhd-images_003.006.002-release.zip

13335 kB / 13335 kB (100%)

Images successfully installed to: /usr/local/share/uhd/images

Done downloading firmware to /usr/local/share/uhd/images

Done function firmware at: mar ene 21 22:49:41 CET 2014

Starting function gnuradio_build at: mar ene 21 22:49:41 CET 2014

/usr/local/lib already in ld.so.conf.d

Doing ldconfig...

Building Gnu Radio...

=====> THIS WILL TAKE QUITE A WHILE <=====

...Doing cmake

...Cmaking

...Building

...Installing

[sudo] password for jjp:

Done building and installing Gnu Radio

GRC freedesktop icons install ...Done

Done function gnuradio_build at: mié ene 22 05:25:15 CET 2014

Starting function rtl_build at: mié ene 22 05:25:15 CET 2014

Building rtl-sdr...Done building rtl-sdr

Building hackrf...Done building hackrf

Building gr-iqbal...Done building gr-iqbal

Building bladeRF...Done building bladeRF

Building gr-osmosdr...Done building gr-osmosdr

Done building/installing rtl-sdr/gr-osmosdr

Done function rtl_build at: mié ene 22 05:40:03 CET 2014

Starting function extras at: mié ene 22 05:40:03 CET 2014

Doing GIT checkout for extra module gr-baz

Building extra module gr-baz

Doing GIT checkout for extra module grextras

Building extra module grextras

Done function extras at: mié ene 22 05:41:02 CET 2014

Starting function mod_groups at: mié ene 22 05:41:02 CET 2014

This script has just modified /etc/group to place your userid ('\$USER') into group 'usrp'

In order for this change to take effect, you will need to log-out and log back

in again. You will not be able to access your USRP1 device until you do this.

If you wish to allow others on your system to use the USRP1 device, you will need to use:

```
sudo usermod -a -G usrp userid
```

For each userid you wish to allow access to the usrp

```
*****
```

Further

```
Done function mod_groups at: mié ene 22 05:41:05 CET 2014
```

```
Starting function mod_udev at: mié ene 22 05:41:05 CET 2014
```

```
Done function mod_udev at: mié ene 22 05:41:09 CET 2014
```

```
Starting function mod_sysctl at: mié ene 22 05:41:09 CET 2014
```

```
Applying updates to /etc/sysctl.conf
```

```
Group 'usrp' now has real-time scheduling privileges
```

```
You will need to log-out and back in again for this to  
take effect
```

```
Done function mod_sysctl at: mié ene 22 05:41:12 CET 2014
```

```
Starting function pythonpath at: mié ene 22 05:41:12 CET 2014
```

```
*****
```

You should probably set your PYTHONPATH to:

```
/usr/local/lib/python2.7/dist-packages
```

Using:

```
export PYTHONPATH=/usr/local/lib/python2.7/dist-packages
```

in your .bashrc or equivalent file prior to attempting to run

any Gnu Radio applications or Gnu Radio Companion.

Done function pythonpath at: mié ene 22 05:41:13 CET 2014

Done all functions at: mié ene 22 05:41:13 CET 2014

All Done

Send success/fail info to sbrac.org?n

jjp@jjp-VirtualBox:~\$