



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Linares

Trabajo Fin de Grado

RECEPCIÓN Y ANÁLISIS DE SEÑAL LTE MEDIANTE RADIO DEFINIDA POR SOFTWARE (SDR)

Alumno: José Manuel Cobo Fernández

Tutor: *Prof. D. Damián Martínez Muñoz*

Depto.: Ingeniería de Telecomunicación

Octubre, 2019

RECEPCIÓN Y ANÁLISIS DE SEÑAL LTE MEDIANTE RADIO DEFINIDA POR SOFTWARE (SDR)

Alumno: José Manuel Cobo Fernández

Tutor: *Prof. D. Damián Martínez Muñoz*

Depto.: Ingeniería de Telecomunicación

Octubre, 2019

FIRMA AUTOR

FIRMA TUTOR

AGRADECIMIENTOS

A lo largo de mi etapa académica he compartido buenos momentos con personas que hacían más ameno el aprendizaje; a mis compañeros y compañeras les agradezco su compañía y atención.

Quiero agradecer el apoyo que me ha brindado mi familia durante esta etapa, en la que también llegan momentos no tan buenos, y a los que debemos hacer frente para conseguir el objetivo.

También quiero agradecer el apoyo de todos los profesores y profesoras, que no sólo han aportado conocimientos técnicos a mi formación, sino también aptitudes para afrontar los retos profesionales.

Cómo no, quiero agradecer a mis amigos y amigas su paciencia con los "No puedo ir hoy, tengo que entregar mañana una práctica" o algún tipo de motivo por el que algún día no podíamos quedar para hacer algo.

No quiero dejar sin mencionar a todas las personas que en este último año he conocido, como han sido nuevos compañeros y compañeras de trabajo que me han ayudado siempre que les he necesitado y a los que considero como amigos, y a otras personas que me han aportado mucho también a nivel personal.

Por último, me gustaría hacer una mención especial a mi tutor del Trabajo de Fin de Grado, D. Damián Martínez Muñoz, por su aportación y su disponibilidad siempre que le he necesitado.

A todas las personas aquí mencionadas, les doy las gracias por todo. ¡Ha sido un placer!

No obstante, es probable que me deje a algún grupo sin mencionar...perdonadme si no os he mencionado.

José Manuel Cobo Fernández

Estudiante del Grado en Ingeniería de Tecnologías de Telecomunicación

Linares, Octubre de 2019.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS	6
ÍNDICE DE TABLAS.....	7
RESUMEN	8
GLOSARIO DE TÉRMINOS.....	9
1. INTRODUCCIÓN	12
1.1. Objetivos.....	12
1.2. Organización del trabajo.....	13
1.3. Estructura del TFG.....	13
2. EVOLUCIÓN DE LOS SISTEMAS DE TELEFONÍA MÓVIL	15
2.1. 1G [6].....	15
2.2. 2G [6].....	15
2.2. 3G.....	17
2.4. 4G.....	19
3. CONCEPTOS TEÓRICOS DE SISTEMAS 4G	21
3.1. Bandas de frecuencia.....	21
• Banda 20 de LTE (800 MHz).....	21
• Banda 3 de LTE.....	22
• Banda 7 y Banda 38 de LTE.....	23
3.1.1. LTE Discovery.....	26
3.2. OFDM	26
3.3. OFDMA como técnica de acceso múltiple para el enlace descendente.....	28
3.4. Desarrollo y arquitectura de Sistemas LTE	29
3.4.1. Core Network	30
3.5. Red de acceso radio	31
3.6. Estructura y características del acceso radio en LTE.....	31
3.6.1. Estructura en el dominio de la frecuencia [7]	32
3.6.2. Estructura en el dominio del tiempo [7].....	32
3.6.3. Recursos físicos [7]	34
3.7. Cell Search como técnica de acceso al medio y canales en LTE [7]	36
3.7.1. Canales Físicos	38
3.7.2. Canales de Transporte	40
3.7.3. Canales Lógicos.....	40

4. SOFTWARE DEFINED RADIO (SDR)	42
4.1. ¿Qué es SDR?	42
4.2. Inicios de SDR	43
4.3. Hardware SDR	43
4.4. Herramientas Software SDR	46
5. MONTAJE UTILIZADO	48
5.1. Esquemático de la placa BladeRF	50
5.2. Primeros pasos con BladeRF y LTE Toolbox	52
6. PUESTA EN MARCHA DEL MONTAJE	55
6.1. Captura de señal LTE	55
6.2. Procesado de señales LTE	57
• Introducción	58
• Carga y procesado de la señal procedente del eNodeB	58
• Búsqueda de celdas, longitud de Prefijo cíclico y detección de modo dúplex	60
• Estimación y corrección del desplazamiento de frecuencia:	60
• Demodulación OFDM y estimación de canal	60
• Demodulación PBCH, Decodificación BCH y Análisis de MIB	61
• Demodulación OFDM en Ancho de banda total	61
• Decodificación de SIB1	61
7. RESULTADOS	63
8. CONCLUSIONES Y LÍNEAS DE FUTURO	69
BIBLIOGRAFÍA	71
ANEXO I: MANUAL DE INSTALACIÓN DE BLADERF	74
ANEXO II: GUIDE DE MATLAB	80
Iniciación a GUIDE	80
GUIDE CREADA PARA EL TF	83

ÍNDICE DE FIGURAS

<i>Figura 1: Evolución de las tecnologías inalámbricas [7]</i>	20
<i>Figura 2: Logo de la App LTE Discovery [14]</i>	26
<i>Figura 3: Captura propia de la interfaz de LTE Discovery</i>	26
<i>Figura 4: Espectro de una subportadora y espectro OFDM [32]</i>	27
<i>Figura 5: Esquema de sistema multiacceso [15]</i>	28
<i>Figura 6: Esquema de acceso múltiple distribuido [15]</i>	29
<i>Figura 7: Arquitectura del Core Network [7]</i>	30
<i>Figura 8: Arquitectura de red de acceso radio [7]</i>	31
<i>Figura 9: Estructura de LTE en el dominio del tiempo [7]</i>	34
<i>Figura 10: Recurso básico tiempo-frecuencia [7]</i>	34
<i>Figura 11: Planificación en el enlace descendente (DL) [7]</i>	35
<i>Figura 12: Posiciones de PSS y SSS en el dominio del tiempo en FDD y TDD [7]</i>	37
<i>Figura 13: Diagrama de bloques de receptor superheterodino [5]</i>	44
<i>Figura 14: Diagrama de bloques de receptor SDR [5]</i>	44
<i>Figura 15: Comparación de equipos SDR [1]</i>	46
<i>Figura 16: Interfaz de SDR-Sharp [4]</i>	47
<i>Figura 17: Dispositivo BladeRF [18]</i>	48
<i>Figura 18: Antena RoHS 2J301M [19]</i>	48
<i>Figura 19: Respuesta en frecuencia Antena RoHS 2J301M</i>	49
<i>Figura 20: Comparación FPGAs de Altera [22]</i>	50
<i>Figura 21: Diagrama de bloques de BladeRF [18]</i>	51
<i>Figura 22: Diagrama de bloques Transceiver Lime Micro [23]</i>	52
<i>Figura 23: Captura Línea de comandos BladeRF [24]</i>	53
<i>Figura 24: Logo NetMonitor [27]</i>	63
<i>Figura 25: Captura Aplicación NetMonitor Pantalla 1</i>	64
<i>Figura 26: Captura NetMonitor Pantalla 3</i>	64
<i>Figura 27: Captura Señal Vodafone (Banda 3)-Samplerate: 18 Mmuestras/s</i>	65
<i>Figura 28: Captura Señal Vodafone (Banda 3)-Samplerate: 30,72 Mmuestras/s</i>	66
<i>Figura 29: Captura Señal Orange (Banda 20)</i>	67
<i>Figura 30: Captura Señal Vodafone (Banda 20)</i>	67
<i>Figura 31: Captura Señal Movistar (Banda 20)</i>	68
<i>Figura 32: Paleta de componentes GUIDE [28]</i>	81
<i>Figura 33: Property Inspector GUIDE [28]</i>	82

<i>Figura 34: Editor de GUIDE [29].....</i>	<i>82</i>
<i>Figura 35: Interfaz de carga de datos desde archivo.....</i>	<i>85</i>
<i>Figura 36: Captura típica Vodafone Señal LTE.....</i>	<i>89</i>
<i>Figura 37: Resultado Botón Generar Aleatorios.....</i>	<i>90</i>
<i>Figura 38: Resultado Botón Capturar datos de dispositivo.....</i>	<i>92</i>
<i>Figura 39: Resultado Botón Informar.....</i>	<i>93</i>
<i>Figura 40: Resultado Botón Acerca de.....</i>	<i>95</i>

ÍNDICE DE TABLAS

<i>Tabla 1: Operadores en Banda 20 LTE [9].....</i>	<i>22</i>
<i>Tabla 2: Operadores en Banda 3 LTE [9].....</i>	<i>22</i>
<i>Tabla 3: Operadores en Bandas 7 y 38 LTE-FDD [9].....</i>	<i>23</i>
<i>Tabla 4: Operadores en Bandas 7 y 38 LTE-TDD [9].....</i>	<i>25</i>
<i>Tabla 5: Parámetros LTE.....</i>	<i>36</i>

RESUMEN

Este Trabajo Fin de Grado tiene como propósito el diseño de una interfaz gráfica de representación de señales y elementos característicos de LTE utilizando la herramienta GUIDE de Matlab.

Para llevar a cabo una extracción de datos y características de LTE, necesitamos la adquisición de datos del exterior. Para ello, se ha usado el dispositivo **BladeRF**, una tarjeta SDR de bajo/medio coste de la compañía Nuand, que puede operar en el rango de LTE.

Para dotar al trabajo de una funcionalidad necesaria a priori para el usuario, se ha implementado la interfaz teniendo en cuenta que se pueden obtener capturas de señal en tiempo real, con los altos recursos de computación que eso conlleva y los cuáles se requieren para un correcto procesado.

GLOSARIO DE TÉRMINOS

AMPS: Advanced Mobile Phone System

NTT: Nippon Telegraph and Telephone Corporation

NMT: Nordic Mobile Telephone

TACS: Total Access Communications System

RTMI: Radioteléfono móvil integrado

GSM: Global System for Mobile Communications

IS: Interim Estándar

CDMA: Code Division Multiple Access

TDMA: Time Division Multiple Access

PDC: Personal Digital Cellular

D-AMPS: Digital Advanced Mobile Phone System

GPRS: General Packet Radio Service

3GPP: The 3rd Generation Partnership Project

WCDMA: Wideband Code Division Multiple Access

TD-SCDMA: Time Division Synchronous Code Division Multiple Access

IMT-A: The International Mobile Telecommunications-Advanced Standard

GMSK: Gaussian minimum shift keying

HSDPA: High Speed Downlink Packet Access

LTE: Long Term Evolution

LTE-Advanced: Long Term Evolution-Advanced

UMTS: Universal Mobile Telecommunications System

IMT-2000: International Mobile Telecommunications-2000

SDR: Software Defined Radio

DL: DownLink

UL: UpLink

FDD: Frequency-division duplexing

TDD: Time-division duplexing

QAM: Quadrature Amplitude Modulation

WiMAX: Worldwide Interoperability for Microwave Access

DVB: Digital Video Broadcasting

OFDM: Orthogonal Frequency Division Multiplexing

OFDMA: Orthogonal Frequency-Division Multiple Access

EPC: Evolved Packet Core

EPS: Evolved Packet System

RAN: Radio Access Network

MME: Mobility Management Entity
NAS: Non-access stratum
AS: Access stratum
S-GW: Serving Gateway
P-GW: Packet Data Network Gateway
QoS: Quality of Service
HSS: Home Subscriber service
IFFT: Inverse Fast Fourier Transform
FFT: Fast Fourier Transform
SFN: System Frame Number
RE: Resource Element
RB: Resource Block
PBCH: Physical Broadcast Channel
PSS: Primary synchronization signal
SSS: Secondary synchronization signal
MIB: Master Information Block
SIB1: System Information Block number 1
PCI: Physical Cell Id
QPSK: Quadrature Phase-Shift Keying
CFI: Control Format Indicator
DCI: Downlink Control Information
PCFICH: Physical Control Format Indicator Channel
PDCCH: Physical Downlink Control Channel
PHICH: Physical Hybrid ARQ Indicator Channel
PBCH: Physical Broadcast Channel
PDSCH: Physical Downlink Shared Channel
BCH: Broadcast Channel
DL-SCH: Downlink Shared Channel
PCH: Paging Channel
MCH: Multicast Channel
BCCH: Broadcast Control Channel
PCCH: Paging Control Channel
CCCH: Common Control Channel
MCCH: Multicast Control Channel
DCCH: Dedicated Control Channel
IEEE: Instituto de Ingenieros Eléctricos y Electrónicos
DAC: digital-to-analog converter

ADC: analog-to-digital converter
eNodeB: Evolved Node B
DDC: Digital Down Converter
FPGA: Field-programmable gate array
CLI: Command Line Interface
MMSE: Minimum Mean Squared Error
FIR: Finite Impulse Response
DSP: Digital Signal Processing
UE: User Equipment
 T_G : Tiempo de Guarda
CP: Prefijo Cíclico
 N_{FG} : Frecuencias de Guarda
ICI: Inter-Carrier Interference
BER: Bit Error Rate
 N_p : Portadoras Piloto
 T_s : Período de símbolo
 Δf : Separación entre subportadoras
 N_{FFT} : Tamaño FFT
 $T_{subframe}$: Tiempo de subtrama
 T_{frame} : Tiempo de trama
 T_{slot} : Tiempo de slot
 F_s : Frecuencia de muestreo
 $N_{Samples}$: número de muestras
 F_c : Frecuencia de la portadora

1. INTRODUCCIÓN

Las formas de comunicación entre personas han evolucionado a lo largo de la historia de la humanidad. La comunicación entre personas, sin usar ningún medio tecnológico, es limitada, ya que es imposible comunicarse con personas que se encuentren a una distancia lejana de ti por cuestiones puramente físicas. [17]

Las comunicaciones móviles han experimentado un crecimiento ininterrumpido desde las últimas décadas, ya que la mayoría de la población tiene en la actualidad un aparato que haga uso de ello; no como hace unas décadas, en que sólo se usaba por una parte minoritaria de la población por el precio elevado de la tecnología.

El mundo de las comunicaciones móviles ha pasado por distintas generaciones, cada una de ellas basada en un sistema, unos estándares, capacidades técnicas y nuevas características que la diferenciaban de la anterior, aunque algunas intentaban reutilizar algunas características y elementos de las anteriores, intentando adaptarse a las necesidades demandadas en cada época.

1.1. Objetivos

Este Trabajo Fin de Grado tiene como objetivos principales:

- La captura de señales LTE en el enlace descendente (DL) mediante la plataforma SDR BladeRF x40, lo que implica que es necesario el aprendizaje de la tecnología SDR y la creación de un marco de trabajo que integra el entorno de la tarjeta SDR con el entorno de trabajo de Matlab para su posterior procesamiento, que constituye el siguiente objetivo.
- Procesado de la señal recibida en el DL mediante el ejemplo que proporciona la Toolbox LTE de Matlab "Cell Search, MIB and SIB1 Recovery"; se realiza el proceso de *Cell Search*, que explicaremos más adelante, y que a modo de resumen, es el proceso por el cual los usuarios pueden encontrar la celda en la que van a operar dentro de la red. En este caso, debemos estudiar ese proceso y adquirir los conocimientos necesarios de características de LTE para poder entender el proceso.
- Programación de una interfaz usando la funcionalidad GUIDE de Matlab y que permite aunar ambos procesos descritos anteriormente.

1.2. Organización del trabajo

En cuanto a la organización del trabajo, éste se puede dividir en etapas, teniendo en cuenta que algunas se solapan en el tiempo y otras están siempre presentes:

- Búsqueda de información y documentación sobre los temas relevantes del trabajo como son LTE y SDR
- Prácticas con tarjeta BladeRF, teniendo en cuenta sus características. Se puede dividir en 2 subetapas:
 - Manejo a través de línea de comandos CLI.
 - Manejo a través de Matlab (*System Object* de Matlab)
- Análisis de código de Matlab "Cell Search, MIB and SIB1 Recovery" para su comprensión y posterior modificación para integrar en la interfaz gráfica.
- Captura de señales para comprobar que el código funciona.
- Desarrollo de la memoria. En este punto, se puede empezar a redactar partes teóricas, ya que las partes más prácticas, se van a completar en etapas posteriores. Aunque se incluya aquí, para indicar que se puede ir realizando poco a poco, se trata de la última etapa del desarrollo de este TFG.
- Análisis y búsqueda de información de herramienta GUIDE de Matlab para la creación de interfaces gráficas.
- Creación de interfaz, con la programación que implica pasar parámetros entre *callbacks* de las funciones implementadas.
- Captura de señales a través de la interfaz.

1.3. Estructura del TFG

En este apartado se realiza un resumen del contenido de cada uno de los capítulos que forman la memoria:

- Resumen
Se lleva a cabo una breve descripción de los propósitos generales del trabajo.
- Capítulo 1: Introducción.
En el primer capítulo se hace una breve introducción sobre la tecnología en la actualidad, y se establecen los objetivos y los puntos de partida para poder empezar a realizar el trabajo.
- Capítulo 2: Evolución de los sistemas de telefonía móvil.

En este capítulo se describen las principales tecnologías de cada una de las generaciones de los sistemas de telefonía móvil a lo largo del tiempo hasta llegar a la tecnología 4G.

- Capítulo 3: Conceptos teóricos de sistemas 4G.

En el tercer capítulo se realiza un estudio de los conceptos teóricos más importantes de los sistemas 4G, necesarios para el buen entendimiento y desarrollo del trabajo.

- Capítulo 4: Software Defined Radio (SDR).

En este capítulo se lleva a cabo una introducción a la tecnología SDR, a la vez que se describen características hardware de estos sistemas y software relacionado con SDR para el análisis de señales.

- Capítulo 5: Montaje utilizado.

En este quinto capítulo se describen los componentes hardware que se han utilizado para el desarrollo del TFG y sus características.

- Capítulo 6: Puesta en marcha del montaje.

En este capítulo se describe la puesta en marcha del montaje, explicando el proceso de *Cell search* así como los parámetros adecuados de entrada al sistema.

- Capítulo 7: Resultados.

En este capítulo se muestra capturas de señal realizadas a través del montaje propuesto para el trabajo.

- Anexo I: Manual de instalación.

Documento explicativo de los pasos a seguir para la instalación del paquete de BladeRF y las consideraciones a tener en cuenta.

- Anexo II: Guide de Matlab.

Se hace una breve descripción de la funcionalidad GUIDE de Matlab y se explica el GUIDE creado para el TFG.

2. EVOLUCIÓN DE LOS SISTEMAS DE TELEFONÍA MÓVIL

2.1. 1G [6]

Los sistemas de primera generación (1G) son sistemas basados en telefonía móvil analógica de los años 80 y dedicados exclusivamente a la voz. Su expansión no fue muy uniforme y en todo el mundo surgieron diferentes sistemas: AMPS (*Advanced Mobile Phone System*) en Estados Unidos, en Japón se desarrollaron 3 estándares desarrollados por NTT, TZ-801, TZ-802 y TZ-803, y en Europa aparecieron diferentes versiones:

- NMT (*Nordic Mobile Telephone*): Dinamarca, Noruega, Holanda, etc.
- TACS (*Total Access Communications System*): Reino Unido y España.
- C450: Alemania Oriental, Portugal.
- RTMI: Italia.
- Radiocom 2000: Francia

2.2. 2G [6]

Los sistemas de segunda generación (2G) aparecieron a principios de los años 90 motivados por las limitaciones de los sistemas de primera generación. Este nuevo sistema seguía basándose en la voz, pero pretendía mejorar las capacidades de los sistemas por el hecho de introducir protocolos de telefonía digital. Como ejemplo, se aumentaba el número de enlaces simultáneos en un mismo ancho de banda y se permitía integrar otros servicios en una misma señal.

Al igual que en los sistemas de primera generación, también se desarrollaron varios estándares:

- GSM (*Global System for Mobile Communications*)
- IS-95/CDMA
- IS-136/TDMA
- PDC (*Personal Digital Cellular*)
- D-AMPS (*Digital Advanced Mobile Phone System*)

De todos ellos, **GSM** fue el que se estandarizó y extendió a nivel mundial.

GSM es una tecnología inalámbrica que permite prestar servicios de voz de alta calidad y servicios de datos conmutados por circuitos.

La tecnología GSM permite que varios usuarios compartan el mismo canal radio por la técnica TDM (multiplexado por división en el tiempo, en castellano), por la división del canal en el dominio del tiempo, es decir, se asigna a cada transmisión de datos una ranura de tiempo dentro de un mismo canal.

Las características de la interfaz radio de GSM son: [33]

- Sistema FDMA/TDMA con FDD
- La separación de los radiocanales es 200 kHz
- Bandas de frecuencias: 900/1800/1900 MHz
- Modulación: GMSK (*Gaussian minimum shift keying*) a 270,833 kb/s
- Potencia máxima de terminal móvil:
 - 900 MHz: 8 W-0,8 W (típico 2 W)
 - 1800 MHz: 1 W-0,25 W (típico 0,25 W)
- Potencia máxima de base:
 - 900 MHz: 320 W-2,5 W
 - 1800 MHz: 20 W-2,5 W
- Distancia de cobertura máxima: 35 km, aunque puede ampliarse

En cuanto a las bandas de frecuencia, nos encontramos con las siguientes:

- GSM 900 (124 radiocanales)-(Europa)
 - UL: 890-915 MHz
 - DL: 935-960 MHz
- E-GSM 900 (49 radiocanales)-(Europa)
 - UL: 880-890 MHz
 - DL: 925-935 MHz
- GSM 1800 (374 radiocanales)
 - UL: 1710-1785 MHz
 - DL: 1805-1880 MHz
- GSM 1900 (300 radiocanales)
 - UL: 1850-1910 MHz
 - DL: 1930-1990 MHz

En el transcurso de los años, estas tecnologías se han extendido para soportar servicios de paquetes de datos, y a estas extensiones se les conoce como 2.5G, ya que ofrece un mayor rango de capacidades que las tecnologías originales. Algunos de los

ejemplos de tecnologías extendidas pueden ser EDGE y GPRS (*General Packet Radio Service*).

GPRS es una tecnología que ofrece mayor eficiencia espectral que GSM para servicio de datos y se trata de una tecnología basada en paquetes, por lo que los datos se transmiten sobre una red IP en ráfagas. Fue definido por ETSI/3GPP como parte de la fase 2+ de especificaciones de GSM. [35]

GPRS aprovecha gran parte de la estructura de red GSM y hace una reserva flexible de canales radio.

La velocidad puede ser de hasta 21,4 kbps por intervalo de tiempo.

2.3. 3G

Durante la década de los 90, la necesidad de soportar tanto servicios de voz como de datos hizo que surgiera la necesidad de desarrollar una nueva generación de tecnologías celulares.

Para que alcanzara un ámbito global el desarrollo de estos nuevos sistemas, se dieron cuenta de que el desarrollo del 3G tenía que tener unas bases globales. Por ello, la **Third-Generation Partnership Project (3GPP)** se formó para desarrollar las tecnologías 3G WCDMA (acceso radio de UMTS) y TD-SCDMA. Al poco tiempo, la organización 3GPP2 se formó para desarrollar la tecnología 3G cdma200, una evolución de la tecnología 2G IS-95.

La primera versión de WCDMA (*release 99*) fue acabada en 1999. Se incluían servicios de voz y video sobre conmutación de circuitos y servicios de datos sobre portadoras de conmutación de paquetes y circuitos.

La red de tercera generación requería mayor capacidad y debía soportar una mayor variedad de servicios con velocidad binaria elevada y variable dinámicamente. Además, debía proporcionar seguridad de acceso y confidencialidad así como una arquitectura abierta a la llegada de nuevas aplicaciones.

UMTS es uno de los principales sistemas de comunicaciones móviles de tercera generación que fue desarrollado dentro del marco de trabajo de la ITU conocido como IMT-2000 (*International Mobile Telecommunications-2000*).

UMTS integra todos los servicios y su velocidad de transferencia varía entre 144 kbps y 2 Mbps.

En cuanto a las características de su interfaz radio, nos encontramos con [34]:

- Separación entre portadoras: 5 MHz
- Modulación BPSK/ QPSK
- Ensanchamiento espectral usando códigos de canalización y aleatorización
- Estructura de trama temporal
- Potencia máxima del terminal móvil: 21 o 24 dBm
- Velocidad binaria variable de forma estática y dinámica
- Traspaso con continuidad y entre portadoras o hacia otros sistemas.

Con respecto a las bandas de frecuencia, nos encontramos con:

- Modo FDD: bandas "emparejadas" (hay otras aparte de las descritas, aunque se utilizan menos) con 12 portadoras para cada sentido de transmisión:
 - UL: 1920-1980 MHz
 - DL: 2110-2170 MHz
- Modo TDD: bandas "no emparejadas" con 7 portadoras en ambos sentidos de transmisión:
 - 1900-1920 MHz
 - 2010-2025 MHz

Las mayores mejoras en WCDMA vinieron con la introducción de HSDPA (*High Speed Downlink Packet Access*) en la *release* 5 seguido de HSPA en el que se contemplaba la mejora del enlace Ascendente (*Enhanced Uplink*), en la *release* 6. HSPA normalmente se conoce como 3.5G, tecnología que permite tasas de transmisión de varios Mbits/s a la vez que mantiene la compatibilidad con las especificaciones 3G originales.

La llegada de estas tecnologías de banda ancha propició el uso de *smartphones* como los dispositivos iPhone y Android, y la llegada a las masas de población hizo que el desarrollo fuera rápido. Por otra parte, el uso masivo de estos dispositivos y un amplio rango de servicios de datos como las redes sociales, el vídeo, *streaming*, tiendas online, etc., requiere de mayores capacidades y una mejora de la eficiencia espectral, ya que los usuarios necesitan mayores tasas de transmisión y una latencia mínima para una experiencia satisfactoria con estos servicios, por lo que estas necesidades llevaron a la evolución de HSPA y el posterior desarrollo de la tecnología 4G.

2.4. 4G

La tecnología LTE fue desarrollada inicialmente para soportar paquetes de datos y no soportar conmutación de circuitos para voz, a diferencia de 3G, en la que HSPA se introdujo para aumentar el rendimiento de los paquetes de datos en una tecnología ya existente. Los servicios móviles de banda ancha requerían grandes tasas de transmisión de datos, baja latencia y una gran capacidad, en los que además se requería flexibilidad en el espectro y una similitud entre las soluciones FDD y TDD. Se desarrolló una nueva arquitectura de red, cuyo núcleo se conoce como *Enhanced Packet Core* (EPC) y reemplazó a la arquitectura que se usaba en GSM y en WCDMA/HSPA. La primera versión de LTE apareció en la *release* 8 de las especificaciones 3GPP y el primer despliegue comercial tuvo lugar en 2009, aunque posteriormente las redes LTE experimentaron un rápido despliegue a nivel mundial.

En cuanto a LTE, el uso de una sola tecnología mundialmente aceptada propicia el rápido desarrollo de nuevos servicios y la reducción de costes tanto para usuarios como para operadores.

No obstante, la designación oficial de tecnología 4G se le concede al 3GPP LTE-Advanced (*release* 10), ya que cumple con los requisitos que establece la ITU para la primera versión del IMT-A (*International Mobile Telecommunications Advanced*), cuyo objetivo es la garantía de la calidad de servicio en la transmisión de servicios de voz y datos.

Desde su introducción comercial, LTE ha mejorado en términos de capacidad, tasas de transmisión, flexibilidad en el desarrollo y espectro, etc, permitiendo picos de transmisión de datos de hasta 300 Mbits/s. En la *release* 13, la evolución de LTE puede soportar hasta picos multi-Gbits/s. [7]. En el siguiente capítulo vamos a ver más en profundidad las características de LTE.

Además, con los avances relativamente recientes en capacidades de computación, procesamiento de señales digitales, inteligencia artificial, radiofrecuencia, diseños hardware, topologías de red y muchos otros elementos, se ha evolucionado a sistemas cada vez más complejos, con inteligencia y de alto rendimiento que se pueden adaptar a entornos cada vez más dinámicos, que presentan gran cantidad de información en tiempo real y con el mínimo número de errores. Por estas características y necesidades, surgió la tecnología de los sistemas de comunicaciones **SDR (Software Defined Radio o Radio Definida por Software)**, que adopta numerosos avances en todos los campos para obtener el máximo rendimiento en emisores y receptores.

En la siguiente figura se muestra la evolución de las tecnologías inalámbricas:

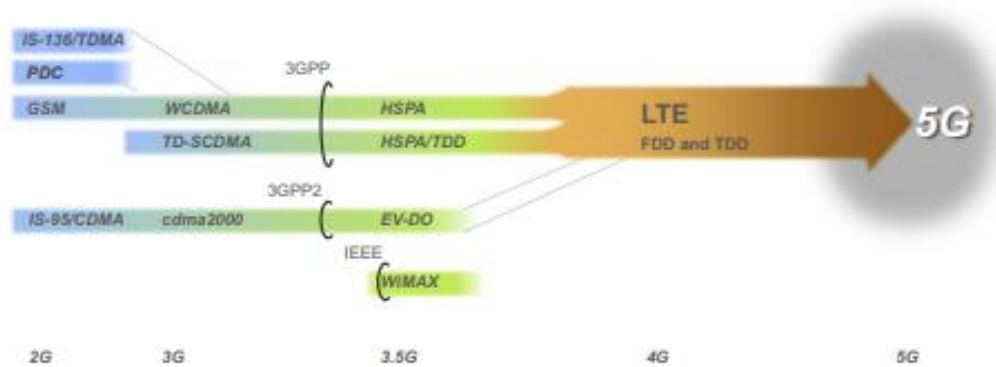


Figura 1: Evolución de las tecnologías inalámbricas [7]

3. CONCEPTOS TEÓRICOS DE SISTEMAS 4G

En este apartado se va a hacer una descripción del servicio LTE en España, las técnicas de acceso múltiple empleadas en LTE, la estructura de las señales LTE en el dominio del tiempo y la frecuencia y los diferentes canales encargados de la transmisión de información dentro del sistema LTE.

3.1. Bandas de frecuencia

Se va a describir el estado en el que se encuentra el servicio LTE en España, teniendo en cuenta las bandas operativas que tienen asignadas cada uno de los principales operadores de este país como son Movistar, Vodafone, Orange y Yoigo para dar cobertura a todo el país:

- **Banda 20 de LTE (800 MHz)**

Se trata de la primera banda que estaba siendo utilizada por la TDT y que se liberó para el impulso de la implantación del 4G. Esta liberación supuso el **primer dividendo digital**.

Tiene gran interés en la tecnología móvil 4G por su alta penetración y propagación en interiores.

También generó polémica ya que muchos terminales móviles eran al principio incompatibles con esta banda. He estado buscando y marcas como Xiaomi y Meizu [12], [13], incluyen esta banda, aunque desconozco si hay algún modelo que no la soporte, aunque me extrañaría ya que me he fijado en los modelos más básicos. No obstante, he encontrado un terminal (Jiayu M2) que directamente no soporta ninguna de las bandas LTE [11]. Ahora es extraño encontrar con un terminal no compatible, pero hace unos años terminales de gama media-alta con el "OnePlus One" no la soportaban.

Los operadores se distribuyen las siguientes frecuencias de la banda, como se aprecia en la tabla 1:

Operador	Bloque	Subida	Bajada	Concesión	Caducidad
Orange	2x 5MHz	832-837 MHz	791-796 MHz	01/04/2015	25/04/2031
Orange	2x 5MHz	837-842 MHz	796-801 MHz	01/04/2015	25/04/2031
Vodafone	2x 5MHz	842-847 MHz	801-806 MHz	01/04/2015	25/04/2031
Vodafone	2x 5MHz	847-852 MHz	806-811 MHz	01/04/2015	25/04/2031
Movistar	2x 5MHz	852-857 MHz	811-816 MHz	01/04/2015	25/04/2031
Movistar	2x 5MHz	857-862 MHz	816-821 MHz	01/04/2015	25/04/2031

Tabla 1: Operadores en Banda 20 LTE [9]

- **Banda 3 de LTE**

Se trata de la banda que más cobertura 4G ofrece en España y opera en modo dúplex FDD, con frecuencias emparejadas. La asignación de las bandas a los distintos operadores es el siguiente:

Operador	Bloque	Subida	Bajada	Concesión	Caducidad
Movistar	2x 20MHz	1710,1-1730,1 MHz	1805,1-1825,1 MHz	24/07/1998	31/12/2030
Vodafone	2x 20MHz	1730,1-1750,1 MHz	1825,1-1845,1 MHz	24/07/1998	31/12/2030
Yoigo	2x 5MHz	1750,1-1755,1 MHz	1845,1-1850,1 MHz	15/06/2011	31/12/2030
Yoigo	2x 5MHz	1755,1-1760,1 MHz	1850,1-1855,1 MHz	15/06/2011	31/12/2030
Yoigo	2x 4,8 MHz	1760,1-1764,9 MHz	1855,1-1859,9 MHz	15/06/2011	31/12/2030
Orange	2x 20MHz	1764,9-1784,9 MHz	1859,9-1879,9 MHz	24/07/1998	31/12/2030

Tabla 2: Operadores en Banda 3 LTE [9]

- **Banda 7 y Banda 38 de LTE**

Esta banda se utiliza para la tecnología LTE en aquellas zonas en las que la banda 3 (1800 MHz) está bastante saturada, como zonas con gran afluencia de público.

Operadores con frecuencias FDD:

Bloque	Subida	Bajada	Operador (Cobertura)	Concesión	Caducidad	Observaciones
2x10 MHz	2500-2510MHz	2620-2630MHz	Movistar (Estatal)	10/10/2011	31/12/2030	-
2x10 MHz	2510-2520MHz	2630-2640MHz	Movistar (Estatal)	10/10/2011	31/12/2030	-
2x10 MHz	2520-2530MHz	2640-2650MHz	Orange (Estatal)	10/10/2011	31/12/2030	-
2x10 MHz	2530-2540MHz	2650-2660MHz	Orange (Estatal)	10/10/2011	31/12/2030	-
2x5 MHz	2540-2545MHz	2660-2665MHz	Vodafone (Estatal)	10/10/2011	31/12/2030	-
2x5 MHz	2545-2550MHz	2665-2670MHz	Vodafone (Estatal)	10/10/2011	31/12/2030	-
2x5 MHz	2550-2555MHz	2670-2675MHz	Vodafone (Estatal)	10/10/2011	31/12/2030	-
2x5 MHz	2555-2560MHz	2675-2680MHz	Vodafone (Autonómica)	10/10/2011	31/12/2030	Todas las autonomías

Bloque	Subida	Bajada	Operador (Cobertura)	Concesión	Caducidad	Observaciones
2x10 MHz	2560- 2570MHz	2680- 2690MHz	Euskaltel (Autonómica)	10/10/2011	31/12/2030	País Vasco
			Jazztel (adquirida por Orange) (Autonómica)	10/10/2011	31/12/2030	Andalucía, Baleares y Canarias
			Orange (Autonómica)	20/05/2016	31/12/2030	Aragón, Cantabria, Castilla y León, Cataluña, Comunidad Valenciana, Extremadura, La Rioja, Murcia, Navarra y Ceuta
			Movistar (Autonómica)	20/05/2016	31/12/2030	Madrid y Melilla
			R Cable (Autonómica)	10/10/2011	31/12/2030	Galicia
			Telecable (Autonómica)	10/10/2011	31/12/2030	Asturias
			Telecom CLM (Autonómica)	10/10/2011	31/12/2030	Castilla-La Mancha

Tabla 3: Operadores en Bandas 7 y 38 LTE-FDD [9]

Operadores con frecuencias TDD:

Cobertura	Bloque	Subida	Bajada	Operador	Concesión	Caducidad	Observaciones
Estatat	1x10 MHz	2575-2585 MHz		Vodafone	20/12/2011	31/12/2030	-
Estatat	1x10 MHz	2585-2595 MHz		Vodafone	20/12/2011	31/12/2030	-
Autonómica	1x10 MHz	2595-2605 MHz		Aire Networks	30/07/2014	31/12/2030	Extremadura y Melilla
				Aire-Alvi-TVHoradada-Wimaz	30/07/2014	31/12/2030	Comunidad Valenciana
				Cota	20/12/2011	31/12/2030	Murcia
				Euskaltel	20/12/2011	31/12/2030	País Vasco
				ib-red	30/07/2014	31/12/2030	Baleares
				MASMOVIL	30/07/2014	31/12/2030	Andalucía, Castilla-La Mancha, Cataluña y Madrid
				R cable	20/12/2011	31/12/2030	Galicia
				Telecable	20/12/2011	31/12/2030	Asturias
				Sin asignar		31/12/2030	Resto de comunidades y Ceuta
Estatat	1x10 MHz	2605-2615 MHz		Aire Networks	20/12/2011	31/12/2030	Revendida por Orange

Tabla 4: Operadores en bandas 7 y 38 LTE-TDD [9]

Estas son las principales bandas que dan cobertura 4G a España; no obstante, hay otras bandas que se encuentran en despliegue o aún no han salido a subasta. [8] [9]

Por ejemplo, la banda 28 (700 MHz) actualmente se utiliza para la difusión de la televisión y está previsto su uso en 5G. Además, también está previsto que va a usar la banda 42 (3500 MHz), que actualmente se emplea para WiMAX.

La banda 32 (1500 MHz) es una banda de solo bajada que va a salir a subasta pública próximamente. [10]

3.1.1. LTE Discovery

Para comprobar qué banda está usando nuestro móvil, podemos descargarnos la app para Android *LTE Discovery*:



Figura 2: Logo de la App *LTE Discovery* [14]

En la siguiente captura realizada por mí, se muestra que en ese momento el dispositivo móvil estaba usando la banda 20 para transmitir y recibir, con frecuencia *downlink* 806 MHz y frecuencia *uplink* de 847 MHz.

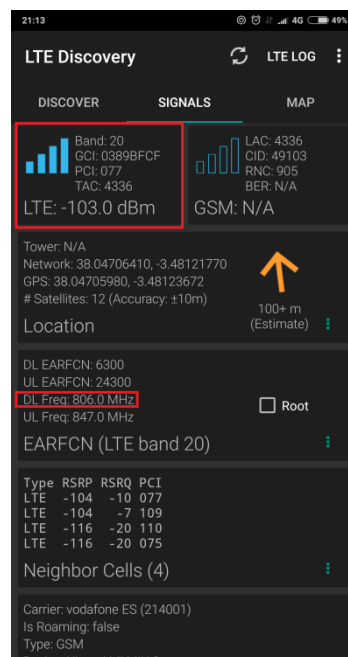


Figura 3: Captura propia de la interfaz de *LTE Discovery*

3.2. OFDM

La modulación por división ortogonal de frecuencia (en inglés, Orthogonal Frequency Division Multiplexing, OFDM) es una modulación que permite enviar información modulando en QAM o en PSK un conjunto de portadoras de diferente frecuencia, es decir, hace uso de diferentes subportadoras ortogonales sobre un ancho de banda amplio. Cada subportadora está separada $f_k = f_0 + k/T$, donde T es el período de símbolo.

OFDM no sólo es usado en LTE, ya que también se usa en otras tecnologías radio como son WiMAX y DVB.

La modulación OFDM es robusta frente al multitrayecto habitual en canales de radiodifusión, y frente al desvanecimiento debido a las condiciones meteorológicas y las interferencias de RF.

Por las características de esta modulación, las señales con diferentes retardos que llegan al receptor pueden contribuir positivamente al proceso de recepción, lo que hace que se puedan construir redes de radiodifusión de frecuencia única en la que las interferencias no sean un problema ni afecten.

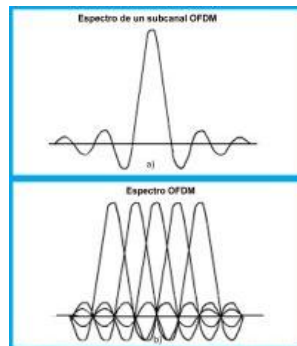


Figura 4: Espectro de una subportadora y espectro OFDM [32]

Para generar la modulación OFDM hay que tener en cuenta parámetros como: el número de subportadoras, el esquema de modulación y la frecuencia de guarda que se va a utilizar.

El número de subportadoras OFDM puede variar y el espaciado entre subportadoras también. Un símbolo OFDM está formado por la suma de N subportadoras moduladas que se implementa mediante la IFFT (Transformada Rápida de Fourier Inversa). OFDM también añade otros parámetros para garantizar su funcionamiento:

- **Tiempo de Guarda (T_G):**

Es el tiempo que se añade a cada símbolo OFDM para evitar que se produzca ISI (Inter-Symbol Interference). Este tiempo debe ser mayor que el retardo máximo de la señal ($T_G > \tau_{\max}$).

- **Prefijo Cíclico (CP):**

Es una extensión que se añade al tiempo de guarda para evitar la pérdida de ortogonalidad entre las subportadoras o ICI (*Inter-Carrier Interference*). Consiste en la replicación del final de cada símbolo OFDM.

- **Frecuencia de Guarda (N_{FG}):**

Pretende conseguir la protección frente a interferencias del canal adyacente dejando sin modular las portadoras del extremo del espectro.

- **Portadoras Piloto (N_P):**

Se utilizan para estimar el canal en la frecuencia y el tiempo en recepción (más adelante se describirá la estructura de la trama LTE).

- **Codificación del Canal:**

OFDM combate el fading selectivo que sufren algunas de las subportadoras a través de una codificación que implica la utilización de códigos de bloque, códigos convolucionales y entrelazado. Hay una alta BER (*Bit Error Rate*) debido a la alta BER de algunas subportadoras.

3.3. OFDMA como técnica de acceso múltiple para el enlace descendente

OFDMA (Orthogonal Frequency-Division Multiple Access) se utiliza para describir el uso de OFDM como técnica de multiacceso, ya que en cada intervalo de símbolo OFDM, las subportadoras se transmiten a diferentes terminales o UE (*User Equipment*). Es decir, la diferencia con OFDM es que en OFDM se da por hecho que un transmisor emite hacia un solo receptor a diferencia de OFDMA.

Hay 2 formas de transmitir los bloques de subportadoras a un terminal:

- De forma consecutiva (Sistema multiacceso):



Figura 5: Esquema de sistema multiacceso [15]

- De forma dispersa (Acceso múltiple distribuido):

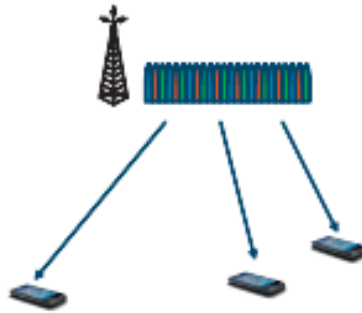


Figura 6: Esquema de acceso múltiple distribuido [15]

Una de las ventajas del acceso múltiple distribuido debido a que la transmisión se distribuye en mayor ancho de banda es el incremento de diversidad en frecuencia.

3.4. Desarrollo y arquitectura de Sistemas LTE

El desarrollo de los sistemas LTE se inició a finales de 2004 con el objetivo principal de proporcionar una nueva tecnología de acceso radio basada en conmutación de paquetes, con las correspondientes mejoras en cuanto a rendimiento y latencia.

Los requerimientos también se establecieron en la evolución de la arquitectura de red, dando lugar a una nueva arquitectura de acceso radio con un solo tipo de nodo, el *eNodeB*, así como una nueva arquitectura del core de red, el *Evolved Packet Core* (EPC).

El EPC junto con la arquitectura RAN forman lo que se conoce como *Evolved Packet System* (EPS).

La arquitectura RAN se encarga de todas las funcionalidades radio de la red, como puede ser el *scheduling*, protocolos de retransmisión, códigos y esquemas de transmisión multiantena.

El EPC, sin embargo, no se encarga de funciones relacionadas con el acceso radio pero se encarga de otras que son necesarias para el desarrollo de la red como la autenticación o la preparación de las conexiones *end-to-end*.

3.4.1. Core Network

El EPC es una radical evolución del core de red GSM/GPRS usado en GSM y WCDMA/HSPA; EPC soporta accesos, ya que no soporta acceso al dominio de conmutación de circuitos, sólo conmutación de paquetes.

El *Mobility Management Entity* (MME) es el nodo del plano de control del EPC. Sus funciones incluyen la conexión y el manejo de portadoras en un dispositivo y el uso del IDLE para activar transiciones. Las distintas funcionalidades operativas que se producen entre el EPC y el dispositivo móvil se le conoce como *non-access stratum* (NAS), para diferenciarlo del que se produce entre el dispositivo y la RAN, al que se le denomina *access stratum* (AS).

El *Serving Gateway* (S-GW) es el nodo del plano de usuario que conecta el EPC al LTE RAN y es fundamental cuando los dispositivos se mueven entre *eNodeBs* u otras tecnologías 3GPP, ya que recopila información y estadísticas necesarias.

El *Packet Data Network Gateway* (PDN Gateway, P-GW) conecta el EPC a Internet. Se encarga de la asignación de la dirección IP a un dispositivo específico y de la aplicación de QoS (calidad de servicio) en base a las reglas controladas por el PCRF (*policy and charging rules function*).

El EPC también cuenta con el nodo *home subscriber service* (HSS), que contiene información de suscripción.

En actuales implementaciones, algunos de ellos se combinan entre sí.

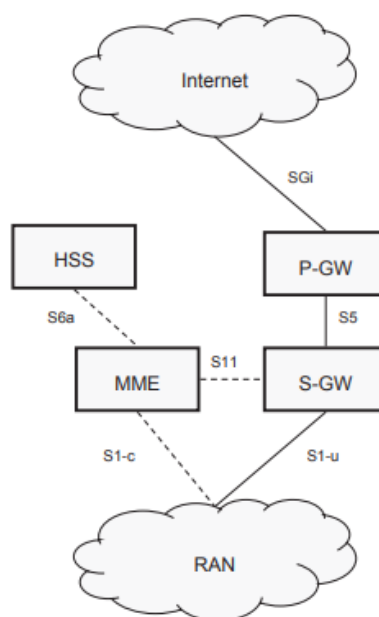


Figura 7: Arquitectura del Core Network [7]

3.5. Red de acceso radio

El LTE RAN usa una arquitectura con un único tipo de nodo, el **eNodeB**, y se encarga de todas las funciones radio en una o varias células. Se trata de un nodo lógico y no tiene una implementación física.

Una implementación común de un eNodeB se basa en un esquema de 3 sectores, donde una estación base soporta transmisiones de 3 células, aunque no es la única implementación.

El eNodeB se conecta al EPC por la interfaz S1, en concreto a la S-GW a través de *S1 user-plane*, S1-u y al MME a través del *S1 control-plane*. Un eNodeB puede ser conectado a múltiples MMEs/S-GWs para repartos de carga y redundancia.

La interfaz X2, que conectada los eNodeBs entre sí, se usa principalmente para soportar movilidad, permitiendo bajas pérdidas por movilidad entre células vecinas por medio de reenvío de paquetes.

La interfaz entre el eNodeB y el dispositivo móvil se conoce como *Uu* interfaz.

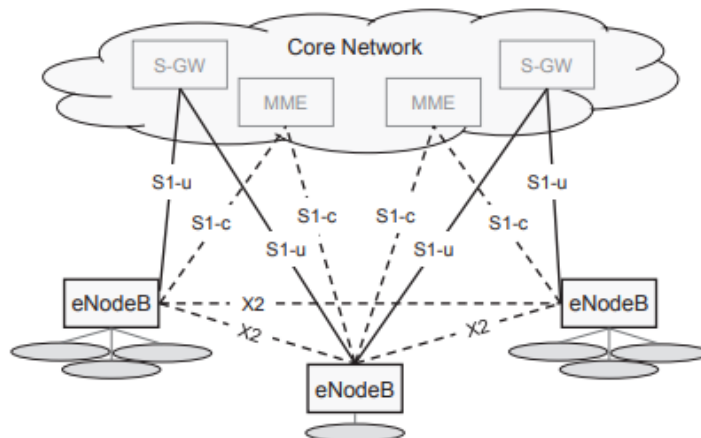


Figura 8: Arquitectura de red de acceso radio [7]

3.6. Estructura y características del acceso radio en LTE

Las tramas LTE tienen una estructura determinada tanto en el dominio del tiempo como de la frecuencia y se lleva a cabo una asignación dinámica de recursos tiempo-frecuencia entre usuarios. Además, OFDM permite la flexibilidad para la planificación o adaptación de las transmisiones.

3.6.1. Estructura en el dominio de la frecuencia [7]

Las especificaciones permiten portadoras con diferentes anchos de banda, y varían desde el mínimo ancho de banda posible $W=1,4$ MHz, equivalente a usar 72 subportadoras, hasta el máximo $W=20$ MHz, equivalente a 1200 subportadoras. No obstante, a partir de la *release* 10, se pueden alcanzar anchos de banda de transmisión de hasta 100 MHz, con el uso de hasta 5 portadoras de diferentes anchos de banda. A esta técnica se le conoce como *Carrier Aggregation*.

La implementación de un transmisor o un receptor se basa en la FFT, por su implementación de baja complejidad y alta eficiencia, y el tamaño correspondiente a la FFT es $N_{FFT}=2048$ para una portadora con ancho de banda $B=20$ MHz.

La frecuencia de muestreo $F_s=\Delta f N_{FFT}$, es $F_s=30,72$ MHz si $N_{FFT}=2048$, donde $\Delta f=15$ kHz es el valor de la separación entre subportadoras. La selección de esta separación entre subportadoras se ha hecho en base a un equilibrio entre el prefijo cíclico contra la sensibilidad de la propagación *Doppler* y otros tipos de errores de frecuencia.

Además de este espacio de 15 kHz entre subportadoras, también se define para LTE un espacio reducido de 7,5 kHz, con el doble de tiempo de símbolo OFDM. Se introdujo este espacio reducido para dedicarse específicamente a *multicast* basada en MBSFN y *broadcast*.

3.6.2. Estructura en el dominio del tiempo [7]

En el dominio del tiempo, las transmisiones LTE están organizadas en tramas (*frames*) de 10 ms. de longitud, cada una de las cuáles se divide en 10 subtramas (*subframes*) de 1 ms. de longitud. Cada subtrama contiene 2 *slots* de longitud 0,5 ms., y cada uno contiene 6 o 7 símbolos OFDM incluyendo el prefijo cíclico (CP), dependiendo de si se trata de prefijo normal (7 símbolos) o extendido (6 símbolos).

Se define una unidad básica temporal $T_s=1/(15000*2048)$ s. como referencia temporal, del que los diferentes intervalos de tiempo definidos anteriormente son múltiplos.

Esta referencia T_s se basa en el tiempo de FFT de un transmisor o receptor implementado con una FFT de tamaño 2048 y permite definir los siguientes intervalos temporales:

$T_{\text{frame}}=307200 \cdot T_s$, $T_{\text{subframe}}=30720 \cdot T_s$, $T_{\text{slot}}=15360 \cdot T_s$ para duración de trama, subtrama y slot respectivamente.

A un nivel superior, cada trama se identifica por medio del SFN (*System Frame Number*) [16]. Este período es importante, ya que en cualquier comunicación, uno de los requerimientos más importantes es que el emisor y el receptor deben estar sincronizados en el tiempo, es decir, los relojes de transmisor y receptor se tienen que sincronizar previamente al establecimiento de la comunicación. En consecuencia a esto, el UE y el eNodeB necesitan mantener la sincronización en el SFN y número de subtrama durante todo el proceso de comunicación.

Además, el reloj en las señales LTE cuentan con 2 intervalos diferenciados, uno que marca cada 10 ms y el otro que marca cada 1 ms.

En el intervalo de 10 ms, los valores de SFN tienen un rango de 0 a 1023 (número de trama del sistema) y en el intervalo de 1 ms tienen un rango de 0 a 9 (número de subtrama). Cuando se alcanza el valor máximo de subtrama, se vuelve de 9 a 0 y el número de SFN aumenta en 1. Así mismo, cuando el valor de SFN alcanza el máximo (1023), éste vuelve a 0.

Por tanto, el período SFN es de 1024, por lo que se repite cada 1024 tramas o equivalentemente 10.24 s. y se utiliza para definir ciclos de transmisión con un período más largo que una trama.

No obstante, LTE ha evolucionado para intentar cubrir las necesidades relacionadas con IoT, lo que ha derivado en que este período de tiempo máximo no fuera lo suficientemente grande como para llevar a cabo la programación de todos los parámetros relacionados con el tiempo, por lo que ha dado lugar a la introducción de otro intervalo de tiempo llamado HFN (*Hyper Frame Number o Hyper SFN*). Constituye el siguiente nivel a SFN y el rango también oscila entre 0 y 1023.

El espacio entre subportadoras de 15 kHz se corresponde con un tiempo de símbolo útil $T_u=2048 \cdot T_s$, equivalente a 66,7 μ s.

En la siguiente figura se muestra la estructura de una señal LTE en el dominio del tiempo:

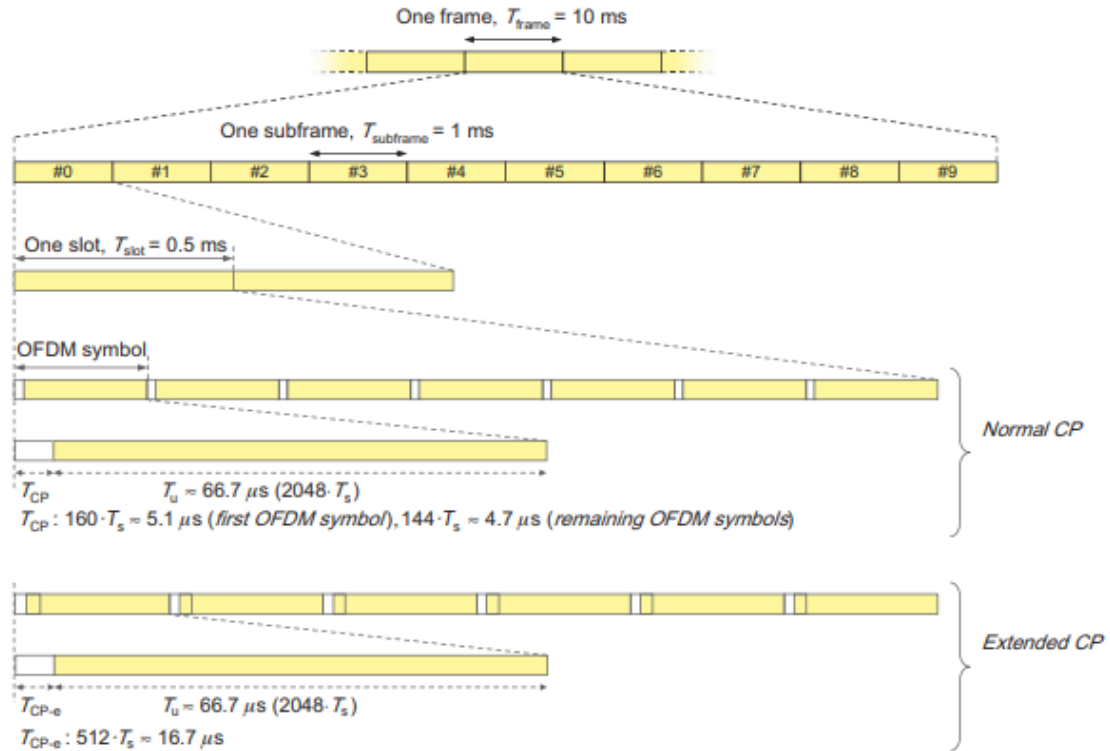


Figura 9: Estructura de LTE en el dominio del tiempo [7]

3.6.3. Recursos físicos [7]

Se trata de una combinación de asignaciones de tiempo y frecuencia. Un *resource element* (RE) consiste en una subportadora modulada con M niveles (M=4, 16, 64 dependiendo de si es QPSK, 16 QAM o 64 QAM) en un símbolo OFDM, y es el recurso físico más pequeño en LTE; a su vez los *resource elements* se agrupan en *resource blocks* (RB).

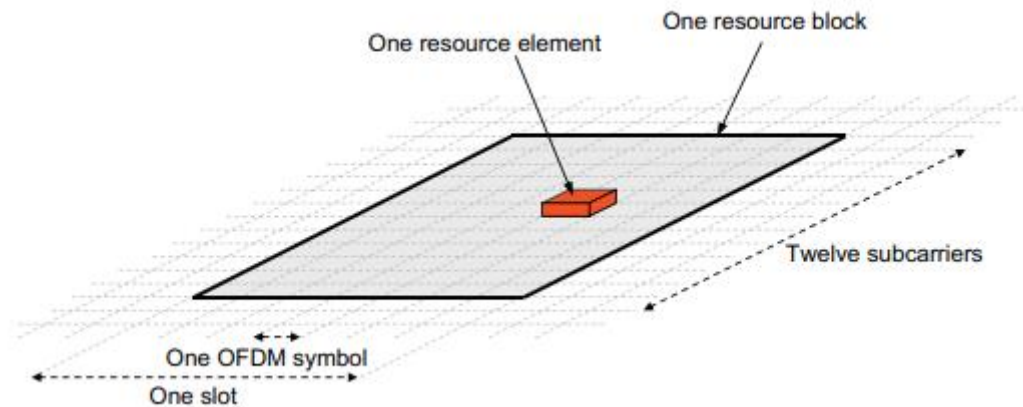


Figura 10: Recurso básico tiempo-frecuencia [7]

Cada *resource block* consiste en 12 subportadoras consecutivas en un intervalo de 0,5 ms (T_{slot}), con un ancho de banda de 180 kHz ($180 \text{ kHz} = 12\Delta_f$), y dependiendo del prefijo cíclico (CP) que se use, cada RB tiene un determinado número de RE: si es CP normal, $1 \text{ RB} = 7 \times 12 = 84 \text{ RE}$ y si es CP extendido, $1 \text{ RB} = 6 \times 12 = 72 \text{ RE}$.

Como una característica básica de LTE está la asignación dinámica de recursos, de forma que un *scheduler* (planificador) asigna recursos tiempo-frecuencia a los usuarios.

A través del *scheduling*, es posible aplicar la diversidad multiusuario en el dominio del tiempo, en la que se tiene en cuenta el estado del canal del usuario para asignar el canal con mejor calidad a éste y así aprovechar al máximo los recursos. En el dominio de la frecuencia también se puede explotar y tiene en cuenta los desvanecimientos selectivos en frecuencia.

El canal se percibe de manera diferente por cada usuario, por lo que se necesita una buena selección de usuario a transmitir para que se produzca una transmisión eficiente.

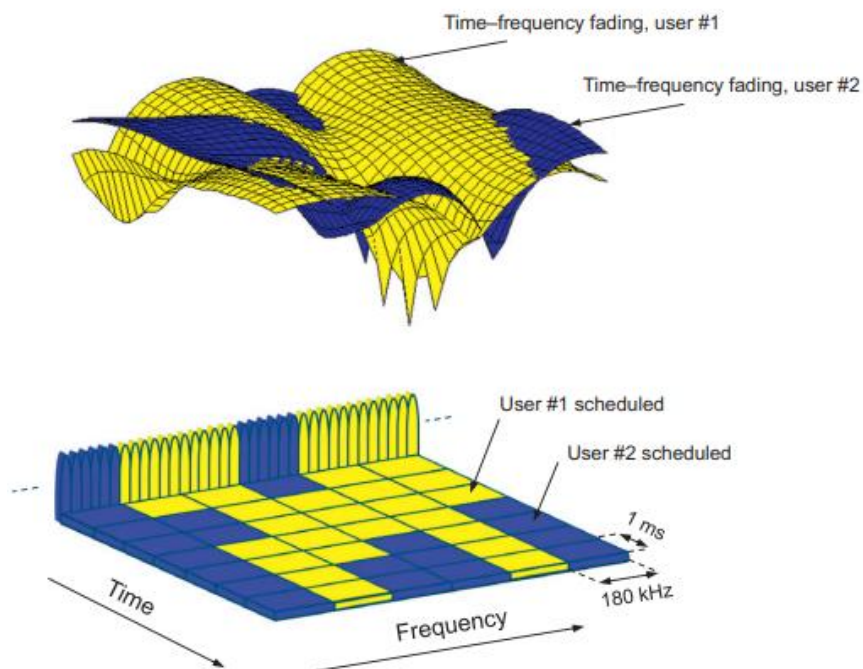


Figura 11: Planificación en el enlace descendente (DL) [7]

Teniendo en cuenta los parámetros y los cálculos de este apartado, se puede construir la tabla siguiente:

Ancho de banda nominal (W en MHz)	1,4	3	5	10	15	20
Tamaño FFT (N_{FFT})	128	256	512	1024	1536	2048
Espacio Subportadoras (Δ_f)	15 kHz					
Tiempo FFT (t)	66,7 μ s					
Número de RB (UL o DL)	6	15	25	50	75	100
Número de subportadoras	72	180	300	600	900	1200
Frecuencia de muestreo (F_s en Mmuestras/s)	1,92	3,84	7,68	15,36	23,04	30,72

Tabla 5: Parámetros LTE

3.7. Cell Search como técnica de acceso al medio y canales en LTE [7]

Antes de que un dispositivo LTE pueda comunicarse y acceder a una red LTE, éste debe encontrar y lograr la sincronización con una celda de la red y además, debe recibir y decodificar la información necesaria para la correcta comunicación dentro de la celda.

El primer paso se le conoce como *Cell Search* y su objetivo es la sincronización con una celda, la determinación del instante en el que comienza una trama en el *Downlink* y el conocimiento de la identidad de capa física de la celda (PCI).

Para el proceso de *Cell Search*, dos señales son transmitidas en cada componente de la portadora del enlace descendente, la señal de sincronización primaria (PSS son sus siglas en inglés) y la señal de sincronización secundaria (SSS). Aunque tienen la misma estructura, las posiciones de las señales de sincronización dentro de la trama varían dependiendo de si se usa FDD o TDD:

- En caso de que se use FDD, la PSS se transmite en el último símbolo del primer slot de las subtramas 0 y 5, mientras que la SSS se transmite en el penúltimo símbolo del mismo slot que el PSS, por lo que es justo el anterior.

- En caso de que se use TDD, la PSS se transmite en el tercer símbolo de las subtramas 1 y 6, mientras que la SSS se transmite en el último símbolo de las subtramas 0 y 5.

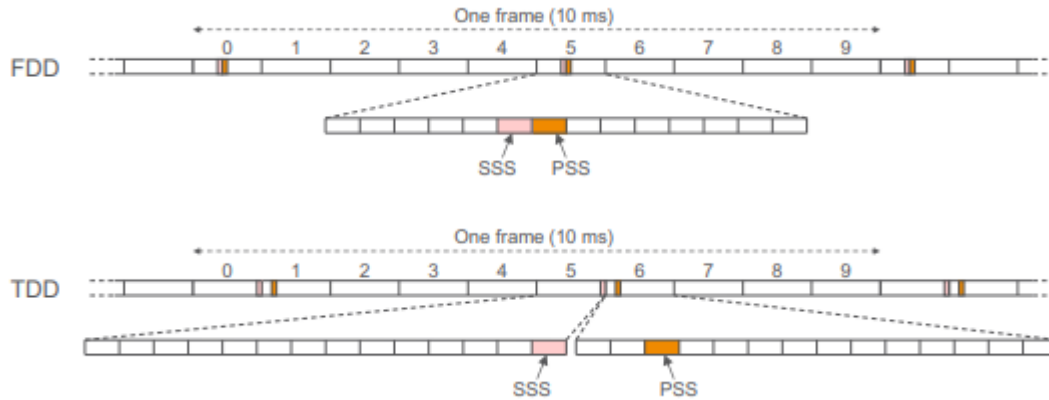


Figura 12: Posiciones de PSS y SSS en el dominio del tiempo en FDD y TDD [7]

La diferencia en la estructura del dominio de tiempo PSS/SSS entre FDD y TDD permite que un dispositivo detecte el modo de operación de la celda si no se conoce previamente.

Para una misma celda, las dos señales PSS de una trama son idénticas y pueden tomar los valores 0, 1 y 2, cuyo valor se denota como *Physical Layer Identity* (N_{ID}). Cuando el usuario ha detectado e identificado la señal PSS de una celda, tiene acceso a información como la *Physical Layer Identity* o la identificación de 5 ms. del tiempo de trama de una celda, por lo que teniendo en cuenta la posición relativa entre la señal PSS y SSS, también puede determinar la posición de la SSS.

En cuanto a las señales SSS, estas no son iguales y cada una puede ser identificada con un valor entre 0 y 167. Una vez conocida una de las señales SSS de una celda, no es necesario que se conozcan las dos, se puede detectar el momento en el que la trama comienza en el DL. Además, también permite determinar el valor del *Physical Layer Cell Identity Group Number* (N_{ID1}).

Con los valores de N_{ID1} y N_{ID2} se puede determinar la identidad física de la celda (PCI), que se calcula como: $N_{IDCELL} = 3N_{ID1} + N_{ID2}$.

Cuando se conoce el valor del PCI, el UE también puede identificar y localizar las señales de referencia que se distribuyen.

Con la información que se obtiene a través de las señales PSS y SSS, el terminal se sincroniza con la celda y conoce acerca del instante en el que se transmite la trama de esa celda en concreto. Posteriormente, el terminal tiene que conseguir la información sobre la celda, que el sistema emite repetidamente desde la red, y así operar correctamente en la celda en la que se ha sincronizado para completar el proceso de *Cell search*.

Más adelante, se va a explicar más detalladamente este proceso a través de un ejemplo de Matlab, pero para entenderlo correctamente, debemos conocer el proceso de transporte de información sobre la interfaz radio de LTE, como la información relativa al campo de control y sincronización, necesaria para la correcta comunicación dentro de la celda

Para el transporte de información, se utilizan distintos tipos de canales:

- Canales físicos
- Canales de transporte
- Canales lógicos

Los de mayor contribución al proceso de *Cell search* son los canales físicos, por lo que son los que se van a explicar con mayor profundidad.

3.7.1. Canales Físicos

Los canales físicos son los encargados de la transmisión de los datos de usuario y los mensajes de control. Hay variación entre los de los enlaces descendente (DL) y ascendente (UL), pero nos vamos a centrar en los de DL, ya que en el trabajo nos centramos en las señales del enlace descendente. [7]

- **Physical Control Format Indicator Channel (PCFICH):**

El PCFICH indica el tamaño del campo de control haciendo alusión al número de símbolos OFDM. Es clave que la información del PCFICH se decodifique correctamente, ya que en caso de que la decodificación sea errónea, el terminal (UE) no sabrá decodificar el resto de canales de control ni tendrá una referencia del comienzo del campo de datos dentro de una subtrama.

El PCFICH está formado por 2 bits de información; éstos corresponden a los tres posibles tamaños de la parte de control de 1, 2 o 3 símbolos. Los bits están codificados, modulados en QPSK y se mapean sobre 16 RE del primer símbolo OFDM de cada

subtrama. El hecho de mapearse sobre el primer símbolo es porque hasta que el PCFICH no se decodifica, el campo de control es desconocido.

- **Physical Hybrid ARQ Indicator Channel (PHICH):**

Indica el estado del Hybrid ARQ (señales HARQ de ACK o NACK) para la correcta recepción de un bloque de transporte.

Se mapea sobre el primer símbolo OFDM del campo de control y 3 grupos de RE, separados un tercio del ancho de banda nominal de transmisión aproximadamente.

- **Physical Downlink Control Channel (PDCCH):**

Este canal físico transporta la información sobre la programación (*scheduling*) a través del *Downlink Control Information* (DCI), que contiene la siguiente información para el UE:

- Programación de recursos para el DL.
- Instrucciones sobre el control de potencia en el UL.
- Concesión de recursos para el UL
- Indicaciones sobre información del sistema.

El formato DCI varía según su tamaño y su uso, y pueden ser de tipo 1, 1A, 1B, 1C, 1D, 2, 2A, 2B.

El PDCCH está en la región de control pero su mapeo sobre los RE se controla a través del *Control Channel Elements* (CCE), cuya estructura está formada por 36 RE y el número de CCEs varía según el tamaño del DCI y las condiciones del canal.

- **Physical Broadcast Channel (PBCH):**

En LTE, la información del sistema se transporta por dos mecanismos diferentes sobre dos canales de transporte también diferentes:

- Master Block Information (MIB): representa la información estática del sistema.
- System Information Blocks (SIBs): representa la información dinámica del sistema.

El MIB es el único que se transporta por el PBCH. Esta mapeado sobre las 72 subportadoras centrales o equivalentemente, los 6 RB centrales independientemente del ancho de banda, que se repite cada 40 ms.

La parte dinámica (SIBs) se va a explicar en el siguiente punto.

- **Physical Downlink Shared Channel (PDSCH):**

LTE define hasta 16 tipos de SIBs en función de la información que contengan y los de orden menor son los que más se repiten al contar con información más esencial.

Nos vamos a centrar en el SIB 1, que es transportado por el PDSCH y se emite en la subtrama 5 cada 80 ms.

La información del SIB 1 es:

- Información sobre si el usuario puede acceder a la red o no.
- Programación del resto de SIBs (scheduling).

3.7.2. Canales de Transporte

Los canales de transporte se encargan de transportar la información desde la capa física hasta la capa MAC (*Media Access Control*). De los cuatro canales de transporte en el DL, los únicos relevantes para este trabajo son los canales BCH y DL-SCH:

- **Broadcast Channel (BCH):**

Transporta la información del canal físico PBCH (MIB).

- **Downlink Shared Channel (DL-SCH):**

Transporta la información del canal físico PDSCH (SIBs).

- **Paging Channel (PCH):**

Se utiliza para proporcionar comunicación al canal lógico PCCH.

- **Multicast Channel (MCH):**

Se utiliza para transportar información del canal lógico MCCH.

3.7.3. Canales Lógicos

Estos canales se dividen entre canales de control y canales de trabajo. Es cierto que los canales de control son los más relevantes para este trabajo pero sólo el BCCH es el verdaderamente importante. Los canales son los siguientes:

- **Broadcast Control Channel (BCCH):**

El eNodeB utiliza este canal lógico para emitir repetidamente la información del sistema (MIB y SIBs), y por tanto, esta información se transporta por el canal BCH en el caso del MIB y por el DL-SCH en el caso de los SIBs.

- **Paging Control Channel (PCCH):**

Se utiliza para clasificar información en el proceso de búsqueda de una unidad de red.

- **Common Control Channel (CCCH):**

Este canal es utilizado para permitir el Access Random Information como puede ser el establecimiento de una conexión.

- **Multicast Control Channel (MCCH):**

Se utiliza para información requerida en un receptor multicast.

- **Dedicated Control Channel (DCCH):**

Este canal se usa para llevar información de control específica de un usuario como pueden ser las acciones de control de potencia.

4. SOFTWARE DEFINED RADIO (SDR)

En cuanto al despliegue y desarrollo de la red radio, la implantación de LTE permite llevar a la práctica el uso de algunos conceptos relacionados con **Software Defined Radio (SDR)** y que permite la modificación de ciertas características y funcionalidades de la red sin necesidad de modificar el hardware. Además, con la evolución en términos de capacidad de computación, se permite plantear la separación de funcionalidades radio y de procesamiento digital, dando lugar a una distribución de unidades radio por cada una de los emplazamientos de todo el territorio mientras que la parte digital podría centralizarse en un centro de gestión que soporte distintos emplazamientos.

Lo que se pretende con estas funcionalidades es reducir el coste de despliegue y operación de la red y compensar así los costes cada vez mayores por nivel de tráfico.

4.1. ¿Qué es SDR?

Todas las organizaciones profesionales proponen definir un marco de referencia común en cuanto a términos y definiciones que permitan una fácil comunicación entre profesionales que trabajan en áreas de investigación o desarrollo de productos. En el caso de las comunicaciones inalámbricas y SDR también es así. El Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) Grupo de Trabajo P1900.1 creó unas definiciones comunes para garantizar que los profesionales usaran la misma terminología como puede ser Radio, Nodo de radio, Software, Capa Física, SDR, Transmisor, Receptor, Forma de onda, etc.

Un sistema SDR es un complejo dispositivo que permite ejecutar diversas acciones simultáneamente para permitir una transmisión y recepción de los datos perfecta, usando reconfiguraciones y reprogramaciones a través de software. En general, un sistema de comunicaciones digital consiste en un conjunto de bloques que captan algún tipo de información procedente de diversas fuentes y la transmiten a través del aire para que un receptor pueda recibir esa información, y procese y decodifique la información para conseguir una versión reconstruida de la señal original. [17]

El creador de SDR, Joseph Mitola, definió los términos básicos relativos a esta tecnología a principios de los 90 y formuló un prototipo ideal de SDR, cuyos componentes físicos eran una antena y un convertidor analógico-digital (ADC) en el receptor y así mismo, el transmisor contaba con un convertidor digital-analógico (DAC) y una antena transmisora. El resto de funciones estaban soportadas por procesos reprogramables.

Como la idea inicial todavía no se había conseguido en los 90, el término SDR se empezó a utilizar para describir a un dispositivo que usa software pero que todavía cuenta con bastantes componentes hardware.

4.2. Inicios de SDR

SDR ha evolucionado, como muchas tecnologías, de entornos militares a entornos civiles. La primera operación con SDR, conocida como *Speakeasy* fue desarrollada por la United States Navy entre 1991 y 1995. Desafortunadamente, la aplicación no se pudo utilizar con otro hardware que no fuera esa para el cual se había hecho, además, de que el dispositivo ocupaba totalmente la parte trasera de un vehículo de transporte.

Su hermano menor, el *Speakeasy II*, tuvo mayor éxito debido principalmente a los avances en la electrónica, los circuitos de comunicaciones inalámbricas y las técnicas de programación modulares.

Desde el primer momento hasta ahora, ha habido un largo camino, en el que ahora tanto el hardware como el software de SDR están disponibles a precios relativamente bajos.

4.3. Hardware SDR

En este apartado se va a analizar la diferencia entre el hardware de receptores tradicionales y receptores SDR.

- **Receptores tradicionales**

Un receptor tradicional realiza varias operaciones:

- Ajuste de frecuencia portadora para selección de la señal deseada.
- Filtro para separación de otras señales recibidas.
- Amplificación para compensar las pérdidas transmitidas
- Demodulación

La mayoría de los receptores tradicionales hacen uso de esquemas de receptores heterodinos. En la siguiente figura se muestra el esquema típico de un receptor superheterodino:

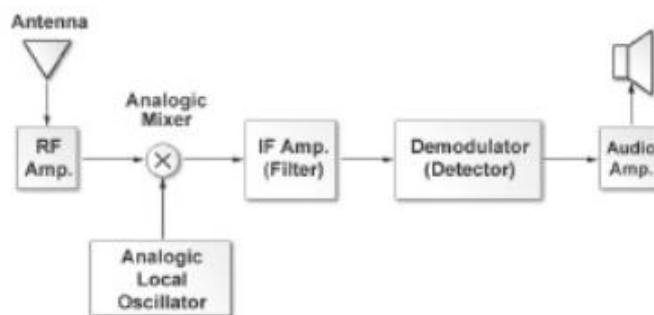


Figura 13: Diagrama de bloques de receptor superheterodino [5]

En este esquema, después de que la señal sea captada por la antena, es amplificada por una etapa RF que opera sólo en las frecuencias del rango deseado. Posteriormente, la señal pasa por un mezclador que recibe la contribución de un oscilador local analógico por otra de sus entradas. La frecuencia del oscilador local se establece mediante el control de sintonización radio, y ahora, el mezclador es capaz de traducir la señal a la frecuencia intermedia (IF).

El oscilador típicamente establece su frecuencia en base al valor de la diferencia entre la frecuencia de la señal deseada y la IF.

El siguiente bloque corresponde a un filtro paso banda que atenúa toda señal excepto una parte del espectro. El ancho de banda de esta etapa limita el ancho de banda de la señal que está siendo recibida.

En la última etapa, el demodulador recupera la señal modulada de la salida del amplificador de IF. En el ejemplo de la figura, la señal demodulada se lleva a un amplificador de audio que se conecta a un altavoz. No obstante, los procesos de señal de esta etapa dependen del propósito para el que esté hecho el receptor.

- **Receptor SDR**

La siguiente figura muestra el diagrama de bloques típicos de un receptor SDR.

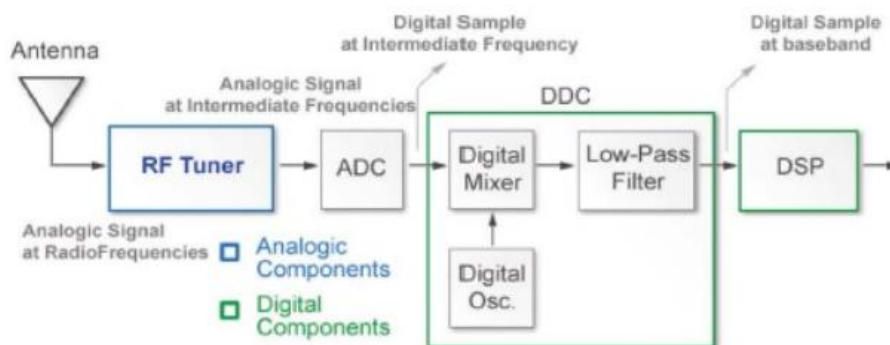


Figura 14: Diagrama de bloques de receptor SDR [5]

Nótese que en el primer bloque el sintonizador de RF convierte la señal analógica a *IF*, por lo que realiza la misma operación que los tres primeros bloques del receptor superheterodino.

Posterior a esto, la señal *IF* se lleva a un convertidor ADC, y en su salida se obtienen muestras digitales. Las muestras pasan al siguiente bloque denominado *Digital Down Converter (DDC)*, que constituye la parte más importante de un sistema SDR. Está compuesto por 3 componentes principales:

- Mezclador digital
- Oscilador local digital
- Filtro paso bajo con respuesta al impulso finita (*FIR*)

Cada uno de estos componentes realizan operaciones similares a sus homólogos analógicos, es decir, el mezclador digital y el oscilador local llevan a cabo un cambio de *IF* a banda base, mientras que el filtro paso bajo *FIR* limita el ancho de banda de la señal final. Para la implementación de cada uno de estos bloques, el DDC incluye un gran número de registros para conseguir unos resultados satisfactorios.

Finalmente, las muestras en banda base pasan al bloque de *Digital Signal Processing (DSP)*, donde se implementan procesos como la demodulación y la decodificación, entre otros.

Este bloque de procesado (PDS, con sus siglas en castellano) puede ser implementado en un FPGA si el sistema está adaptado a una aplicación específica. Sin embargo, las etapas de DSP son normalmente implementadas en un software específico que añade versatilidad a la solución. [5]

En cuanto a equipos comerciales SDR, hay diversos equipos con características y precios diferentes. En la siguiente figura, se puede ver una comparación de los principales productos disponibles en el mercado:

SDR	Tune Low (MHz)	Tune Max (MHz)	RX Bandwidth (MHz)	ADC Resolution (Bits)	Transmit? (Yes/No)	Price (\$USD)
RTL-SDR (R820T)	24	1766	3.2 / 2.56 Stable	8	No	~20
Funcube Pro+	0.15 410	260 2050	0.192	16	No	~200
Airspy	24	1800	10	12	No	199
SDRPlay	0.1	2000	8	12	No	149
HackRF	30	6000	20	8	Yes	299
BladeRF	300	3800	40	12	Yes	400 & 650
USRP 1	DC	6000	64	12	Yes	700

Figura 15: Comparación de equipos SDR [1]

Como podemos apreciar en la figura, dependiendo de características como el rango de frecuencias, el ancho de banda o el hecho de que pueda actuar de transmisor o no, se elige un equipo u otro.

Además de los equipos mencionados en la figura, hay otros modelos o variantes de algunos equipos cuyas características varían como por ejemplo, dentro de *SDRPlay*, dos de los modelos que no encontramos son el modelo *SDRPlay RSP2* y *SDRPlay RSP1A* cuyas diferencias radican básicamente en el rango de frecuencia, la resolución del ADC y consecuentemente, el precio. [2]

Además, en el mercado han ido apareciendo a lo largo del tiempo otros equipos con características variadas como pueden ser *Seedstudio KiwiSDR Kit Software Defined Radio with BeagleBone Green* o *Ubertooth One*, aunque como se indica, la gama es bastante amplia. [3]

4.4. Herramientas Software SDR

Anteriormente, hemos analizado la parte de hardware, por lo que ahora vamos a analizar el software que hay disponible para el análisis de señales de radiofrecuencia. Hay multitud de aplicaciones desarrolladas para grandes empresas y universidad que han dedicado mucho dinero a la investigación. No obstante, también hay software libre que nos permite analizar las señales. Un ejemplo de este software es **SDR-Sharp**:

SDR-Sharp muestra en pantalla señales en tiempo real que han sido adquiridas por el dispositivo SDR. A continuación, se muestra una captura de la interfaz de *SDR-Sharp* en funcionamiento:

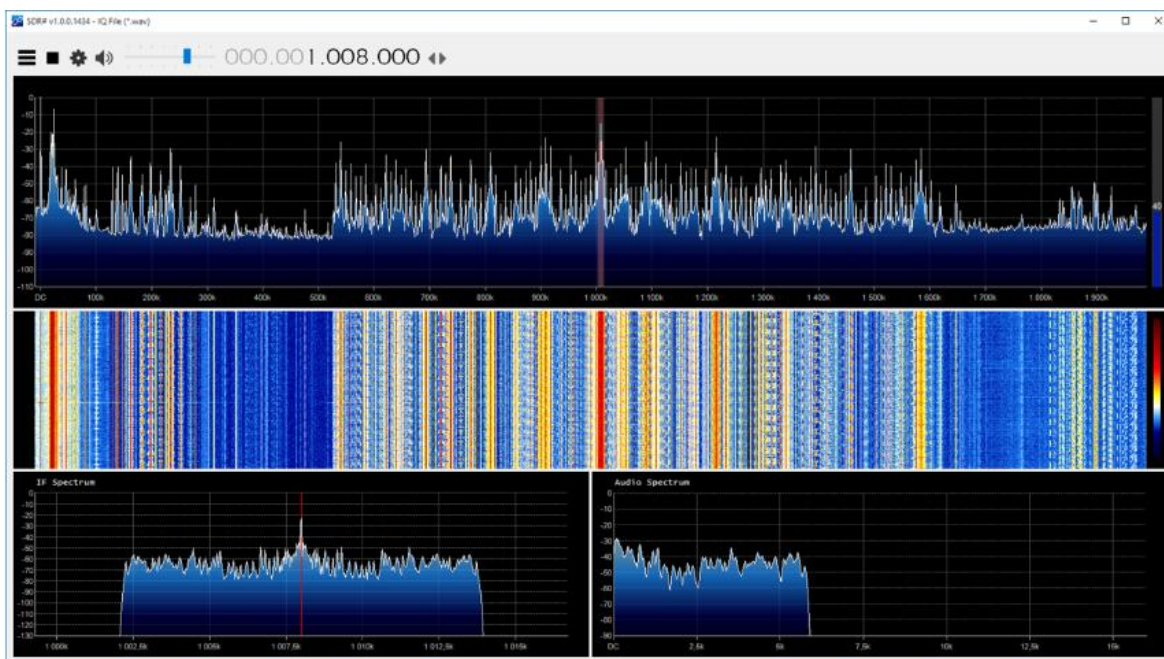


Figura 16: Interfaz de SDR-Sharp [4]

Como podemos ver en la figura, la aplicación muestra en las 2 gráficas superiores el espectro en tiempo real y el comportamiento de la señal en el tiempo. Además, en las 2 gráficas inferiores, se muestra el ancho de banda seleccionado y separa en espectro de IF y en espectro de distribución en frecuencia de la voz demodulada; es más, si tenemos un altavoz conectado al ordenador, podemos oír su contenido acústico.

SDR-Sharp trabaja adecuadamente en ordenadores con *single core* de 2 GHz con al menos 1 GB de RAM. No obstante, hay algunas operaciones que consumen más recursos.

Además, *SDR-Sharp* es una herramienta muy útil cuando es necesario detectar una interferencia. [5]

5. MONTAJE UTILIZADO

BladeRF es una plataforma de radio definida por software diseñada para un uso doméstico y profesional de esta tecnología.

El modelo usado para este proyecto fue la bladeRF x40 (nombrada como bladeRF de aquí en adelante) y es desarrollada por Nuand. Es una plataforma abierta, con un coste en su página web de 420 \$ y está diseñada para permitir a estudiantes y profesionales de radio una inmersión en el mundo de las tecnologías inalámbricas. [18]



Figura 17: Dispositivo BladeRF [18]

La BladeRF es un dispositivo con alimentación por USB que no necesita ningún tipo de conexión adicional para su funcionamiento. Si se necesita hacer procesamiento de host, USB 3.0 Super Speed proporciona un alto rendimiento y una interfaz de baja latencia para la conexión del PC y la antena.

La antena utilizada para recepción ha sido la RoHS 2J301M-300RG174-C20N. [19].



Figura 18: Antena RoHS 2J301M [19]

Se trata de una antena que puede trabajar en un rango de frecuencias en torno a 800 MHz hasta unos 2 GHz.

Tiene una impedancia de 50 Ohms, su polarización es vertical y tiene una ganancia de 2.2dBi máxima. [20].

La respuesta en frecuencia de la antena se muestra en la figura siguiente:

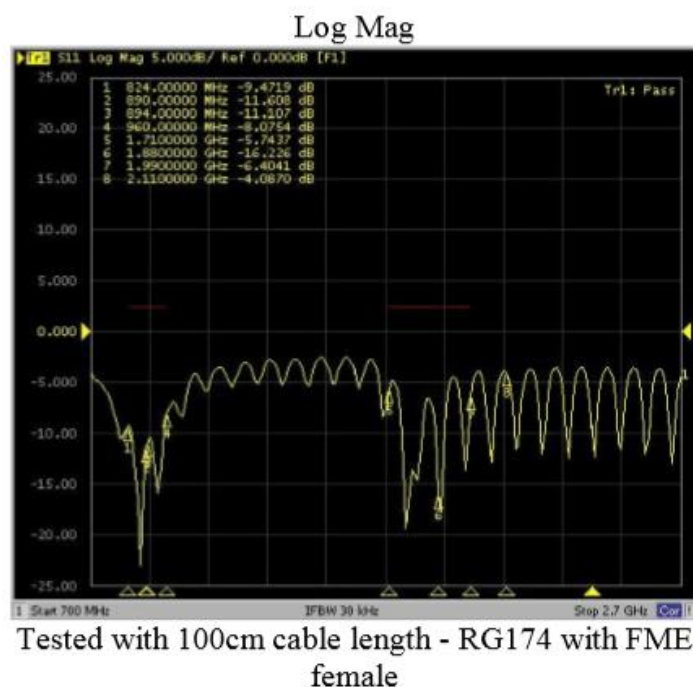


Figura 19: Respuesta en Frecuencia Antena RoHS 2J301M

Con el dispositivo BladeRF se permite tanto la transmisión como la recepción de señales de RF. El rango de frecuencia que abarca el dispositivo va desde 300 MHz hasta 3.8 GHz, que se puede extender hasta las bandas HF y VHF con el XB-200 Transverter Module, y el ancho de banda seleccionable va desde 1.5 MHz hasta 28 MHz. Tiene un conversor AD de 12 *bit* y cuenta con un FPGA 40KLE Cyclone 4. [21].

Un FPGA es un circuito programable que contiene bloques lógicos que nos permiten configurar en el momento a través de un lenguaje apropiado y nos crea una lógica de control, que permite controlar desde funciones sencillas hasta complejos sistemas de control.

Esta tarjeta se fabrica en dos modelos cuya única diferencia es el FPGA con el que cuentan: el modelo x40 tiene un FPGA Cyclone 4E de Altera EP4CE40 y el modelo

x115 contiene el FPGA de la misma familia pero diferente modelo, el Cyclone 4E de Altera EP4CE115. El modelo x115 tiene un FPGA con más recursos, por lo que hace del modelo usado en el proyecto una opción más económica. [22].

En la siguiente figura, se pueden observar las diferencias entre ambos elementos:

Resources	EP4CE6	EP4CE10	EP4CE15	EP4CE22	EP4CE30	EP4CE40	EP4CE55	EP4CE75	EP4CE115
Logic elements (LEs)	6,272	10,320	15,408	22,320	28,848	39,600	55,856	75,408	114,480
Embedded memory (Kbits)	270	414	504	594	594	1,134	2,340	2,745	3,888
Embedded 18 × 18 multipliers	15	23	56	66	66	116	154	200	266
General-purpose PLLs	2	2	4	4	4	4	4	4	4
Global Clock Networks	10	10	20	20	20	20	20	20	20
User I/O Banks	8	8	8	8	8	8	8	8	8
Maximum user I/O (M)	179	179	343	153	532	532	374	426	528

Figura 20: Comparación FPGAs de Altera [22]

5.1. Esquemático de la placa BladeRF

Este dispositivo hardware cuenta con diferentes elementos, y se pueden diferenciar 4 grandes bloques:

- El bloque de adquisición y transmisión de datos que se compone principalmente por el transceiver LMS6002D de Lime Micro.
- El bloque de control y DSP integrado por el FPGA 40KLE Cyclone 4E de Altera (EP4CE40).
- El bloque de comunicaciones *Hardware-PC* integrado por el controlador USB SuperSpeed USB CYUSB301X de Cypress.
- El bloque que genera el reloj, que se compone por un generador de reloj programable Si5338 de Silicon Labs, un oscilador de cristal de temperatura compensada y un conversor DAC. [18].

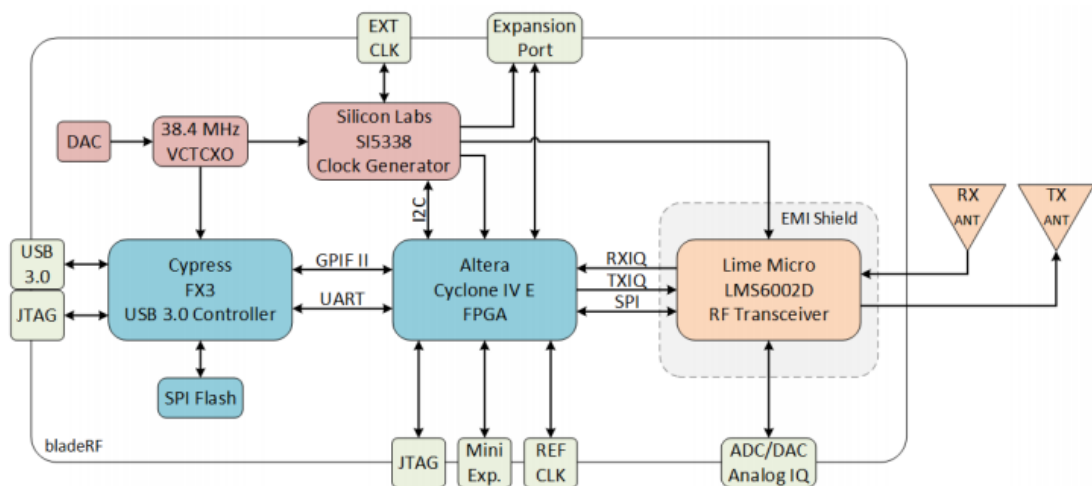


Figura 21: Diagrama de bloques de BladeRF [18]

La placa puede ser alimentada por el USB o por una fuente de continua externa a 5 V DC, ya que cuenta con una conector hembra tipo *jack*.

El *transceiver* integrado en la tarjeta trabaja en el rango de 0.3 a 3.8 GHz, recibiendo o transmitiendo muestras a una frecuencia de muestreo configurable máxima de 40 millones de muestras por segundo. Las muestras son cuantizadas con 12 bits con signo, en el rango de valores [-2048, 2047]. En la figura 21 se muestra un esquema del *transceiver* de la tarjeta.

Para la recepción de datos, hay tres entradas diferentes con un amplificador dedicado LNA. La señal que se adquiere en cada puerto se amplifica en una primera etapa de amplificación de bajo ruido programable (RXLNA) y posteriormente, la señal de radio frecuencia es mezclada con el PLL para convertirse en banda base. Luego, la demodulación continúa con otra etapa de amplificación de ganancia variable (RXVGA1) previa al filtro paso bajo programable (RXLPF), y la señal recibida IQ es amplificada por otro amplificador programable (RXVGA2) para prevenir que el conversor AD se sature.

Cada parte de la señal se convierte al dominio digital en 12 bits y en cada par de flancos de RX_CLK se entrega I y Q.

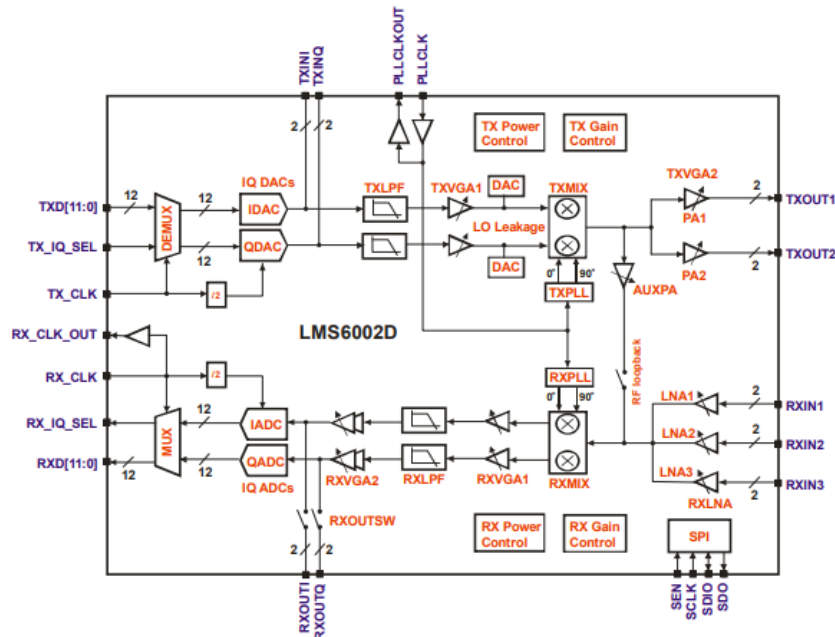


Figura 22: Diagrama de bloques Transceiver Lime Micro [23]

Los datos y una señal de dato válido son entregados al FPGA, y cada dos períodos de reloj, se obtiene una muestra válida. El FPGA proporciona flexibilidad para las configuraciones de usuario a futuro.

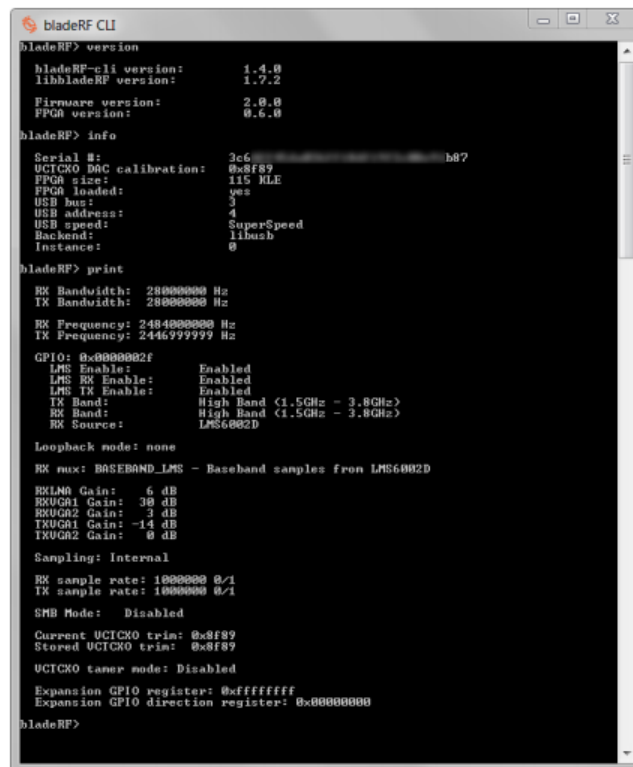
En cuanto al bloque de comunicaciones Hardware-PC, el chip FX3 es un controlador USB de alta velocidad (*SuperSpeed*), cuya interfaz SPI permite cargar el firmware del FX3 desde una memoria flash de inicialización; éste puede ser actualizado desde la página web de Nuand.

El FX3 tiene integradas capas físicas para USB 2.0 y USB 3.1 generación 1.

Además, la placa cuenta con un oscilador VCTCXO de 38,4 MHz que se controla por un DAC. Se trata del reloj de entrada del FX3 y se utiliza como entrada de referencia del chip Si5338, aunque éste último genera otros relojes a utilizar. [18].

5.2. Primeros pasos con BladeRF y LTE Toolbox

Para comenzar a manejar la tarjeta Blade RF, se utiliza la interfaz CLI (Windows y Linux). Se trata de una interfaz entre el ordenador y la tarjeta SDR que permite insertar y modificar todos sus parámetros.



```
bladeRF CLI
bladeRF> version
bladeRF-cii version: 1.4.0
libbladeRF version: 1.7.2
Firmware version: 2.0.0
FPGA version: 0.6.0

bladeRF> info
Serial #: 3c6...b87
UCTCKO DAC calibration: 0x8f89
FPGA size: 115 MLE
FPGA loaded: yes
USB bus: 3
USB address: 4
USB speed: SuperSpeed
Backend: libusb
Instance: 0

bladeRF> print
RX Bandwidth: 20000000 Hz
TX Bandwidth: 20000000 Hz
RX Frequency: 2404000000 Hz
TX Frequency: 2446999999 Hz

GPIO: 0x0000002f
LMS Enable: Enabled
LMS RX Enable: Enabled
LMS TX Enable: Enabled
TX Band: High Band (1.5GHz - 3.8GHz)
RX Band: High Band (1.5GHz - 3.8GHz)
RX Source: LMS6002D

Loopback mode: none
RX mux: BASEBAND_LMS - Baseband samples from LMS6002D
RXLNA Gain: 6 dB
RXUGA1 Gain: 30 dB
RXUGA2 Gain: 3 dB
TXUGA1 Gain: -14 dB
TXUGA2 Gain: 0 dB

Sampling: Internal
RX sample rate: 100000000 0/1
TX sample rate: 100000000 0/1

SMB Mode: Disabled
Current UCTCKO trim: 0x8f89
Stored UCTCKO trim: 0x8f89
UCTCKO taper mode: Disabled
Expansion GPIO register: 0xffffffff
Expansion GPIO direction register: 0x00000000

bladeRF>
```

Figura 23: Captura Línea de comandos CLI BladeRF [24]

A través de CLI (*Command Line Interface*) se puede insertar los parámetros que va a configurar tanto emisor como receptor y también permite enviar y recibir archivos en formato *.bin* o *.csv* para su posterior tratamiento de esos datos. En cuanto a los parámetros que se pueden configurar están la ganancia (tx y rx), ancho de banda, frecuencia de muestreo, frecuencia central, *loopback*, etc.

El comando *print* permite visualizar las configuraciones de los diferentes parámetros de la placa que tenga en ese momento y el comando *set* permite ajustar los diferentes parámetros de la placa.

Además, la API correspondiente disponible en Nuand para implementar el código basado en *System Objects* de Matlab se ha instalado en el directorio de BladeRF, y a partir de este momento, podemos empezar a usar la tarjeta a través de Matlab, que es la funcionalidad que nos interesa para este trabajo.

Matlab proporciona una toolbox para LTE, la **LTE Toolbox**, cuyo contenido ofrece funciones para el diseño, simulación y verificación de sistema LTE y LTE-Advanced.

A través de la toolbox, se puede configurar, simular, medir y analizar sistemas de comunicaciones basados en este estándar.

En este trabajo, combinando el uso del ejemplo "Cell Search, MIB and SIB1 Recovery" de Mathworks, función propia de LTE Toolbox con el hardware apropiado para la captura de señales LTE (tarjeta BladeRF), podemos configurar un receptor LTE para el tratamiento de la señal.

6. PUESTA EN MARCHA DEL MONTAJE

Una vez montado el dispositivo para empezar a trabajar con la tarjeta, en este capítulo se va a describir el código necesario para procesar la señal capturada utilizando la *toolbox* LTE de Matlab. Además, también se describe en 6.1., los parámetros necesarios que debemos asignar a la tarjeta para que empiece a capturar.

6.1. Captura de señal LTE

Este código es importante ya que nos va a permitir definir el marco de parámetros para un correcto procesamiento de la señal y, en consecuencia, una correcta toma de muestras en tiempo real; no obstante, depende de los recursos computacionales de los que dispongamos.

En esta parte es donde entra en juego el dispositivo BladeRF, que constituye el hardware principal del montaje incluyendo la antena.

Una vez montada la parte hardware, como explicábamos en el apartado anterior, debemos integrar en Matlab la API correspondiente disponible en Nuand para implementar el código basado en *System Objects* de Matlab.

Con este código implementado, se puede proceder al proceso de captura de señal LTE y a continuación se explican los parámetros de recepción más importantes a tener en cuenta del código, sobre todo aquellos que dependen de la señal a capturar y que por tanto representan la parte dinámica de parámetros. Estos parámetros son:

- **Frecuencia de muestreo (F_s):**

Teniendo en cuenta la tabla 5, se observa que el mayor ancho de banda nominal es de 20 MHz. Si la frecuencia de muestreo se define como $F_s = \Delta f N_{FFT}$, por tanto $F_s = 30,72$ Mmuestras/s si $N_{FFT} = 2048$ ($N_{FFT} = 2048$ para $W = 20$ MHz).

Si nos encontramos en la banda 3 de LTE podemos tener en cuenta estos parámetros. Sin embargo, si la banda de la que recibimos es la banda 20 cuyo ancho de banda nominal es de 5 MHz, los parámetros varían de la siguiente forma:

$F_s = 7,68$ MMuestras/s si $N_{FFT} = 512$ ($N_{FFT} = 512$ para $W = 5$ MHz).

- **Número de muestras (NSamples):**

El MIB corresponde a un bloque de transporte BCH y esta función se va a analizar más adelante. El BCH *Transmission Time Interval* (TTI) es el tiempo necesario para la transmisión de un bloque de transporte y es 40 ms o 4 tramas.

El BCH se transmite en 4 partes, que se mapean en la subtrama 0; por tanto, para asegurar que esta subtrama se recibe, la señal debe contener mínimo 11 tramas. Si las 4 partes del TTI son necesarias por condiciones adversas de la señal, se necesitarían al menos 41 subtramas.

En cuanto al SIB1, se transmite en la subtrama 5 de cada trama par (con 4 RVs diferentes transmitidas de forma consecutiva), lo que nos da un período global de 8 tramas (80 ms). Si sólo se necesita asegurar la recepción de un único RV de SIB1, sólo serían necesarias 21 subtramas de captura, aunque llegarían a ser 81 si fueran necesarios los 4 RVs.

Teniendo en cuenta esto, la captura debe ser de 81 subtramas mínimo (contemplamos el caso peor en condiciones adversas de nivel de señal) o equivalentemente 81 ms.

Dependiendo del ancho de banda nominal con el que trabajemos, asociado a la banda LTE en la que estemos trabajando, tenemos los siguientes casos:

- Para $W=20$ MHz: $F_s=30,72$ Mmuestras/s, por tanto, $NSamples \geq 2488320$ Muestras ($NSamples=F_s \cdot T$).
- Para $W=5$ MHz: $F_s=7,68$ Mmuestras/s, por tanto, $NSamples \geq 622080$ Muestras ($NSamples=F_s \cdot T$).

- **Frecuencia de la portadora (F_c)**

Este parámetro depende la banda LTE con la que estemos trabajando. En el trabajo se ha utilizado la banda 20 y la banda 3 dependiendo de donde se hayan tomado las capturas.

Vamos a hacer una distinción entre la banda 20 y la banda 3, ya que en la banda 3 de LTE (1800 MHz) tomamos como frecuencia de la portadora la frecuencia central del intervalo del bloque en el enlace descendente (DL) correspondiente a cada operador (véase *tabla 2*).

Sin embargo, como vamos a ver en el capítulo de *Capturas Realizadas*, en la banda 20 de LTE, (véase *tabla 1*), es más difícil de ajustar frecuencia de la portadora

y en los casos en los que se obtenían los mejores resultados, se ha tomado los extremos del intervalo del bloque de frecuencias de cada operador.

En cuanto a otros parámetros, como el ancho de banda depende de la banda de frecuencia que estemos utilizando y la ganancia la podemos ajustar teniendo en cuenta que hay 2 etapas de amplificación.

No obstante, hay un parámetro del dispositivo hardware que debemos tener en cuenta y es el tema del *buffer*, es decir, el tamaño del buffer por el número de *buffers* debe mayor que el número de muestras que tengamos en cada captura señal, para asegurar que todas las muestras se puedan almacenar.

6.2. Procesado de señales LTE

El ejemplo "Cell Search, MIB and SIB1 Recovery" de Mathworks [25] nos ayuda a establecer el punto de partida para empezar a realizar el trabajo fin de grado (TFG), ya que proporciona los procesos y funciones necesarias para procesar y analizar las señales LTE del enlace descendente que podemos capturar a través de nuestro equipo hardware *Nuand BladeRF 3.0*. Este trabajo ya está implementado por Mathworks pero es necesario entender el código y entender las funciones relacionadas con LTE para la posterior modificación de alguna parte del código.

Este ejemplo de Mathworks muestra cómo a través de la LTE Toolbox se puede sincronizar, demodular y decodificar una señal *eNodeB* en vivo o que hayamos capturado previamente. Previamente a que el *User Equipment* (UE) o Equipo de Usuario pueda comunicarse con la red, éste debe realizar procedimientos de búsqueda y selección de celdas y debe obtener información inicial del sistema. Esto significa que debe adquirir una sincronización a nivel de slot y de trama, descubriendo la identidad de la celda y decodificando el *Master Information Block* (MIB) y los *System Information Blocks* (SIBs). Este ejemplo solo decodifica MIB y SIB1, es decir, sólo el primer bloque de información del sistema.

Para decodificar MIB y SIB1 se necesita un receptor completo, ya que tiene que ser capaz de demodular y decodificar la mayoría de los canales y señales del enlace descendente (*downlink*).

A continuación, se hace una descripción de las partes del código al que se hace mención [25]:

- **Introducción**

El UE debe obtener la información básica del sistema para poder comunicarse con la red. Esta información la transporta el MIB y los SIB. El MIB cuenta con la información más esencial del sistema:

- Ancho de banda
- *System Frame Number* (SFN)
- Programación (*scheduling*) de los SIBs.
- La configuración del canal PHICH

En cuanto a la relación de los bloques MIB y SIB con los canales implicados en LTE, el MIB se transporta en el canal de transmisión (BCH) mapeado en el canal de transmisión física (PBCH). Se transmite con un esquema fijo de codificación y modulación, y puede decodificarse posteriormente al procedimiento de búsqueda de la celda de inicio. Con esta información que el UE obtiene de MIB puede decodificar el Indicador de Formato de Control (CFI), que indica la longitud del canal de control del enlace descendente físico (PDCCH). Permite además, decodificar el PDCCH y buscar mensajes de control de enlace descendente (DCI). Un mensaje DCI CRC enmascarado con el SI-RNTI (Identificador temporal de Red de Radio de Información del Sistema), indica que un SIB es transportado en la misma subtrama.

Los SIB se transmiten en el canal lógico *Broadcast Control Channel* (BCCH), y generalmente, los mensajes BCCH se transportan en el canal compartido de enlace descendente (DL-SCH) y se transmiten en el canal compartido de enlace descendente físico (PDSCH). La asignación de recursos de la transmisión PDSCH se indica mediante un mensaje DCI en el PDCCH.

El primer *System Information Block* (SIB1) se transmite para especificar la programación de otra información adicional del sistema, además de otras características de identidad de la célula e identidad de la Red pública de telefonía móvil terrestre (PLMN). SIB1 se transmite en un horario fijo pero la asignación de recursos del PDSCH del SIB1 es dinámica y se indica en un mensaje DCI asociado.

- **Carga y procesamiento de la señal procedente del eNodeB**

El ejemplo nos proporciona la posibilidad de poder usar una señal LTE generada por la LTE Toolbox, y dichas muestras están almacenadas en el archivo "eNodeBOutput.mat". Las muestras almacenadas en este archivo corresponderían a una señal capturada en condiciones ideales.

Se trata de una captura I/Q de antena única de un *eNodeB* con 2 antenas de transmisión. Esta captura se realiza a 15,36 Mmuestras/s, por lo que se puede muestrear a los anchos de banda válidos hasta 10 MHz: 1,4 MHz, 3 MHz, 5 MHz, 10 MHz.

El MIB corresponde a un bloque de transporte BCH. El intervalo de tiempo de transmisión BC (TTI), es decir, el tiempo necesario para transmitir 1 bloque de transporte es de 40 ms o 4 tramas. El BCH se transmite en 4 partes, y cada parte se asigna a la primera subtrama de una trama (subtrama 0); es posible que cada transmisión pueda ser decodificable independientemente, aunque depende de las condiciones de la señal. Para garantizar que se reciba la subtrama 0, la captura debe tener 11 subtramas, para que no se pierda si se inicia a capturar señal durante la subtrama 0.

Cuando las condiciones para la señal son deficientes, pueden llegar a ser necesarias las 4 partes del TTI, por lo que en este caso, la captura debe tener como mínimo 41 subtramas.

Con respecto a SIB1, se transmite en la subtrama 5 de cada trama par, con 4 versiones de redundancia (RV) diferentes que se transmiten consecutivamente, dando lugar a un período de 80 ms u 8 tramas, por tanto, al igual que con MIB, se necesitan 21 subtramas de captura para garantizar la recepción de 1 RV de SIB, pero en caso de que las RV deban combinarse por las condiciones de la señal, se necesitan hasta 81 subtramas.

Antes de decodificar el MIB, el UE no conoce el ancho de banda completo del sistema. Las señales de sincronización primaria (PSS) y secundaria (SSS) y el PBCH (que contiene el MIB) se encuentran en las 72 subportadoras centrales (6 bloques de recursos) del ancho de banda del sistema, por lo que inicialmente el UE demodula solo esta región central. Por ello, inicialmente se establece el ancho de banda en 6 bloques de recursos y la señal I/Q se debe volver a muestrear de nuevo en consecuencia.

La velocidad de muestreo para la búsqueda de celdas de inicio se establece usando la función *lteOFDMInfo*, configurado para 6 bloques de recursos. "enb.CyclicPrefix" aparece en la llamada a *lteOFDMInfo* para suprimir un aviso de valor por defecto.

- **Búsqueda de celdas, longitud de Prefijo cíclico y detección de modo dúplex**

En esta parte se llama a la función *lteCellSearch* para conseguir la identidad de celda y el desplazamiento de tiempo "offset" para la primera cabecera de trama. La búsqueda de celdas se repite para cada combinación de longitud de prefijo cíclico y modo dúplex, y la combinación con mayor correlación permite identificar estos parámetros descritos anteriormente.

Se representa un gráfico de la correlación entre la señal recibida y las señales PSS/SSS para la identificación de la celda detectada.

La PSS se detecta mediante la correlación en el dominio del tiempo y la SSS es detectada mediante la correlación en el dominio de la frecuencia. Antes de detectar SSS, se lleva a cabo la estimación y corrección de desplazamiento de frecuencia usando la correlación de prefijo cíclico. La detección de PSS depende de si los desplazamientos en frecuencia son pequeños o grandes; en este último caso, la correlación PSS puede verse afectada y degradarse.

- **Estimación y corrección del desplazamiento de frecuencia:**

Previamente a la demodulación OFDM, cualquier desplazamiento de frecuencia selectivo se debe eliminar. El desplazamiento de frecuencia en la forma de onda I/Q se estima y corrige mediante las funciones *lteFrequencyOffset* y *lteFrequencyCorrect*. El desplazamiento de frecuencia se estima mediante la correlación con el prefijo cíclico, y por ello, puede estimar desplazamientos de hasta de hasta +/- la mitad del espaciado entre subportadoras (+/- 7.5 KHz).

- **Demodulación OFDM y estimación de canal**

La forma de onda I/Q OFDM, tras bajar a una frecuencia de muestreo inferior, se demodula para producir una malla de recursos "rxgrid", y se utiliza para realizar una estimación de canal. "hest" es la estimación del canal, "nest" es la estimación del ruido (para ecualización MMSE) y "cec" es la configuración del estimador de canal.

Para la estimación del canal, este ejemplo supone 4 señales de referencia específicas a las celdas, por lo que hay disponibles estimaciones de canal para cada antena receptora a partir de todos los puertos de señal de referencia de específicos de celda posibles. Este número de puertos real no se conoce todavía. La estimación del canal sólo se realiza en la primera subtrama.

Se utiliza una ventana promedio piloto conservadora de 13x9 en frecuencia y tiempo, y así se puede reducir la influencia del ruido en las estimaciones piloto durante el proceso de estimación de canal.

- **Demodulación PBCH, Decodificación BCH y Análisis de MIB**

Se decodifica tanto el MIB como el número de puertos de señal de referencia específicos de la celda, que se transmiten como una máscara en el CRC BCH. La función *ltePBCHDecode* establece un tiempo de trama módulo 4 y lo devuelve en el parámetro `[nfmod4]`. Además, también devuelve los bits de MIB en el vector "mib" y el número de puertos de señal de referencia de la celda, que se asigna a "enb.CellRefP". Si se decodifica el número de puertos como "enb.CellRefP=0", se ha producido un error al decodificar el BCH. La función *lteMIB* sirve para analizar el vector de bits "mib" y añadir campos que sean de utilidad a la estructura de configuración "enb". Tras la decodificación del MIB, el ancho de banda que se detecta se almacena en "enb.NDLRB".

- **Demodulación OFDM en Ancho de banda total**

Como ya conocemos el ancho de banda de la señal, ésta se vuelve a muestrear a la velocidad de muestreo nominal que usa la LTE Toolbox para ese ancho de banda.

La estimación y corrección del desplazamiento en frecuencia se lleva a cabo en la señal que se muestrea nuevamente; posteriormente se realiza la sincronización temporal y la demodulación OFDM.

- **Decodificación de SIB1**

Los siguientes pasos se realizan en este bloque:

- Demodulación del canal indicador de formato de control físico (PCFICH) y decodificación de CFI
- Decodificación PDCCH
- Búsqueda de PDCCH a ciegas
- Recuperación de bits SIB: demodulación PDSCH y decodificación DL-SC
- Almacenamiento intermedio y restablecimiento del estado DL-SCH Harq

Después de la recuperación, el SIB CRC debe ser 0.

Estos pasos de decodificación se realizan en bucle para cada aparición de una subtrama que transporte SIB1 en la señal que se recibe.

La señal de referencia SIB1 se transmite en la subtrama 5 de las tramas pares, así que la señal de entrada se verifica primero para establecer que por lo menos se ha producido una aparición de SIB1. Para cada subtrama SIB1, se dibuja la respuesta de magnitud de la estimación del canal, además de la constelación del PDCCH reciba recibido.

Analizado el mensaje de DCI, proporciona la configuración del PDSCH que lleva el SIB1, se demodula el PDSCH y se decodifican los bits de DL-SCH que proporcionan los bits del SIB1.

7. RESULTADOS

Las capturas que podemos tomar en cada zona dependen de las bandas de frecuencia en servicio de LTE, ya que dependiendo de la zona, habrá servicio para una determinada banda o para varias.

Si no estamos seguros, podemos usar la aplicación LTE Discovery, descrita anteriormente, para saber en todo momento qué banda de frecuencia está usando nuestro terminal.

Además, si supiéramos la ubicación de las antenas a las que se conecta nuestro terminal, podríamos tomar señales más fiables y puras debido al menor desvanecimiento por culpa de la distancia.

Una de las herramientas que podemos usar para conocer la ubicación de las antenas es *NetMonitor*.



Figura 24: Logo NetMonitor [27]

En esta aplicación se puede determinar la ubicación de la antena, la señal que recibimos de ella (si hay varias cercanas, podemos recibir la información de todas), el tipo de tecnología que tenemos habilitada en ese momento (GPRS, 4G, etc.) y una serie de identificativos.

En mi caso, se puede ver como la antena de la que está recibiendo información nuestro terminal se identifica por el número "214 01", y según indica pertenece a Vodafone ES.



Figura 25: Captura Aplicación NetMonitor Pantalla 1

Si deslizas a la derecha, hay un registro de algunas de las antenas de las que se ha recibido información recientemente y si deslizas de nuevo, aparecerán todas las antenas que hay a nuestro alcance con su correspondiente ubicación en el mapa, para identificarlas gráficamente:



Figura 26: Captura NetMonitor Pantalla 3

La más cercana a mi pueblo (Canena) es la que aparece en la captura de pantalla y se encuentra a las afueras del pueblo.

No obstante, el trabajo se ha intentado desarrollar en equipos del Laboratorio Radiofrecuencia y Microondas (L-137) siempre que mi disponibilidad lo ha permitido por contar estos equipos con mayor capacidad de computación.

En esta ubicación, se ha intentado capturar señales tanto en banda 20 como en banda 3 de LTE, y hay algunos operadores que nos ofrecen información, pero otros no tenemos nivel de señal válido.

Los mejores resultados se han obtenido en banda 3 LTE en la banda de frecuencia del operador *Vodafone* ($F_c=1835,1$ MHz), como se muestra en la figura de la página siguiente:

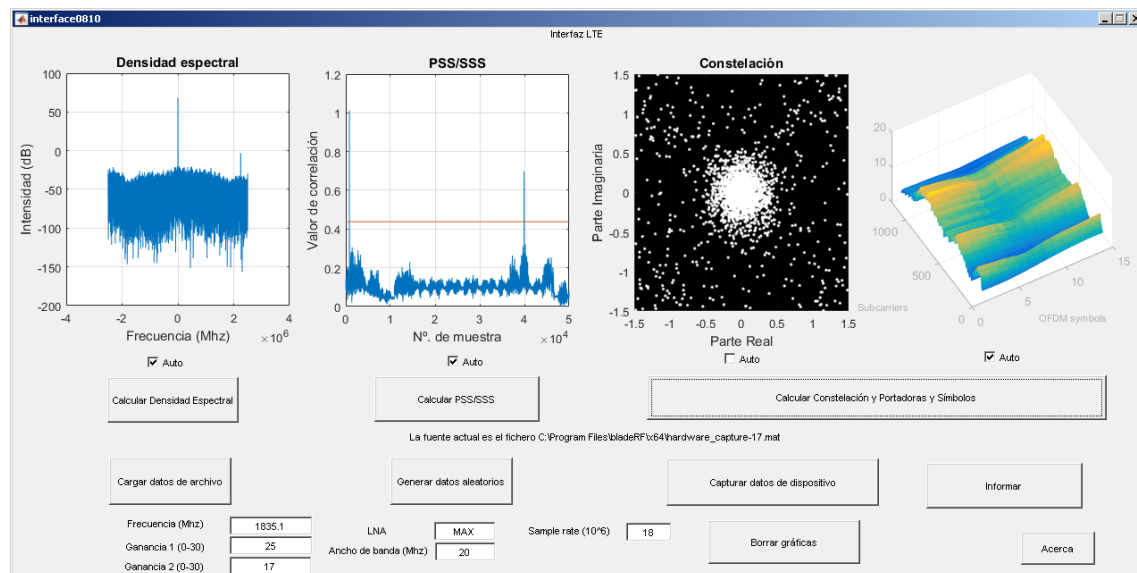


Figura 27: Captura Señal Vodafone (Banda 3)-Samplerate: 18 Mmuestras/s

He incluido la captura completa del interfaz para ver los parámetros de configuración de la tarjeta BladeRF. Como podemos apreciar, la frecuencia de muestreo (*samplerate*) elegida no cumple con los requisitos de la señal LTE, por lo que procedemos a realizar una captura aumentando el *samplerate* y obtenemos el siguiente resultado:

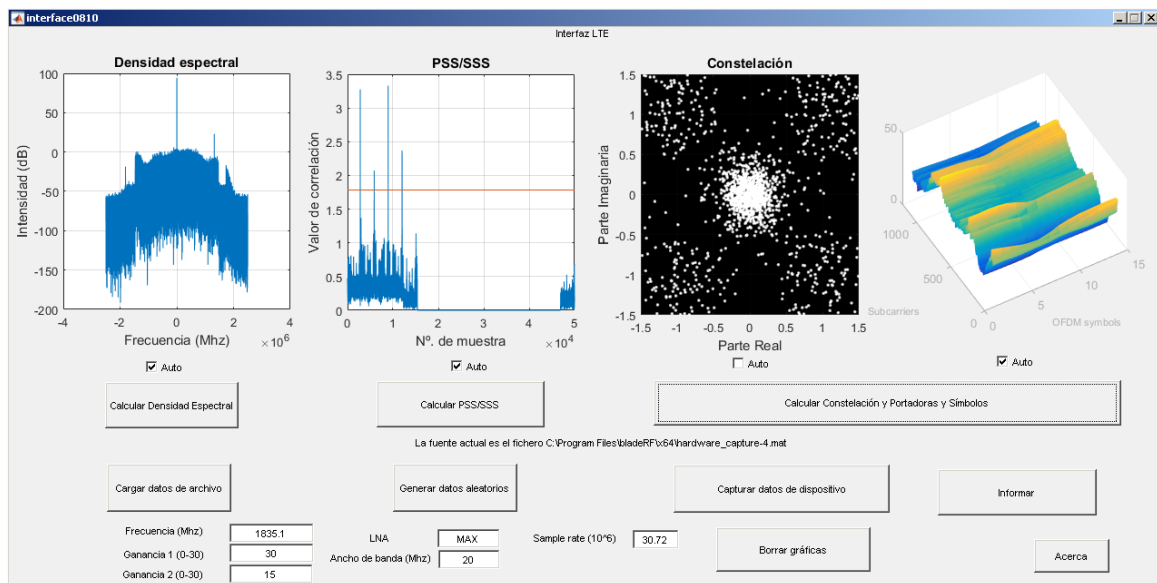


Figura 28: Captura Señal Vodafone (Banda 3)-Samplerate: 30,72 MMuestras/s

:

Aquí podemos observar que la correlación entre PSS y SS ha aumentado, y en consecuencia, los siguientes parámetros han mejorado como la distribución de los símbolos PDCCH.

En la banda 3 de los demás operadores presentes no se han encontrado resultados satisfactorios, ya que probablemente la estación base no daba servicio a esa zona de cobertura.

En la banda 20, Movistar, Vodafone y Orange tienen presencia en nuestra zona y se obtienen los siguientes resultados:

- Captura Orange:

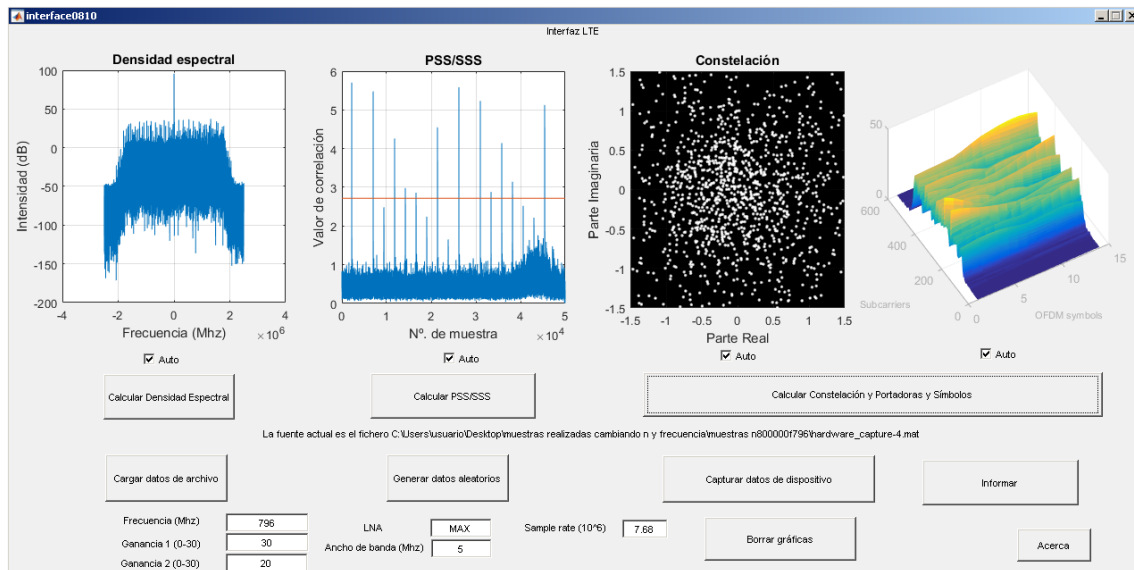


Figura 29: Captura Señal Orange (Banda 20)

- Captura Vodafone:

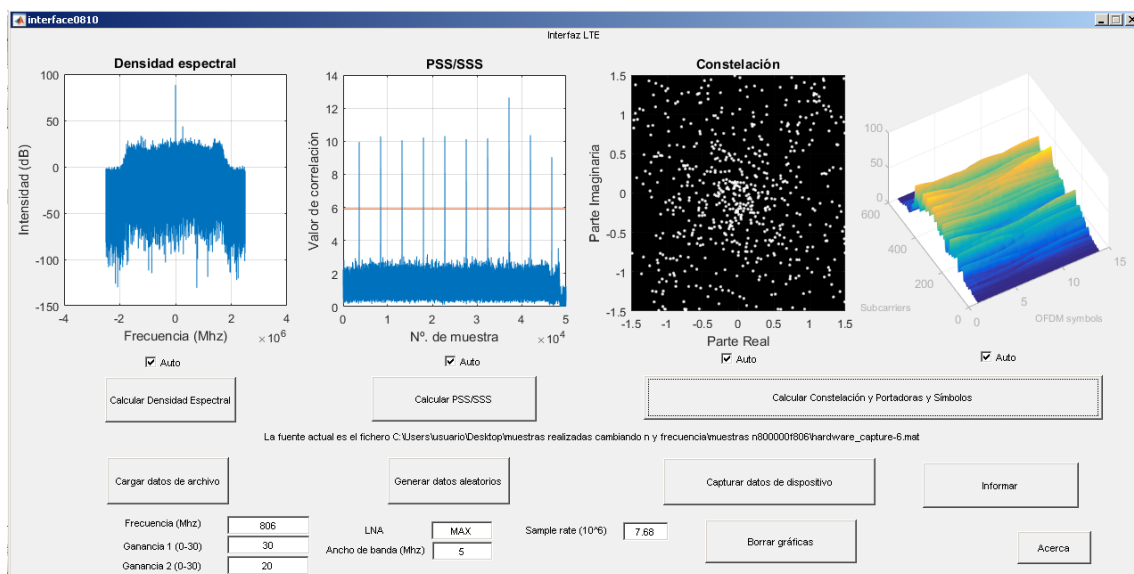


Figura 30: Captura Señal Vodafone (Banda 20)

- Captura Movistar:

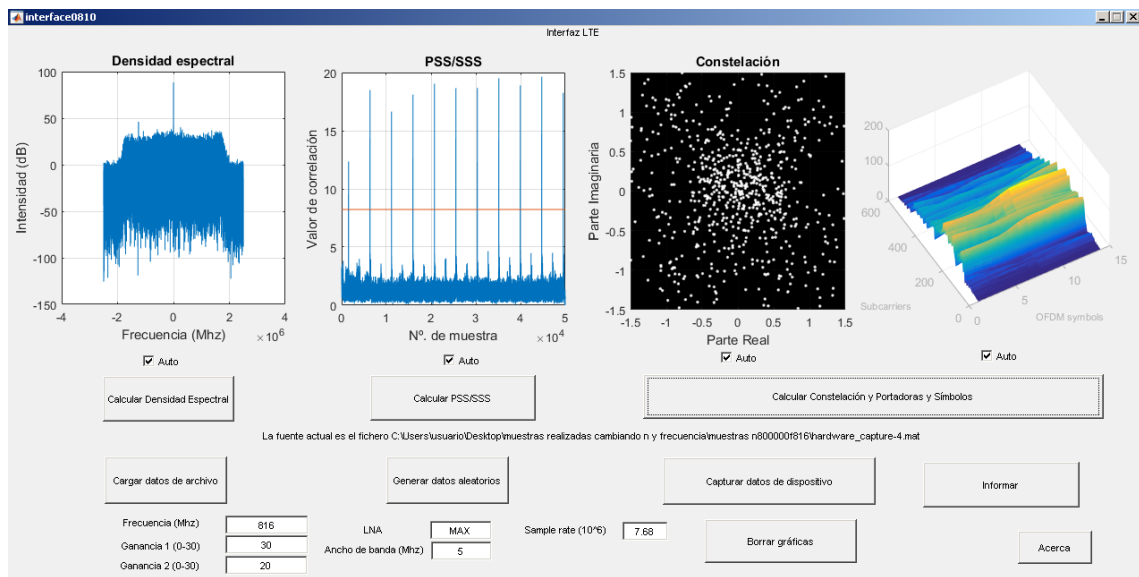


Figura 31: Captura Señal Movistar (Banda 20)

Se puede observar en esta banda que aunque parezca que estamos usando 600 subportadoras según el gráfico, sólo usamos 300 como indica en el estándar LTE ya que en las restantes no hay información.

Estas 3 capturas se han hecho con $n \geq 622080$, en este caso, 800000.

En general, en todos los casos, se tuvo problemas con la determinación del número de muestras a capturas en cada llamada a la tarjeta, ya que según el estándar LTE, se exige un número mínimo de muestras, pero eso conlleva que en la banda 3 (ancho de banda nominal de 20 MHz), el número de muestras mínimas a capturas es de casi 2,5 millones de muestras y si nuestros recursos de computación son limitados, podemos tener problemas a la hora de ejecutar el programa.

Además, en el caso de la banda 3, se tuvo que configurar la tarjeta aumentando el tamaño de sus buffers, ya que el tamaño que viene por defecto se queda corto.

8. CONCLUSIONES Y LÍNEAS DE FUTURO

Conclusiones

Este trabajo me ha permitido explorar y conocer el campo de las tecnologías inalámbricas más a fondo y en especial, todas las características y particularidades de la tecnología LTE.

En el campo de SDR, también se han adquirido nociones hasta ahora desconocidas para mí, aunque es cierto que había tenido alguna formación en una tecnología relacionada como es SDN (*Software Defined Network*), pero en este caso, se aplica la misma filosofía a las comunicaciones inalámbricas.

En cuanto al desarrollo del trabajo, en una primera fase se trabajó con la tarjeta desde su línea de comandos CLI. Posteriormente, se buscó la manera de cómo integrarla con Matlab para trabajar con el *System Objects* y que posteriormente se pudiera hacer una interfaz gráfica integrada. Aquí tuve un problema en mi ordenador personal, ya que al integrar el directorio de *BladeRF* (API *libbladerf*) en Matlab e intentar establecer comunicación con la tarjeta y el PC, Matlab se cerraba y al intentarlo de nuevo, seguía sin funcionar. Por tanto, tuve que usar otro dispositivo que ya sí me permitió esa comunicación, y a priori, había seguido los mismos pasos de instalación de los *drivers* y el API de la tarjeta.

En la siguiente fase, se procedió al entendimiento del código y la información que proporcionaba el archivo proporcionado por Mathworks del proceso de *Cell Search*, el "Cell Search, MIB and SIB1 Recovery". Este archivo ya implementado ha sido muy útil, pero su modificación ha sido necesaria para crear la interfaz gráfica, ya que dentro del archivo se llamaba a objetos creados de Mathworks que no permitían su modificación para incluir en una interfaz gráfica.

En cuanto a la captura de señal LTE, se ha podido analizar en ambas bandas, banda 3 y banda 20 de LTE, las señales que podemos obtener de cada uno de los operadores y las diferencias de parámetros de una banda a otra. De hecho, el número de muestras mínimas, como se comentaba en el trabajo, varía entre las dos bandas, por lo que el análisis de la banda 3 requiere más recursos de procesamiento. En cuanto a la calidad de las señales, no son óptimas pero sí nos pueden dar información útil del proceso de *Cell Search*, ya que en esos casos se obtenía información en línea de comandos de cada uno de los parámetros que intervenían en el proceso.

La parte de la interfaz gráfica ha sido una tarea interesante en la que he aprendido conceptos de diseño de interfaces a través de Matlab, un lenguaje potente y usado para procesamiento de señal, por lo que agrupa todos los procesos que implica la programación de esa interfaz.

Líneas de futuro

Este proceso de *Cell Search* que se analiza en el TFG, no es el único proceso que el terminal de usuario tiene que procesar para conectarse a la red, ya que en el trabajo sólo se decodifica el SIB1. Por tanto, se podría realizar los siguientes procesos en un trabajo posterior a futuro:

- Completar el proceso de *Cell Search*, como la decodificación de los SIBs restantes, ya que una vez decodificado el SIB1 se tiene información sobre el resto de bloques. Esto nos daría la oportunidad de disponer de mayor información específica del sistema.
- Obtener capturas de señales en bandas no usuales de encontrar como puede ser la banda 7.
- Estudiar las señales del enlace ascendente (UL) ya que el dispositivo usado en el trabajo lo permite. Consecuentemente, se tendría que realizar un estudio teórico de las tramas en el enlace ascendente.

BIBLIOGRAFÍA

- [1] *Comparisons with other common Wideband Commercial Software Defined Radios*
Disponible en: <https://www.rtl-sdr.com/about-rtl-SDR/>
- [2] *Roundup of Software Defined Radios*, 2014. Disponible en: <https://www.rtl-sdr.com/roundup-software-defined-radios/>
- [3] *10 Popular Software Defined Radios (SDRs) of 2019*. Bliley Technologies, 2018.
Disponible en: <https://blog.bliley.com/10-popular-sdrs-software-defined-radios-2018>
- [4] Quick Start Guide. © airspy.com 2019. Disponible en: <https://airspy.com/quickstart/>
- [5] MACHADO-FERNANDEZ, José Raúl. *Software Defined Radio: Basic Principles and Applications*. Rev. Fac. ing. [online]. 2015, vol.24, n.38, pp.79-96. Disponible en: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0121-11292015000100007&lng=en&nrm=iso. ISSN 0121-1129.
- [6] *Sistemas de telefonía y comunicaciones móviles*. Máster Móviles UA. Disponible en: <https://mastermoviles.gitbook.io/tecnologias2/sistemas-de-telefonía-y-comunicaciones-moviles>
- [7] DAHLMAN, Erik, PARKVALL, Stefan, SKÖLD, Johan. *4G, LTE-Advanced Pro and The Road to 5G. Third Edition*. Boston: Elsevier, 2016
- [8] Consulta del Registro Público de Concesiones. Ministerio de Economía y Empresa. Disponible en: https://sedeaplicaciones.minetur.gob.es/RPC_Consulta/FrmConsulta.aspx
- [9] Frecuencias telefonía móvil, Oct. 2019 Disponible en: https://wiki.bandaancho.st/Frecuencias_telefon%C3%ADa_m%C3%B3vil
- [10] Frecuencias y bandas LTE en España, Sept. 2019. Disponible en: https://wiki.bandaancho.st/Frecuencias_y_bandas_LTE_en_Espa%C3%B1a
- [11] Descripción móvil Jiayu. Disponible en: <https://www.jiayu.es/smartphone/210-jiayu-m2.html>
- [12] Descripción móvil Xiaomi. Disponible en: <https://www.mi.com/es/redmi-note-5/>
- [13] Descripción móvil Meizu. Disponible en: <http://www.movilesymas.es/movil/meizu/m8c-16gb-2gb-ram.html#.XbSAJW5uJ7c>

- [14] Logo LTE Discovery:
<https://play.google.com/store/apps/details?id=net.simplyadvanced.ltediscovery&hl=es> 41
9
- [15] DAHLMAN, Erik, PARKVALL, Stefan, SKOLD, Johan. *4G LTE/LTE-Advanced for Mobile Broadband*. Boston: Elsevier, 2011
- [16] LTE Quick Reference. Disponible en:
https://www.sharetechnote.com/html/Handbook_LTE_SF_N.html
- [17] COLLINS, Travis F., GETZ, Robin, PU, Di, WYGLINSKI, Alexander M.. *Software-Defined Radio for Engineers*. EEUU: Artech House, 2018
- [18] Datasheet BladeRF. Disponible en: <https://www.nuand.com/bladeRF-brief.pdf>
- [19] Descripción Antena. Disponible en: <https://es.rs-online.com/web/p/antenas-wifi/9026818/>
- [20] Datasheet Antena. Disponible en: <https://docs-emea.rs-online.com/webdocs/14b1/0900766b814b18b7.pdf>
- [21] Modelo BladeRF x40 (FPGA). Disponible en:
<https://www.nuand.com/product/bladerf-x40/>
- [22] Comparación FPGAs de Altera. *Cyclone IV FPGA Device Family Overview, Cyclone IV Device Handbook, Volume 1, Chapter 1., Marzo 2016*. Disponible en:
<https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/hb/cyclone-iv/cyiv-51001.pdf>
- [23] Datasheet Lime-Micro. Disponible en:
<https://limemicro.com/app/uploads/2015/10/LMS6002Dr2-DataSheet-1.2r0.pdf>
- [24] Guía de instalación y primeros pasos con BladeRF. Disponible en:
http://www.nuand.com/bladeRF-doc/guides/bladeRF_windows_installer.pdf
- [25] Mathworks Matlab Examples, "Cell Search, MIB and SIB1 Recovery". Disponible en: <https://www.mathworks.com/help/lte/examples/cell-search-mib-and-sib1-recovery.html>
- [26] Mathworks Synchronization Signals (PSS and SSS). Disponible en:
<https://es.mathworks.com/help/lte/ug/synchronization-signals-pss-and-sss.html>
- [27] Logo NetMonitor:
<https://play.google.com/store/apps/details?id=com.parizene.netmonitor&hl=es>

- [28] Mathworks "Adding componets to the gui". Disponible en:
https://www.mathworks.com/help/matlab/creating_guis/adding-components-to-the-gui.html
- [29] Mathworks GUI Example. Disponible en:
https://www.mathworks.com/help/matlab/creating_guis/about-the-simple-guide-gui-example.html
- [30] Fernández de Córdoba Martos, Gonzalo. *Creación de Interfaces Gráficas de Usuario (GUI) con MatLab*. Salamanca: Septiembre de 2007. Disponible en:
<http://webpersonal.uma.es/de/gfdc/docencia/GuiSection.pdf>
- [31] MOORE, Andrew, WILSON, Ron. *FPA's for Dummies, 2nd Intel Special Edition*. New Jersey: John Wiley & Sons, Inc., 2017
- [32] Jiménez, Leonardo, Parrado, Joaquín, Quiza, Carlos, Suárez, Carlos. Modulación multiportadora OFDM. Ingeniería, 2001-00-00 vol:6 nro:2 pág:30-34. Disponible en:
<https://revistas.udistrital.edu.co/index.php/reving/article/view/2699/3891>
- [33] Martínez Muñoz, Damián. *Apuntes asignatura Comunicaciones Móviles. Tema 5: Sistema GSM*, Universidad de Jaén, Escuela Politécnica Superior de Linares
- [34] Martínez Muñoz, Damián. *Apuntes asignatura Comunicaciones Móviles. Tema 8: Sistema UMTS*, Universidad de Jaén, Escuela Politécnica Superior de Linares
- [35] Martínez Muñoz, Damián. *Apuntes asignatura Comunicaciones Móviles. Tema 6: Sistema GPRS*, Universidad de Jaén, Escuela Politécnica Superior de Linares

ANEXO I: MANUAL DE INSTALACIÓN DE BLADERF

El manual de instalación se encuentra en la página: www.nuand.com/bladeRF-doc/guides/bladeRF_windows_installer.pdf

Los requerimientos mínimos del PC en donde se va a instalar y configurar la tarjeta son:

- Procesador *quad-core* de 64-bit (3 GHz)
- 4 GB RAM
- Controlador USB 3.0
- Windows 7 64-bit

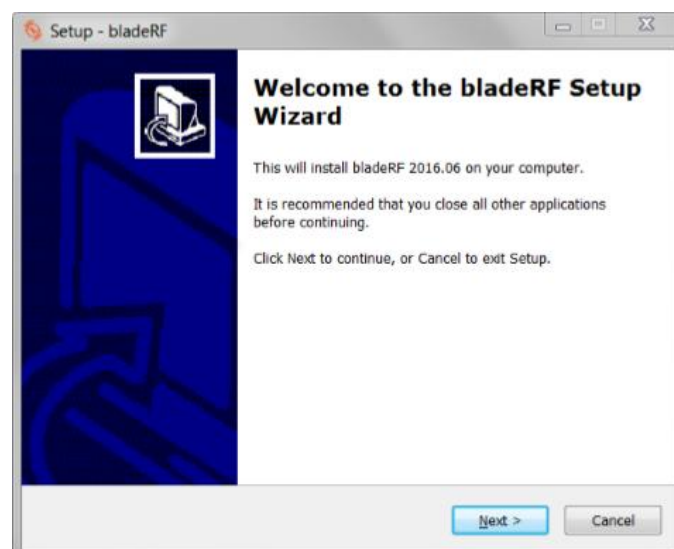
Las versiones de Windows soportadas son: XP, Vista, 7 y 8.1 (todos en versiones 32-bit y 64-bit)

Las versiones de Matlab soportadas son: 2014b, 2015a, 2015b y 2016a. No obstante, en este trabajo no se puede usar una versión inferior a 2016a, ya que es la primera versión que introduce la LTE Toolbox.

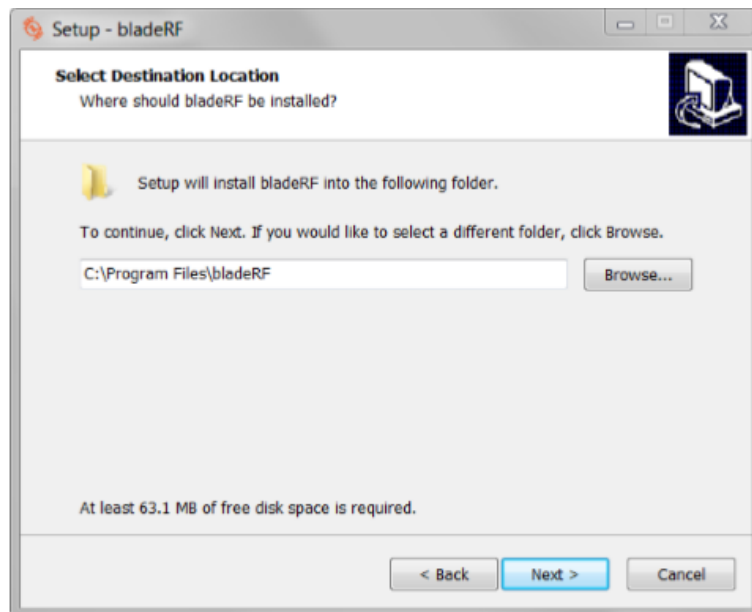
Para proceder a la instalación, la última versión disponible se encuentra en: https://nuand.com/windows_installers/bladeRF-win-installer-latest.exe

Ejecutamos el programa instalable .exe y nos aseguramos que la tarjeta no se encuentra conectada al sistema. No la conectamos hasta que el proceso haya finalizado o hasta que la consola nos lo pida, por ejemplo para la actualización del *firmware*.

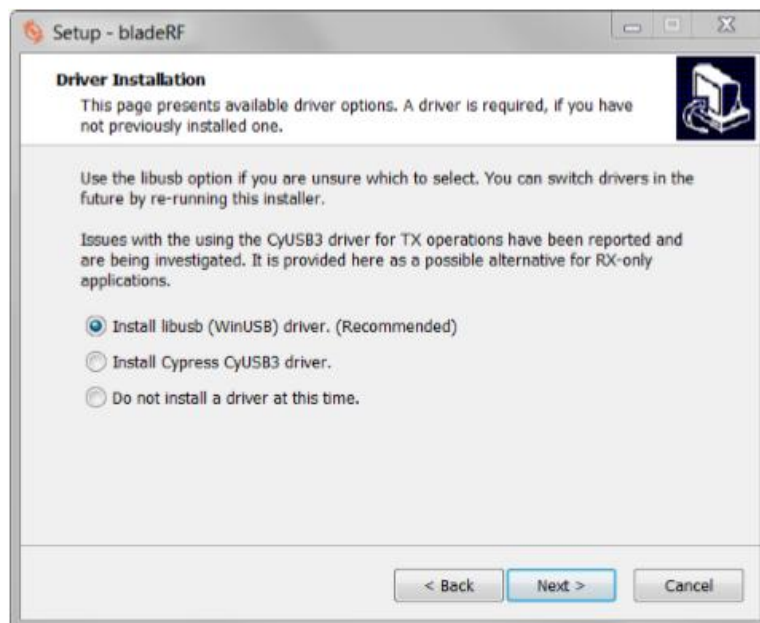
La primera pantalla que se presenta es la siguiente, por lo que procedemos a pulsar el botón *Next* **(todas las figuras proceden del archivo de ayuda a la instalación: www.nuand.com/bladeRF-doc/guides/bladeRF_windows_installer.pdf):**



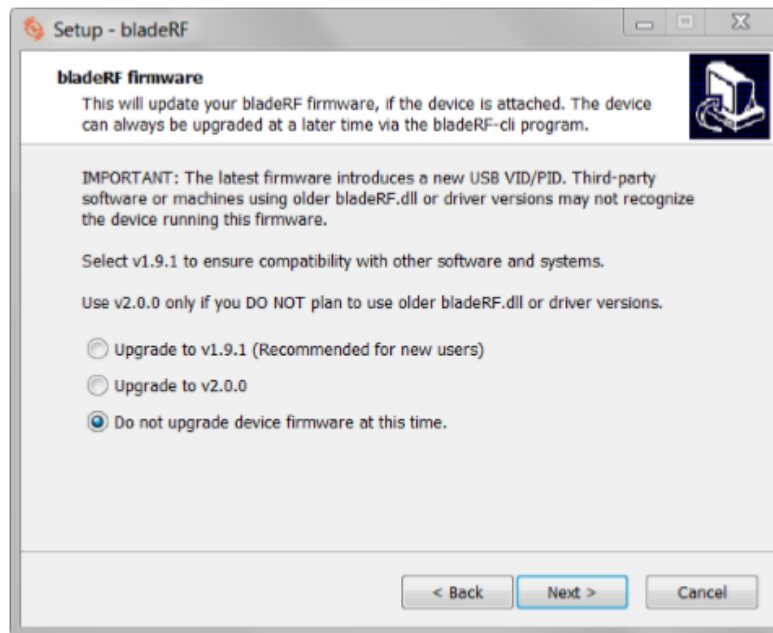
Seguidamente, tenemos que elegir el directorio de instalación. Una vez configurado este campo, podemos seguir el proceso de instalación:



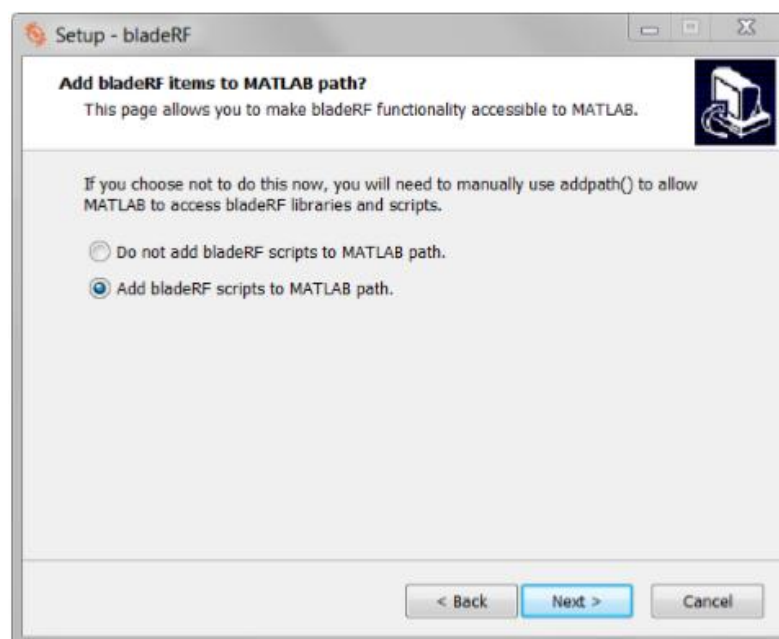
En la siguiente figura, hay tres posibilidades de instalación de los drivers. En este caso, dejamos seleccionada la opción recomendada del *libusb driver* ya que en el caso del CyUSB3, es para uso de capacidades en transmisión, no en recepción.



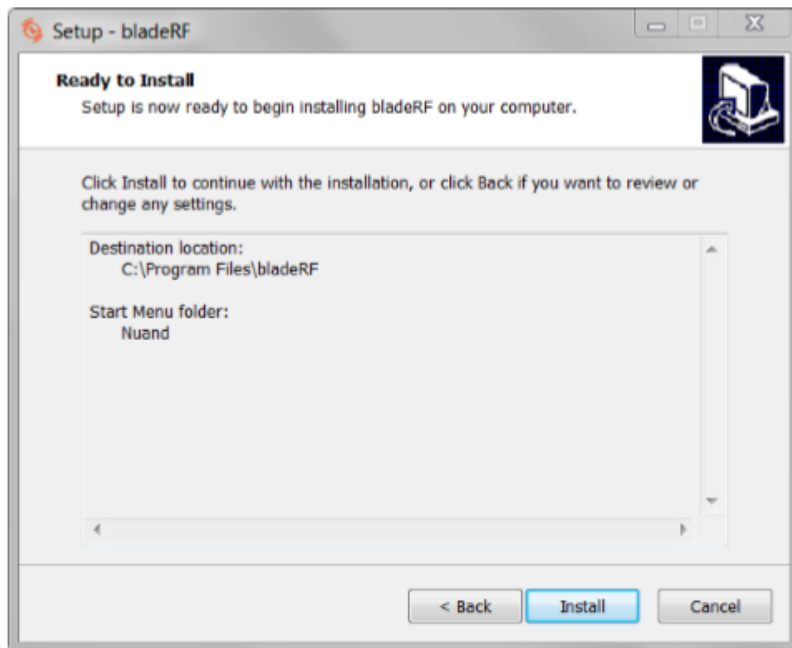
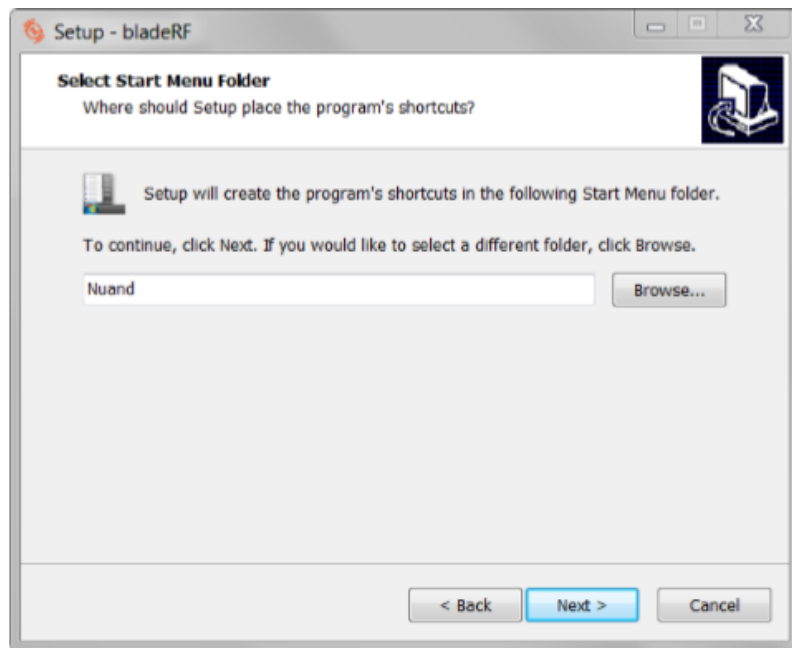
El siguiente paso da la posibilidad de actualizar el firmware, aunque si no se hace, posteriormente se puede actualizar usando el bladeRF-cli. Si queremos mantener la compatibilidad con otros sistemas y otro tipo de software, tenemos que seleccionar la opción "Upgrade to v1.9.1".



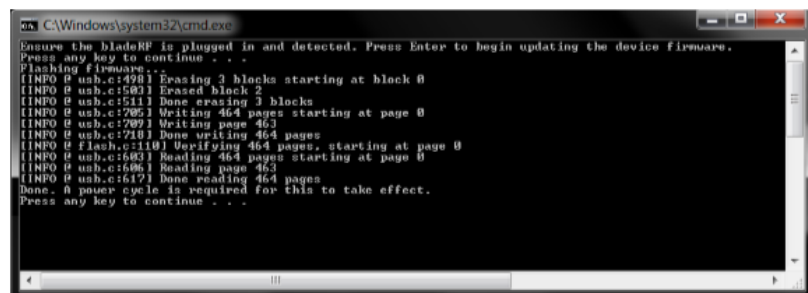
El siguiente paso es añadir las librerías de BladeRF al directorio de Matlab (*Path*), cuya pantalla aparecerá si se detecta Matlab de 64-bit:



Las siguientes pantallas permiten establecer el directorio de instalación de la tarjeta y dónde se va a encontrar el programa de desinstalación. En la siguiente, se muestra que ya está todo listo para instalar:

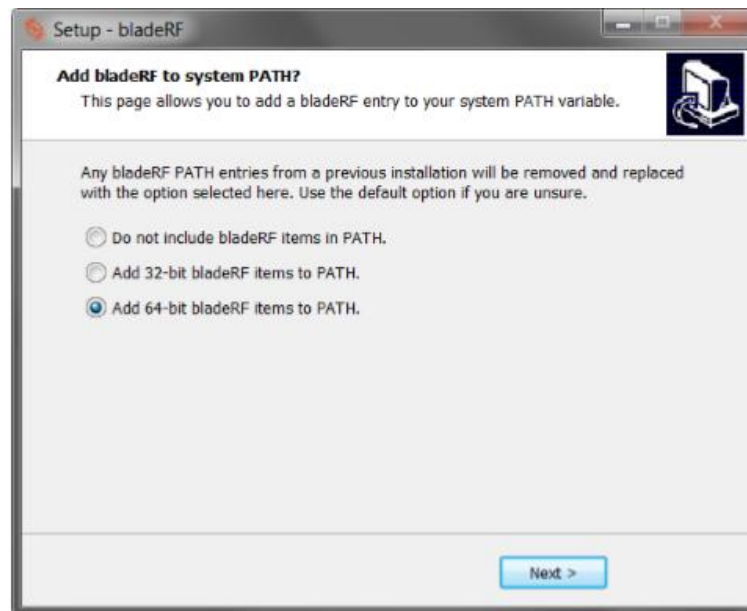


Si se selecciona la actualización del *firmware*, se mostrará una pantalla similar a la siguiente:

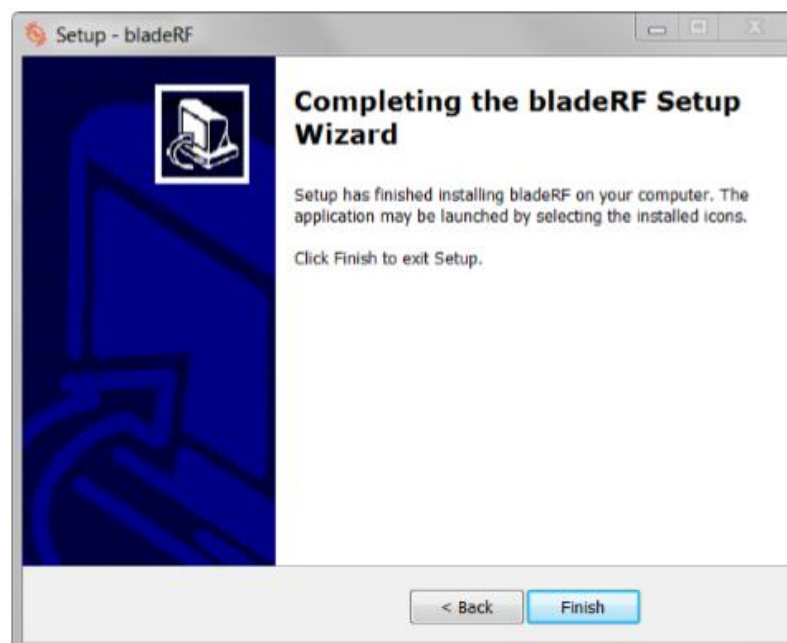


En este momento, se nos indica por pantalla que debemos conectar la tarjeta para que empiece el proceso y no debemos desconectarla; en el peor de los casos, en el que desconectemos por accidente, el dispositivo entrará en modo de recuperación *bootloader*.

Al final de la instalación, nos aparecerá esta pantalla en la que se da la posibilidad de añadir los elementos de bladeRF 32-bit o 64-bit al *Path*



La última pantalla de instalación completada nos aparecerá y es importante que reiniciemos el sistema para que se apliquen los cambios realizados en el sistema con respecto a los directorios que se añaden, incluyendo el de Matlab.



Cuando empecemos a trabajar con Matlab, tenemos que tener en cuenta que los archivos que queramos ejecutar se encuentren en el directorio deseado.

Por ello, los archivos que creamos en Matlab, los guardaremos en la carpeta "x64" de BladeRF y para que no haya problemas de directorios podemos usar la función de Matlab ***addpath.***

A esta función debemos pasarle como argumentos de entrada las carpetas que tengamos que tener en el directorio. En este caso, hay que tener las carpetas bladeRF \ x64 y bladeRF \ matlab.

ANEXO II: GUIDE DE MATLAB

Iniciación a GUIDE

Una interfaz gráfica es una herramienta que permite al usuario interaccionar con un programa de forma más sencilla a través de menús o métodos, ya que facilitan la entrada y salida de datos.

El hecho de crear una buena GUI es importante ya que permiten a usuarios inexpertos facilitar el uso de ese programa; también les puede ayudar a usuarios avanzados a que puedan realizar acciones de forma más rápida.

Muchos lenguajes de programación nos permiten crear GUIs como pueden ser Python, Visual Basic o Matlab, que es el que usamos en este trabajo.

Matlab nos permite hacer GUIs usando la herramienta llamada GUIDE (*GUI Development Environment* o *Graphical User Interface Development Environment*).

Esta herramienta nos permite diseñar la presentación de los controles de la interfaz al usuario y personalizar propiedades. Cuando ya están colocados los controles, hay que editar las funciones de llamada (Callback) de cada uno de ellos, para que al accionarlos, se ejecute la acción que previamente hemos programado.

Uno de los componentes de GUIDE es el editor de propiedades (property editor), que permite modificar propiedades de los controles que hayamos introducido.

En las GUIs, el usuario determina el orden en que se ejecutan las acciones y funciones desarrolladas en el programa a través de su interacción con el interfaz.

En el desarrollo de GUIs podemos diferenciar entre:

- Diseño de componentes (controles, menús, gráficas, ejes) que van a formar parte de la GUI.
- Programación de la acción que se va a ejecutar en cada uno de los componentes cuando el usuario interaccione con la interfaz.

A GUIDE se puede acceder de varias maneras, la primera de ellas y más rápida y sencilla es tecleando guide en la ventana de comandos (es la que yo uso, ya que a mi me resulta la más rápida):

```
>> guide
```

Otra forma de acceder a GUIDE es a través de la opción *File* y selección *New GUI*.

Podemos crear 4 tipos de GUI: *Blank GUI (Default)*, *GUI with Uicontrols*, *GUI with Axes and Menu* y *Modal Question Dialog*.

En este trabajo, se ha optado por crear la primera opción (Interfaz en blanco) en la que podemos añadir los controles y botones que nos hagan falta, ya que las otras opciones tienen ejemplos u opciones ya predefinidas.

Los componentes principales de GUIDE son:

- Barra de Menús: funciones básicas de edición de GUIs.
- Paleta de componentes: aquí están los uicontrols, componentes que permiten seleccionar los controles que queramos añadir. [28]

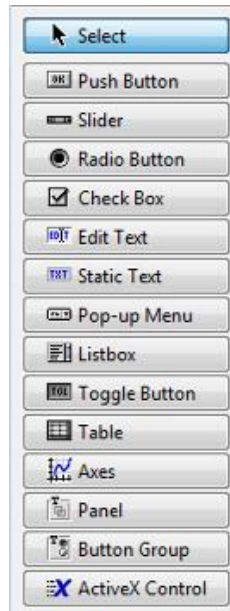


Figura 32: Paleta de componentes GUIDE [28]

- Barra de Herramientas: cuenta con los botones de ejecución, alineación de componentes, inspector de propiedades, navegador de objetos y editor de menús.

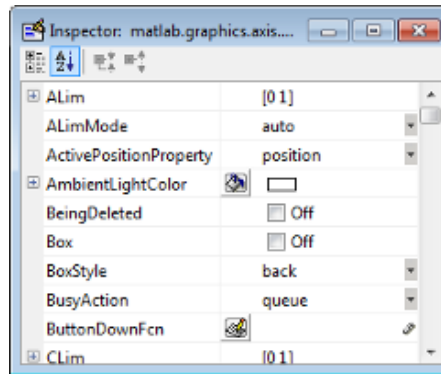


Figura 33: Property Inspector GUIDE [28]

- Callback: es el que controla la acción y se trata de una rutina que se ejecuta cuando se activa un objeto de la clase uicontrol. Define la rutina como una cadena cuya expresión se ejecuta en el espacio de trabajo de Matlab.

El editor de GUIDE permite diseñar interfaces arrastrando y soltando componentes en el área de diseño de la GUI.

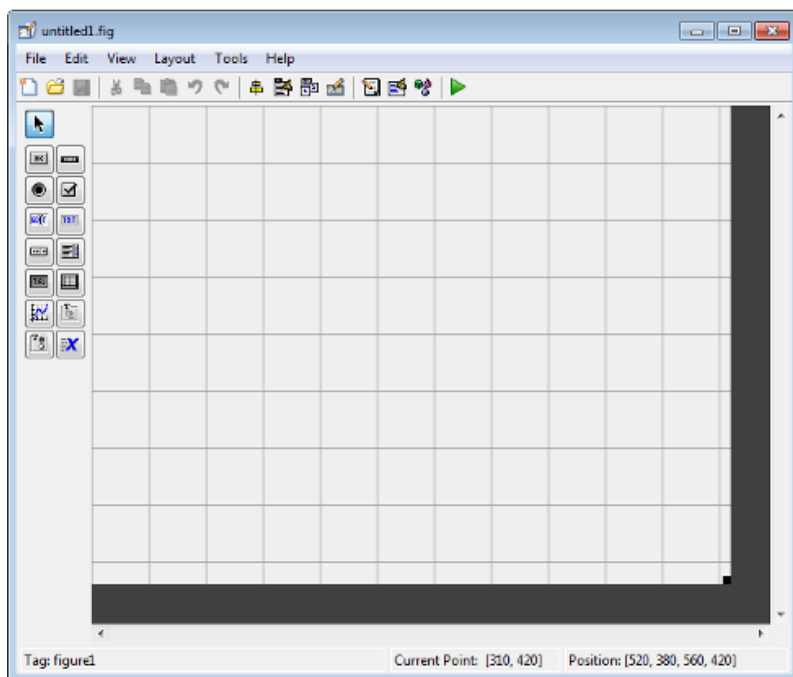


Figura 34: Editor de GUIDE [29]

Esta interfaces gráficas (GUIs) creadas con GUIDE empiezan con la función inicial (callback) descrita anteriormente, que se llama cuando se invoca la interfaz.

Cuando se guarda este archivo, automáticamente se genera un fichero .m y un fichero .fig.

GUIDE CREADA PARA EL TFG

La GUIDE creada cuenta con las siguientes funciones

```
function varargout = interface0810(varargin)
```

Es la función de inicialización y creación de la interfaz gráfica que crea Matlab por defecto al crear un archivo .fig asociado al archivo .m (el archivo .fig y .m tienen que tener el mismo nombre).

```
function interface0810_OpeningFcn (hObject, eventdata, handles, varargin)
```

Es la función que se ejecuta justo antes de que la interfaz (interface0810) sea visible. En esta función, además del código que crea Matlab, he introducido un registro de fecha y hora de entrada al programa: *handles.runstarttime = now();*

```
function BOTON_CARGAR_MAT_Callback(hObject, eventdata, handles)
```

Esta función nos permite cargar un archivo que previamente hayamos guardado, es decir, señales capturadas con la tarjeta y que hemos almacenado en un **archivo .mat**. Se ejecuta, al igual que todos los *callbacks*, cuando presionamos el botón en la interfaz gráfica.

La función comprueba que el archivo sea correcto de diferentes formas:

- Si no se ha especificado ningún archivo, muestra por pantalla:

Intento n° 1:

El usuario no ha especificado archivo y ha cancelado.

- Si se ha cargado un archivo .mat, comprueba que mínimo tenga algunas variables requeridas en el sistema como pueden ser la señal *nodob* o el *samplerate*. Por tanto, si cargamos un entorno que no cuenta con estas variables, muestra por pantalla lo siguiente:

Intento n° 1:

Se va a transferir el entorno cargado hacia la variable handle del interfaz GUIDE

Las variables existentes en el entorno cargado son:

'nodob'

```

'samplerate'
Cargando el archivoC:\Program
Files\bladeRF\x64\grafica3v2.mat
La var bandwidth no existe y por eso la creamos antes de
cargarla

```

En este caso, la variable bandwidth no existe en el entorno, y por eso se tiene que crear.

Además, cada vez que cargamos algún entorno, en la interfaz gráfica aparece la siguiente información:

La fuente actual es el fichero C:\Program Files\bladeRF\x64\grafica3v2.mat

- Si el fichero cuenta con estas variables, se sale de la función al cargar correctamente el archivo, y la información que se muestra en la ventana de comandos es la siguiente:

```

Intento nº 1:
Se va a transferir el entorno cargado hacia la variable
handle del interfaz GUIDE
Las variables existentes en el entorno cargado son:
    'bandwidth'
    'frequency'
    'lna'
    'nodob'
    'samplerate'
    'vga1'
    'vga2'

```

```

Cargando el archivoC:\Program
Files\bladeRF\x64\hardware_capture-1.mat
La var bandwidth sí existe porque se cargó o generó antes

```

- Si se ha cargado un fichero con extensión errónea (sólo se admiten ficheros .mat), aparece de nuevo la pantalla para seleccionar archivo y se le solicita al usuario que vuelva a cargar un archivo que cumpla con las especificaciones. Por pantalla aparece lo siguiente:

```

Intento nº 1:

Fichero con extensión errónea .m. Seleccione otro archivo.

```

En definitiva, sólo se sale de la función si cargamos un archivo correcto o si el usuario no especifica archivo y cancela. Si no es así, se va a pedir al usuario que introduzca un archivo válido.

Posteriormente, se transfiere lo cargado a los *handles* y se muestra los valores cargados del fichero en los *edit text* de la interfaz.

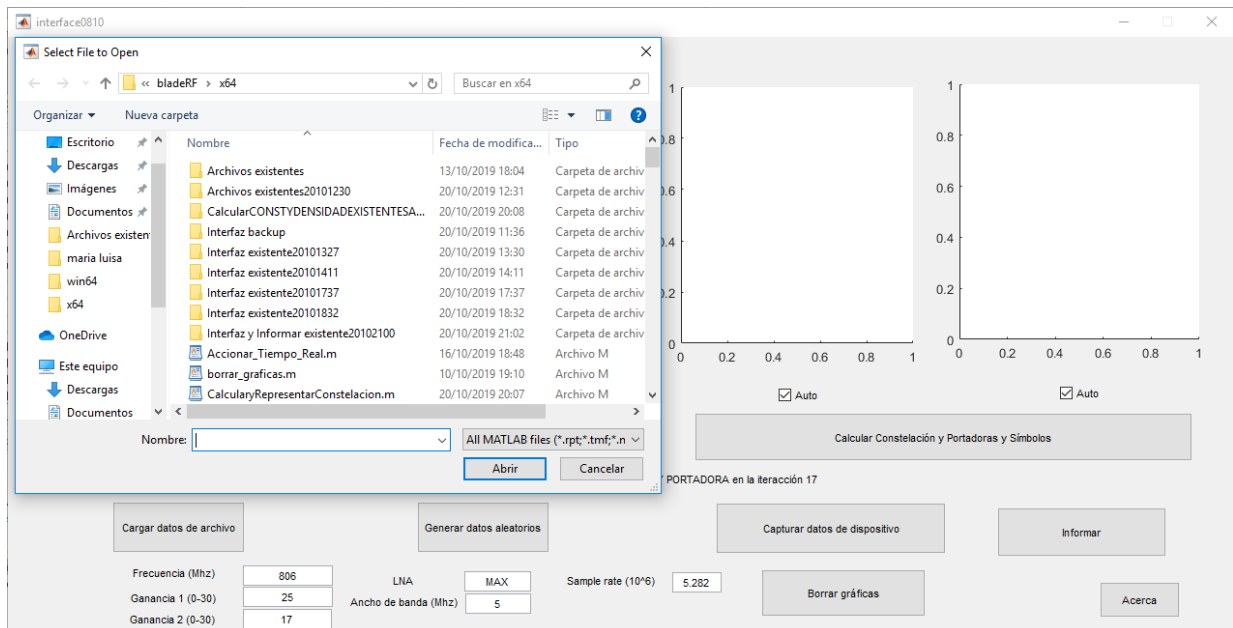


Figura 35: Interfaz de carga de datos desde archivo

9.1. Ejemplo con captura real

Usamos el archivo `hardware_capture-2.mat`, previamente cargado con los parámetros:

- Frecuencia: 806 MHz
- Ganancia 1: 25
- Ganancia 2: 17
- LNA: MAX
- Ancho de banda: 5 MHz
- Samplerate: 5.282 MHz

Al pulsar el botón Cargar datos de archivo, por ventana de comandos sale lo siguiente:

Intento n° 1 :

Se va a transferir el entorno cargado hacia la variable `handle` del interfaz GUIDE

Las variables existentes en el entorno cargado son:

```
'bandwidth'
'frequency'
'lina'
'nodob'
'samplerate'
'vga1'
```

'vga2'

Cargando el archivo C:\Program Files\bladeRF\x64\hardware_capture-2.mat

- La var bandwidth sí existe porque se cargó o generó antes

Los siguientes botones cargan las gráficas pero no muestran información relevante: Botón CALCULAR DENSIDAD ESPECTRAL y Botón CALCULAR PSS/SSS

El botón que muestra información relevante es:

Botón CALCULAR CONSTELACIÓN Y PORTADORAS Y SÍMBOLOS.

Por ventana de comandos aparece lo siguiente:

```
Resampling from 5.282Ms/s to 1.920Ms/s for cell search / MIB decoding...
```

```
Performing cell search...
```

```
Timing offset to frame start: 16898 samples
```

```
Cell-wide settings after cell search:
```

```
    NDLRB: 6
```

```
    DuplexMode: 'FDD'
```

```
    CyclicPrefix: 'Normal'
```

```
    NCellID: 77
```

```
    NSubframe: 0
```

Estas son las características de la celda después del proceso de Cell Search, dónde el parámetro más difícil de deducir con un simple vistazo puede ser NDLRB denota el número de *Resource Blocks* en el enlace descendente, cuyo rango varía entre 6 y 110.

```
Performing frequency offset estimation...
```

```
Frequency offset: 21.127Hz
```

```
Performing OFDM demodulation...
```

Se obtiene la frecuencia de offset

```
Performing MIB decoding...
```

```
Cell-wide settings after MIB decoding:
```

```
    NDLRB: 50
```

```
DuplexMode: 'FDD'
CyclicPrefix: 'Normal'
NCellID: 77
NSubframe: 0
CellRefP: 2
PHICHDuration: 'Normal'
Ng: 'One'
NFrame: 470
```

Estas son las características de la celda después de decodificar el MIB (nótese que ahora el NDLRB es 50 en vez de 6).

Restarting reception now that bandwidth (NDLRB=50) is known...

Resampling from 5.282Ms/s to 15.360Ms/s...

Performing frequency offset estimation...

Frequency offset: 39.100Hz

Performing timing offset estimation...

Timing offset to frame start: 135188 samples

Se vuelve a empezar la recepción conocido el NDLRB y se obtienen parámetros como la frecuencia de offset o las muestras de offset.

Performing OFDM demodulation...

SIB1 decoding for frame 470

Decoding CFI...

Decoded CFI value: 2

Se conforma la demodulación OfdM y se decodifica SIB1 para la trama 470. El valor del CFI decodificado es 2.

Punto de ruptura

PDCCH search for SI-RNTI...

DCI message with SI-RNTI:

```
        DCIFormat: 'Format1A'
        CIF: 0
AllocationType: 0
Allocation: [1x1 struct]
ModCoding: 3
        HARQNo: 0
        NewData: 0
        RV: 1
        TPCPUCCH: 1
        TDDIndex: 0
        SRSRequest: 0
        HARQACKResOffset: 0
```

Se hace una búsqueda PDCCH para SI_RNTI y el DCI tiene los campos y valores que se reflejan arriba.

PDSCH settings after DCI decoding:

```
        RNTI: 65535
        PRBSet: [8x1 uint64]
NLayers: 2
        CSI: 'On'
Modulation: {'QPSK'}
        RV: 1
        TxScheme: 'TxDiversity'
NTurboDecIts: 5
```

Decoding SIB1...

PDSCH RMS EVM: 316.474%

PDSCH Peak EVM: 1734.142%

SIB1 CRC: 1

La interfaz muestra la siguiente información gráfica:

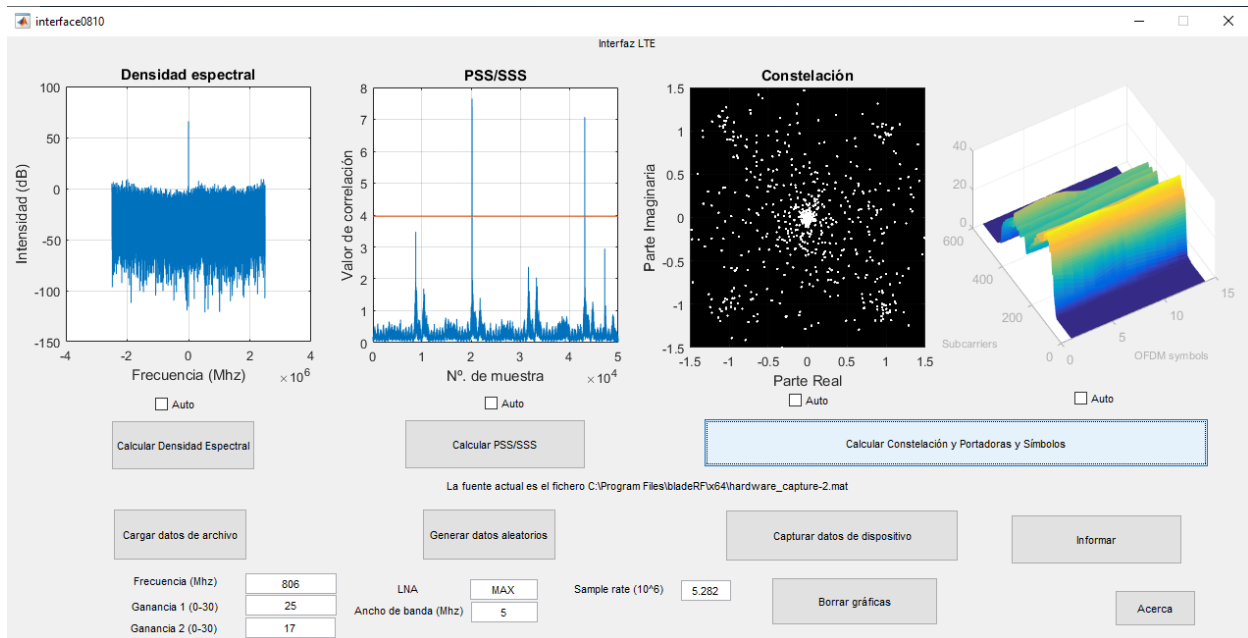


Figura 36: Captura típica Vodafone Señal LTE

```
function BOTON_GENERAR_ALEAT_Callback(hObject, eventdata, handles)
```

Esta función permite generar una señal con parámetros aleatorios, que están dentro de unos rangos que se han establecido, cada vez que se pulse el botón. Además, almacena los datos en los *handles* y muestra estos valores en los *edit text* de la interfaz gráfica.

En la ventana de comandos se muestra lo siguiente:

Generando datos aleatorios en memoria:

Los puntos en *nodob* son 150000 y su media total es de $0.499727 + 0.500330 \cdot i$.

El valor almacenado en *handles.bandwidth* es 15000000.

El valor almacenado en *handles.frequency* es 1779000000.

El valor almacenado en *handles.samplerate* es 6114073.

El valor almacenado en *handles.vga1* es 22.

El valor almacenado en `handles.vga2` es 25.

Cada vez que pulsamos el botón, generamos unos parámetros aleatorios dentro del rango que hemos definido.

Esta función nos permite comprobar que señales puramente aleatorias no nos permiten obtener datos relacionados con LTE, ya que no se trata de una señal real capturada y no permite su análisis de procesamiento de la señal.

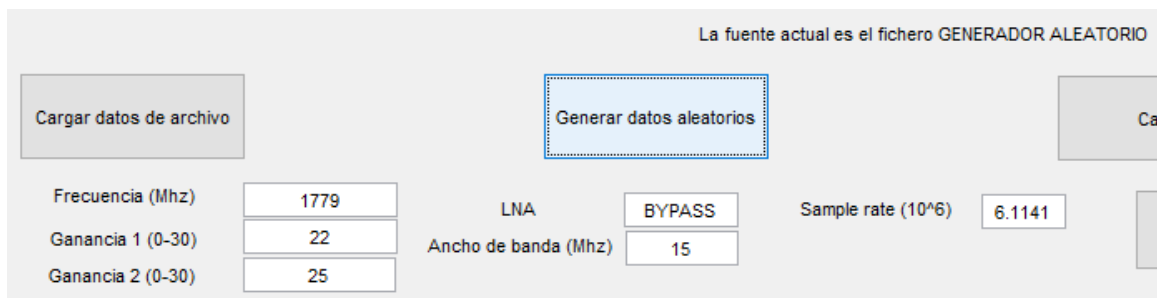


Figura 37: Resultado Botón Generar Datos Aleatorios

```
function BOTON_DENSIDAD_Callback(hObject, eventdata, handles)
```

Es el *callback* del botón "**Calcular Densidad Espectral**" de nuestra interfaz y se ejecuta cuando presionamos en el botón. En este caso, se hace una llamada a la función "**CalcularRepresentarDensidadEspectral.m**".

Esta función calcula la densidad espectral de energía de la señal de la señal previamente cargada de un archivo *.mat* o de una señal creada aleatoriamente.

```
function BOTON_PSSSSS_Callback(hObject, eventdata, handles)
```

Se trata del *callback* del botón "**Calcular PSS/SSS**" de nuestra interfaz que se ejecuta al presionar el botón.

Esta función mide el nivel de correlación entre las señales PSS y SSS con respecto a un umbral. Si el nivel de correlación entre las señales está por encima del umbral, vamos a poder seguir procesando la señal LTE; en caso de que ese nivel sea bajo, nos muestra un *warning* indicando que el nivel de correlación entre las señales es débil.

```
function BOTON_CONSTELACION_Callback(hObject, eventdata, handles)
```

Es el *callback* del botón "**Calcular Constelación y Portadoras y Símbolos**" de la interfaz.

Esta función hace una llamada a otra función: "**CalcularRepresentarConstelacion.m**", en la que se calcula y representa la constelación de puntos en el plano imaginario (parte real y parte imaginaria y también calcula y representa la figura en 3 dimensiones de las subportadoras y los OFDM símbolos.

```
function BOTON_HARDWARE_Callback(hObject, eventdata, handles)
```

Esta función integra todo el procesamiento y gráficos resultantes del análisis de una señal LTE que se está capturando a través del hardware conectado al puerto USB 3.0 de nuestro equipo.

A través de los botones Calcular Densidad Espectral, Calcular PSS/SSS y Calcular Constelación y Portadoras y Símbolos, se puede representar cada uno de los gráficos asociados a cada función, pero en este caso, al pulsar el botón **Capturar datos de dispositivo**, permite una representación de hasta los 4 gráficos simultáneamente. Todo depende del *checkbox* que esté activado en ese momento perteneciente a cada una de las gráficas. (tenemos 4 *checkbox* con la función *callback*: `function checkbox_axesn_Callback(hObject, eventdata, handles` con $n=1, 2, 3$ y 4)) Mientras el hardware está capturando señales en tiempo real, podemos activar o desactivar cualquier *checkbox* y así visualizamos la gráfica que deseemos en cada instante.

Una vez pulsado el botón, la función está programada para que realice lo siguiente en este orden (sólo se describe lo más importante):

- Borra las gráficas que haya previamente representadas en la interfaz.
- Desactiva los demás botones para que no se puedan pulsar durante la ejecución.
- Borra los datos que haya precargados de los distintos parámetros y la señal
- Lee los parámetros introducidos por el usuario en la interfaz, teniendo en cuenta que estos datos se han analizado a través de las funciones de control para no pasarle unos parámetros fuera de rango al hardware de captura.
- Se le pasan los diferentes valores para que pueda iniciar una captura de la señal que tenga los parámetros deseados.

- A través del comando `b.receive()`, se empieza a capturar señal y se representa lo que el usuario desee, como he descrito anteriormente.

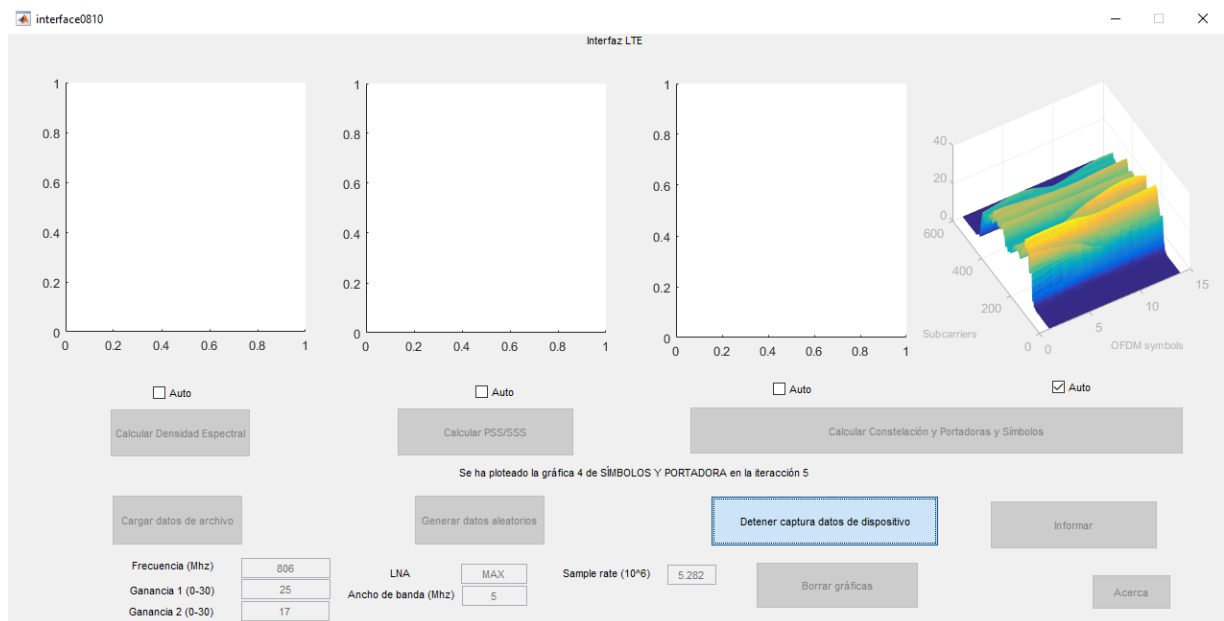


Figura 38: Resultado Botón Capturar datos de dispositivo

Cada una de las capturas se van almacenando en un archivo `.mat`, ya que nos pueden servir para cargarlo posteriormente y llevar a cabo su análisis más detenido que no nos permite la captura en tiempo real. Por tanto, para hacer pruebas y que no se generaran gran cantidad de archivos, se ha puesto un valor máximo de iteraciones.

No obstante, si el usuario pulsa de nuevo el botón, ahora nombrado como **Detener captura datos de dispositivo**, puede detener la captura sin tener que esperar a que el programa realice el máximo número de iteraciones.

En este caso, al pulsar el botón en este estado, los demás botones se habilitan para que el usuario pueda realizar la misma o cualquier otra acción.

```
function BOTON_INFORMAR_Callback(hObject, eventdata, handles)
```

Esta función invoca a la función **"InformarEstadoGeneral.m"** y permite mostrar información bastante información general del programa:

- Si la captura de la tarjeta está activada o detenida

- El número de puntos o muestras de *nodob*
- El valor de parámetros como frecuencia, ancho de banda, *samplerate* o ganancia, de señal almacenada y cargada, señal aleatoria o señal capturada por la tarjeta. En este último caso, mostrará los parámetros que nosotros hayamos introducido en el *edit text* de cada uno de estos parámetros previamente a iniciar la captura del dispositivo.
- Un registro de tiempo de entrada al programa y tiempo de ejecución.

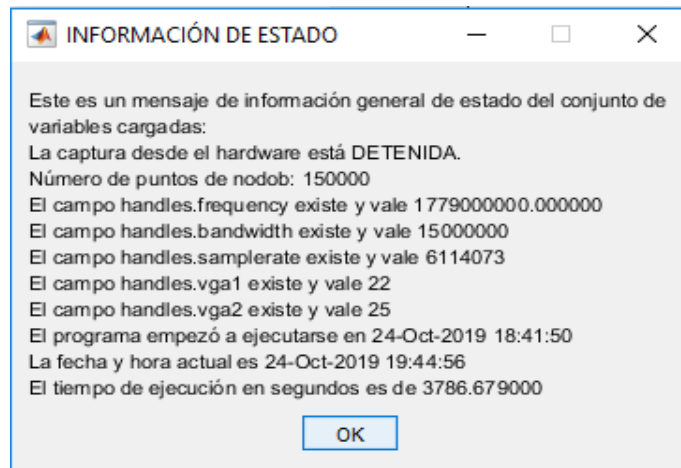


Figura 39: Resultado Botón Informar

```
function BOTON_BORRARGRAFICAS_Callback(hObject, eventdata, handles)
```

Esta función llama a "**borrar_graficas.m**".

Se encarga de borrar las gráficas existentes en cada uno de los axes de la interfaz (*handles.axes1*, *handles.axes2*, *handles.axes3* y *handles.axes4*).

```
function controlLNA_Callback(hObject, eventdata, handles)
```

En esta función se controla que el parámetro LNA exista y tenga un valor apropiado. Los valores apropiados de esta variable son: 'BYPASS', 'MID' y 'MAX'.

Además, si el usuario introduce el valor en minúscula, el valor se transforma a mayúscula a través de la función de Matlab ***upper***.

Si el usuario ha introducido un valor correcto, se muestra el valor correcto en la interfaz; en caso contrario, si el usuario ha introducido un valor que no es correcto, se genera un valor aleatoriamente entre los 3 valores posibles.

Se distinguen en la función 2 opciones: si contamos con una versión moderna como es la R2018b, la sintaxis de código es diferente que cuando usamos la versión R2016a, que es la que principalmente he usado en este proyecto, ya que es la primera versión en la que se introduce la *LTE Toolbox*.

```
function controlGANANCIA2_Callback(hObject, eventdata, handles)
```

Esta función controla que el valor introducido por el usuario este comprendido entre los valores dentro del rango que soporta la tarjeta; si el usuario introduce un valor que no está dentro del rango, se asigna el valor mínimo o máximo. En este caso, el rango de la ganancia 2 (parámetro *vga2* de la tarjeta) va de 1 a 30.

```
function controlGANANCIA1_Callback(hObject, eventdata, handles)
```

Al igual que la función anterior, controla que el valor de la ganancia 1 (parámetro *vga1* de la tarjeta) esté dentro del rango 1 a 30.

```
function controlFRECUENCIA_Callback(hObject, eventdata, handles)
```

En esta función se controla que la frecuencia central (parámetro *frequency* de la tarjeta) introducida por el usuario en la interfaz esté dentro del rango de frecuencias que maneja el hardware. La frecuencia mínima es 300 MHz y la frecuencia máxima es 3.8 GHz, por tanto, si el usuario introduce un valor fuera de rango, se asigna un valor dentro del rango. El usuario tiene que introducir el valor en MHz, al igual que el ancho de banda y la frecuencia de muestreo.

```
function controlANCHO_DE_BANDA_Callback(hObject, eventdata, handles)
```

En este caso, la función controla el parámetro ancho de banda (*bandwidth* de la tarjeta) que va a manejar el hardware. Se verifica que esté dentro del rango 1.5 MHz a 28 MHz, que es el ancho de banda máximo que nos permite.

```
function controlSAMPLERATEMHZ_Callback(hObject, eventdata, handles)
```

En este caso, se controla que la frecuencia de muestreo (parámetro *samplerate* de la tarjeta) esté dentro del rango 1 MHz a 40 MHz.

```
function BOTON_ACERCA_Callback(hObject, eventdata, handles)
```

Muestra información del autor del trabajo, versión de Matlab y fecha en un mensaje que se muestra al pulsar el botón.

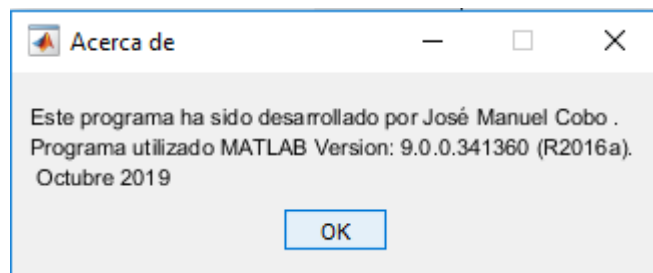


Figura 40: Resultado Botón Acerca de