Manual de uso e instalación de OpenBTS

(La liberalización de la telefonía celular)

Autor: Dario Flores, CLO S.A.
dflores@clo.cl
dario010675@gmail.com

Modificaciones realizadas:	Realizado por:	Fecha:	Versión
Generación de primera de documentación de la solución usrp1+OpenBTS+Asterisk.	Dario Flores	Agosto 12, 2011	0.1
Se agregó el punto 15 sobre envío/recepción de SMS con el servidor smqueue.	Dario Flores	Septiembre 15, 2011	0.2

1 Introducción

Este documento pretende dar una rápida vista sobre la solución OpenBTS y compararla con una solución de red GSM tradicional así como sus pasos de instalación, configuración y puesta en marcha exitosa con terminales GSM de bajo nivel tecnológico junto a terminales smartphones de última generación de diferentes marcas y operadores.

El objetivo principal es implementar una celda(s) de telefonía GSM 2G y presentar una interface de aire a terminales de tipo GSM sin importar su nivel tecnológico o antigüedad, el cual su vez usa el aplicativo de central telefónica Asterisk PBX para conectar las llamadas entre los usuarios de la red y el mundo exterior.

Una celda OpenBTS de telefonía GSM 2G puede funcionar en las bandas de frecuencias de 850, 900, 1800 o 1900 MHz, así como dar servicios de mensajería corta SMS entre terminales (basado en SIP).

Este documento no cubre las capacidades de OpenBTS relacionadas al hacking GSM como por ejemplo:

- Seguimiento de terminales activos entre celdas
- IMSI Catchers
- Almacenamiento de datos TMSIS de celdas para dinámica de usuarios en celdas. (Comportamiento y seguimiento)
- Spoofing de antenas BTS (suplantación)
- Generación de Denial of Services (DOS)
- Grabación de llamadas
- Otros

2 Red GSM convencional

Una red GSM es un sistema complejo compuesto por varios componentes. El último tramo de este sistema es la antena BTS (Base Statiob Transceiver). La BTS es la responsable de transmitir y recibir las señales de radio frecuencia (RF) al terminal del usuario (teléfono celular, PDA, módem, etc.) Las BTS son controladas por una BSC (Base Station Controller) que está conectado a un MSC y VLR (Mobile Switching Center y Visitor Location Register). Básicamente, el MSC/VLR son responsables de autenticar al usuario contra la base de datos HLR -Home Location Register- y el AUC -Centro de Autenticación de usuarios- de la red del operador.

A continuación se muestra una imagen de los elementos clave de una red GSM:

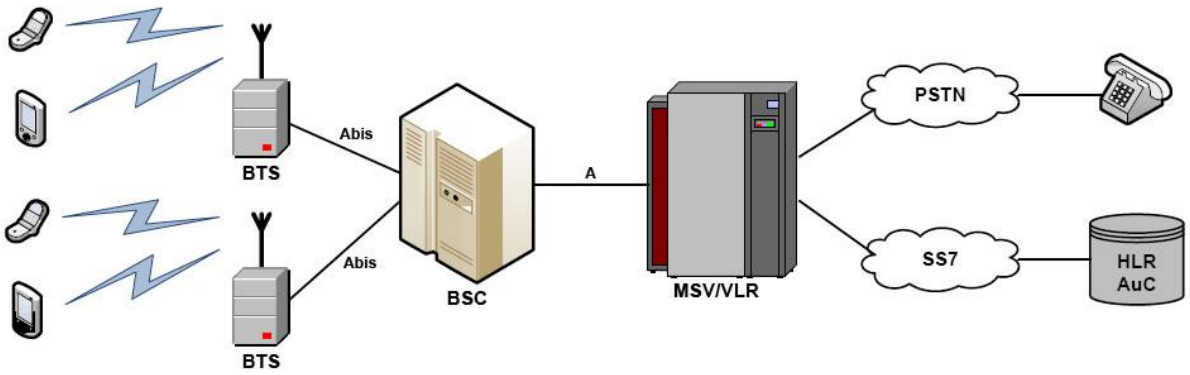


Diagrama de una red GSM convencional

3 El proyecto OpenBTS

El proyecto OpenBTS es un esfuerzo para construir una aplicación de código abierto y comercial licenciada de Unix que utiliza el software Universal Software Radio Peripheral (USRP) para presentar una interfaz GSM de aire a la norma de teléfonos GSM utilizando el software de central telefónica Asterisk PBX para conectar las llamadas. OpenBTS utiliza el hardware llamado usrp para recibir y transmitir la señal GSM, esto se hace utilizando el framework de GNU Radio. Asterisk se utiliza para conectar las llamadas entre los teléfonos GSM celulares en la red OpenBTS. Cualquier otro dispositivo que pueda conectarse a Asterisk puede ser también utilizado.

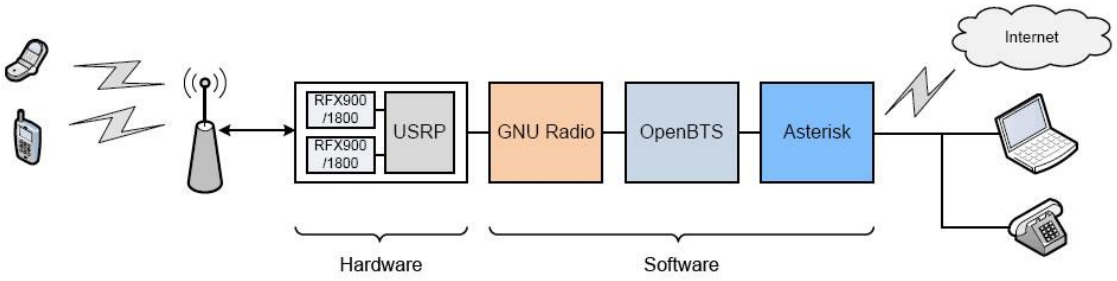


Diagrama de una red OpenBTS

Nota: USRP y usrp no tienen relación entre sí. USRP se refiere al framework de software de radio frecuencia y usrp, usrp1, usrp2, usrp E100, usrp 200 y usrp 210 se refiere al hardware para radio frecuencia.

4 Beneficios de una red GSM basada en OpenBTS

OpenBTS está diseñado para proveer sistemas de comunicación GSM tanto indoor como outdoor bajo configuración de red privada. Cualquier empresa privada, gobierno, fuerzas armadas o bien operadores de telefonía pueden utilizar OpenBTS como a una alternativa a soluciones de grandes marcas, y de esta manera implementar redes para comunicaciones corporativas privadas, locales o de emergencias en modo de espera o “standby”, entre sus ventajas técnicas destacan:

- Rápida capacidad de poner en marcha una red con una celda.
- Capacidad de múltiples celdas en múltiples zonas geográficas compartiendo un backhaul Vo-IP único. (Core Vo-IP)
- Celdas con bajo consumo de energía. (desde 100 Watt)
- Infraestructura basada en HW Open Source y/o de reconocidas marcas con disponibilidad inmediata con proveedores locales o extranjeros vía plataformas de e-commerce.
- SW puede es en su mayoría de tipo Open Source con un bajo costo de TOC.
- Permite un rápido ROI en el mediano plazo a las empresas de Telco.
- Ideal para extender servicios de voz corporativos (anexos) fuera de la oficina.
- Permite a una empresa de privada o telco crear su red 2G en un modo orgánico con niveles de inversiones y operaciones bajos.
- Una nueva opción para empresas de telecomunicaciones pequeñas o medianas que desean competir en el mercado de la telefonía celular a bajo costo.

5 Actuales versiones de OpenBTS

A la fecha de redacción de este documento, la última versión oficial no-comercial de los autores originales *David Burgess* y *Harvind Samra* de OpenBTS es la 2.6 conocida como Mamou. Esta versión se puede encontrar bajo la URL: http://sourceforge.net/projects/openbts/, así como sus versiones más antiguas.

Desde Mayo 2011, existe una nueva versión independiente, paralela y que toma como punto de partida la última versión oficial 2.6 (Mamou). Esta versión paralela posee soporte UHD (Universal Hardware Device) y se denomina OpenBTS-UHD. OpenBTS-UHD incluye todas las funcionalidades de la versión pública principal con las siguientes características:

- Soporte de todos los productos usrp de Ettus Research

- Soporte para dispositivos usrp "embedded" E100 con procesador ARM Cortex-A8
- Soporte para tarjetas hijas no-RFX como las WBX, SBX, DBSRX, y DBSRX2
- Permitir referencias externas de señal (en dispositivos soportados) para alta precisión de sincronización.
- Transmitir y recibir control de ganancia en dispositivos no registrados en 52 Mhz.
- Parches adicionales no incluidos en repositorio central público de OpenBTS.
- Productos usrp E100, N200 y N210 no requieren modificaciones físicas del componente oscilador. (comparado con la unidad usrp1)
- Mas información en: http://gnuradio.org/redmine/projects/gnuradio/wiki/OpenBTSUHD

Para mayores informaciones sobre las funcionalidades de la versión comercial de OpenBTS, puede visitar la página de Kestrel Signal Processing: http://www.kestrelsp.com/OpenBTS.html

6 Terminologías GSM a conocer de antemano

Algunos los conceptos del mundo GSM que deberá dominar de antemano para entender mejor las diferentes posibilidades de configuración y uso de OpenBTS son las siguientes:

- **MCC y MNC:** Corresponden a las siglas en inglés de **Mobile Country Code** y **Mobile network code**, son dos códigos numéricos usados conjuntamente para identificar el país y los operadores de telefonía móvil que utilizan ya sea GSM, CDMA, UMTS y ciertas redes satelitales. Ambos códigos corresponden a los primeros 5 a 6 dígitos del total de 15 dígitos de un código **IMSI**.
- **IMSI:** Es el acrónimo de **I**nternational **M**obile **S**ubscriber **I**dentity (*Identidad Internacional del Abonado a un Móvil*). Es un código de identificación único para cada dispositivo de telefonía móvil, e integrado en la tarjeta SIM del terminal GSM, que permite su identificación a través de las redes GSM y UMTS.

  Ejemplo de código IMSI: IMSI730011835026703

  En donde 730 corresponde al MCC asignado a Chile y 01 a una de las compañias de telefonía móvil con operación dentro de Chile. A modo de ejemplo cada vez que algún usuario sale fuera de su país y solicita servicio de Roaming a su operador local, este internamente solicita al operador del país de destino permitir llamadas a un código IMSI de un chip de su red.

- **TMSI:** La "Identidad temporal del abonado móvil" (TMSI) es la identidad más enviadas entre el móvil y la antena BTS más cercana). El TMSI se asigna al azar por el VLR a todos los móviles en la zona, en el momento que se encienden y buscan registrarse en una antena BTS. El número es local a un área de ubicación, por lo que tiene que ser actualizado cada vez que el móvil se mueve a una nueva área geográfica con otra antena BTS.
- **ARFCN**: **A**bsolute **R**adio-**F**requency **C**hannel **N**umber (Radio Frecuencia absoluta de número de canal), especifica un par

canales de operadores de radio y físico utilizados para la transmisión y recepción en la **interface Um** de redes celular GSM, una para la señal de enlace ascendente (up-link) y otro para la señal de enlace descendente (down-link).

Cada ARFCN tiene un ancho de banda de 270,833 kHz y los ARFCN´s utilizan una separación entre canales de 200 kHz en cualquier banda de GSM dada. ARFCN`s se utilizan para el componente de frecuencia en función del esquema de acceso múltiple por GSM (**FDMA** Frecuencia de Acceso Múltiple por División). Junto con el componente basado en el tiempo (**TDMA** Acceso Múltiple por División de Tiempo) el canal físico se define por la selección de un determinado ARFCN y una cierta franja de tiempo. **Nota**: no confundir este canal físico con los canales lógicos.

7 Hardware usado

Hardware	
Tipo	**Especificaciones**
Computador	Laptop IBM Lenovo Intel P4 2.8 Ghz, 160 GB HD, 2 GB RAM, tarjeta de red y puerto USB.
USRP1	Tarjeta madre Rev 4.5- adquirda en www.ettus.com (Modificada localmente para soportar a un clock externo de 52 MHz.)
Tarjeta hijas	• 2 WBX 50 MHz a 2.2 GHz Transceiver. Poder de transmisión de señal: hasta 100 mW.
Antenas	VERT 900, 824-960 MHz, 1710-1990 MHz Quad-band Cellular/PCS con 3dBi de ganancia.
Reloj externo	• ClockTamer-1.2 adquirido en: http://shop.fairwaves.ru/clock-tamer/
Terminales GSM	1 Nokia 6120 (entel), 1 Sony Ericsson 310 (Movistar), 2 Samsung Galaxy II (Entel y Movistar), 1 Apple iPhone (Claro), 1 Huawei G7210 (entel).

Las unidades USRP1 provistas por Ettus LLC cuentan de un reloj interno oscilador de 64MHz de tipo SMD (Surface Mounted Device), el cual sirve para aplicaciones de radio frecuencia de todo tipo pero no para la sincronización de dispositivos GSM. La modificación a la tarjeta madre de la unidad usrp1 es mandatorio para el uso de un nuevo oscilador de frecuencia externo de tipo USB (Clocktamer), el cual puede ser adquirido por internet en el sitio de e-commerce: http://shop.fairwaves.ru/clock-tamer/.

7.1 Procedimiento para deshabilitación del oscilador interno

- Mover micro resistencia R2029 a R2030, esto desconecta el oscilador de 64 MHz de la unidad USRP. R2029 es una resistencia de 0-Ohm.
- Mover capacitor C925 a C926
- Remover C924

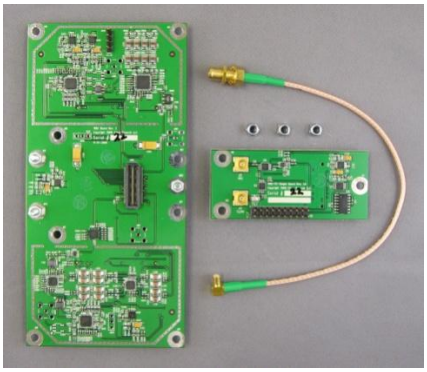
7.2 Recomendaciones al usar el reloj externo de 52 Mhz

Fabricar un mini PCB con dos resistencias de 100 Ohms en configuración paralela (resistencia equivalente de 50 Ohms) las cuales van soldadas en la parte inferior de la tarjeta madre del USRP en los bornes del conector SMA "Clock in" o "External Clock In". Un extremo de la resistencia en el borne central y el segundo extremo de la resistencia en cualquier de los bornes externos. Esto previene posible picos de tensiones que pudiese dañar la tarjeta madre.

Se recomienda que la modificación sea realizada por un técnico experto en electrónica de tecnología SMD. En el caso de la actual unidad USRP estas tareas fueron realizadas por Olimex Chile Ltda. (http://www.olimex.cl). Estas modificaciones son obligatorias si se adquiere una unidad USRP1.

Actualmente existe una nueva línea de productos USRP de nueva generación modelo N200 y N210 los cuales usan sincronización mediante software UDH (Universal Hardware Device) y no mediante driver del software Gnuradio, por lo cual el cambio de oscilador ya no es necesario.

Unidad usrp1 con tarjeta madre y dos tarjetas hijas WBX (50 Mhz a 2.1 GHz de frecuencia) y oscilador externo ClockTamer



Tarjetas hijas WBX

Vista frontal de la unidad usrp1 sin antenas.

Oscilador externo de 52 Mhz "Clocktamer".

8 Software usado

Software	
Ubuntu Desktop	10.10
Asterisk	1.4.42
Gnuradio (última versión de desarrollo desde git con soporte UHD)	3.3.5
Kal (calibrador de frecuencia)	0.41
libosip2	3.3.5
OpenBTS-UHD (Última versión con soporte UHD via git)	2.6

8.1 Observaciones del software usado

8.1.1 Versión de Gnuradio a usar

Para esta instalación en particular se ha usado la versión ubicada en el repositorio git de Gnuradio http://gnuradio.org/git/gnuradio.git basada en versión experimental 3.3.5. Sin embargo si se desea utilizar la versión oficial estable de Gnuradio 3.3.0, esta puede ser descargada desde: http://ftp.gnu.org/gnu/gnuradio/gnuradio-3.3.0.tar.gz .

8.1.2 Mejor versión de Asterisk a usar

Se recomienda usar sólo versiones de Asterisk basadas en 1.4.xx y 1.6.xx, las nuevas versiones basadas en 1.8.xx poseen un “bug” aún indeterminado a nivel de SIP el cual provoca que las llamadas se terminen abruptamente luego de 32 segundos debido a una supuesta

falta de trafico RTP entre los SIP "end points", en este caso los terminales GSM.

8.1.3 Mejor distribución de Linux a usar

Se recomienda usar distribuciones basadas en Linux Ubuntu 10.04 LTS Desktop o 10.10 Desktop debido a que estas distribuciones poseen el mejor soporte de dependencias para poder posteriormente configurar, compilar e instalar Gnuradio, OpenBTS y Asterisk. Distribuciones Linux basadas en Fedora, Mandrake, Suse o incluso Debian no son recomendables debido a la inexistencia de ciertos paquetes.

8.1.4 OpenBTS-UHD

Para esta instalación se ha usado una versión con soporte UHD en: http://gnuradio.org/redmine/projects/gnuradio/wiki/OpenBTSUHD la cual es una versión paralela a la versión oficial de los autores originales ubicada en Sourceforge.net en: http://sourceforge.net/projects/openbts/ . OpenBTS-UHD a diferencia de la versión oficial, provee soporte para tarjetas de radiofrecuencia WBX y otras no RFX.

9 Proceso de instalación

Se recomienda tener experiencia avanzada en el uso de Linux y Asterisk a nivel de consola junto con la capacidad de entender y resolver cualquier tipo de error de dependencia que pudiese aparecer durante el proceso de instalación. Se recomienda tener un pc, o laptop de buena velocidad basado en Intel Celeron o superiores, así como AMD Sempron o superiores con 1 GB de RAM, disco duro con un mínimo de 40 GB y conexión a internet permanente. (y mucha paciencia)

9.1 Pasos

I. Instalar Ubuntu 10.10 Desktop y abrir una ventana de terminal como root y crear directorio donde ser descargaran todos las dependencias y programas a instalar en forma manual:

- ```
  sudo –i (luego clave)
  cd /root
  mkdir software (ejemplo)
  cd software
  ```

II. Realizar "update" de repositorio de Ubuntu y luego un "upgrade" general de las dependencias del sistema:

- ```
  sudo apt-get update
  ```
- ```
  sudi apt-get upgrade
  ```

III. Instalar las dependencias para Gnuradio y OpenBTS:

- ```
  sudo apt-get -y install vim ssh libfontconfig1-dev
  libxrender-dev libpulse-dev swig g++ automake autoconf
  libtool python-dev libfftw3-dev libcppunit-dev
  libboost-all-dev libusb-dev fort77 sdcc sdcc-libraries
  libsdl1.2-dev python-wxgtk2.8 git-core guile-1.8-dev
  libqt4-dev python-numpy ccache python-opengl libgsl0-
  dev python-cheetah python-lxml doxygen qt4-dev-tools
  libqwt5-qt4-dev libqwtplot3d-qt4-dev pyqt4-dev-tools
  python-qwt5-qt4 libortp-dev latext git-core cmake
  libxml2-dev libortp-dev libusrp-dev libusrp0 gawk
  ```

IV. Descargar e instalar el paquete "libosip" para soporte de comunicación SIP entre OpenBTS y Asterisk.

- wget http://ftp.gnu.org/gnu/osip/libosip2-3.5.0.tar.gz
  gunzip libosip2-3.5.0.tar.gz
  tar -xvf libosip2-3.5.0.tar
  ./configure
  make
  make install

V. Descargar e instalar Asterisk 1.4.42:

- wget
  http://downloads.asterisk.org/pub/telephony/asterisk/releases/asterisk-1.4.42.tar.gz
  gunzip asterisk-1.4.42.tar.gz
  tar -xvf asterisk-1.4.42.tar
  ./configure
  make menuselect
  make
  make install
  make samples
  make config

VI. Descargar e instalar soporte UHD desde git, solo para unidades USRP serie N2xx o E100, no aplica para unidades usrp1 (saltar este paso):

- git clone git://code.ettus.com/ettus/uhd.git
  cd uhd/host
  mkdir build
  cd build
  cmake ../
  make
  make test
  sudo make install
  #Buscar donde esta la ruta de UHD con:
  find |grep libuhd
  #exportar a la nueva ruta (path):
  export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/lib

VII. Descargar e instalar Gnuradio 3.3.5 desde git y modificar código fuente del driver de velocidad del clock interno de Gnuradio:

- git clone http://gnuradio.org/git/gnuradio.git
  #Modificar código fuente del driver del clock interno de Gnuradio:

- ```
  cd  /gnuradio/usrp/host/lib
  vi usrp_basic.cc
  #Ir a la línea 110 y cambiar el valor 64000000 por
  52000000
  #Línea original:
  "d_verbose (false),d_fpga_master_clock_freq(64000000),
  d_db(2)"
  #Línea modificada:
  "d_verbose (false),d_fpga_master_clock_freq(52000000),
  d_db(2)"
  # Salvar cambios y salir del editor VIM.
  # exportar la nueva ruta (path):
  export
  PKG_CONFIG_PATH=/usr/local/lib/pkgconfig:${PKG_CONFIG_
  PATH}

  #Configurar, compilar e instalar:
  ./bootstrap
  #En el caso de unidades USRP serie N2xx o E100 usar:
  ./configure --enable-gr-uhd
  #En el caso de unidades usrp1 usar:
  ./configure --with-usrp1
  make
  make check
  sudo make install
  ```

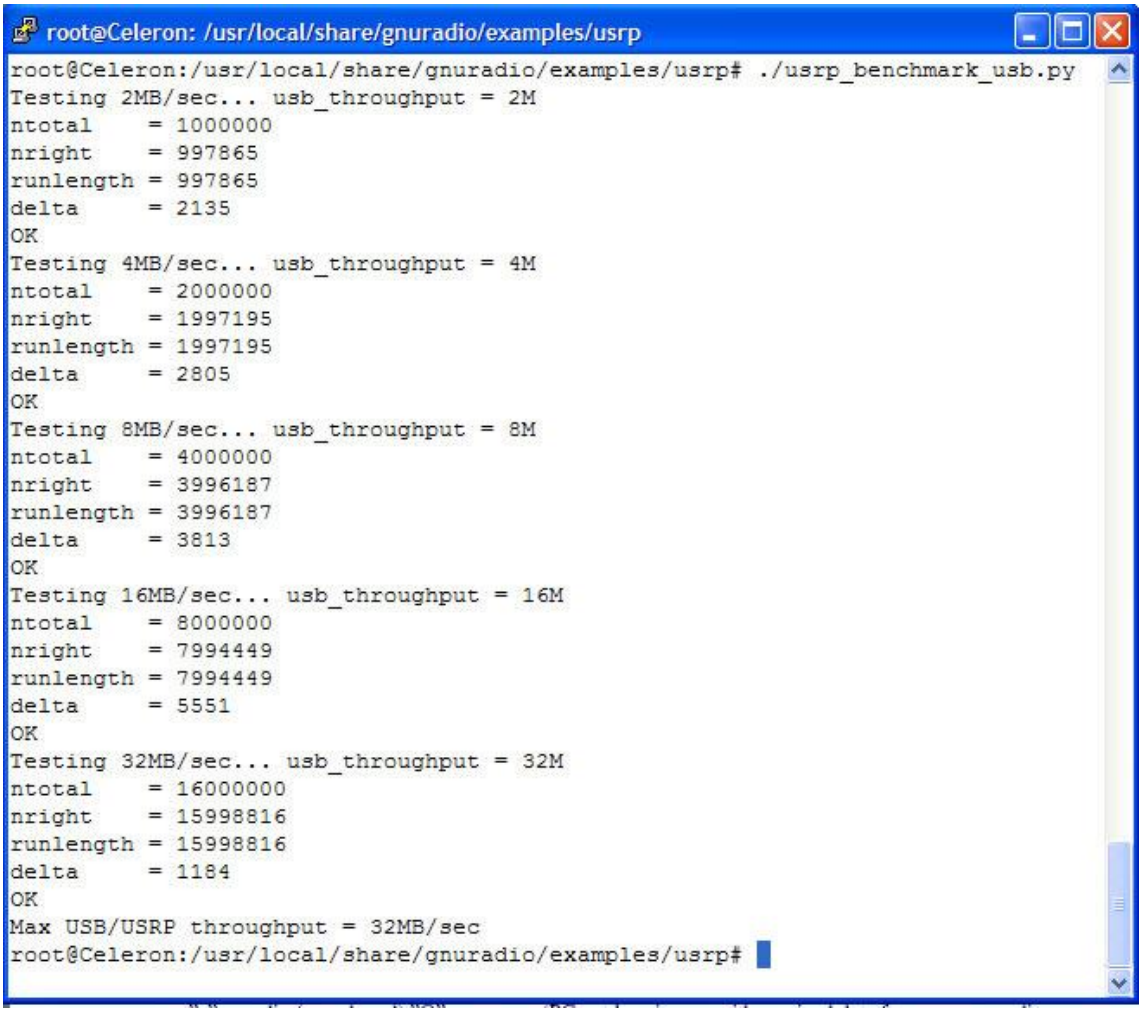
VIII. Descargar e instalar el calibrador de oscilador externo "Kalibrate" para la unidad usrp1:

- wget http://thre.at/kalibrate/kal-v0.4.1.tar.bz2
  # En caso de usar unidad USRP E100 o N2XX existe una versión especial denominada Kalibrate-UHD la cual puede ser descargada desde: http://ttsou.github.com/kalibrate-uhd/
  bzip2 -d kal-v0.4.1.tar.bz2
  tar -xvf kal-v0.4.1.tar
  ./bootstrap
  ./configure
  make
  make install

  # Si ha adquirido tarjetas hijas WBX, puede ejecutar los siguientes
  ```

comandos para escanear todas las bandas de frecuencias GSM y detectar las antenas BTS de los operadores, y tambien confirmar el buen funcionamiento del nuevo oscilador externo de su unidad USRP1:

```
# Para banda de 850 MHz
kal -s GSM850 -F 52000000 -R B
# Para banda de 900 MHz
kal -s GSM900 -F 52000000 -R B
# Para banda de 1800 MHz
kal -s DCS -F 52000000 -R B
# Para banda de 1900 MHz
kal -s PCS -F 52000000 -R B
```

IX. Finalmente instalar y descargar OpenBTS-UHD desde la página de gnuradio.org:http://gnuradio.org/redmine/projects/gnuradio/wiki/OpenBTSUHD

- ```
  git clone git://github.com/ttsou/openbts-uhd.git
  # o desde,
  git clone http://github.com/ttsou/openbts-uhd.git
  # luego configurar, compilar e instalar:
  cd openbts/public-trunk/
  ./configure
  make
  sudo make install
  reboot now
  # ¡¡¡Fin de la instalación, ☺ reiniciar el sistema y todo listo para la configuración!!!
  ```
  ```

10 Prueba de la conexión USB con la unidad usrp1

Realizada la instalación de Gnuradio y reiniciado el sistema, puede ejecutar el script python `./usrp_benchmark.py` localizado en: `/usr/local/share/gnuradio/examples/usrp` para confirmar el correcto funcionamiento de la comunicación entre Gnuradio y la unidad USRP1 mediante el puerto USB, el cual debe arrojar el siguiente resultado:



10.1 Error típico

Si al ejecutar la aplicación "`./usrp_benchmark_usb.py`" se obtiene el error mostrado más abajo, significa que el intérprete Python no encuentra su ruta o path de ejecución al directorio de binarios.

```
root@P4:/usr/local/share/gnuradio/examples/usrp#
./usrp_benchmark_usb.py
Traceback (most recent call last):
  File "./usrp_benchmark_usb.py", line 30, in <module>
    from gnuradio import gr
  File "/usr/local/lib/python2.6/dist-
packages/gnuradio/gr/__init__.py", line 43, in <module>
    from gnuradio_core import *
  File "/usr/local/lib/python2.6/dist-
packages/gnuradio/gr/gnuradio_core.py", line 23, in
<module>
    from gnuradio_core_runtime import *
  File "/usr/local/lib/python2.6/dist-
packages/gnuradio/gr/gnuradio_core_runtime.py", line 24, in
<module>
    _gnuradio_core_runtime = swig_import_helper()
  File "/usr/local/lib/python2.6/dist-
packages/gnuradio/gr/gnuradio_core_runtime.py", line 20, in
swig_import_helper
    _mod = imp.load_module('_gnuradio_core_runtime', fp,
pathname, description)
ImportError: libgnuradio-core-3.4.1git.so.0: cannot open
shared object file: No such file or directory
```

Para resolver este error, debe ejecutar el comando "`ldconfig`" dentro del directorio raíz "`/sbin`".

- `cd /sbin`
  `./ldconfig`

Este comando crea las uniones necesarias entre las librerías compartidas que se encuentran en los directorios especificados en las líneas de comando al usar la consola de terminal local o ssh remota.

10.2 Errores de voltaje

No se recomienda usar hub USB, estos no entregan el correcto voltaje hacia el oscilador externo clocktamer y unidad USRP1, estos últimos deben ser conectados directamente a los puertos USB 2.0 del PC o

Laptop. (Con USB 1.2 obtendrá el mismo error de conexión en el script python, algunos modelos laptops pueden ser incompatibles.)

11 Escaneo de bandas GSM con Kalibrator

Al usar el programa Kalibrator usted podrá escanear las diferentes bandas de frecuencias GSM y descubrir los distintos canales usados por las diferentes antenas BTS de los operadores de telefonía local.

- `kal -s GSM850 -F 52000000 -R B`
- `kal -s GSM900 -F 52000000 -R B`
- `kal -s DCS -F 52000000 -R B`
- `kal -s PCS -F 52000000 -R B`

Scanned Band	Threshold	Channel Range	Channel	Frecuency	Power
GSM850	7556dot703151	128 to 251	178	879.2MHz + 255Hz	214785.07
			179	879.4MHz + 248Hz	60489.92
			180	879.6MHz + 253Hz	148005.61
			181	879.8MHz + 258Hz	35883.31
			234	890.4MHz + 260Hz	41449.75
			235	890.6MHz + 261Hz	11994.92
GSM900	233dot282949	1 to124	121	959.2MHz + 32.279kHz	707.72
			122	959.4MHz + 32.489kHz	703.88
DCS (1800)	94dot97452	512 to 885	583		132.34
			704	1843.6MHz + 612Hz	243.27
			742	1851.2MHz - 18.688kHz	119.39
			822	1867.2MHz + 362Hz	243.27
			824	1867.2MHz + 333Hz	225.20
			824	1867.6MHz + 231Hz	119.19
			843	1871.4MHz - 9.362kHz	162.19
			844	1871.6MHz - 9.421kHz	116.30
PCS (1900)	145dot197085	512 to 810	513	1930.4MHz + 590Hz	1322.31
			518	1931.4MHz + 563Hz	375.39
			521	1932.0MHz + 558Hz	230.50
			525	1932.8MHz + 631Hz	865.19
			526	1933.0MHz + 626Hz	228.65
			530	1933.8MHz + 601Hz	510.34
			531	1934.0MHz + 676Hz	157.14
			573	1942.4MHz + 611Hz	812.56
			584	1944.6MHz + 491Hz	357.21
			588	1945.4MHz + 589Hz	10871.53
			589	1945.6MHz + 597Hz	2959.15
			590	1945.8MHz + 588Hz	618.00
			591	1946.0MHz + 606Hz	208.81
			595	1946.8MHz + 555Hz	505.34
			599	1947.6MHz + 559Hz	176701.45
			602	1948.2MHz + 578Hz	271.78
			603	1948.4MHz + 557Hz	254.72
			606	1949.0MHz + 1.644kHz	1773.49
			607	1949.2MHz + 568Hz	452.08
			608	1949.4MHz + 554Hz	463.29
			633	1954.4MHz - 407Hz	392.56
			749	1977.6MHz + 546Hz	307.23

12 Tabla MCC y MNC para Chile

De acuerdo a la página web Wikipedia los códigos de las redes de telefonía celular MCC (Mobile Country Code) y MNC (Mobile Network Code) en Chile usados por los operadores más conocidos son el 730 (para todos los operadores) y desde el 01 al 99:

http://en.wikipedia.org/wiki/Mobile_Network_Code

Chile - CL

MCC	MNC	Brand	Operator	Status	Bands (MHz)
730	01	entel	Entel PCS Telecomunicaciones S.A.	Operational	GSM 1900 / UMTS 1900
730	02	movistar	Telefónica Móvil de Chile	Operational	GSM 850 / UMTS 850 / UMTS 1900
730	03	Claro	Claro Chile S.A.	Operational	GSM 1900 / CDMA2000 1900 / UMTS 850
730	04	Nextel	Centennial Cayman Corp. Chile	Operational	iDEN 800
730	08	VTR Móvil	VTR S.A.	Not operational	UMTS 1700 / UMTS 2100
730	09	Nextel	Centennial Cayman Corp. Chile	Operational	UMTS 1700 / UMTS 2100
730	10	entel	Entel Telefonía Móvil S.A.	Operational	GSM 1900 / UMTS 1900
730	99	Will	WILL Telefonía	Operational	GSM 1900 / UMTS 1900 (Residential).

13 Configurando OpenBTS

La configuración de OpenBTS reside en un archivo maestro con el nombre de "`OpenBTS.config`", el cual se encuentra localizado en el directorio "`/apps`" de la raíz de la instalación de OpenBTS.

13.1 Definición del tipo de red GSM

Se deben especificar lo códigos de la red GSM, en este caso con valores de red de prueba o desarrollo con los valores **001** y **01** para el **MCC** y **MNC** respectivamente, quedando el archivo de la siguiente manera:

```
# 001 = Valor MCC de red de prueba sin corresponder a
ningun país.
GSM.MCC 001

# MNC = 01 Valor MCC de código red de prueba sin
corresponder a ninguna compañía operadora.

GSM.MNC 01
```

Los valores de códigos de red de prueba permiten que cualquier terminal GSM de cualquier operador pueda conectarse a la antena BTS, sin importar los códigos de red MCC y MNC del chip SIM. (MCC 730 para Chile, MNC 01, 02, 03 hasta 99 para los diferentes operadores de telefonía móvil de Chile)

Configurados los valores MCC y MNC se necesita finalmente especificar la banda de frecuencia a usar. Para no interferir con las bandas de frecuencias de los operadores más usadas (850 Mhz y 1900 MJHZ) se recomienda usar bandas de frecuencia no usadas por estos, como las banda de 900 o 1800 MHz en la opción `GSM.Band`.

```
GSM.Band 900
$static GSM.Band
GSM.ARFCN 50
$static GSM.ARFCN
GSM.Neighbors 51 55
```

13.2 Selección de ARFCN

Para los valores del ARFCN a ser usado por la antena BTS en la línea `GSM.ARFCN`, se tomará un valor de canal no usado por ningún operador en la banda de los 900 MHz. Tomando como referencia la tabla del escaneo previo de las bandas GSM con Kal, podemos usar el canal 50 el cual esta distante a los canales 121 y 122 detectados por la unidad USRP previamente.

Finalmente podemos especificar valores de canales de celdas vecinas como `51 y 55` en `GSM.Neighbors`. En nuestro caso, estas celdas son inexistentes, pero es necesario especificarlas para el funcionamiento de OpenBTS.

13.3 Nombre de red

OpenBTS permite asignar un nombre a nuestra nueva red. Por defecto el archivo "`OpenBTS.conf`" tiene el nombre OpenBTS en varios lugares del archivo. Puede reemplazar y ocupar el nombre que desee como por ejemplo:

- OpenGSM
- FreeGSM
- TestGSM

Solo debe asegurarse de que el nombre no tenga dos palabras o espacios en blanco.

13.4 Configuración de la ruta de Asterisk

Si va a emplear un servidor Asterisk en forma local junto con OpenBTS, no se requieren mayores modificaciones al archivo OpenBTS.config. Solo debe dejar la IP que viene por defecto intacta "`127.0.0.1`".

14 Configuración de Asterisk

Instalada la aplicación de central Vo-IP Asterisk, se deben intervenir dos archivos, los cuales son sip.conf y extensions.conf para crear la nueva numeración para los terminales GSM y rutas de discado entre terminales GSM de la celda y conexión a red pública ya sea por trama E1 o por algún proveedor vo-ip ITSP (Internet Telephony Service Provider) mediante SIP Trunking.

14.1 Obtención de códigos IMSI de los terminales GSM

Existen dos maneras de obtener los códigos IMSI de los chips SIM de cualquier terminal GSM:

1. Mediante el uso de cualquier lector USB de tarjetas de memoria con capacidad de chip SIM, el cual puede ser encontrado en muchos comercios locales de hardware. Recomendamos usar el programa Sim Manager de Dekart Software, el cual puede ser descargado en: http://www.dekart.com/products/card management/sim manager/ con libre uso por 30 días.
2. Si OpenBTS se encuentra funcionando y sin importar si Asterisk se encuentra configurado, cualquier terminal GSM que busque y detecte en forma manual la celda activa de OpenBTS en su lista de operadores disponibles y solicite conectarse a esa red, OpenBTS generará un archivo temporal TMSI el cual contiene el código IMSI del terminal GSM.

```
OpenBTS> tmsis
TMSI       IMSI            IMEI              age  used
0x4e4036e6 730025900111944                ?   97h   94h
0x4e4036e7 730011835026703                ?   97h   94h
0x4e403ab1 730011605081998                ?   97h   96h
0x4e403ab2 730011001425862                ?   96h   95h
0x4e403f7d 310410419846448                ?   96h   95h
0x4e403f7e 730027300177160                ?   96h   94h

6 TMSIs in table
OpenBTS>
```

Tabla TMSIS de OpenBTS con los registros IMSI y TMSI de los terminales que solicitaron registrarse a la celda BTS.

14.2 Aprovisionamiento de numeración

Una vez obtenidos los códigos IMSI de los terminales GSM deseados, dentro del archivo "`sip.conf`", se crean los anexos SIP necesarios con al siguiente estructura asignando un número de callerid diferente a cada

código IMSI. Cada anexo SIP debe por norma general tener los (15) quince dígitos del código IMSI, ejemplo;

[IMSI730011605081998] callerid=1000 canreinvite=no type=friend context=openbts disallow=all allow=ulaw allow=alaw allow=gsm host=dynamic dtmfmode=info	[IMSI730011835026703] callerid=1001 canreinvite=no type=friend context=openbts disallow=all allow=ulaw allow=alaw allow=gsm host=dynamic dtmfmode=info

14.3 Rutas de discado

Finalizado el aprovisionamiento de los anexos o números en "`sip.conf`" se debe proceder a generar las rutas de discado en "`extensions.conf`" de la siguiente manera:

```
[sip-local]
exten => 1000,1,Dial(SIP/IMSI730011605081998)
exten => 1001,1,Dial(SIP/IMSI730011835026703)
exten => 1002,1,Dial(SIP/IMSI730025900111944)
exten => 1003,1,Dial(SIP/IMSI730030201450875)
```

Finalizado estos pasos de aprovisionamiento solo debe conectarse a la consola de CLI de Asterisk y realizar un reload de ambos archivos. A partir de este momento queda habilitado para ejecutar llamadas en su celda OpenBTS (¡Felicitaciones!).

15 Servidor SMS de OpenBTS (smqueue)

A partir de la versión 2.5.6 (Lacassine) de OpenBTS, este posee un módulo independiente de servidor de SMS denominado smqueue. El cual permite enviar y recibir mensajes de texto corto entre terminales GSM.

Esta solución no tiene relación con las soluciones implementada por los operadores de telefonía GSM tradicionales debido a que funciona bajo protocolo SIMPLE (SIP for Instant Messaging and Presence Leveraging Extensions)

15.1 Configuración de smqueue

Para habilitar el servicio de SMS en OpenBTS solo se requiere editar el archivo "`smqueue.conf`" en ubicado en el directorio "`/smqueue`" de la raíz de instalación de OpenBTS y especificar la IP del servidor asterisk el cual puede ser local o remoto.

```
# Local SIP config
SIP.myPort 5063
# smqueue's addresses, as seen by the Asterisk server
SIP.myIP 127.0.0.1
SIP.myIP2 192.168.0.102
```

Para iniciar el servidor smqueue sólo se debe ejecutar el comando: "`./smsqueue`" desde el terminal SSH obteniendo el siguiente resultado:

smqueue también permite recibir mensajes de texto desde otros servidores SIP mediante método de SIP global relay o integrarse a Gateway HTTP para envío/recepción de SMS desde servicios externos como clickatell.com.