

Trabajo Fin de Grado
Grado en Ingeniería Electrónica, Robótica y
Mecatrónica

Diseño, Fabricación y Testeo de PCB basada en el
Transceptor SX1257, para la Experimentación con
Comunicaciones Digitales en FPGA

Autor: Carlos Cabrera Maldonado

Tutor: Hipólito Guzmán Miranda

Dpto. de Ingeniería Electrónica
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2025



DINEL
DPTO. INGENIERÍA ELECTRÓNICA

Trabajo Fin de Grado
Grado en Ingeniería Electrónica, Robótica y Mecatrónica

Diseño, Fabricación y Testeo de PCB basada en el Transceptor SX1257, para la Experimentación con Comunicaciones Digitales en FPGA

Autor:

Carlos Cabrera Maldonado

Tutor:

Hipólito Guzmán Miranda

Profesor titular

Dpto. de Ingeniería Electrónica
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla
Sevilla, 2025

Trabajo Fin de Grado: Diseño, Fabricación y Testeo de PCB basada en el Transceptor SX1257, para la Experimentación con Comunicaciones Digitales en FPGA.

Autor: Carlos Cabrera Maldonado

Tutor: Hipólito Guzmán Miranda

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2025

El Secretario del Tribunal

A Julián, Clara, Víctor y Jesús.

*Algún día entenderéis todo lo que
está aquí escrito.*

AGRADECIMIENTOS

A mi familia, que desde que tengo uso de razón, ha perseguido mi felicidad. En especial a mi padre, que me alentó a ser ingeniero y me ha apoyado en este y muchos otros proyectos.

A mi grupo scout, que me enseñó una forma diferente de vivir. Gracias a Marco, Javi y Pablo por todas esas risas en las tiendas.

A mis amigos de Córdoba, que tuve el placer de conocer con cuatro años y aún forman parte de mi vida. Carlos y Fabio, sois para mí una suerte inmensurable. Entre los tres vamos a llegar muy lejos.

A mis profesores del Colegio Almanzor y a los de la Universidad, de los que tanto he aprendido. Por supuesto, gracias a Hipólito, mi tutor, por su paciencia y energía infinitas.

A Rafa, Hugo, Ignacio y Adrián, grandes aliados a la hora de hacer trabajos y de salir a tomar algo. A Edu, Tomeu y Jesús, por hacer de 2023 un año inolvidable. A Álvaro, aunque solo coincidimos en verano, no me olvido de ti.

Y a Miriam, que, en muy poco tiempo, ha llegado a significar tanto. Gracias por hacerme las cosas fáciles.

En conjunto, todos habéis contribuido a moldear la persona que soy hoy, de la cual estoy muy orgulloso. Sin vosotros, este trabajo sencillamente no es posible. Sin nada más que añadir,

Carlos

RESUMEN

Este trabajo pretende contribuir a la mejora de la calidad de enseñanza en el Máster U. en Ingeniería de Telecomunicación de la Universidad de Sevilla, mediante la implementación de una PCB como periférico compatible con tarjetas de desarrollo de FPGA, que permita dejar atrás la simulación y experimentar con la transmisión y recepción de señales en entorno real.

El proyecto lo componen diversas tareas, que abarcan de principio a fin la realización de un sistema de telecomunicación digital, desde la exposición del mercado actual de sistemas de radiofrecuencia y el estudio de una solución económica, hasta el diseño de la PCB, su fabricación y su testeo. Finalmente, se lleva a cabo una demostración que da pie a una serie de conclusiones y trabajos futuros.

Como valor añadido, se ahonda brevemente en la física de las telecomunicaciones y se estudia el manejo de la tarjeta con el uso exclusivo de herramientas libres, dando una visión general del trabajo muy completa y enfocada al usuario que le dará uso: el alumnado.

ABSTRACT

This work aims to contribute to the improvement of teaching quality in the University Master's Degree in Telecommunication Engineering at the University of Seville, through the implementation of a PCB as a peripheral compatible with FPGA development boards. This approach makes it possible to move beyond simulation and to experiment with signal transmission and reception in a real-world environment.

The project consists of several tasks that cover, from start to finish, the development of a digital telecommunications system: from an overview of the current radio-frequency systems market and the study of a cost-effective solution, to the PCB design, its fabrication, and testing. Finally, a demonstration is carried out, leading to a set of conclusions and proposals for future work.

As an added value, the project briefly delves into the physics of telecommunications and examines the operation of the board using exclusively open-source tools, providing a comprehensive overview of the work that is clearly oriented towards its end users: the students.

ÍNDICE

| | |
|--|--------------|
| Agradecimientos | IX |
| Resumen | XI |
| Abstract | XIII |
| Índice | XV |
| Índice de Tablas | XVII |
| Índice de Figuras | XIX |
| Abreviaturas y Acrónimos | XXIII |
| 1 Introducción | 1 |
| 1.1 Problema | 1 |
| 1.2 Estado del arte | 4 |
| 1.3 Solución | 6 |
| 1.4 Alcance y estructura | 9 |
| 2 Entorno de trabajo | 11 |
| 2.1 Sistema operativo | 11 |
| 2.2 KiCad | 12 |
| 2.3 Micro-Cap 12 | 14 |
| 2.4 iCEBreaker | 14 |
| 2.4.1 Hardware | 14 |
| 2.4.2 NEORV32 | 20 |
| 2.4.3 Programación de la FPGA | 22 |
| 3 Fundamentos de telecomunicaciones | 25 |
| 3.1 Introducción | 25 |
| 3.2 Teorema de Fourier | 26 |
| 3.3 Transmisores y receptores | 29 |
| 3.4 Modulación y demodulación | 32 |
| 3.5 Sistema de comunicación digital | 35 |
| 3.6 SDR y GDR | 36 |
| 3.7 Modulación Sigma-Delta | 37 |
| 3.8 Protocolos de comunicación digital | 39 |
| 3.8.1 UART | 39 |
| 3.8.2 SPI | 40 |
| 3.8.3 I ² C | 41 |
| 4 Descripción del PMOD | 43 |
| 4.1 Visión general | 43 |
| 4.2 Electrónica básica | 47 |
| 4.3 El transceptor SX1257 | 48 |
| 4.3.1 Descripción del circuito | 48 |
| 4.3.2 Bloque de transmisión | 50 |
| 4.3.3 Bloque de recepción | 52 |
| 4.3.4 Registros | 52 |
| 4.3.5 Otras funciones | 57 |
| 4.4 SC18IS602B: El puente I ² C-SPI | 59 |
| 4.5 Circuitería de las antenas | 63 |

| | |
|--|-----------|
| 4.6 Componentes escogidos y presupuesto | 64 |
| 5 Pruebas con el PMOD | 69 |
| 5.1 Búsqueda de herramientas de software | 69 |
| 5.2 Pruebas con la revisión B | 71 |
| 5.3 Análisis y modificaciones de la revisión C | 76 |
| 5.4 Puesta a punto de la revisión D | 78 |
| 6 Demostración práctica con el PMOD | 81 |
| 6.1 Preparaciones previas | 81 |
| 6.2 Demostración en montaje local | 84 |
| 6.3 Demostración en montaje completo | 86 |
| 7 Conclusiones y trabajo futuro | 89 |
| 7.1 Conclusiones | 89 |
| 7.2 Trabajo futuro | 90 |
| Referencias | 91 |

ÍNDICE DE TABLAS

| | |
|---|----|
| Tabla 1. LUT para una puerta AND. | 15 |
| Tabla 2. Fases de QPSK. | 33 |
| Tabla 3. Valores de los componentes en fase y cuadratura. | 34 |
| Tabla 4. Listado de pines de E/S de la revisión D. | 44 |
| Tabla 5. Configuración de los GPIOs. | 62 |
| Tabla 6. Hoja de componentes. | 66 |
| Tabla 7. Hoja de gastos. | 67 |
| Tabla 8. Configuración del SC18IS602B. | 72 |
| Tabla 9. Configuración del SX1257. | 73 |

ÍNDICE DE FIGURAS

| | |
|---|----|
| Figura 1. Propuesta para la asignatura. Bloques a diseñar y simular por el alumnado. | 2 |
| Figura 2. Presupuesto del Departamento de Ingeniería Electrónica. | 3 |
| Figura 3. XRF16 Xilinx RFSoc Gen3 System-On-Module. Ofertado por Avnet. | 4 |
| Figura 4. Zynq UltraScale + RFSoc. Ofertado también por Avnet. | 4 |
| Figura 5. RFSoc 4x2. Ofertado por Real Digital. | 5 |
| Figura 6. HamGeek 70MHz-6GHz Zynq7020+AD9363 SDR. Ofertado por HamGeek. | 5 |
| Figura 7. HamGeek 1MHz to 6GHz HackRF One R10C V2.0.x. Ofertado también por HamGeek. | 6 |
| Figura 8. Diseño de la iCEBreaker en KiCad. | 7 |
| Figura 9. Diferentes PMODs para la iCEBreaker. | 7 |
| Figura 10. Diseño KiCad de la revisión C del PMOD. | 8 |
| Figura 11. Bloques a diseñar por el alumnado, llevados a la práctica usando el PMOD SX1257. | 8 |
| Figura 12. Esquemático de la iCEBreaker. | 12 |
| Figura 13. Layout de la iCEBreaker. | 13 |
| Figura 14. Pistas de cobre de la iCEBreaker. | 13 |
| Figura 15. FPGA de la tarjeta iCEBreaker. | 15 |
| Figura 16. Puerta AND de 4 entradas. | 15 |
| Figura 17. Diagrama de un CLB estándar. | 16 |
| Figura 18. Celda SRAM. | 17 |
| Figura 19. FT2232H de la iCEBreaker. | 18 |
| Figura 20. Bucle de control de un PLL. | 18 |
| Figura 21. Oscilador en anillo. | 19 |
| Figura 22. Arquitectura de Von Neumann. | 20 |
| Figura 23. Microcontrolador. | 20 |
| Figura 24. Esquema de la NEORV32. | 21 |
| Figura 25. Ejemplo. Puerta AND en VHDL. | 22 |
| Figura 26. Ejemplo. Puerta AND en formato JSON. | 23 |
| Figura 27. Sistema de comunicación I. | 25 |
| Figura 28. Sistema de comunicación II. | 26 |
| Figura 29. Señal analógica versus señal digital. | 26 |
| Figura 30. Construcción de una señal cuadrada mediante armónicos. | 27 |
| Figura 31. Transformada de Fourier de un pulso. | 29 |

| | |
|---|----|
| Figura 32. Ancho de banda. | 29 |
| Figura 33. Onda electromagnética. | 30 |
| Figura 34. Generación de una onda electromagnética. | 31 |
| Figura 35. Proceso de modulación (en este caso, la AM). | 32 |
| Figura 36. ASK, FSK y PSK. | 33 |
| Figura 37. Diagrama de constelación QPSK. | 34 |
| Figura 38. Diagrama de constelación 16-QAM. | 34 |
| Figura 39. Sistema de comunicación digital. | 36 |
| Figura 40. Modulación Sigma-Delta. | 37 |
| Figura 41. Modulador Sigma-Delta. | 38 |
| Figura 42. Esquema de conexión del protocolo UART. | 39 |
| Figura 43. Esquema de conexión del protocolo SPI. | 40 |
| Figura 44. Esquema de conexión del protocolo I ² C. | 41 |
| Figura 45. Protocolo I ² C. | 42 |
| Figura 46. Diseño KiCad de la Revisión D, vista superior. | 43 |
| Figura 47. Diseño KiCad de la Revisión D, vista inferior. | 44 |
| Figura 48. Layout de la revisión D, capa superior. | 45 |
| Figura 49. Esquemático de la revisión D. | 46 |
| Figura 50. Revisión D. Electrónica básica. | 47 |
| Figura 51. Revisión D. SX1257. | 48 |
| Figura 52. Esquema del SX1257 [44]. | 49 |
| Figura 53. Revisión D. Cristal de cuarzo. | 49 |
| Figura 54. SX1257. Front-end del transmisor. | 50 |
| Figura 55. SX1257. Front-end del receptor. | 52 |
| Figura 56. SX1257. Protocolo SPI. | 53 |
| Figura 57. SX1257. Registros. | 53 |
| Figura 58. SX1257. Registros de modo de funcionamiento. | 54 |
| Figura 59. SX1257. Registros del PLL. | 54 |
| Figura 60. SX1257. Registros de revisión. | 55 |
| Figura 61. SX1257. Registros del transmisor. | 55 |
| Figura 62. SX1257. Registros del receptor. | 56 |
| Figura 63. SX1257. Registros de los puertos DIO (Digital Input/Output). | 56 |
| Figura 64. SX1257. Registros adicionales. | 57 |
| Figura 65. SX1257. Loop-back. | 58 |
| Figura 66. SX1257. Mapa de DIOs. | 58 |
| Figura 67. Revisión D. SC18IS602B. | 59 |
| Figura 68. SC18IS602B. Dirección de esclavo. | 59 |

| | |
|---|----|
| Figura 69. SC18IS602B. Escritura y lectura en el buffer. | 60 |
| Figura 70. SC18IS602B. FunctionID para comunicación SPI. | 60 |
| Figura 71. SC18IS602B. Byte de configuración para reloj SPI. | 61 |
| Figura 72. SC18IS602B. Byte de habilitación de GPIOs. | 62 |
| Figura 73. SC18IS602B. Byte de configuración de GPIOs. | 63 |
| Figura 74. Revisión D. Circuitería de las antenas. | 63 |
| Figura 75. Revisión D del PMOD. | 64 |
| Figura 76. Declaración de las señales del TWI. | 70 |
| Figura 77. Esquema de códigos y señales I. | 71 |
| Figura 78. Sobreescribiendo un registro del SX1257. | 73 |
| Figura 79. Errores de la revisión B. | 75 |
| Figura 80. Si514. | 76 |
| Figura 81. Dudas en el esquemático. | 77 |
| Figura 82. Simulación en Micro-Cap 12. Arriba: 10 mH. Abajo: 10 nH. | 77 |
| Figura 83. Analizador de espectros. | 78 |
| Figura 84. Diagrama de radiación de las antenas. | 79 |
| Figura 85. Sistema completo. | 79 |
| Figura 86. SX1257. Propuesta del modulador Sigma-Delta. | 81 |
| Figura 87. Esquema de códigos y señales II. | 83 |
| Figura 88. Montaje local. | 84 |
| Figura 89. Demostración en montaje local. | 84 |
| Figura 90. Señal piramidal atenuada y distorsionada, recibándose aún sin haber antena de recepción. | 85 |
| Figura 91. Montaje completo. Half-duplex. | 86 |
| Figura 92. Señal triangular solapándose con una señal piramidal. | 86 |
| Figura 93. Transmisión de señal triangular distorsionada. | 87 |
| Figura 94. Demostración en montaje completo. Half-duplex. | 87 |
| Figura 95. Montaje completo. Full-duplex. | 88 |
| Figura 96. Demostración en montaje completo. Full-duplex. | 88 |

ABREVIATURAS Y ACRÓNIMOS

| | |
|--------|--|
| MIT | Máster U. en Ingeniería de Telecomunicación |
| GIIT | Grado en Ingeniería de las Tecnologías de Telecomunicación |
| VHDL | VHSIC Hardware Description Language |
| VHSIC | Very High Speed Integrated Circuit |
| TFG | Trabajo de Fin de Grado |
| BOUS | Boletín Oficial de la Universidad de Sevilla |
| SDR | Software Defined Radio |
| GDR | Gateway-Defined Radio |
| FPGA | Field Programmable Gate Array |
| EEES | Espacio Europeo de Educación Superior |
| RFSoc | Radio Frequency System on Chip |
| IDE | Integrated Development Environment |
| PCB | Printed Circuit Board |
| PMOD | Peripheral Module |
| THT | Through-Hole Technology |
| SMD | Surface-Mount Device |
| GND | Ground |
| VEE | Voltage at the Emitter |
| VCC | Voltage at the Collector |
| BOM | Bill Of Materials |
| CPL | Component Placement List |
| CLB | Configurable Logic Block |
| LUT | Look Up Table |
| FF | Flip-Flop |
| SRAM | Static Random Access Memory |
| FIFO | First Input First Output |
| MOSFET | Metal Oxide Semiconductor Field-Effect Transistors |
| USB | Universal Serial Bus |
| SPI | Serial Peripheral Interface |
| UART | Universal Asynchronous Receiver/Transmitter |
| IC | Inter-Integrated Circuit |
| PLL | Phase-Locked Loop |
| VCO | Voltage Controlled Oscillator |
| DSP | Digital Signal Processing |
| IP | Intellectual Property |
| PWM | Pulse Width Modulation |
| LED | Light Emitting Diode |
| CPU | Central Processing Unit |
| ALU | Arithmetic/Logic Unit |

| | |
|-------|--|
| CU | Control Unit |
| ISA | Instruction Set Architecture |
| RISC | Reduced Instruction Set Computing |
| GPIO | General Purpose Input/Output |
| TWI | Two Wire Interface |
| JSON | JavaScript Object Notation |
| RTLIL | Register Transfer Level Intermediate Language |
| ASC | ASCII Configuration |
| ASCII | American Standard Code for Information Interchange |
| AM | Amplitud Modulation |
| FM | Frecuency Modulation |
| PM | Phase Modulation |
| ASK | Amplitude Shift Keying |
| FSK | Frequency Shift Keying |
| PSK | Phase Shift Keying |
| BSK | Binary Phase Shift Keying |
| QPSK | Quadrature Phase Shift Keying |
| QAM | Quadrature Amplitude Modulation |
| OFDM | Orthogonal Frequency Division Multiplexing |
| UIT | Unión Internacional de Telecomunicaciones |
| VHF | Very High Frequency |
| UHF | Ultra High Frequency |
| TX | Transmit |
| RX | Receive |
| MOSI | Master Out Slave In |
| MISO | Master In Slave Out |
| SCLK | Serial Clock |
| SS | Slave Select |
| SDA | Serial Data |
| SCL | Serial Clock |
| ACK | Acknowledgement |
| NACK | Not Acknowledgement |
| DC | Direct Current |
| AC | Alternating Current |
| EMI | ElectroMagnetic Interference |
| IF | Intermediate Frequency |
| ISM | Industrial, Scientific, Medical |
| PDS | Power Distribution Scheme |
| LNA | Low Noise Amplifier |
| CPOL | Clock Polarity |
| CPHA | Clock Phase |
| DIO | Digital Input/Output |
| VDD | Voltage at the Drain |
| VSS | Voltage at the Source |

| | |
|-----|-----------------------------|
| BJT | Bipolar Junction Transistor |
| MSB | Most Significant Bits |
| LSB | Least Significant Bits |
| SNR | Signal to Noise Ratio |

1 INTRODUCCIÓN

Pensar es fácil, actuar es difícil, y poner los pensamientos de uno mismo en acción es lo más difícil del mundo.

- Goethe -

Las telecomunicaciones siempre han sido una parte indispensable en el desarrollo tecnológico que envuelve a la humanidad. La posibilidad de intercambiar información de forma telemática permite al ser humano comportarse como una única sociedad repartida por lo ancho y largo del planeta, en la que cada avance obtenido conforma un nuevo escalón sobre el que apoyarse, evitando que cada civilización suba por escaleras diferentes.

El margen de mejora de esta tecnología es inmenso. Desde las señales de humo utilizadas por las civilizaciones antiguas, hasta la invención del teléfono por Alexander Graham Bell en 1876 [1]. Pero fue en 1897 cuando Guglielmo Marconi logra enviar la primera señal transatlántica sin el uso de ningún cable [2], iniciando la era de las telecomunicaciones inalámbricas.

En la actualidad, numerosas tecnologías ampliamente utilizadas, como pueden serlo Bluetooth, Wi-Fi o los servicios de telefonía móvil, dan forma a lo que hace poco más de un siglo era ciencia ficción: transmitir y recibir información *por el aire*. No obstante, no se trata de magia, sino de física, y abordaré estos fenómenos en el tercer capítulo de este trabajo.

El Máster Universitario en Ingeniería de Telecomunicación de la Universidad de Sevilla [3] forma a personas para afrontar la infinidad de problemas que existen tras atender una llamada telefónica, enviar un mensaje de texto o acceder a una página web. Para comprender lo que hay detrás de estas acciones cotidianas se requiere un dominio absoluto de las matemáticas y la física, una base sólida de teoría de la señal y sesiones prácticas de transmisión y recepción de señales, entre muchas otras cosas. Es esta demanda de una enseñanza de calidad lo que origina la motivación de este trabajo.

1.1 Problema

El Máster Universitario en Ingeniería de Telecomunicación (MIT) es la continuación del Grado en Ingeniería de las Tecnologías de Telecomunicación (GITT) [4], y actúa como lanzadera al mundo laboral, capacitando a los estudiantes para desenvolverse con eficacia en el ámbito profesional de las telecomunicaciones.

Ya en el GITT, se imparte en múltiples asignaturas diseño digital con lenguajes de descripción hardware¹, siendo VHDL (VHSIC Hardware Description Language)² el más practicado dada su importancia en el entorno industrial europeo. Dichas competencias se retoman en el MIT con Electrónica Digital para Comunicaciones, donde se les da un uso más especializado y enfocado en las telecomunicaciones.

En esta asignatura, coordinada por D. Hipólito Guzmán Miranda, se describen en VHDL algunos de los bloques que conforman un sistema de telecomunicación digital, para posteriormente simular su respuesta y evaluar su desempeño, lo que permite detectar posibles errores e introducir mejoras. Actualmente solo se implementa el *estimador de canal*, pero en un futuro se pretende abarcar un sistema de radiofrecuencia completo, como el que se muestra en la Figura 1.

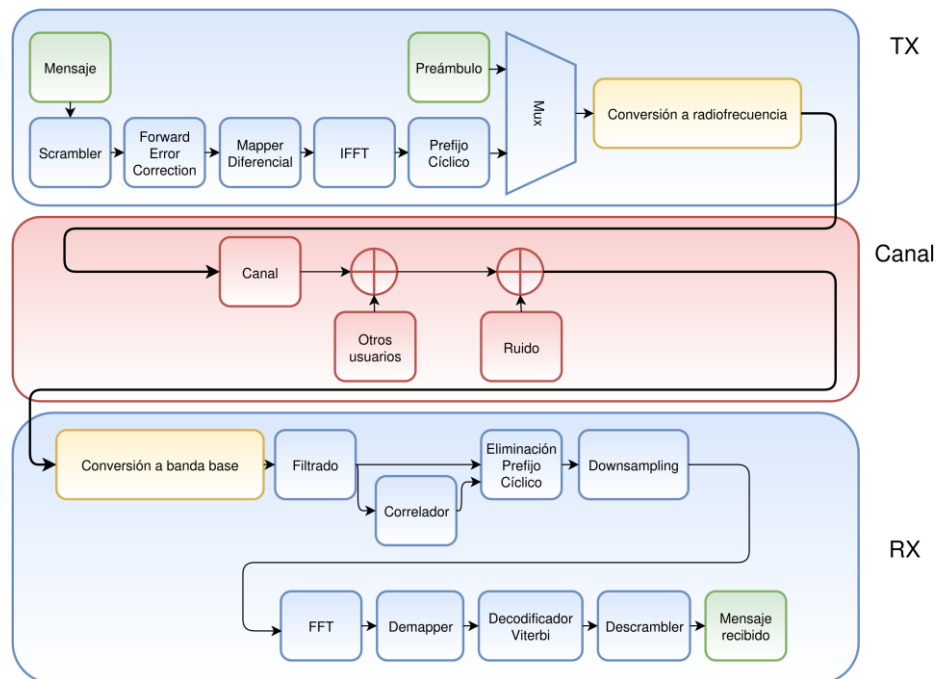


Figura 1: Propuesta para la asignatura. Bloques a diseñar y simular por el alumnado [5].

Esta innovadora propuesta, daría un enfoque de muy alto valor a la asignatura y permitiría al estudiante comprender e implementar el proceso que hay detrás del intercambio de información al completo. No obstante, el hecho de que los diseños realizados solo se lleven a la simulación es un gran limitante, pues no se están abarcando la infinidad de problemas que surgen al pasar de lo teórico a lo experimental. Para la correcta inserción del estudiante en el ámbito profesional, sería muy conveniente no solo afianzar los fundamentos teóricos en el entorno de simulación, si no plasmarlos también en un ensayo real, pues son estas prácticas las que dan la intuición y experiencia propias de un ingeniero.

D. Hipólito, tutor de este Trabajo de Fin de Grado (TFG) y muy involucrado en el mismo, se percató de este problema y planteó el desarrollo de este proyecto para dar respuesta a dichas necesidades. No obstante, proveer a todo el alumnado de los medios necesarios para poder transmitir y recibir señales de radio requiere una capacidad económica que muchas veces escapa a los medios del centro docente.

En la Universidad de Sevilla, todos los datos económicos son de dominio público y quedan plasmados el Anexo IV del BOUS (Boletín Oficial de la Universidad de Sevilla), anexo correspondiente al presupuesto del centro

¹ Lenguaje especializado que permite describir la estructura y el comportamiento de un circuito electrónico digital. En lugar de programar instrucciones que ejecuta un procesador (como en un lenguaje de programación tradicional), se definen cómo están conectados los componentes y cómo reaccionan las señales dentro de un circuito. Se modela y diseña el hardware antes de fabricarlo, de ahí su nombre.

² Un lenguaje de descripción hardware ampliamente utilizado. Las siglas VHSIC corresponden a Very High Speed Integrated Circuit.

para el año 2025 [6]. En el capítulo diez, Presupuesto de Gastos, se incluye una entrada para el Departamento de Ingeniería Electrónica, departamento encargado de la docencia de Electrónica Digital para Comunicaciones.

| D. INGENIERÍA ELECTRÓNICA | | | | 26.613,00 |
|---------------------------|-----|----|--|-----------|
| 18802810 | 422 | 2 | | 6.613,00 |
| 18802810 | 422 | 60 | | 10.000,00 |
| 18802810 | 422 | 66 | | 10.000,00 |

Figura 2: Presupuesto del Departamento de Ingeniería Electrónica [6].

Como puede observarse, los números son relativamente bajos, aún tratándose de una institución de renombre en España como lo es la Universidad de Sevilla. Considerando que el Departamento de Ingeniería Electrónica coordina decenas de asignaturas al mismo tiempo, tras deducir todos los gastos necesarios durante el ciclo lectivo quedará una cantidad muy escasa para innovar.

Una solución que permitiría llevar los conceptos de transmisión y recepción de señales al plano físico sin requerir equipamiento costoso es la radio definida por software (Software Defined Radio, SDR). Esta tecnología permite implementar mediante software algunas funciones que tradicionalmente se realizaban con hardware específico (como moduladores, codificadores o demoduladores, bloques que se comentarán en los siguientes capítulos), lo que ofrece una gran flexibilidad a bajo coste.

Sin embargo, en Electrónica Digital para Comunicaciones no se busca diseñar estas funciones en software, sino en un lenguaje de descripción hardware. Por lo tanto, este proyecto no se limita a SDR en su forma clásica, si no a una variación menos estandarizada: la Gateware-Defined Radio (GDR, término acuñado de expertos en el sector [7]), en la cual los bloques funcionales de la radio se implementan directamente en una FPGA (Field Programmable Gate Array)³, al reconfigurarla con su software⁴ correspondiente mediante lenguajes de descripción hardware como VHDL.

Por tanto, este trabajo parte de la necesidad de encontrar una solución económica que permita implementar GDR en la asignatura.

Por otra parte, aún disponiendo de los medios suficientes, no cualquier solución es óptima a largo plazo. La Universidad de Sevilla forma parte del EEES (Espacio Europeo de Educación Superior) y adapta su estructura académica a sus principios y directrices desde hace años. Un principio clave de esta iniciativa es el *aprendizaje de por vida*, es decir, promover la formación continua a lo largo de toda la vida, más allá de los estudios universitarios formales.

Esto queda muy bien reflejado en *Planteamientos Docentes e Investigadores y Proyecto de Investigación* [8]: si la docencia se impartiese con software propietario, cuyas licencias comerciales son especialmente costosas, como suele ser el caso del software de diseño electrónico, los egresados de la universidad no serán capaces de seguir aprendiendo de manera autónoma, salvo que tengan la suerte de trabajar en una empresa o institución que pueda permitirse dichas licencias.

El uso de herramientas de software libre (es decir, el autor libera su trabajo bajo una licencia que permite usarlo y modificarlo sin necesidad de pagar) y de código abierto (también llamado *open source*, se refiere a la disponibilidad total del código fuente) permite a los alumnos seguir aprendiendo independientemente de si siguen siendo alumnos de la universidad o no, si están trabajando en empresas que pueden costearse las licencias de software o no, o incluso si deciden emprender sus propios negocios.

En resumen, no solo es necesaria una solución económica, si no que además debe estar alineada con los principios del EEES, para lo que es necesario el uso de software y hardware libre.

³ En oposición al microcontrolador, con lógica interna fija, una FPGA es un chip digital reconfigurable que permite crear hardware a medida, sin necesidad de fabricar un circuito físico nuevo. Al reprogramarse, toda la lógica se modifica.

⁴ Cada FPGA está asociada a un programa informático específico para su configuración y uso, aunque algunas son compatibles con herramientas de terceros.

1.2 Estado del arte

Una forma eficiente de implementar GDR en Electrónica Digital para Comunicaciones, solventando así la falta de aprendizaje en entorno real, sería mediante la adquisición de varias réplicas de un RFSoc (Radio Frequency System on Chip) apropiado para el ámbito académico. Estas tarjetas integran en un solo chip un sistema completo de transmisión y recepción de señales por radiofrecuencia, reconfigurable a través de una FPGA. Se trata de una solución flexible que podría prestarse a los estudiantes durante todo el periodo lectivo del curso y así potenciar su aprendizaje autónomo.

A continuación, se hará un estudio del estado del mercado actual y se analizará si las soluciones encontradas encajan con el problema descrito.

Los RFSoc de mayor calidad ofrecen una funcionalidad excelente, pero requieren una inversión relativamente importante, y como se explicó en la sección anterior, el centro docente podría no disponer de los medios necesarios.

Avnet es una de las principales distribuidoras globales de componentes electrónicos y soluciones tecnológicas, actuando de puente entre los fabricantes y los consumidores. Podemos encontrar en su página web diferentes RFSoc a la venta.



Figura 3: XRF16 Xilinx RFSoc Gen3 System-On-Module. Ofertado por Avnet [9].

Como es de esperar, este producto proporciona una serie de funcionalidades demasiado avanzadas para el uso académico que se le daría a la placa, con un coste unitario de 29,995.00 €. Lo mismo ocurre con la tarjeta mostrada a continuación.

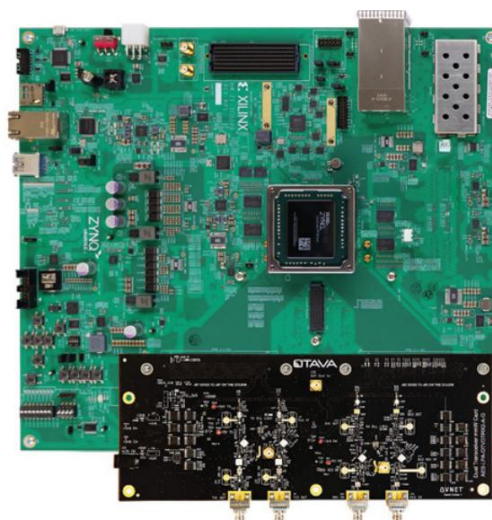


Figura 4: Zynq UltraScale + RFSoc. Ofertado también por Avnet [10].

Su coste es de 21,999.00 €. El patrón se repite en muchos más productos que no se nombrarán aquí. Como se indicó en la Figura 2, el Departamento de Ingeniería Electrónica apenas tiene presupuesto para comprar una de estas tarjetas, por lo que no es una solución viable.

Realizando una búsqueda más profunda, se encuentran productos mucho más económicos a costa de perder prestaciones. Real Digital es una empresa dedicada al diseño, fabricación y distribución de placas de circuitos, enfocando sus productos al uso académico en estudiantes universitarios.

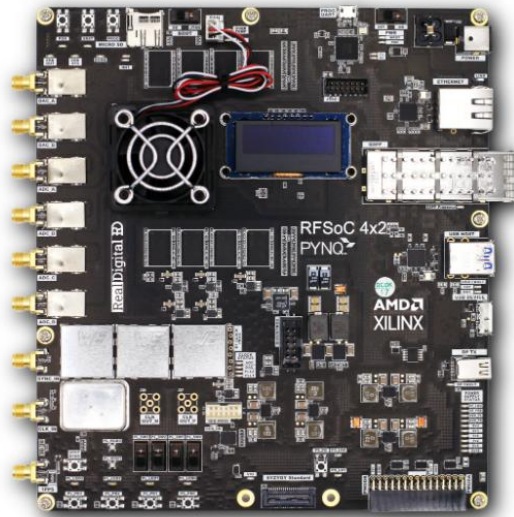


Figura 5: RFSoc 4x2. Ofertado por Real Digital [11].

Esta solución sería más que idónea para dotar al alumnado de flexibilidad para hacer toda la experimentación que se deseara. No obstante, el precio de cada placa alcanza los 2,499.99 €, por lo que sigue escapando a la capacidad económica de la universidad, aunque se repartan en grupos de varios alumnos.

Si el precio sigue descendiendo, se encuentran soluciones mucho más simples a costa de una pérdida significativa de calidad, fiabilidad y prestaciones. HamGeek es una marca china que produce y vende equipos de radiofrecuencia y accesorios relacionados. Muchos de sus productos son réplicas económicas de otros fabricantes.



Figura 6: HamGeek 70MHz-6GHz Zynq7020+AD9363 SDR. Ofertado por HamGeek [12].

Con un precio unitario de 108.26 €, esta tarjeta es totalmente asequible para la asignatura, aún tratándose de un grupo numeroso de alumnos matriculados. Hay productos incluso más baratos.



Figura 7: HamGeek 1MHz to 6GHz HackRF One R10C V2.0.x. Ofertado también por HamGeek [13].

A pesar de su coste de 79.68 €, la pérdida de prestaciones es demasiado importante como para pasarse por alto (entre muchas desventajas, destaca que esta tarjeta es *half-duplex*⁵).

No obstante, las FPGAs de estas dos últimas tarjetas pertenecen al fabricante Xilinx, por lo que es necesaria una licencia de Vivado⁶ para reprogramarlas. Actualmente puede utilizarse Vivado WebPACK, que es el entorno de desarrollo⁷ gratuito oficial de Xilinx, y se usa como estrategia comercial para atraer desarrolladores, investigadores y estudiantes hacia su ecosistema. No obstante, esta situación puede cambiar si Xilinx opta por otro modelo de negocio, obligando a usar la versión de pago.

Durante la etapa universitaria esto no es un problema, pues el centro se encarga de comprar paquetes de licencias informáticas para todo el alumnado. Sin embargo, al finalizar los estudios, se deberá adquirir por medios propios. Como se explicó en la sección anterior, lo ideal sería una tarjeta basada en software y hardware libre, asegurando al ingeniero que pueda seguir formándose durante toda su vida.

En resumen, el mercado actual no es del todo acorde a la solución que se busca. Si bien es cierto que existen soluciones económicas y con las prestaciones necesarias que podrían implementarse en Electrónica Digital para Comunicaciones, el estudiante podría estar aprendiendo procedimientos que no será capaz de replicar al finalizar sus estudios.

1.3 Solución

Dado que, a día de hoy, no existe un RFSoc que cumpla con todos los requisitos antes mencionados, este TFG estudia una solución alternativa.

El Departamento de Ingeniería Electrónica ya se encuentra en posesión de la iCEBreaker, una tarjeta basada en software y hardware libre, a excepción de su FPGA. Está diseñada especialmente para proyectos educativos y experimentales en electrónica digital.

⁵ No se puede transmitir y recibir a la vez. En caso contrario, recibe el nombre de *full-duplex*.

⁶ Entorno de desarrollo creado por Xilinx para diseñar, simular y programar FPGAs.

⁷ Un entorno de desarrollo (también llamado IDE, por sus siglas en inglés *Integrated Development Environment*) es un conjunto de herramientas de software donde el desarrollador puede escribir código, compilarlo o sintetizarlo, simularlo y detectar errores sin necesidad de recurrir a múltiples programas separados.

D. Hipólito, localizó en un repositorio de GitHub⁹ los archivos de fabricación de un PMOD [16], diseñado por el desarrollador Konrad Beckmann, para implementar GDR en tarjetas de desarrollo de FPGA. Basado en el transceptor¹⁰ SX1257, el diseño integra en una sola PCB (Printed Circuit Board)¹¹ toda la funcionalidad que requiere un sistema de telecomunicación digital.

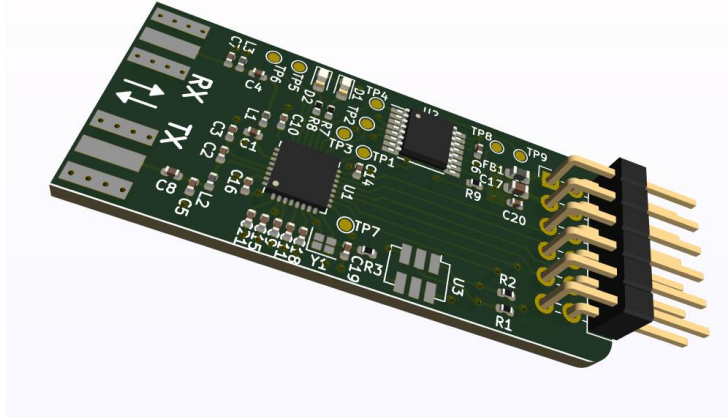


Figura 10: Diseño KiCad de la revisión C del PMOD [16].

El proyecto para fabricar este PMOD, que había pasado ya por tres revisiones (A, B y C), quedó en un segundo plano hace seis años. Konrad nunca llegó a conseguir las prestaciones que quería. No obstante, esta tarjeta, en combinación con la iCEBreaker de la que ya se disponía, podría tratarse de la solución idónea, al estar basada en software y hardware libre y ser su fabricación muy económica, como se explicará en el cuarto capítulo.

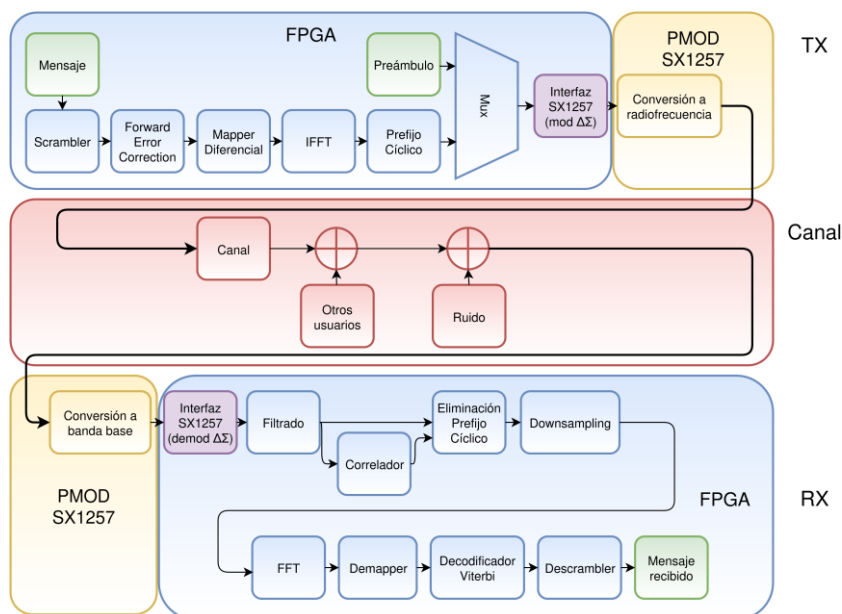


Figura 11: Bloques a diseñar por el alumado, llevados a la práctica usando el PMOD SX1257 [5].

⁹ GitHub es una plataforma en línea para alojar, compartir y colaborar en proyectos de software (aunque también se usa para documentación, hardware, datos y mucho más). Se basa en Git, un sistema de control de versiones que permite registrar todos los cambios en un proyecto. Otra plataforma muy similar y que aloja los archivos de este trabajo es GitLab.

¹⁰ Un transceptor es un dispositivo que cuenta con un transmisor y un receptor que comparten parte de la circuitería o se encuentran dentro del mismo encapsulado.

¹¹ Placa de circuito impreso en español, es una placa, generalmente de material aislante, sobre la que se montan y conectan componentes electrónicos.

1.4 Alcance y estructura

Dado que D. Hipólito ya disponía de una réplica de la revisión B del PMOD, el alcance de este TFG podría resumirse en los puntos siguientes:

1. Localizar las herramientas de software adecuadas para programar la tarjeta iCEBreaker.
2. Desarrollar el código necesario para configurar la revisión B.
3. Hacer pruebas con la revisión B y verificar su falta de prestaciones.
4. Estudiar la revisión C, realizar las modificaciones pertinentes y mandarla a fabricar.
5. Puesta a punto de la nueva tarjeta.
6. Transmitir y recibir una señal de forma satisfactoria.
7. Realizar un sistema de telecomunicación digital completo, donde se experimentase con distintos tipos de modulaciones.

El séptimo objetivo, aunque de alto interés y sin lugar a duda muy instructivo, queda fuera del alcance de este TFG, ya que requeriría de un tiempo excesivo para poder desarrollarlo adecuadamente. El cumplimiento de los seis primeros objetivos ya excedió considerablemente el plazo establecido en un inicio, pues surgieron multitud de problemas con los que lidiar entre los que cabe destacar:

- Falta de conceptos básicos: Al tratarse de una rama de la ingeniería totalmente desconocida para un ingeniero robótico, fue necesaria una extensa formación autónoma previa al desarrollo del proyecto. También se trató de diseñar los bloques de la radio en VHDL, sin que se fuese consciente de la magnitud de esta tarea (Electrónica Digital para Comunicaciones hace uso de un cuatrimestre completo para llevar a cabo una parte del receptor). Como se ha comentado antes, se optó por asignar esta tarea a futuras continuaciones de este trabajo.
- Problemas con las herramientas de software: Al tratarse de software libre, su mantenimiento recae en la comunidad, lo que lo hace más propenso a presentar fallos o limitaciones. En numerosas ocasiones, la carencia de determinadas prestaciones pasaba inadvertida y llegaba a confundirse con errores de programación cometidos por el alumno. Fue necesario evaluar distintas herramientas y versiones de las mismas, hasta identificar aquella que mejor se adaptaba a los requerimientos del proyecto.
- Demora en la fabricación de la PCB: La falta de comunicación, agravada por el inicio del verano, entre el alumno, la universidad y la tienda de electrónica que gestionaba el pedido llevó a realizar un primer encargo de una forma errónea sin que el alumno tuviera conocimiento de ello, lo que retrasó la finalización de este trabajo en más de un mes.

No obstante, el contenido de valor reside indudablemente en los objetivos que sí se han cumplido, los cuales se abordan de forma estructurada en los siguientes capítulos:

- Capítulo 1: Introducción.
Análisis de la problemática, las soluciones existentes actuales y en qué innova la solución aquí propuesta. Alcance y estructuración del proyecto.
- Capítulo 2: Entorno de trabajo.
Descripción de los recursos hardware y software utilizados, incluyendo la tarjeta iCEBreaker, las herramientas libres para su configuración y otros programas complementarios.

- Capítulo 3: Fundamentos de telecomunicaciones.
Conceptos básicos de las telecomunicaciones. SDR y GDR. Protocolos de comunicación digital.
- Capítulo 4: Descripción del PMOD.
Explicación de toda la electrónica que envuelve al sistema, con especial hincapié en los circuitos integrados. Datos de fabricación y hoja de costes.
- Capítulo 5: Pruebas con el PMOD.
Experimentación con la tarjeta, errores detectados y modificaciones propuestas. Puesta a punto de la nueva versión del PMOD.
- Capítulo 6: Demostración práctica con el PMOD.
Se verifica el funcionamiento del sistema realizando una transmisión y una recepción satisfactoria.
- Capítulo 7: Conclusiones y trabajo futuro.
Reflexiones finales del proyecto.

2 ENTORNO DE TRABAJO

Si buscas resultados distintos, no hagas siempre lo mismo.

- Albert Einstein -

Antes de profundizar en el mundo de las telecomunicaciones, resulta conveniente dedicar un capítulo a explicar todos los recursos, tanto software como hardware, empleados en el desarrollo de este TFG, pues estas herramientas no son pocas ni fáciles de comprender si no se está familiarizado con este tipo de entornos. En este apartado, se presenta la tarjeta iCEBreaker, junto con algunas de sus principales prestaciones, así como las distintas herramientas libres utilizadas para su configuración. Además, se describen primero otros programas complementarios empleados a lo largo del proyecto.

2.1 Sistema operativo

Puesto que es la plataforma más extendida para programar software libre, se ha escogido Linux¹² como el sistema operativo que soportará la mayoría de los programas informáticos del proyecto, y en concreto, la distribución¹³ Ubuntu.

Además, con el objetivo de optimizar el uso de la memoria del ordenador, se ha optado por emplear una máquina virtual en lugar de realizar una partición en el disco duro. Mediante VirtualBox es posible disponer de un entorno Linux y del espacio requerido para el desarrollo del proyecto, reservando únicamente 30 GB de la memoria principal.

Por otra parte, aunque todos los software empleados son soportados en Linux, los programas KiCad y Micro-Cap 12 han sido utilizados en Windows por simplicidad.

¹² Linux es un sistema operativo libre y de código abierto, basado en Unix y creado en 1991 por Linus Torvalds.

¹³ Las distribuciones de Linux (o distros) son diferentes versiones del sistema operativo Linux que incluyen, además del núcleo (kernel), un conjunto de programas, bibliotecas, entornos de escritorio y herramientas específicas adaptadas a distintos fines.

2.2 KiCad

KiCad es un software libre y de código abierto utilizado para el diseño de PCBs. Se trata de una herramienta muy completa e intuitiva que se organiza en varias fases:

- Primero, se dibuja el esquema eléctrico con símbolos normalizados, representando los distintos componentes y sus conexiones sin concretar en el espacio físico que ocupan.
- A continuación, se vuelca el esquema del circuito electrónico a la herramienta de diseño del *layout* o distribución de la PCB, donde a cada elemento del circuito se le asocia una huella o *footprint*. La huella define el tamaño y la forma de los *pads*¹⁴ del componente, además de indicar el método de ensamblado del mismo. En líneas generales, pueden ser de montaje traversante (THT, Through-Hole Technology) o de montaje superficial (SMD, Surface-Mount Device).
- Tras definir las dimensiones de la PCB, las huellas se colocan en la posición deseada y se realiza el conexionado de los *pads* mediante pistas¹⁵. En ocasiones, no se pueden realizar todas las conexiones sin que las pistas solapen unas con otras. En ese caso, es necesario que las pistas recorran la PCB en distintas capas. Una práctica muy utilizada es reservar una capa completa como plano de tierra (GND, Ground, también llamada a veces VEE, Voltage at the Emitter) y otra como plano de alimentación (VCC, Voltage at the Collector). De esta forma, se disminuye considerablemente el número de pistas, pues si se quiere alimentar o conectar a tierra un componente basta con taladrar *hacia abajo*.
- Una vez que todo está conectado siguiendo el esquema eléctrico, se dibuja la serigrafía¹⁶, que vendrá impresa sobre la máscara de soldadura¹⁷. La PCB está ahora lista para fabricarse.

Con este software se diseñó la tarjeta iCEBreaker, cuyo esquemático y *layout* se muestran a modo de ejemplo.

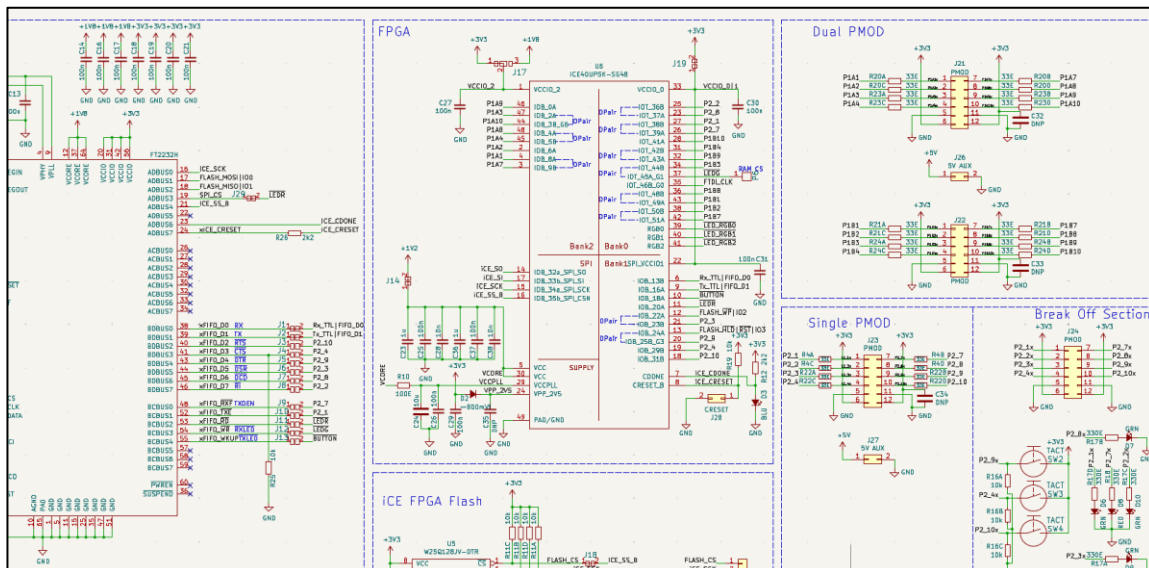


Figura 12: Esquemático de la iCEBreaker [17].

¹⁴ Son las superficies metálicas donde se sueldan los pines o terminales de los componentes electrónicos. Véase la Figura 13, donde las zonas rectangulares de color morado oscuro definen las huellas de los componentes SMD y las zonas circulares definen las huellas de los componentes THT.

¹⁵ Son las líneas de cobre impresas en la PCB que conectan eléctricamente los distintos componentes, funcionando como "cables planos" dentro de la placa. Véase la Figura 14 (caminos rojos, largos y estrechos).

¹⁶ Es la capa de impresión sobre la PCB que incluye textos, símbolos y marcas de referencia, útil para identificar componentes, orientaciones y facilitar el ensamblaje. Véase la Figura 13 (texto y líneas amarillas).

¹⁷ Capa protectora (habitualmente verde, aunque puede ser de otros colores) que cubre las pistas de cobre de la PCB y las protege de cortocircuitos, desgaste y oxidación, a la par que deja los pads al descubierto. Véase la Figura 8 (todo el material azul).

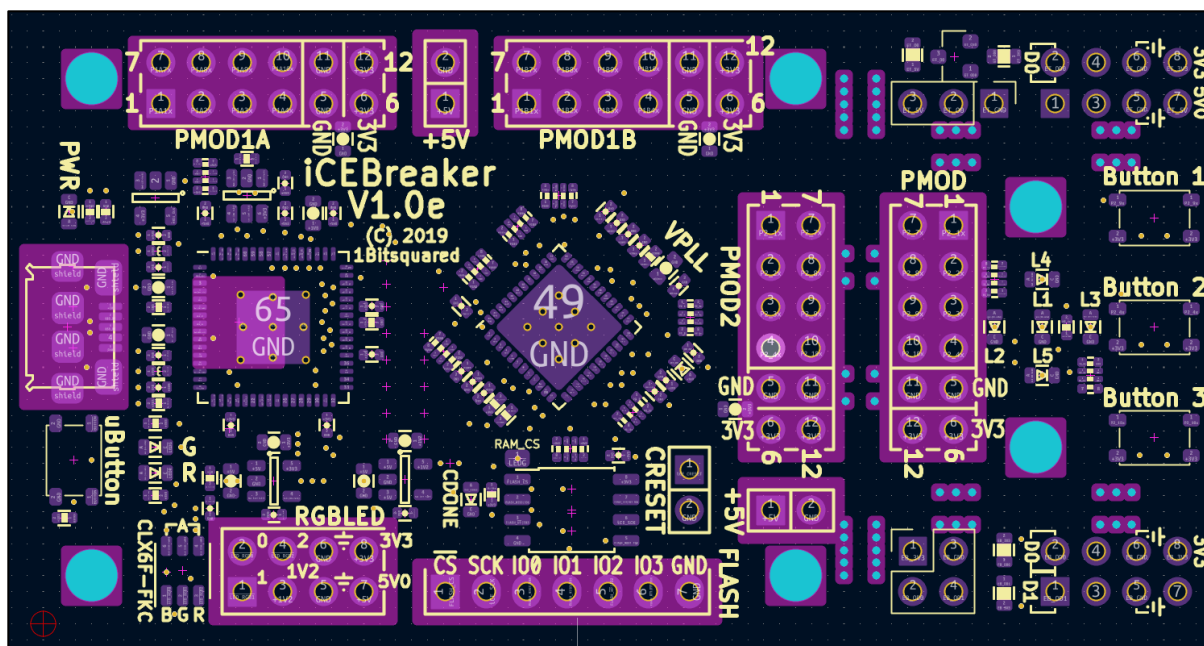


Figura 13: Layout de la iCEBreaker [17].

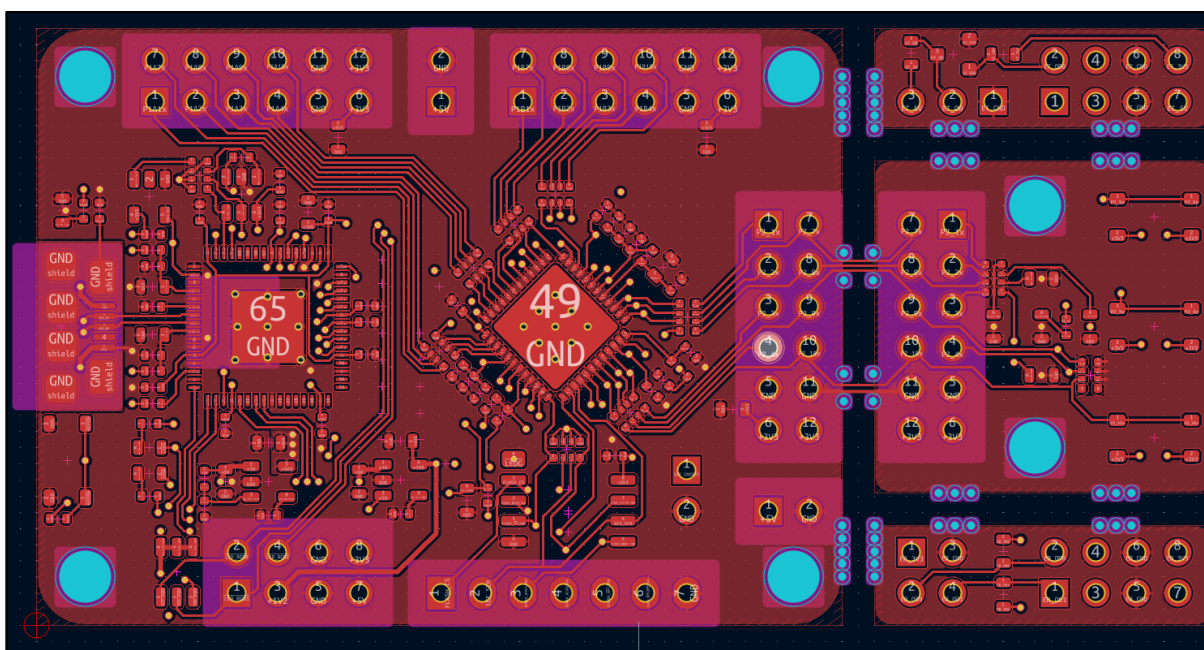


Figura 14: Pistas de cobre de la iCEBreaker [17].

Dejando a un lado el diseño a nivel de circuito, KiCad también es muy útil en el proceso de fabricación de una PCB, donde aparecen nuevas limitaciones:

- ✓ Cada fabricante tiene unas reglas de diseño específicas acordes a la tecnología de la que disponen. Por ejemplo, hay fabricantes que pueden asegurar diámetros de perforación o separación entre pistas más pequeños que otros.
- ✓ Es esencial que se respeten, o de otra forma puede que la manufactura de la PCB no se realice correctamente e influya de forma irreparable en el comportamiento de la misma.

- ✓ Estas reglas de diseño pueden implementarse en KiCad, de forma que el programa indicará si se cumplen o no, una propiedad de lo más útil.
- ✓ Además, con KiCad pueden generarse automáticamente los tres archivos que necesita el fabricante antes de iniciar el proceso de fabricación y ensamblado de una PCB:
 1. Archivos Gerber: Contienen la información gráfica de cada capa de la PCB, donde se definen las pistas, los *pads*, la serigrafía y la máscara de soldadura, entre otros.
 2. BOM (Bill Of Materials): Es la lista completa de componentes necesarios para ensamblar la PCB, incluyendo referencias, cantidades y características técnicas de cada pieza.
 3. CPL (Component Placement List): Indica la posición exacta y la orientación de cada componente en la placa, lo que permite al fabricante automatizar el proceso de montaje con máquinas de inserción y soldadura.

2.3 Micro-Cap 12

Micro-Cap 12 es un programa de simulación de circuitos eléctricos y electrónicos que combina análisis analógico y digital dentro de un mismo entorno. Permite construir esquemas y realizar diferentes tipos de análisis (como barridos en frecuencia, respuesta temporal...), ofreciendo representaciones gráficas precisas del comportamiento del circuito.

Aunque fue originalmente un software comercial desarrollado por Spectrum Software, desde el año 2020 su última versión se distribuye de forma totalmente gratuita tras el cierre de la empresa.

Por su facilidad de uso, precisión y carácter gratuito, Micro-Cap 12 se empleó en este trabajo como herramienta de apoyo para la simulación previa de ciertos bloques analógicos, verificando su comportamiento teórico antes de la implementación final en hardware.

2.4 iCEBreaker

La tarjeta iCEBreaker es un proyecto *open source* diseñado específicamente para la enseñanza y la experimentación con FPGAs. Su carácter abierto garantiza la compatibilidad con un amplio conjunto de herramientas libres, lo que la convierte en una plataforma ideal para el desarrollo de este trabajo.

En concreto, se ha utilizado la versión 1.0e, muy reciente y con buenas prestaciones.

2.4.1 Hardware

La iCEBreaker está basada en la iCE40UP5K del fabricante Lattice Semiconductor, encapsulada en un formato de QFN48. Se puede obtener mucha información de esta FPGA con tan solo descomponer el nombre:

- Familia¹⁸ iCE (Integrated Circuit Engine), serie 40 (conocida por ser de bajo consumo y bajo coste).
- UltraPlus, que es una subfamilia de la iCE40 con más memoria y capacidades avanzadas respecto a las iCE40 originales.

¹⁸ Grupo de dispositivos con la misma arquitectura, pero diferentes capacidades y aplicaciones.

- 5K, que indica la cantidad aproximada de celdas lógicas disponibles (alrededor de 5000).
- Formato QFN48 (Quad Flat No-lead 48), es decir, 48 pines repartidos en los cuatro lados de un encapsulado plano y sin patillas sobresalientes.

De hecho, el nombre de la tarjeta iCEBreaker es un juego de palabras: *ice breaker* en inglés significa romper el hielo, haciendo referencia a tener un primer contacto con las FPGAs de la familia iCE.

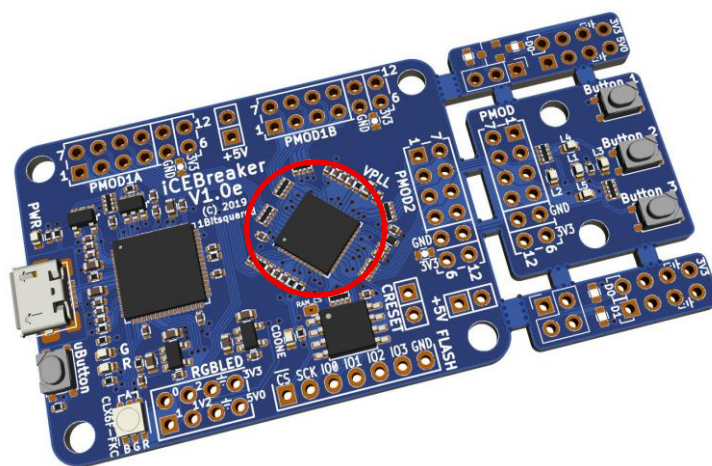


Figura 15: FPGA de la tarjeta iCEBreaker [14].

En concreto, la FPGA posee 5280 celdas lógicas (CLB, Configurable Logic Block), cada una formada por:

1. 4-LUT (Look Up Table de 4 entradas): Es una pequeña memoria que emula una tabla de verdad de 4 entradas y una salida.

Por ejemplo, si se quiere construir una puerta¹⁹ AND (la salida solo está a nivel alto cuando todas las entradas lo están), la tabla quedaría así:

| Entradas | Salida |
|----------|--------|
| 0 0 0 | 0 |
| 0 0 1 | 0 |
| 0 0 1 0 | 0 |
| 0 0 1 1 | 0 |
| ... | ... |
| 1 1 1 1 | 1 |

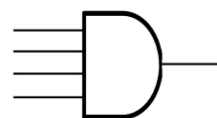


Tabla 1: LUT para una puerta AND.

Figura 16: Puerta AND de 4 entradas [18].

Estos componentes son los que permiten que la FPGA sea reconfigurable. En lugar de implementar

¹⁹ En electrónica digital, una puerta lógica es un conjunto de componentes (resistencias, diodos y transistores) dispuestos de tal forma que solo se obtiene la salida deseada para ciertas combinaciones de las entradas.

físicamente la lógica de cada circuito combinacional, se utiliza la información almacenada en la memoria interna de las LUTs. Así, para cada combinación posible de entradas existe un valor de salida predefinido, de la misma forma que en una tabla de verdad de un circuito booleano convencional. La diferencia es que dicho valor no se calcula a partir de la lógica cableada, sino que se obtiene directamente al leer la LUT.

Al reprogramar la FPGA, se modifican las LUTs, emulando puertas lógicas nuevas (o combinaciones de ellas) y obteniendo una funcionalidad totalmente diferente a la anterior.

2. FF (Flip-Flop): Es un tipo de biestable²⁰ que almacena el resultado de una operación de forma sincronizada con una señal de reloj²¹.

De esta forma, cada celda lógica de la FPGA integra una LUT para lógica combinacional²² y un *flip-flop* para lógica secuencial²³. Un multiplexor²⁴ decide si la salida de la LUT se dirige directamente al enrutado o si por el contrario debe almacenarse primero en el biestable para salir del CLB.

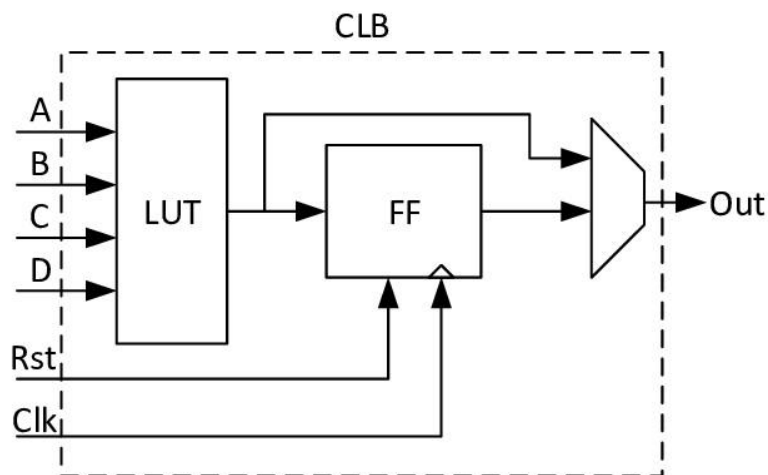


Figura 17: Diagrama de un CLB estándar [19].

3. Carry: Es un componente opcional que actúa como un camino rápido dedicado al cálculo de acarreo²⁵ en operaciones aritméticas. Gracias a este recurso, la FPGA puede implementar sumadores y otros circuitos matemáticos mucho más rápido y eficiente que si se construyeran únicamente con LUTs.

La FPGA cuenta con 128 KB de memoria SRAM (Static Random Access Memory) que puede usarse para almacenar la información de configuración. Se añaden 16 KB extra con una utilidad más: son accesibles desde dos puertos, lo que las hace muy interesantes a la hora de trabajar con colas FIFO (First Input First Output).

²⁰ Se llama así porque tiene dos estados estables (0 o 1) y puede permanecer indefinidamente en uno de ellos hasta que una señal lo cambie.

²¹ Una señal de reloj es una onda periódica que alterna entre nivel bajo (0) y alto (1) de forma regular, marcando el ritmo de funcionamiento de un circuito digital. El ritmo será más o menos rápido según la frecuencia con la que se repite la señal. Puede considerarse el flanco de bajada o el flanco de subida.

²² El resultado depende únicamente de las entradas actuales (no tiene memoria).

²³ El resultado depende de las entradas actuales y del estado previo (tiene memoria). Se necesita un elemento de almacenamiento.

²⁴ Dispositivo electrónico, que permite seleccionar una de entre varias entradas de datos y enviarla a una única salida.

²⁵ El acarreo ocurre cuando al sumar dos números binarios de N bits, se obtiene un resultado de $N + 1$ bits. Equivale a llevarse uno en decimal ($9 + 1 = 10$), pero en base 2.

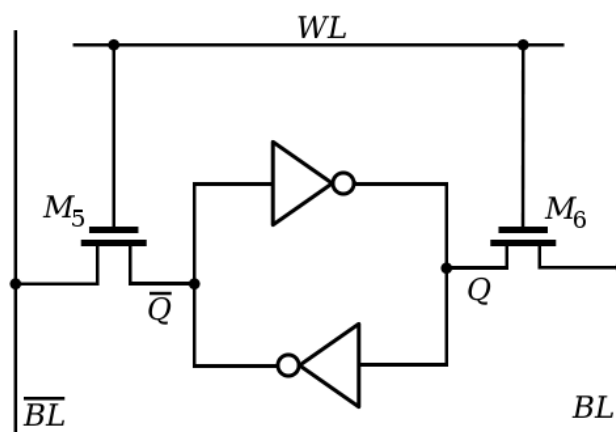


Figura 18: Celda SRAM [20].

En las celdas SRAM, la información queda atrapada en un bucle formado por dos inversores²⁶. La línea de escritura (WL) aplica tensión a los transistores MOSFET (Metal Oxide Semiconductor Field-Effect Transistors)²⁷ cuando se quiere leer o escribir, que permiten que habilitan el camino a la línea de bit (BL) para que se transmita la información.

La red de conexión de la FPGA está llena de multiplexores (como el que se vio en la Figura 17) controlados por una celda SRAM. Esos bits deciden qué caminos están conectados y cuáles no. Las LUTs también son realmente grandes bloques de celdas SRAM y multiplexores.

No obstante, esta memoria es volátil, es decir, al dejar de alimentar la FPGA se pierde la información. Esto obligaría a reprogramarla con cada uso.

La iCEBreaker contaba con esta limitación, por lo que añade una memoria externa: Una memoria Flash, la cual es no volátil gracias al uso de dos niveles de polisilicio en las puertas de los transistores, un material que queda cargado de electrones al aplicarle tensión y puede llegar a almacenar información durante años. De esta forma, el *bitstream*²⁸ queda almacenado en la Flash y se carga en la FPGA automáticamente cada vez que se enciende o se reinicia, solventando el problema.

Otra cuestión a resolver sería como comunicar toda esta información de forma ordenada y estructurada, para lo que existen numerosos protocolos de comunicación. Un protocolo de comunicación digital es un conjunto de normas y procedimientos estandarizados que permiten a los dispositivos electrónicos compatibilizar el intercambio de datos. Estos protocolos definen la sintaxis (cómo se estructuran los datos), la semántica (el significado de los datos) y la sincronización de la comunicación, asegurando que tanto el emisor como el receptor puedan interpretar la información correctamente.

Para cada tarea se trabaja con un protocolo específico:

- El ordenador utiliza USB (Universal Serial Bus) para comunicarse con los periféricos conectados, como puede serlo la iCEBreaker.
- La FPGA espera el *bitstream* con el protocolo SPI (Serial Peripheral Interface).
- Para muchas funciones, como imprimir caracteres por pantalla, se trabaja con UART (Universal Asynchronous Receiver/Transmitter).

²⁶ También llamados puertas NOT. Invierten el estado de la señal de entrada, transformando un 0 en 1 y un 1 en 0.

²⁷ Aunque son complejos de explicar, a grandes rasgos pueden definirse como un interruptor. Si se aplica tensión suficiente en la puerta permiten el paso de corriente, si no, ofrecen una gran resistencia al paso de la corriente. Existen otros tipos de transistores, pero los MOSFETs son los más empleados.

²⁸ El *bitstream* (flujo de bits) es el archivo binario que contiene toda la información necesaria para configurar una FPGA.

- Como ya se explicará, el PMOD que se ha diseñado se comunica a través de IC (Inter-Integrated Circuit).

Se ahondará en algunos de estos protocolos en los capítulos siguientes.

La iCEBreaker cuenta con el FT2232H, un encapsulado de la empresa FTDI (Future Technology Devices International), que se encarga de hacer la conversión de USB a cualquiera de los otros protocolos nombrados y viceversa.

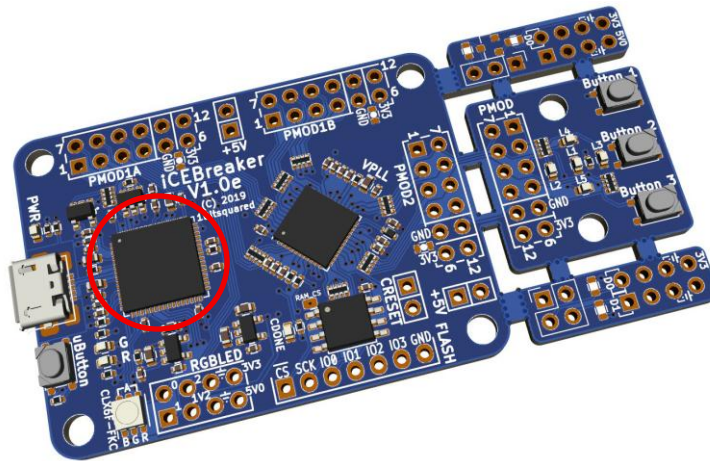


Figura 19: FT2232H de la iCEBreaker.

Para dirigir toda la lógica de la iCEBreaker, se utiliza un cristal de cuarzo²⁹, que es capaz de oscilar de forma estable a 12 MHz. Aunque la FPGA posee dos osciladores RC (basados en resistencias y condensadores, no en cristales) internos de 10 kHz y 48 MHz (divisible a 24, 12 y 6 MHz) cada uno, estos tienen muy poca precisión y estabilidad a largo plazo. Son suficientes para tareas internas sencillas como los temporizadores, pero para funciones complejas es más apropiado utilizar la señal de reloj de un cristal externo.

No obstante, en muchas ocasiones puede no ser adecuada una señal de reloj que oscile a 12 MHz. Según el diseño, pueden ser necesarias frecuencias más altas o bajas (aunque en el segundo caso puede utilizarse simplemente un divisor de frecuencia). Para solucionarlo, se implementan los PLL (Phase-Locked Loop), circuitos de control en lazo cerrado capaces de obtener señales a otras frecuencias.

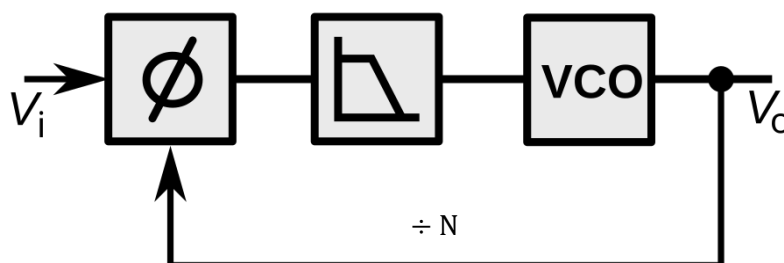


Figura 20: Bucle de control de un PLL [21].

²⁹ Es un componente electrónico que vibra al aplicarle tensión. Esa vibración es muy estable en frecuencia, lo que lo convierte en una referencia de tiempo excelente.

El VCO (Voltage Controlled Oscillator) es un oscilador en anillo poco preciso. Un oscilador en anillo es básicamente una cadena de inversores³⁰ conectados en lazo cerrado. La frecuencia de oscilación depende del tiempo de propagación³¹ de cada inversor y del número de etapas.

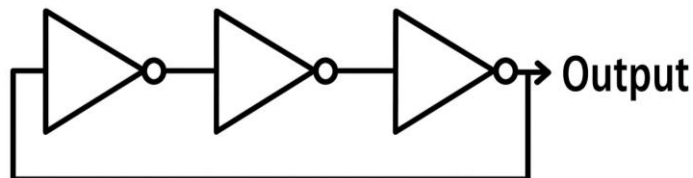


Figura 21: Oscilador en anillo [22].

Cada inversor se construye con transistores MOSFET, cuyo retardo varía en función de la tensión aplicada a la puerta de estos componentes. Por lo tanto, la frecuencia de oscilación del oscilador se decide mediante este voltaje, de ahí el nombre.

Naturalmente, estos inversores no son lo suficientemente precisos y estables, por lo que se complementan con un bucle de control realimentado: el PLL. Es aquí donde entra el detector de fase (primer bloque de la Figura 20), que compara la fase de la señal generada (V_o) con la de la referencia establecida (V_i), en este caso, el reloj de 12 MHz. En realidad, la referencia no se compara directamente con la señal del VCO, sino con una versión dividida por una constante N . Por ejemplo, si se están generando 48 MHz se divide la realimentación entre 4, para que esté a la misma frecuencia que la referencia.

El error de fase detectado se usa como una señal de control que, tras ser suavizada por el filtro de lazo (segundo bloque de la Figura 20), ajusta al VCO. Al forzar la coincidencia de fase no solo se garantiza que la frecuencia sea igual, sino también la alineación temporal entre ambas señales. De esta forma, se consigue una señal de reloj a la frecuencia que se desee, tan estable y fiable como la señal de referencia.

Otras prestaciones muy útiles de la FPGA son los DSPs (Digital Signal Processing), 8 bloques hardware especializados que se incluyen dentro del chip para acelerar operaciones matemáticas, en especial las que aparecen en procesamiento digital de señales. Esto evita tener que gastar muchas LUTs en operaciones matemáticas costosas.

Ofrece también diferentes bloques *hard IP* (Intellectual Property) *core*³², para SPI, I²C y PWM (Pulse Width Modulation)³³, aunque no se le han dado uso en este TFG, pues existen otras vías más intuitivas para implementar estas funciones.

Por último, decir que la iCEBreaker trae consigo una gran variedad de LEDs (Light Emitting Diode)³⁴, botones, pines y conectores (donde pueden conectarse PMODs), para realizar toda clase de experimentación. Se necesita de un ordenador que entregue 5 V por el puerto USB (que luego puede regularse a otros valores, como 3.3 V) para alimentar toda la circuitería. Este puerto USB también sirve para intercambiar información entre el ordenador y la tarjeta.

³⁰ Es un componente electrónico que vibra al aplicarle tensión. Esa vibración es muy estable en frecuencia, lo que lo convierte en una referencia de tiempo excelente.

³¹ Tiempo que tarda una señal en llegar desde su origen al destino. En este caso, se refiere al tiempo que pasa entre que la señal llega al inversor y sale invertida.

³² Una IP core es un bloque de lógica ya diseñado por otro autor, que se puede reutilizar en otros proyectos, facilitando y acelerando el desarrollo. Pueden estar basados en hardware (*hard IP*) o en lenguajes de descripción hardware (*soft IP*).

³³ Modula el ciclo de trabajo de una señal, es decir, qué porcentaje de tiempo la señal está en alto en un período completo.

³⁴ A grandes rasgos, un diodo es un dispositivo que solo permite el paso de la corriente en un sentido. En el caso de un LED, funciona de tal forma que con el paso de la corriente se emite luz.

2.4.2 NEORV32

A pesar de todo lo dicho anteriormente, una FPGA no es siempre la solución más indicada. Está pensada para lógica combinatorial, imitando al hardware fijo, no para algoritmos de software. Aunque con una FPGA se puede construir cualquier tipo de lógica secuencial, cada modificación en el algoritmo requiere resintetizarla, haciendo el proceso excesivamente largo y tedioso, además de no ser el método más eficiente.

En esos casos, es preferible el uso de un procesador o CPU (Central Processing Unit), tal y como la que utiliza un ordenador, pues está diseñada para trabajar de forma secuencial. Una CPU consiste en:

- Una ALU (Arithmetic/Logic Unit), encargada de realizar las operaciones matemáticas y lógicas del procesador.
- Registros, pequeñas memorias internas de la CPU, muy rápidas, usadas para almacenar temporalmente datos e instrucciones. Se construyen con *flip-flops*.
- Una CU (Control Unit), el bloque que organiza y dirige todo el funcionamiento de la CPU.

Si la CPU se complementa con una unidad de memoria, periféricos de entrada/salida y buses³⁵ para transmitir la información entre todos estos componentes, se construye la clásica arquitectura de Von Neumann.

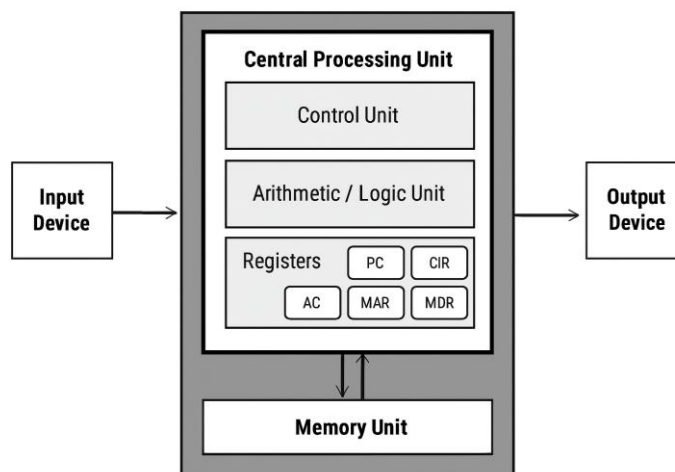


Figura 22: Arquitectura de Von Neumann [23].

En electrónica, si se quiere esta clase de prestaciones se suele trabajar con microcontroladores, encapsulados que integran en un solo chip todos los elementos descritos.



Figura 23: Microcontrolador [24].

³⁵ Son conjuntos de líneas (cables) que transportan información dentro del procesador o entre la CPU y otros componentes.

Sin embargo, esto no es del todo necesario. Lo cierto es que, si se destinasen buena parte de los recursos de la FPGA (celdas lógicas, parte del espacio de la memoria SRAM...) a construir todos estos bloques, se puede obtener la misma funcionalidad que un microcontrolador en la iCEBreaker.

Aquí entra en juego la NEORV32, un proyecto *open source*, compartido con la comunidad a través de Github [25]. Se trata de un *soft processor*, es decir, un procesador descrito en VHDL. Permite ejecutar programas en lenguaje C o ensamblador³⁶, como cualquier microcontrolador, pero corriendo dentro de la FPGA. Si se descompone el nombre:

- NEO hace referencia a moderno o nuevo.
- RV indica que está basada en RISC-V (Reduced Instruction Set Computing, 5^o generación). La ISA (Instruction Set Architecture) es la descripción formal del conjunto de instrucciones que un procesador entiende y puede ejecutar: formato de las instrucciones, modos de direccionamiento, comportamiento ante interrupciones... La arquitectura RISC-V, destaca por ser una ISA de carácter libre y abierto, diseñada con menos instrucciones de lo habitual, simples y eficientes.
- La arquitectura de 32 bits es un tipo de diseño de procesador que maneja datos y direcciones de memoria en palabras (trozos) de 32 bits a la vez.

La NEORV32 no solo consiste en una CPU, si no que añade una amplia gama de periféricos que dan cierta autonomía al proyecto:

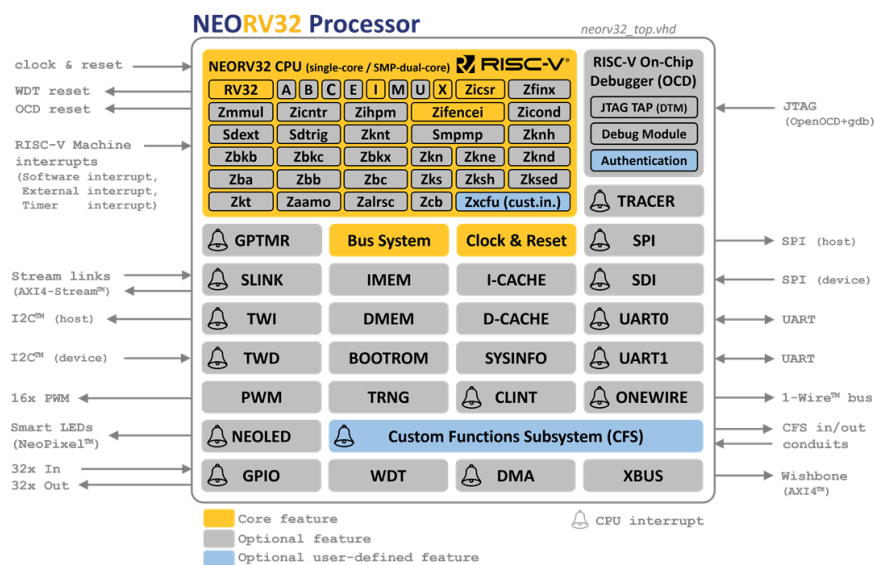


Figura 24: Esquema de la NEORV32 [25].

Como puede observarse, ya se incluye una memoria de datos (DMEM) y otra de instrucciones (IMEM), junto a sus respectivas memorias caché y una memoria de solo lectura para el *bootloader*³⁷ (BOOTROM). Contiene también un bus propio y la posibilidad de usar uno externo (XBUS): el bus Wishbone, otro proyecto libre de licencias que permitiría conectar otros IP *cores* compatibles al procesador.

³⁶ Los lenguajes de programación actúan como un puente de comunicación entre el programador y la máquina. En lugar de escribir código directamente en lenguaje máquina (binario), surgió el lenguaje ensamblador, que servía para programar de una forma más intuitiva y comprensible para los humanos. Dicho código se traduce a binario a través de un ensamblador. Más adelante, surgieron los lenguajes de alto nivel (como C), facilitando aún más esta tarea.

³⁷ Es un programa especial que inicializa lo básico del procesador y, si no hay ningún programa en la memoria principal, permite cargar un ejecutable a través de una interfaz de comunicación.

Al sintetizar el procesador, se pueden habilitar una serie de funcionalidades de suma importancia para el trabajo, destacando GPIO (General Purpose Input/Output)³⁸ y TWI (Two Wire Interface), el nombre alternativo que se le dio al protocolo I²C para evitar posibles implicaciones legales.

Además, se desarrollaron una serie de funciones en C destinadas a facilitar el uso de estos periféricos, por lo que no es necesario profundizar en aspectos muy técnicos, como el mapeado de la memoria principal.

Las facilidades que se ofrecen, el trabajo previo del alumno con esta herramienta y la extensa documentación que hay disponible, fueron las principales razones que llevaron a utilizar la NEORV32 para implementar ciertas funcionalidades en el proyecto, en lugar de utilizar los *hard IP cores* que integra la FPGA, pues requerirían una mayor inversión de tiempo. No obstante, esta decisión traería consigo nuevos problemas que se desarrollarán más adelante.

2.4.3 Programación de la FPGA

Para configurar la FPGA, es necesario realizar una serie de pasos. Por supuesto, todos los programas que se nombran aquí están basados en software libre.

✓ Descripción en VHDL:

Primero, se utiliza un lenguaje de descripción hardware, en este caso VHDL, para definir el comportamiento del circuito digital que se desea implementar. Este código especifica la lógica que será sintetizada después en la FPGA. Pese a que no es obligatorio, se ha utilizado Visual Studio Code para escribir código.

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Entidad: define la interfaz (entradas y salidas)
entity PuertaAND is
  Port (
    A : in STD_LOGIC;    -- Entrada A
    B : in STD_LOGIC;    -- Entrada B
    Y : out STD_LOGIC    -- Salida Y = A AND B
  );
end PuertaAND;

-- Arquitectura: describe el comportamiento interno
architecture ArquitecturaPuertaAND of PuertaAND is
begin
  Y <= A and B;
end ArquitecturaPuertaAND;

```

Figura 25: Ejemplo. Puerta AND en VHDL.

³⁸ Son pines digitales configurables que el procesador expone hacia el exterior. Se llaman "de propósito general" porque no tienen una función fija: el programador decide en su diseño si actúan como entrada (leer un valor lógico) o como salida (escribir un valor lógico).

✓ Síntesis:

Una *netlist* (lista de redes) es una representación intermedia del circuito digital en forma de lista de componentes básicos (puertas lógicas, *flip-flops*, multiplexores, etc.) y de las conexiones entre ellos. En lugar de describir el comportamiento a alto nivel, como se hace en VHDL, la *netlist* refleja directamente qué elementos lógicos existen y cómo están conectados entre sí.

La herramienta de síntesis libre más extendida es Yosys (Yosys Open SYnthesis Suite)³⁹. Yosys genera la *netlist* en formato JSON (JavaScript Object Notation), un formato de texto plano que sirve para representar datos estructurados de forma sencilla. No obstante, este programa no trabaja en VHDL, sino RTLIL (Register Transfer Level Intermediate Language), que es una forma de describir un diseño digital en términos de registros y transferencias de datos entre ellos. Es necesaria una expansión: GHDL (GNU⁴⁰ Hardware Description Language), que analiza el código, comprueba que la sintaxis y semántica son correctas y lo traduce a RTLIL. Por lo tanto, sirve también como depurador de errores.

```
{
  "modules": {
    "PuertaAND": {
      "cells": {
        "P1": {
          "type": "AND",
          "connections": {
            "A": "A",
            "B": "B",
            "Y": "Y"
          }
        }
      }
    }
  }
}
```

Figura 26: Ejemplo. Puerta AND en formato JSON.

✓ Colocación y enrutamiento:

La *netlist* producida por Yosys se entrega a nextpnr (Next Place & Route). Dos pasos:

- Colocación. Primero, se decide en qué bloques de la FPGA (LUTs, *flip-flops*, bloques de SRAM, DSPs...) se va a implementar cada elemento lógico de la *netlist*.
- Enrutamiento. A continuación, se establece qué caminos de la red de interconexión de la FPGA deben usarse para conectar esos bloques entre sí.

³⁹ Es muy común entre los creadores de software libre el uso de siglas recursivas.

⁴⁰ Las siglas de GNU's Not Unix. De nuevo, usan la recursividad y expresan de forma humorística que, aunque la funcionalidad sea parecida a Unix, no son lo mismo, pues GNU es 100% libre.

Todo ello queda definido en un archivo ASC (ASCII Configuration), que es una representación textual de cómo queda configurada la FPGA en código ASCII (American Standard Code for Information Interchange)⁴¹.

✓ Generar el *bitstream*:

Por último, la herramienta Icestorm (llamada así por la familia iCE de FPGAs) traduce el archivo ASC a código binario y lo envía a la iCEBreaker por el puerto USB. El *bitstream* queda almacenado en la memoria Flash.

Todos estos pasos pueden resumirse en un solo comando si se utilizan *makefiles*, ficheros diseñados para automatizar procesos que requieren muchas instrucciones.

Por otra parte, si se ha sintetizado la NEORV32 en la FPGA, serán también necesarias el conjunto de herramientas de GNU para compilar y depurar código en C. El *bootloader* del procesador, está diseñado para esperar la recepción de un ejecutable válido (el código que se quiere ejecutar). Para transmitírselo, puede utilizarse CuteCom, un programa que puede utilizarse para comunicar por UART el ordenador y la NEORV32. Una vez que se está ejecutando el programa en el procesador, este software también será de utilidad para que el usuario envíe los datos que se soliciten a través del puerto serie.

Como puede observarse, deben tenerse en cuenta una gran variedad de herramientas a la hora utilizar una FPGA y más aún si se va a incluir la NEORV32. Cada una de ellas está en continuo desarrollo. Llegado a cierto punto, algunas versiones comienzan a estar desactualizadas o ser incompatibles entre sí, lo que fue un gran limitante en el desarrollo de este trabajo. A fecha de entrega de este TFG, basta con utilizar la última versión disponible de las herramientas para obtener el funcionamiento esperado. Para más indicaciones, véase el repositorio del proyecto [26].

⁴¹ Es un código de 7 bits que asigna valores numéricos a caracteres como letras, números y símbolos para que los ordenadores, que solo entienden números, puedan procesar y mostrar texto de forma comprensible.

3 FUNDAMENTOS DE TELECOMUNICACIONES

Solo sé que no sé nada.

- Sócrates -

Una vez comprendidas las herramientas y la electrónica que envuelven al trabajo, el lector podría sumergirse inmediatamente en la descripción de todo el hardware de la tarjeta PMOD que se ha empleado para este Trabajo de Fin de Grado. No obstante, su comprensión será más llevadera si se dedica un capítulo a explicar una serie de conceptos previos, que resumen brevemente el mundo de las telecomunicaciones.

3.1 Introducción

La razón de ser de los sistemas de comunicación puede resumirse en una frase: conseguir que la información viaje desde su origen hasta su destino, utilizando para ello un tercer elemento denominado canal o medio de la información.



Figura 27: Sistema de comunicación I [27].

El esquema anterior puede desarrollarse algo más si se introduce la figura del transmisor y el receptor, que serán los bloques encargados de lograr que la información viaje a través del canal.



Figura 28: Sistema de comunicación II [27].

Aquí hay que señalar un matiz importante: este sistema de comunicación puede ser de cualquier índole, por ejemplo, dos personas conversando (el canal sería el aire). Cuando se habla de un sistema de telecomunicación, se está concretando en los sistemas que utilizan la ingeniería para dar la posibilidad de transmitir la información a largas distancias (aunque en la práctica, esta distancia puede ser de unos metros). Esta transmisión puede ser tanto no guiada (inalámbrica) como guiada (por cable).

La señal que genera la fuente de información puede ser de dos tipos: analógica, donde se tienen infinitos estados o valores, y digital, donde solo aparecen dos⁴².

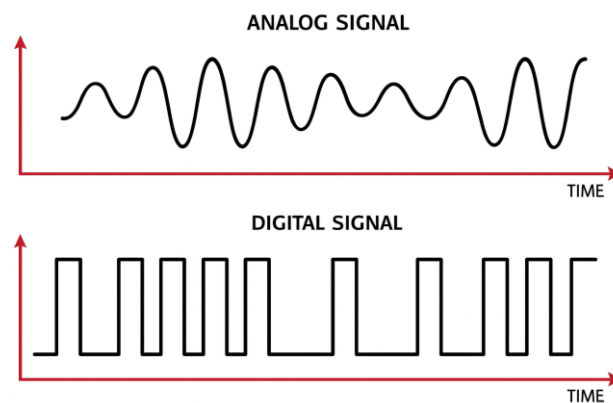


Figura 29: Señal analógica versus señal digital [28].

Este TFG consiste en un sistema de telecomunicación digital. No obstante, antes de profundizar en ellos, se explican una serie de conceptos básicos de alto valor.

3.2 Teorema de Fourier

Sin duda, el principio más importante que envuelve a las telecomunicaciones es el Teorema de Fourier, que dice así: toda señal, por compleja que sea, puede descomponerse en una suma de ondas sinusoidales (senos y cosenos) que oscilan a distintas frecuencias.

⁴² Realmente, el término digital hace referencia a un número de estados finito. Si la señal es además binaria, los estados se reducen a dos. Con el tiempo, ambos términos comenzaron a usarse indistintamente.

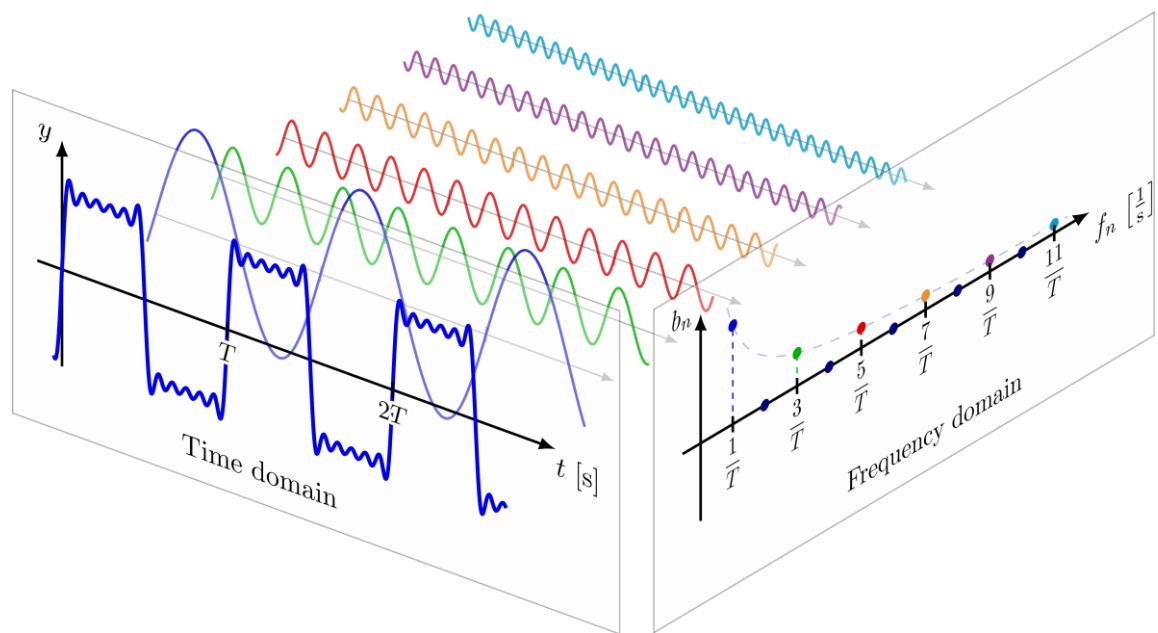


Figura 30: Construcción de una señal cuadrada mediante armónicos [29].

La gráfica de la izquierda de la Figura 30 corresponde al dominio del tiempo de la señal y la de la derecha, al dominio de la frecuencia. Las ondas sinusoidales que aparecen se denominan armónicos. El armónico de frecuencia más baja (1º armónico) se denomina armónico fundamental, pues se parte de esta onda para crear la señal final.

Este teorema se planteó en principio para señales periódicas, es decir, señales que se repiten en un periodo de tiempo. La denominada Serie de Fourier, responde a esta fórmula:

$$x(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} [a_n \cos(n\omega_0 t) + b_n \sin(n\omega_0 t)]$$

Siendo:

- $x(t)$ una señal periódica en el tiempo. El tiempo que tarda en repetirse la señal se denomina periodo, y es la inversa de su frecuencia (cuantas veces se repite por segundo). Esta es también la frecuencia a la que oscila el armónico fundamental.

$$T = \frac{1}{f_0}$$

- ω_0 la frecuencia angular fundamental de la señal. Puesto que el 1º armónico consiste en una onda sinusoidal, que por cada repetición pasa de los 0 a los 360 grados (que equivalen a 2π radianes), la frecuencia angular fundamental corresponde al número de veces que completa una vuelta por segundo.

$$\omega_0 = 2\pi f_0 = \frac{2\pi}{T}$$

- n el número del armónico.
- a_n y b_n la amplitud de los cosenos y senos respectivamente.

$$a_n = \frac{2}{T} \int_0^T x(t) \cos(n\omega_0 t) dt \quad n = 0, 1, 2, 3, \dots$$

$$b_n = \frac{2}{T} \int_0^T x(t) \sin(n\omega_0 t) dt \quad n = 1, 2, 3, \dots$$

La integral multiplica la señal $x(t)$ por el coseno/seno correspondiente y la promedia en un periodo. Matemáticamente, esto es una proyección: se está midiendo cuánto se parece la señal al coseno de frecuencia $n f_0$. A mayor similitud, significará que $x(t)$ se compone en buena parte de este coseno/seno, por lo que los coeficientes a_n y b_n serán más altos.

Por lo tanto, el término $\frac{a_0}{2}$ corresponde al valor medio de la señal, mientras que el sumatorio representa una serie teóricamente infinita de armónicos oscilando a distintas frecuencias. Aunque se sume un seno y un coseno por cada armónico, realmente se trata de un único coseno desfasado.

$$a_n \cos(n\omega_0 t) + b_n \sin(n\omega_0 t) = A_n \cos(n\omega_0 t - \varphi_n)$$

No obstante, la Serie de Fourier no es tan interesante, pues la inmensa mayoría de las señales no son periódicas. En ese caso, entra en juego la Transformada de Fourier, que sigue el mismo principio, pero considerando que la señal tiene un periodo infinito. Convierte la señal de un dominio a otro, mostrando qué frecuencias la componen y con qué intensidad.

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi f t} dt$$

$X(f)$ es la representación de la señal en frecuencia. El término $e^{-j2\pi f t}$, aunque pueda parecer complicado de entender, la Fórmula de Euler dice lo contrario:

$$e^{-j2\pi f t} = \cos(2\pi f t) - j \sin(2\pi f t)$$

Viene a ser lo mismo, una suma de ondas sinusoidales. El signo negativo es una convención utilizada para simplificar la gráfica resultante. El uso de números complejos es tan solo una estrategia matemática para facilitar las operaciones y formular de una manera más compacta.

Por lo tanto, la Transformada de Fourier sigue el mismo principio que los términos a_n y b_n : proyecta qué cantidad de cada frecuencia es necesaria para formar la señal.

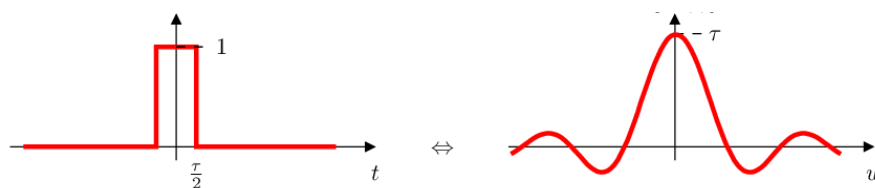


Figura 31: Transformada de Fourier de un pulso [30].

La transformada tendrá la mitad de amplitud que la señal original y será simétrica respecto al eje de coordenadas, debido al uso de números complejos y al signo negativo del término de Euler (si el signo fuese positivo, aparecería una simetría impar, menos visual). A efectos prácticos, puede considerarse tan solo la parte positiva.

Dejando a un lado las matemáticas, la Transformada de Fourier tiene una gran utilidad: definir el ancho de banda de una señal, es decir, el rango de frecuencias que la componen.

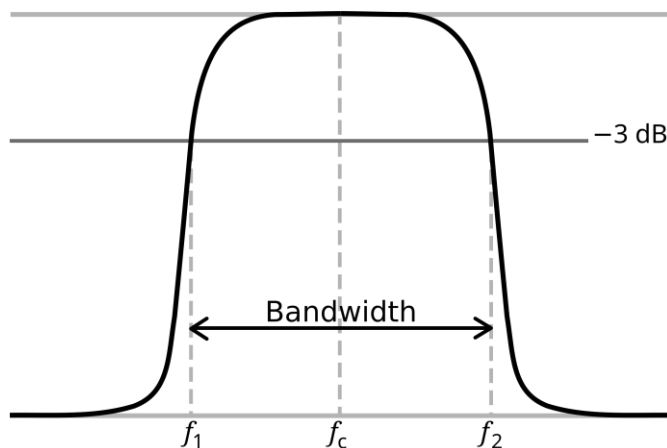


Figura 32: Ancho de banda [31].

Dejan de considerarse parte del ancho de banda las frecuencias que caen en más de 3 dB (que equivale a una pérdida del 70% de la ganancia, es decir, ya no es significativo) respecto a la amplitud máxima.

3.3 Transmisores y receptores

A nivel teórico, en la fase previa al procesamiento de las señales, un transmisor es equivalente a un receptor. Lo único que cambia es el uso que se le da. De hecho, en telecomunicaciones pueden generalizarse los términos de transmitir y recibir si se habla de radiar una señal.

Para simplificar, existen dos tipos fundamentales: los cables (si la radiación es guiada) y las antenas (si es inalámbrica).

Como es evidente, los cables son los más sencillos. Transportan señales eléctricas, transmitiendo la información mediante variaciones de voltaje o intensidad de dicha señal. Estas variaciones pueden ser de naturaleza

analógica, como es el caso de un micrófono conectado con un jack de audio; o digital, como en un cable Ethernet o un USB, donde las tensiones representan bits según una codificación. Por ejemplo, en electrónica digital, es muy común que el 0 se represente con 0 V y el 1 con 3.3 V. El ancho de banda asequible por un cable, es decir, qué frecuencias puede transmitir, puede llegar a ser muy grande, aunque no infinito. A partir de ciertas frecuencias, la atenuación es demasiado importante como para transmitir señales de forma exitosa.

Las antenas, por su parte, merecen algo más de atención. Para comprenderlas, es preciso partir de la Ley de Maxwell-Faraday: Cuando un campo magnético varía con el tiempo, induce a su vez un campo eléctrico y viceversa. Una onda eléctrica no puede existir de forma aislada a una onda magnética. Es por ello que reciben el nombre de ondas electromagnéticas.

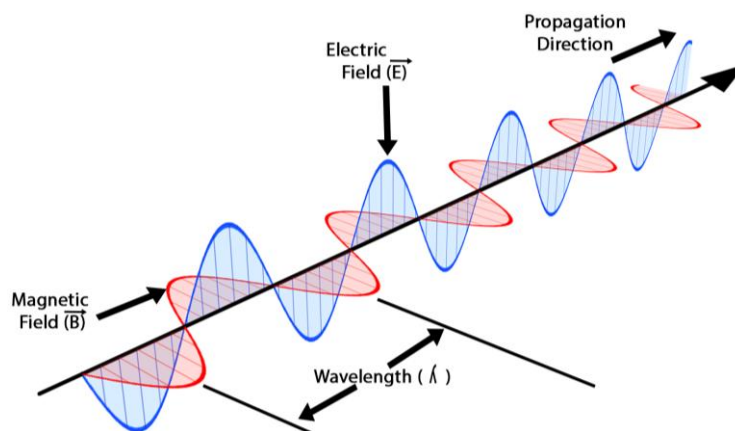


Figura 33: Onda electromagnética [32].

Una antena es, en esencia, un dispositivo relativamente sencillo: se trata de dos conductores por los que circula una corriente alterna. Al variar en el tiempo, esta corriente genera un campo eléctrico cambiante, que a su vez induce un campo magnético asociado. La interacción de ambos campos origina una onda electromagnética que se irradia al espacio y se propaga a su alrededor.

En el proceso inverso, durante la recepción, la onda electromagnética incide sobre la antena y provoca el movimiento oscilatorio de los electrones en el conductor, reproduciendo así la señal original.

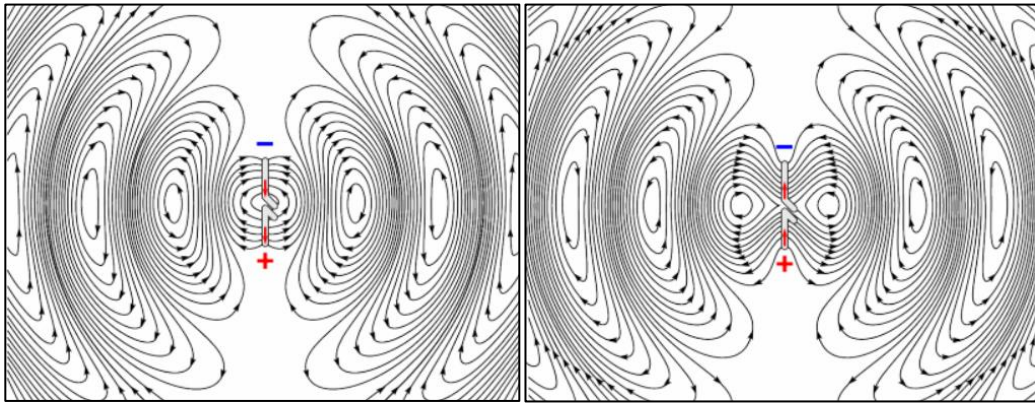


Figura 34: Generación de una onda electromagnética [33].

El tiempo que tardan los electrones en viajar de un extremo a otro de la antena y volver define la frecuencia a la que se está transmitiendo. Existe una relación aproximada entre el tamaño de la antena y la longitud de onda:

$$L \approx \frac{\lambda}{2}$$

Y también entre la longitud de onda y la frecuencia:

$$f = \frac{c}{\lambda}$$

Siendo c la velocidad de la luz ($3 \times 10^8 \frac{\text{m}}{\text{s}}$).

Por lo tanto, cada antena está diseñada para una frecuencia en específico, llamada frecuencia de resonancia. No obstante, también radia correctamente las frecuencias vecinas. Aquí es donde entra en juego el ancho de banda de la señal, y la necesidad de que este no sea muy grande, o de lo contrario no se transmitirá la información correctamente.

Existen estrategias para que una antena pueda utilizarse para radiar a más de una frecuencia de resonancia. En los inicios de las telecomunicaciones, se utilizaban las antenas extensibles. Si se aumenta la longitud de la antena, se está variando también la longitud de onda de la señal que transmite, y, por lo tanto, su frecuencia. Aunque esta práctica sigue observándose hoy en día en sistemas más simples, como los coches radiocontrol, es más común el uso de sintonizadores, circuitos basados en condensadores y bobinas, que merman o acentúan el tiempo que tardan los electrones en recorrer la antena. Es por esto que, con una única radio, pueden abarcarse una gran variedad de frecuencias.

Sin embargo, ocurre una nueva problemática. Si se quiere transmitir una señal de muy baja frecuencia (del orden de Hz), sería necesaria una antena de cientos de kilómetros. Además, aparecen fenómenos físicos que hacen que unas frecuencias sean mejores que otras a la hora de transmitir a ciertas distancias.

3.4 Modulación y demodulación

Como se venía diciendo anteriormente, no siempre es posible transmitir directamente la señal que contiene la información original⁴³. Muchas veces su frecuencia es demasiado baja, sufre demasiadas interferencias o no se adapta bien al medio de transmisión. Para solucionar este problema se emplea la modulación, proceso por el cual una señal llamada *portadora*, normalmente una onda sinusoidal de alta frecuencia, es modificada en alguna de sus características en función de la señal de información, llamada *señal moduladora*. La portadora puede ser tanto analógica como digital, aunque solo se tratará el primer tipo aquí.

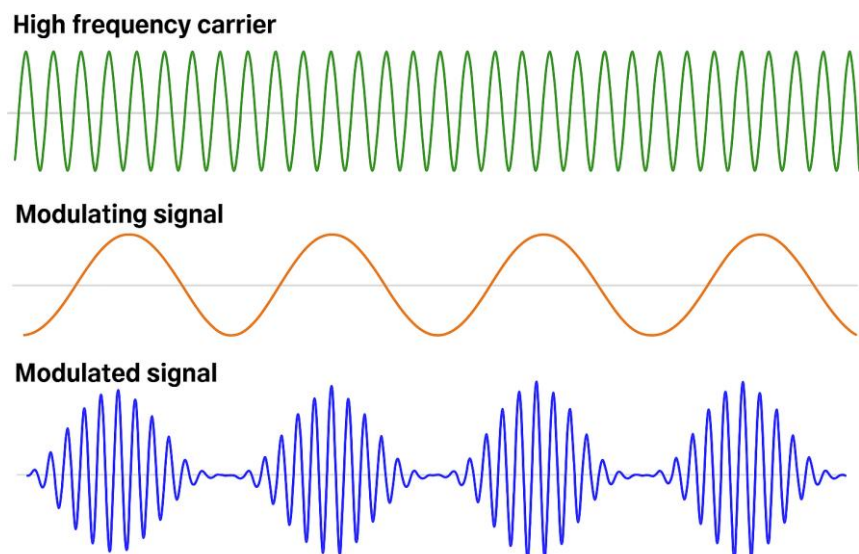


Figura 35: Proceso de modulación (en este caso, la AM) [34].

En el mundo analógico, existen diferentes formas de modulación, que se clasifican según la propiedad de la portadora que se altera:

- Modulación en amplitud (AM, Amplitude Modulation): la amplitud de la portadora varía en función de la señal moduladora. Es sensible al ruido atmosférico y eléctrico.
- Modulación en frecuencia (FM, Frequency Modulation): esta vez se modifica la frecuencia de la portadora. Es más robusta que la AM, razón por la que se emplea en radiodifusión comercial y en comunicaciones de mayor calidad.
- Modulación en fase (PM, Phase Modulation): la fase de la portadora se modifica en función de la señal de información.

⁴³ El rango de frecuencias a la que oscila una señal en su frecuencia original se denomina banda base.

En sistemas digitales, estas modulaciones dan lugar a técnicas discretas como ASK (Amplitude Shift Keying), FSK (Frequency Shift Keying) o PSK (Phase Shift Keying), que constituyen la base de protocolos modernos como Wi-Fi o Bluetooth.

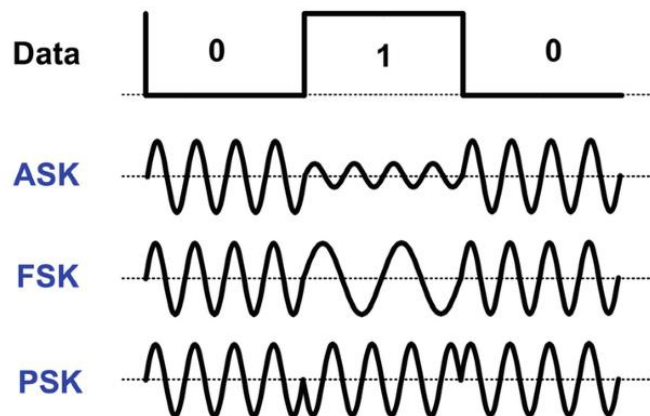


Figura 36: ASK, FSK y PSK [35].

A raíz de la modulación PSK o, mejor dicho, BPSK (Binary Phase Shift Keying), pues solo se trataba con un bit de información mediante dos posibles fases de la portadora (0° y 180°), surge QPSK (Quadrature Phase Shift Keying) que trata con dos bits de información mediante cuatro fases distintas. De esta manera, se reduce a la mitad el tiempo de transmisión sin aumentar el ancho de banda ocupado, lo que resulta de gran interés. Las cuatro fases típicas empleadas en QPSK son:

| Fase | Bits |
|-------------|------|
| 45° | 11 |
| 135° | 01 |
| 225° | 00 |
| 315° | 10 |

Tabla 2: Fases de QPSK.

Generar una portadora que cambie bruscamente de fase no es trivial en hardware. Cada salto de fase introduce distorsiones y la demodulación también se vuelve complicada. Para resolverlo, la portadora se descompone en dos señales ortogonales⁴⁴:

- ✓ La componente I (In-phase, en fase), asociada a un coseno.
- ✓ La componente Q (Quadrature, en cuadratura), asociada a un seno desfasado 90° .

⁴⁴ Dos señales son ortogonales si son matemáticamente independientes y no hay correlación entre ellas, lo que significa que su producto escalar es cero. Esto permite que diferentes señales compartan el mismo medio de transmisión o espacio de señal sin interferir entre sí, pudiéndose recuperar por separado al recibirlas.

De este modo, la portadora no cambia de fase, si no que la suma de las componentes I y Q, con amplitud variable, resultará en la fase deseada. Matemáticamente, la señal QPSK puede expresarse como:

$$s(t) = I(t) \cos(2\pi f_c t) - Q(t) \sin(2\pi f_c t)$$

Donde f_c es la frecuencia de la portadora.

Cada par de bits se asigna a una combinación específica de valores de I y Q, que puede representarse en un diagrama de constelación. Por ejemplo:

| Bits | I(t) | Q(t) |
|------|------|------|
| 11 | 1 | 1 |
| 01 | -1 | 1 |
| 00 | -1 | -1 |
| 10 | 1 | -1 |

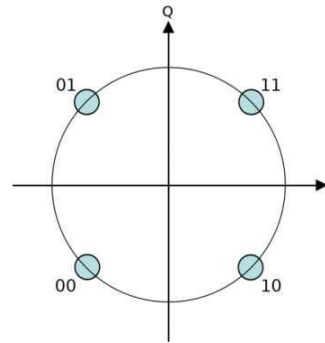


Tabla 3: Valores de los componentes en fase y cuadratura. **Figura 37:** Diagrama de constelación QPSK [36].

Más adelante surgió la modulación QAM (Quadrature Amplitude Modulation), que seguía el mismo principio, pero los valores de I y Q ya no se limitan a un solo nivel, si no a varios. Eso significa que el vector resultante en el plano I/Q no solo cambia de fase, sino también de amplitud. Esto permite una tasa de transmisión bastante grande, con un tolerancia al ruido todavía admisible. La modulación QPSK puede denominarse también 4-QAM.

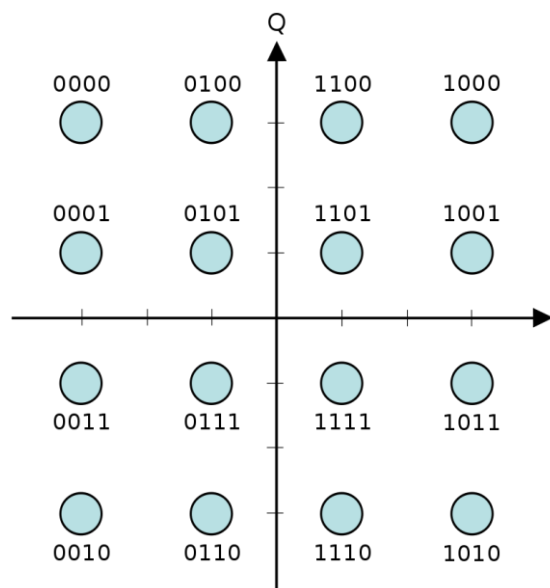


Figura 38: Diagrama de constelación 16-QAM [37].

Además, en lugar de envolver todos los datos en una sola portadora que acapare toda la banda, existe la posibilidad de emplear muchas de ellas al mismo tiempo, repartidas por todo el intervalo del espectro. Esta técnica se conoce como OFDM (Orthogonal Frequency Division Multiplexing) y, a pesar de que las portadoras oscilan a frecuencias muy poco separadas entre sí, nunca llegan a interferir debido a la ortogonalidad. De esta forma, se aprovecha de forma eficiente el espectro del canal y se mejora su robustez, a la par que se consiguen tasas de transmisión altísimas.

La frecuencia de la portadora se asocia a distintas bandas del espectro radioeléctrico, delimitadas por la Unión Internacional de Telecomunicaciones (UIT), con propiedades físicas muy diferentes:

- ✓ Onda larga (30 – 300 kHz): muy baja frecuencia, permite la propagación a grandes distancias siguiendo la curvatura terrestre (ondas de superficie). Se usa en radiobalizas y comunicaciones marítimas.
- ✓ Onda media (300 kHz – 3 MHz): aquí se ubica la radio AM comercial. Sus ondas pueden reflejarse en la ionosfera durante la noche, alcanzando cientos o miles de kilómetros.
- ✓ Onda corta (3 – 30 MHz): aprovecha al máximo la reflexión ionosférica, permitiendo comunicaciones internacionales con equipos relativamente simples.
- ✓ VHF (Very High Frequency, 30 – 300 MHz): donde se sitúa la radio FM (88 – 108 MHz) y la televisión analógica tradicional. Las ondas VHF se propagan prácticamente en línea recta, por lo que la cobertura depende de la visibilidad entre antenas.
- ✓ UHF (Ultra High Frequency, 300 MHz – 3 GHz): empleada en televisión digital, telefonía móvil y Wi-Fi. Permite altas tasas de transmisión, aunque la propagación está más limitada por obstáculos (edificios, relieve).
- ✓ Microondas (3 – 30 GHz): utilizadas en satélites, radares y redes 5G. Su gran ancho de banda permite altísimas velocidades, pero la señal se atenúa rápidamente con la lluvia (fenómeno conocido como *rain fade*).

El proceso inverso a la modulación se denomina demodulación. En este caso, el receptor debe extraer la señal moduladora a partir de la señal modulada recibida. Para ello se utilizan circuitos denominados demoduladores, que reproducen el comportamiento inverso del modulador empleado en el transmisor. El correcto diseño del sistema modulador-demodulador (conocido como módem) es fundamental para asegurar la integridad de la información transmitida.

3.5 Sistema de comunicación digital

Tras todo lo dicho anteriormente, un sistema de comunicación digital puede resumirse en el siguiente esquema:

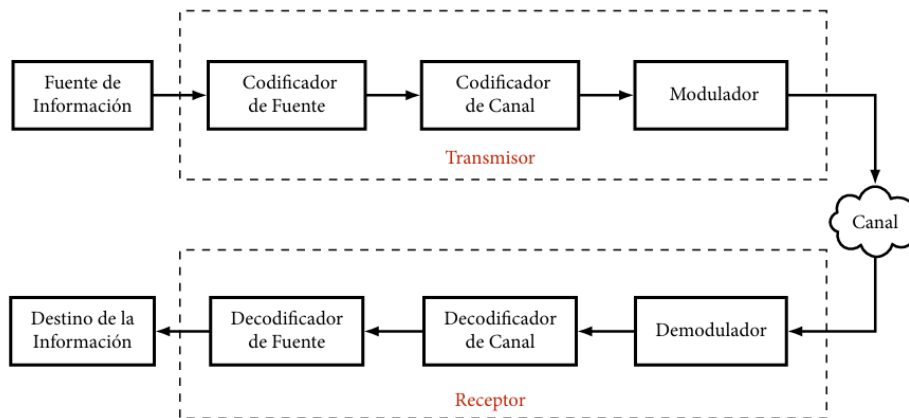


Figura 39: Sistema de comunicación digital [27].

El codificador de fuente es el responsable de representar (codificar) mediante una secuencia de dígitos binarios la señal emitida por la fuente. El codificador de canal, en cambio, tiene como objetivo proteger la información contra los errores introducidos por el canal (ruido, interferencias, atenuación, distorsiones, etc.).

Una vez que se ha codificado la información, los bits se agrupan en símbolos, la unidad mínima de información en telecomunicaciones, se modulan y se transmiten por el canal. En el caso de QPSK, los símbolos son de dos bits. Cuando se recibe la información, se realiza el proceso inverso.

El número de símbolos que llega al modulador por unidad de tiempo se llama tasa de símbolo, y se mide en símbolos/s o baudios. Es la inversa del tiempo de símbolo (el tiempo que tarda un símbolo en llegar al modulador).

$$R_s = \frac{1}{T_s}$$

Como ya se mencionó, en este TFG no se lleva a cabo ninguna codificación ni modulación, sino que se reserva esta tarea para trabajos futuros.

3.6 SDR y GDR

Los bloques mostrados en la Figura 39 (junto a muchos otros más avanzados que consideran un sistema no ideal), se implementan tradicionalmente a través de hardware. Esto vinculaba el sistema a una codificación y modulación específicas.

Como ya se introdujo en el capítulo uno, la Software Defined Radio trata de poner remedio a este problema. La idea principal de la SDR es desplazar parte de estas funciones al dominio del software, de manera que los mismos dispositivos puedan reconfigurarse para distintos estándares y aplicaciones con tan solo modificar el programa que los gobierna.

En la práctica, una SDR emplea un hardware mínimo compuesto por antenas, un conversor analógico-digital (ADC, Analog-to-Digital Converter) y un conversor digital-analógico (DAC, Digital-to-Analog Converter), delegando en el procesador la mayor parte del procesado de señal. Gracias a ello, un mismo equipo puede, por ejemplo, operar como receptor de FM comercial, transmisor Wi-Fi o estación de radioaficionado, simplemente cambiando el software cargado.

Sin embargo, el uso de SDR presenta limitaciones: el procesamiento en tiempo real requiere CPUs muy potentes, y la latencia ⁴⁵ introducida puede ser crítica en ciertos escenarios. Además, la conversión de señales de muy alta frecuencia sigue exigiendo circuitería analógica especializada, por lo que no es posible prescindir por completo del hardware dedicado.

De esta necesidad de optimización nace el concepto de Gateway Defined Radio, término acuñado por Dan Gisselquist [7]. En lugar de trasladar las funciones de radio al software de un procesador, la GDR propone implementarlas directamente en una FPGA mediante lenguajes de descripción hardware como VHDL. De este modo, se obtiene la flexibilidad propia de la reconfiguración, pero manteniendo la velocidad de ejecución y el paralelismo característicos del hardware digital.

3.7 Modulación Sigma-Delta

Esta técnica es muy utilizada en los convertidores analógico-digitales, por lo que se presentará brevemente en esta sección.

La modulación Sigma-Delta ($\Sigma\Delta$) consiste en representar una señal analógica mediante un flujo digital de muy alta frecuencia, normalmente de un solo bit, en el que la amplitud se reconstruye a partir de la densidad de pulsos. Como puede observarse en la Figura 40, las regiones donde la señal tiene valores cercanos al máximo la salida se mantiene a nivel alto más tiempo, mientras que en los mínimos predomina el nivel bajo.

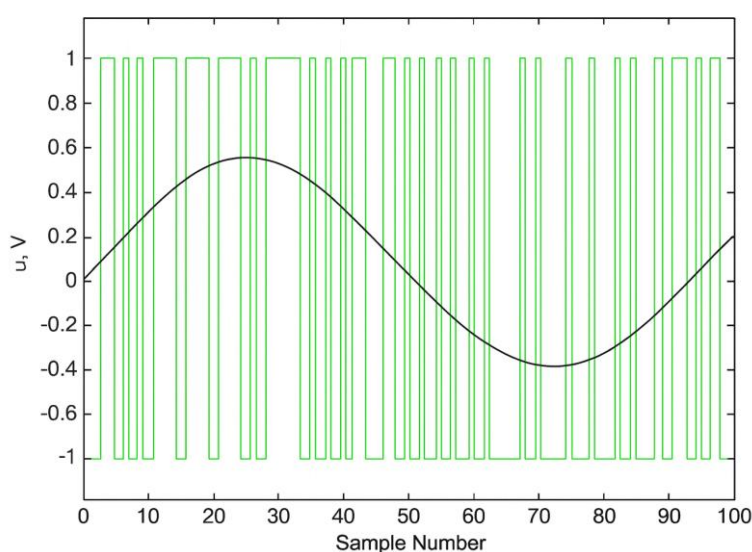


Figura 40: Modulación Sigma-Delta [38].

⁴⁵ Tiempo que tarda en generarse una salida a partir de una entrada.

Los moduladores Sigma-Delta son también conocidos como *noise shapers* o moldeadores de ruido. Esto se debe a que el error de cuantización, en lugar de distribuirse en toda la banda (incluyendo la de interés), se desplaza hacia frecuencias altas, donde resulta mucho más fácil eliminarlo mediante un filtro paso bajo⁴⁶.

Un ejemplo intuitivo: supongamos que se desea digitalizar una señal senoidal de 1 Hz utilizando únicamente dos niveles, -1 y 1 . Se obtendrá un resultado similar a una onda cuadrada. En este caso siempre aparecerá un error de cuantización significativo, que en el dominio de Fourier se manifiesta como ruido en bajas frecuencias, solapándose con la señal senoidal original. En cambio, un modulador Sigma-Delta corrige este error de cuantización en cada iteración, concentrando el ruido en frecuencias mucho más altas. De este modo, la densidad espectral de la señal digitalizada se aproxima mucho mejor a la señal original, y tras aplicar un filtro paso-bajo, la senoidal queda representada con gran fidelidad.

El esquema básico de un modulador Sigma-Delta de primer orden es como el que sigue:

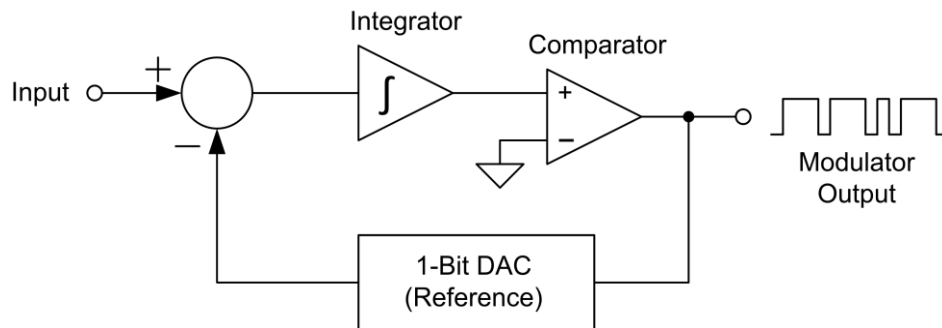


Figura 41: Modulador Sigma-Delta [39].

Primero, el sumador calcula el error que se ha cometido entre la entrada y la señal de retroalimentación. El integrador (Σ) acumula este error. A continuación, el comparador (Δ) decide en cada instante si la señal integrada está por encima de cierto umbral. Es decir, si el error acumulado es positivo su salida es de nivel alto, mientras que si es negativo la salida es de nivel bajo. La salida binaria del comparador vuelve a convertirse en una señal analógica de referencia, para que pueda volver a calcularse el error.

Un segundo integrador mejorará el funcionamiento, pues el error se acumula dos veces: el primer integrador guarda el error inmediato, y el segundo acumula cómo ese error ha ido evolucionando en el tiempo. No solo reacciona al error actual, sino también a la tendencia. Si se añaden más integradores, la eliminación de los ruidos de baja frecuencia será cada vez más agresiva, hasta que estos sean prácticamente nulos. Es lo que se conoce como el orden del modulador.

Al contrario de lo que se pueda pensar, no existe un demodulador Sigma-Delta. Dependiendo del caso, bastará con un filtro paso bajo y un diezmado (leer menos muestras por segundo), eliminando de esta forma los ruidos de alta frecuencia y recuperando la señal original.

⁴⁶ Solo deja pasar frecuencias bajas.

3.8 Protocolos de comunicación digital

Se dedica este último apartado a explicar algunos protocolos de comunicación que se utilizan en este TFG. Estos protocolos pueden entenderse como una forma específica de transmitir la información, de forma que tanto el transmisor como el receptor se entiendan entre sí (pues la Figura 39 representa un sistema ideal y no tiene en cuenta esta clase de problemas).

Se trata de comunicación en serie, es decir, los bits de información se transmiten por un único cable (en contraposición de la comunicación en paralelo, que emplea un cable por cada bit). Cada protocolo define su propia manera de organizar los datos, sincronizar el intercambio y verificar que no se produzcan errores durante la transmisión.

3.8.1 UART

El protocolo UART (Universal Asynchronous Receiver/Transmitter) es uno de los más sencillos y extendidos en sistemas digitales. Se denomina *asíncrono* porque no requiere de una señal de reloj compartida entre emisor y receptor; en su lugar, ambos dispositivos acuerdan previamente una velocidad de transmisión (*baud rate*).

La comunicación se establece mediante dos líneas:

- ✓ TX (Transmit): línea por la que el dispositivo emisor envía los datos.
- ✓ RX (Receive): línea por la que el dispositivo receptor recibe los datos.

Los bits se transmiten de forma secuencial en tramas que incluyen:

- ✓ Bit de inicio (*start bit*): indica el comienzo de la transmisión.
- ✓ Bits de datos: suelen ser 8, aunque pueden configurarse otros tamaños.
- ✓ Bit de paridad (opcional): permite una comprobación básica de errores.
- ✓ Bit de parada (*stop bit*): marca el final de la trama.

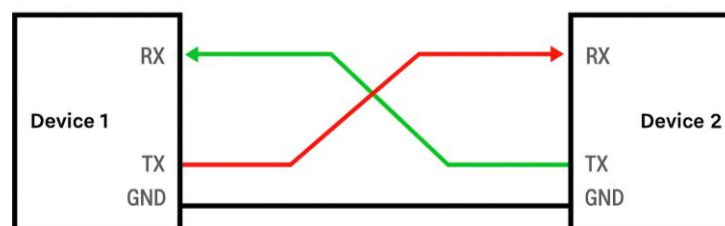


Figura 42: Esquema de conexión del protocolo UART [40].

Su simplicidad lo convierte en un protocolo ideal para depuración y comunicación entre dispositivos a corta distancia, aunque su velocidad máxima es limitada respecto a otros protocolos.

3.8.2 SPI

El protocolo SPI (Serial Peripheral Interface) es un estándar síncrono, es decir, requiere una señal de reloj compartida entre maestro y esclavos. Es ampliamente utilizado por su alta velocidad y simplicidad en la conexión de periféricos.

Las líneas principales son:

- ✓ MOSI (Master Out Slave In): datos enviados desde el maestro al esclavo.
- ✓ MISO (Master In Slave Out): datos enviados desde el esclavo al maestro.
- ✓ SCLK (Serial Clock): reloj generado por el maestro.
- ✓ SS (Slave Select): línea que habilita la comunicación con un esclavo específico.

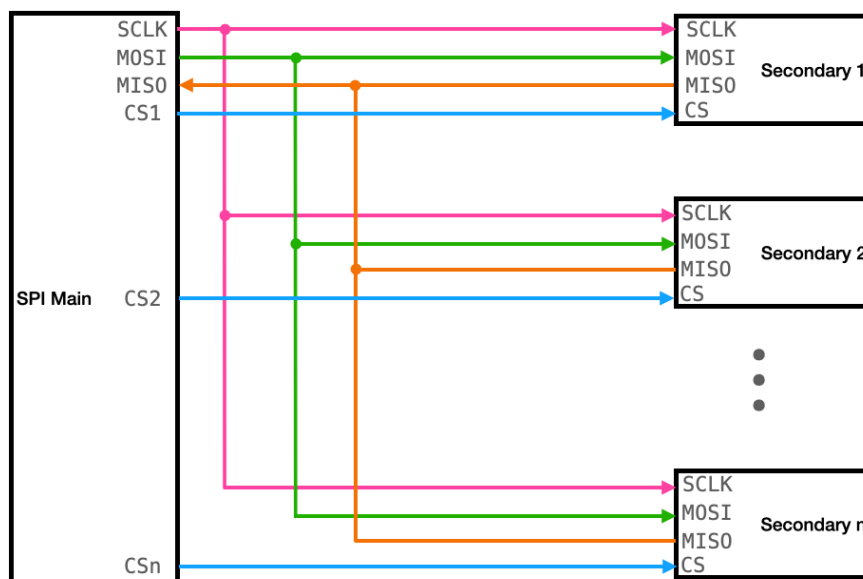


Figura 43: Esquema de conexión del protocolo SPI [41].

SPI permite comunicación *full-duplex*, ya que el envío y la recepción de datos ocurren simultáneamente. Su principal ventaja es la velocidad, pero a costa de requerir más pines que otros protocolos.

3.8.3 I²C

El protocolo I²C (Inter-Integrated Circuit), también conocido como TWI (Two Wire Interface), es un estándar síncrono de comunicación serie que emplea únicamente dos líneas:

- ✓ SDA (Serial Data): línea por la que circulan los datos.
- ✓ SCL (Serial Clock): línea de reloj, generada por el maestro.

Cada dispositivo conectado al bus I²C posee una dirección única, lo que permite conectar múltiples periféricos con tan solo dos cables.

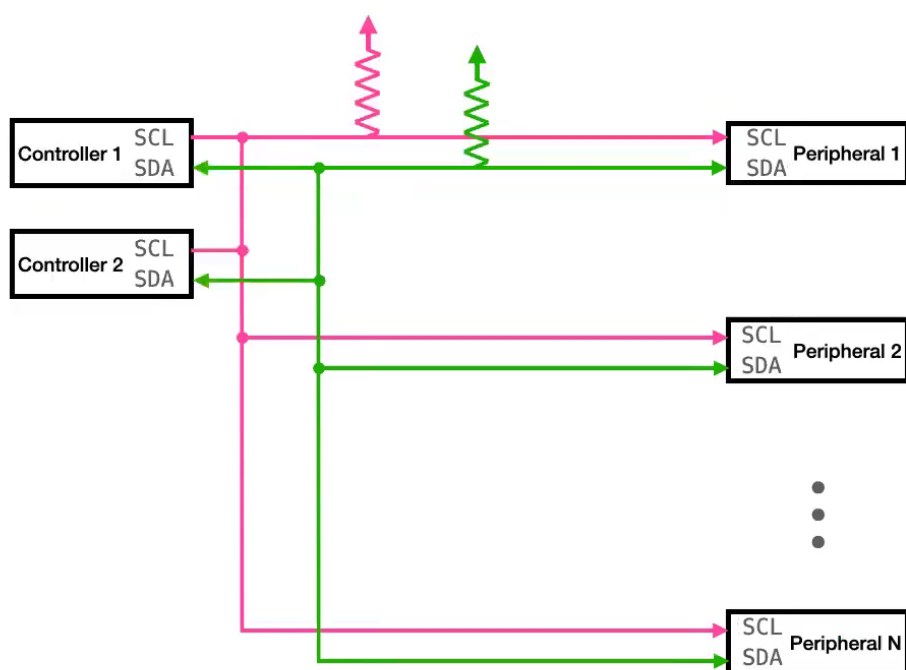


Figura 44: Esquema de conexión del protocolo I²C [42].

Como puede observarse en la Figura 44, las líneas SDA y SCL deben mantenerse a nivel alto mientras no exista ninguna comunicación, por lo que son necesarias dos resistencias de *pull-up*.

El maestro inicia la comunicación enviando una condición de inicio, seguida de la dirección del dispositivo y un bit que indica si se trata de una operación de lectura o escritura. El esclavo con dicha dirección responde con un ACK (*Acknowledge* o reconocido) poniendo la SDA a nivel bajo. A continuación, se transmite un byte de información. Tras cada byte transmitido, el receptor (que puede ser el esclavo en la escritura y el maestro en la lectura) debe enviar un ACK para confirmar la recepción correcta. Si el maestro desea parar de leer, debe enviar un NACK (*Not Acknowledge* o no reconocido) poniendo la SDA a nivel alto. Una vez finalizada la comunicación, el maestro genera la condición de parada.

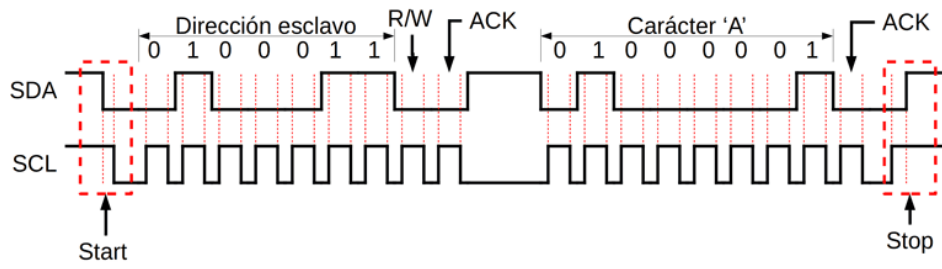


Figura 45: Protocolo I²C [43].

4 DESCRIPCIÓN DEL PMOD

La ciencia puede divertirnos o fascinarnos, pero es la ingeniería la que cambia el mundo.

- Isaac Asimov -

Una vez expuesto todo el contexto e investigación previa que hay detrás del proyecto, se describe en este capítulo el PMOD con el que se ha trabajado al mínimo detalle. Se explicará toda la electrónica que envuelve a la PCB y su funcionalidad específica dentro del sistema. Además, se ahondará en los componentes finales escogidos y otros datos económicos.

4.1 Visión general

La PCB modificada y enviada a fabricar por el alumno recibe el nombre de revisión D, siguiendo así con la cronología de Konrad Beckmann. Este diseño está mayoritariamente basado en la revisión C del PMOD [16], aunque las modificaciones realizadas se reservan para el siguiente capítulo.

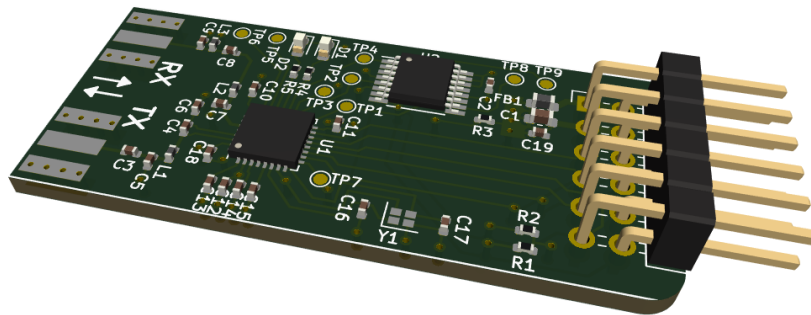


Figura 46: Diseño KiCad de la Revisión D, vista superior.

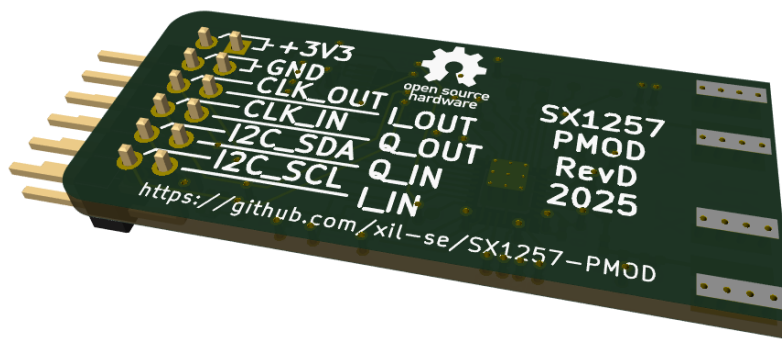


Figura 47: Diseño KiCad de la Revisión D, vista inferior.

En el modelado 3D de la tarjeta no se visualizan algunos componentes, como los conectores de las antenas. Ello se debe a que no existe en la biblioteca de KiCad ningún diseño para los mismos, aunque su huella sí que queda representada en el renderizado.

Como puede observarse en la Figura 47, en todo momento se respeta el trabajo del creador, incluyendo el enlace al repositorio del PMOD en la serigrafía de la tarjeta. Se mantiene también el nombre que se le dio (SX1257 PMOD) y el logo de *open source hardware*. Tan solo se especifica que se trata de la revisión D, diseñada en 2025.

El PMOD se conecta y comunica con la FPGA a través de 12 pines, con la siguiente distribución:

| Nombre | Entrada/Salida | Descripción |
|---------|----------------|---|
| +3V3 | - | Alimenta toda la tarjeta a 3.3 V. Usa dos pines. |
| GND | - | Toma de tierra. Usa dos pines. |
| CLK_IN | Entrada | Reloj de la FPGA. |
| CLK_OUT | Salida | Reloj del SX1257. |
| I2C_SDA | Ambos | Línea de datos del protocolo I ² C. |
| I2C_SCL | Ambos | Línea de reloj del protocolo I ² C. |
| I_IN | Entrada | La componente en fase de la señal a transmitir. |
| I_OUT | Salida | La componente en fase de la señal recibida. |
| Q_IN | Entrada | La componente en cuadratura de la señal a transmitir. |
| Q_OUT | Salida | La componente en cuadratura de la señal recibida. |

Tabla 4: Listado de pines de E/S de la revisión D.

De cara a la FPGA, las prestaciones que ofrece el PMOD pueden entenderse como una caja negra, que genera unas salidas a partir de unas entradas. Su utilización queda resumida en los siguientes pasos:

- Primero, se configura el funcionamiento de la tarjeta deseado mediante los pines designados para el protocolo I²C.
- A continuación:
 - ✓ Si se actúa como transmisor, se genera una señal de información, se codifica, se modula⁴⁷ y se envían sus componentes en fase y en cuadratura a través de los pines correspondientes.
 - ✓ Si se trabaja como receptor, la señal llegará por los pines de salida, de nuevo descompuesta en dos componentes: fase y cuadratura. Debe demodularse y decodificarse para extraer la información.
- La tasa de símbolo entre el PMOD y la FPGA debe estar definida en ambos. Es decir, el primero debe recoger datos a la misma velocidad que el otro los envía y viceversa, o por el contrario, se perderá información. Para conseguir esta sincronización, entran en juego los pines específicos para las señales de reloj. Se ahondará en ellos más adelante.

El *layout* de la PCB está organizado en cuatro capas:

- ✓ La superior, donde se sueldan todos los componentes en sus *pads* correspondientes y se facilita su colocación mediante la serigrafía. La mayoría de las pistas de cobre recorren la PCB en esta capa.
- ✓ Dos internas, reservadas como plano de tierra y plano de alimentación (técnica que ya se explicó en el capítulo dos). De esta manera, se facilita el conexionado.
- ✓ Y la inferior, destinada tan solo a realizar anotaciones y la soldadura de componentes THT. También se hallan aquí algunas pistas de cobre, puesto que no se podían unir algunos de los *pads* sin que las pistas solapasen.

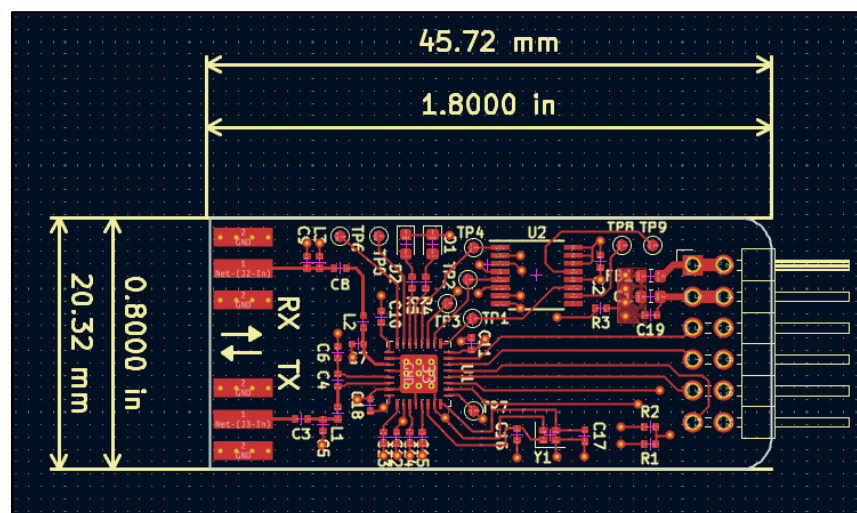


Figura 48: *Layout* de la revisión D, capa superior.

⁴⁷ El sistema admite cualquier modulación que permita descomponer la señal en las componentes I y Q (como QAM, QPSK y OFDM).

No se hará un análisis exhaustivo, aunque se comentarán algunos detalles:

- Los círculos sombreados de rojo, indicados como TP en la serigrafía, corresponden a los *test points*. No están pensados para soldar ningún componente sobre ellos, es simplemente cobre expuesto, muy útil para hacer pruebas. Puede saberse el nivel de tensión o intensidad con un multímetro o incluso la forma de onda de la señal que pasa por esa pista, si se dispone de un osciloscopio.
- En el conector, GND y VCC usan dos pines cada uno en lugar de solo uno. Esto es una técnica bastante común, para reducir riesgos y disminuir el ruido: cada pin y cada pista tienen un límite de corriente seguro. Al duplicarlos, la corriente se reparte y se evita que un único pin se sobrecaliente o que la caída de tensión sea demasiado alta. Este procedimiento se repite un poco más abajo, donde se usan cuatro vías para comunicar un polígono de cobre con el plano de alimentación.
- En general, se han tratado de colocar los componentes pasivos⁴⁸ lo más cerca posible del componente activo o pista que los requiere. Esto mejora el funcionamiento y disminuye las distorsiones.

Para organizar el *layout*, fue necesario primero darle forma al circuito a partir del esquemático de la Figura 49.

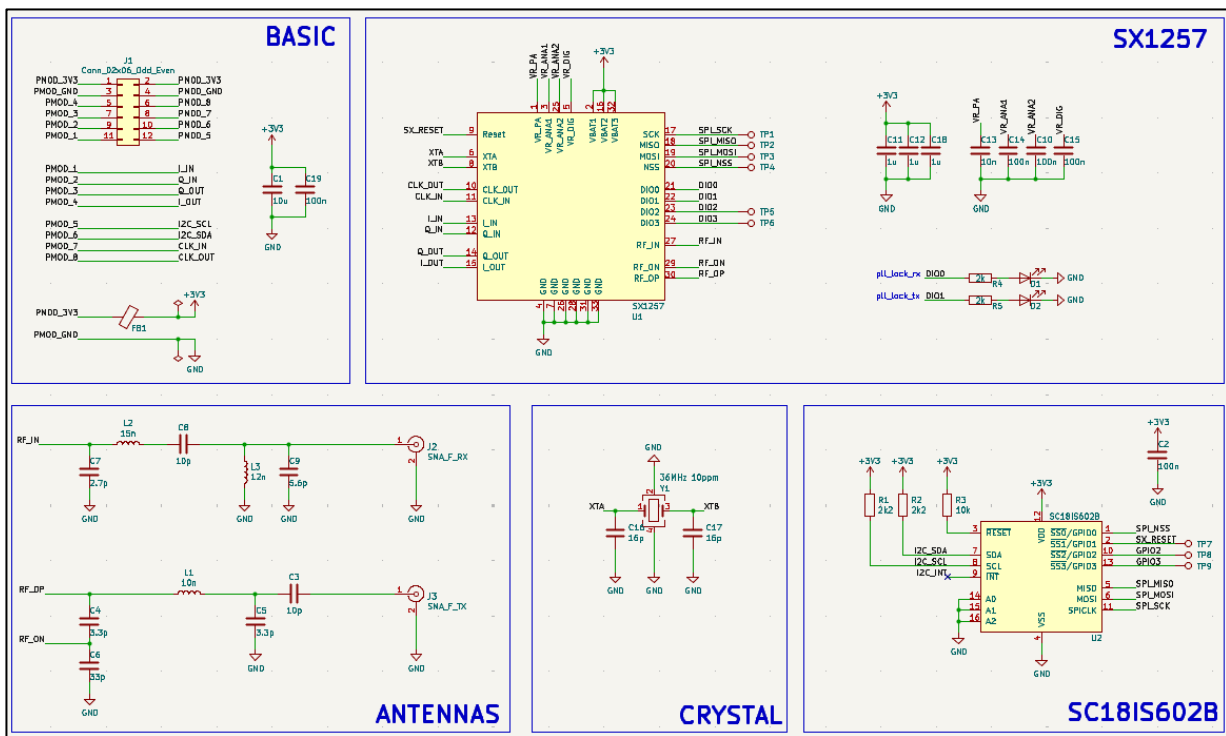


Figura 49: Esquemático de la revisión D.

En las siguientes secciones del capítulo, se desarrollarán cada uno de los bloques de electrónica en los que se divide el esquemático al mínimo detalle. Se reservará la elección concreta de componentes pasivos para más adelante, y tan solo se especificará el valor que deben tener.

⁴⁸ No generan energía, solo la consumen, almacenan o disipan (resistencias, condensadores, bobinas...). Los componentes activos en cambio pueden amplificar, controlar o generar señales; aunque necesitan alimentación externa para funcionar.

4.2 Electrónica básica

La conforman el conector de doce pines junto a dos condensadores de 100 nF y 10 uF y una ferrita, que en conjunto estabilizan la señal de alimentación.

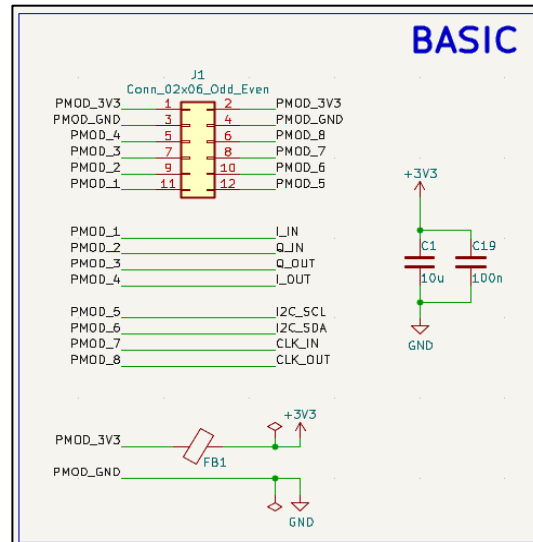


Figura 50: Revisión D. Electrónica básica.

El conector sirve como punto de unión entre la FPGA y la PCB. Tiene la distribución estandarizada de un PMOD, repartido en seis filas y dos columnas.

Los condensadores están conectados a la alimentación de 3.3 V debido a una razón estratégica: son condensadores de desacoplo. Su utilización es crucial, pues, aunque la señal de alimentación se espera en DC (Direct Current, Corriente Continua), esto nunca es así. Siempre aparecerán ruidos a altas frecuencias.

Un condensador es un componente pasivo que almacena y libera energía en forma de campo eléctrico. Está formado por dos placas conductoras separadas por un dieléctrico (material aislante). Cuando se aplica una tensión entre las placas, los electrones se acumulan en una y se vacían de la otra.

El módulo de la impedancia⁴⁹ de un condensador se rige idealmente por la siguiente fórmula:

$$|Z_c| = \frac{1}{\omega C} = \frac{1}{2\pi f C}$$

Siendo C su capacidad de almacenar energía, medida en Faradios. Como puede observarse, si la corriente es puramente continua (la frecuencia es cero), la impedancia tiende a infinito y no permite el paso de electrones: actúa como un circuito abierto. En cambio, sí que permite el paso de la corriente alterna, según la frecuencia a la que oscile y la capacidad del condensador. En concreto, un condensador de 10 uF ofrece una impedancia realmente baja para frecuencias del orden de kHz en adelante, cortocircuitando a tierra los ruidos de alta frecuencia y evitando que afecten al resto del circuito. No obstante, esto es solo teóricamente. Realmente, a frecuencias muy alejadas (para este condensador podría tratarse de decenas de MHz) dejan de ser efectivos. Es por esto que se suelen colocar varios de ellos en paralelo, con distintas capacidades, para filtrar correctamente todos los ruidos. En este caso, se ha combinado el condensador de 10 uF con uno de 100 nF, que es muy efectivo contra los ruidos del orden de MHz.

La ferrita por su parte, es un material cerámico hecho de óxidos de hierro mezclados con otros metales. También

⁴⁹ La impedancia es el concepto que describe la oposición que un circuito presenta al paso de una corriente alterna (AC, Alternating Current). Es una extensión de la resistencia, pero válida para señales que varían en el tiempo.

filtra los ruidos de alta frecuencia, pero en lugar de desviarlos a tierra como hace el condensador, simplemente impide que pasen a partir de ese punto. Además, es especialmente efectiva contra el ruido EMI (ElectroMagnetic Interference)⁵⁰.

4.3 El transceptor SX1257

Este circuito integrado protagoniza toda la funcionalidad que ofrece la PCB. Toda la información expuesta aquí al respecto del encapsulado ha sido extraída de su *datasheet* (hoja de datos) [44].

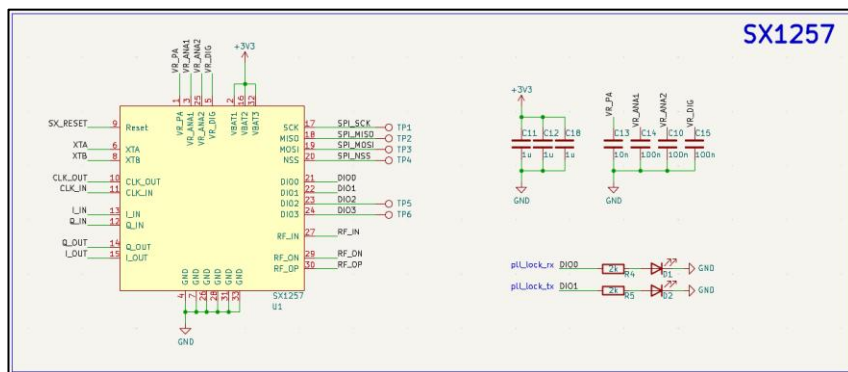


Figura 51: Revisión D. SX1257.

4.3.1 Descripción del circuito

El SX1257 de Semtech es un transceptor de radiofrecuencia de arquitectura Zero-IF⁵¹, diseñado para operar en la banda ISM (Industrial, Scientific, Medical) de Europa, Norte América y Japón, abarcando desde los 862 a los 960 MHz. Este chip, recibe las señales I y Q ya moduladas en la FPGA y se encarga de convertirlas a RF para su transmisión. A la inversa, en la recepción, toma la señal de la antena, la baja directamente a banda base y la digitaliza. Es Multi-PHY: el chip puede trabajar con múltiples tipos de capa física, es decir, con varios esquemas de modulación (FSK, QPSK, OFDM...) y protocolos físicos distintos, sin necesitar hardware adicional.

Según el valor de sus registros, el ancho de banda en el que radia puede ser:

- ✓ De banda estrecha o *narrow-band*: señales más lentas y sensibles, pero que alcanzan mayores distancias. Suelen ser intervalos menores a 25 kHz, aunque el chip alcanza un mínimo de 209 kHz.
- ✓ De banda ancha o *wide-band*: señales rápidas y robustas, aunque más adecuadas en aplicaciones de poca distancia. Se ofrece un máximo de 858 kHz. No obstante, solo se obtendrá un alto rendimiento en frecuencias inferiores a 500 kHz.

Sigue el siguiente esquema:

⁵⁰ Son señales eléctricas parasitarias, provenientes de circuitos cercanos, que se propagan a través de ondas electromagnéticas.

⁵¹ En los receptores clásicos (*superheterodinos*), la señal de RF recibida se mezcla con un oscilador local para bajarla a una frecuencia intermedia (IF, Intermediate Frequency). Esa IF se filtra y amplifica antes de digitalizarse. En un receptor Zero-IF, la señal de RF se mezcla directamente con un oscilador local de la misma frecuencia que la portadora (en banda base). Esto elimina las etapas intermedias, lo que reduce coste y tamaño del sistema.

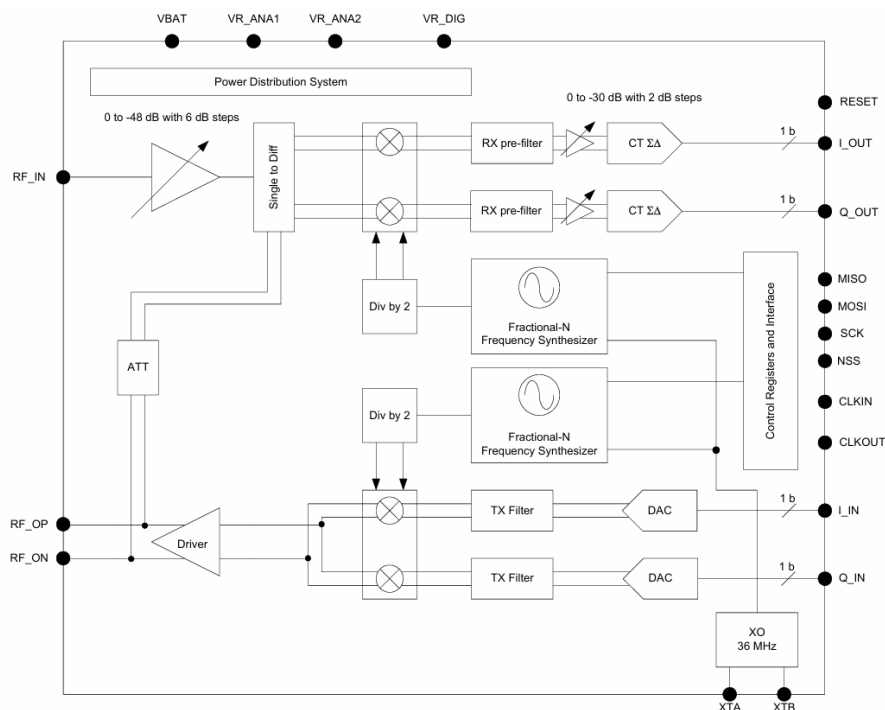


Figura 52: Esquema del SX1257 [44].

El encapsulado se alimenta a 3.3 V por tres puertos diferentes, indicados como VBAT. Para filtrar los ruidos de la alimentación, se utilizan tres condensadores de desacoplo de 1 uF, colocados lo más cerca posible de los tres *pads* de alimentación del encapsulado.

Posee un Power Distribution System (PDS) que regula internamente la energía suministrada por la señal de alimentación y la distribuye de manera estratégica. Para optimizar el funcionamiento de los reguladores de voltaje, repartidos por los distintos bloques del encapsulado, el PDS se apoya también en condensadores de desacoplo, que se conectan a los puertos de voltaje regulado: 100 nF para VR_ANA1, VR_ANA2 y VR_DIG, y 10 nF para VR_PA (que por errata no se muestra en el diagrama).

La referencia de tiempo principal de todo el chip la provee un oscilador local (XO), sirviéndose de un cristal de cuarzo al que excita para que oscile a 36 MHz con una tolerancia de 10 ppm (partes por millón, que son 360 Hz). Puesto que un cristal genera ruidos de muy alta frecuencia, se utilizan dos condensadores de desacoplo de 16 pF. El cristal y el oscilador local se comunican a través de los pines XTA y XTB.

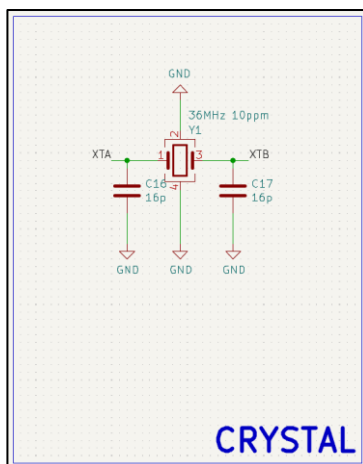


Figura 53: Revisión D. Cristal de cuarzo.

Una vez que se consigue una señal de reloj estable, se activarán y calibrarán los PLLs y se habilitará la señal CLK_OUT (una de las salidas del PMOD), que puede utilizarse para depuración o sincronización.

Los PLLs serán los encargados de elevar nuestra señal de información a radiofrecuencia. Los VCOs están centrados en torno a los 1.9 GHz, de forma que, al dividir la señal entre dos, se obtienen dos portadoras en la banda de interés, desfasadas exactamente 90 grados una de la otra. En un PLL Integer-N clásico, el divisor N solo podría tomar valores enteros, por lo que solo se podrían conseguir frecuencias en pasos de 36 MHz.

$$f_{salida} = N \cdot f_{referencia}$$

Siendo la frecuencia de referencia de 36 MHz. No obstante, al tratarse de un PLL Fractional-N, se resuelve esta limitación. Realmente, no es posible que N sea de naturaleza racional, pero se consigue esa ilusión utilizando un modulador Sigma-Delta de 19 bits de resolución. Lo que significa que:

$$f_{paso} = \frac{f_{referencia}}{2^{19}} = 68.66 \text{ Hz}$$

La frecuencia de salida final se decide mediante un valor repartido en tres registros (véase la sección de registros). Es necesario escribir en el tercer registro (bits menos significativos) para que se efectúen los cambios realizados en los dos primeros.

$$f_{salida} = f_{paso} \cdot RegFrfRx|Tx$$

Es responsabilidad del usuario escoger una frecuencia que cumpla con la normativa estatal. En el caso de Europa, la banda ISM abarca desde los 863 a los 870 MHz.

El ancho de banda del filtro de lazo también puede decidirse con los registros TxPllBw y RxPllBw. En general, para un ancho de banda estrecho, se atenuarán los ruidos de forma contundente, pero se tardará más en enganchar la fase, mientras que un ancho de banda grande significará un enganche de fase rápido, pero el sistema no será tan efectivo frente a ruidos.

4.3.2 Bloque de transmisión

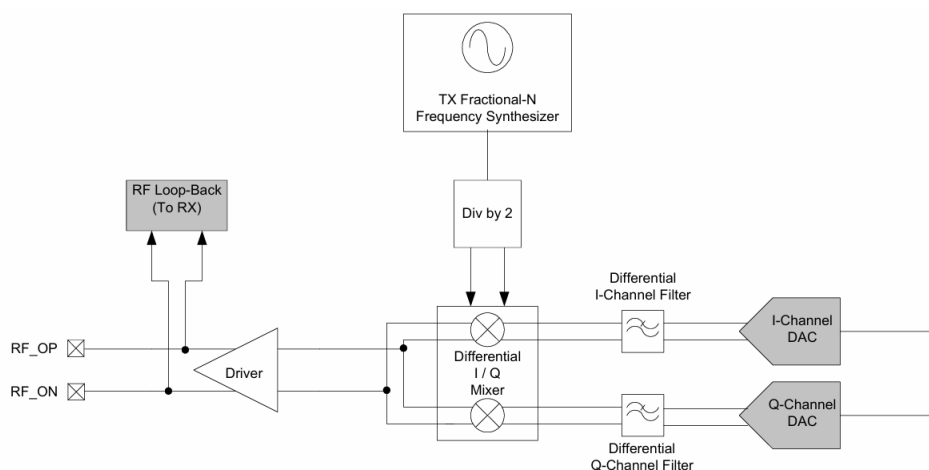


Figura 54: SX1257. *Front-end*⁵² del transmisor [44].

⁵² Es la parte de un sistema que está en contacto directo con la señal del mundo exterior. El back-end en cambio, correspondería a la parte digital de procesamiento y control.

El transmisor recibe la señal a transmitir descompuesta en las señales I_IN y Q_IN, ya moduladas en Sigma-Delta. En concreto, se espera un *bitstream* de 32 o 36 M de muestras por segundo, pues los datos se leen durante el flanco de bajada de una señal de reloj oscilando a una de esas dos frecuencias. Si se quiere usar la de 32 M, será necesario utilizar una señal de reloj externa, proporcionada a través de CLK_IN.

El DAC tiene 8 bits de resolución y un ancho de banda configurable mediante el registro TxDacBw, desde los 290 a los 500 kHz. Convierte cada componente a una señal analógica, atenuando el ruido de cuantización de forma gruesa y económica. El ancho de banda se decide con el número de coeficientes del filtro digital:

$$N = 24 + 8 \cdot TxDacBw$$

Teniendo un mínimo de 24 y un máximo de 64. A más coeficientes, ancho de banda más estrecho. El *datasheet* solo especifica 450 kHz para 32 coeficientes y 290 kHz para 64, aunque pueden interpolarse los demás.

A la salida del DAC, se añade un filtro analógico, que no solo eliminará los ruidos analógicos y de cuantización residuales, sino que estará delimitando la banda de transmisión. Esta banda puede calcularse a partir de esta expresión (resultado en MHz):

$$BW_{3dB} = \frac{17.15}{2 \cdot (41 - RegTxBwAna)}$$

La banda de transmisión debe situarse algo por debajo del ancho de banda programado en el DAC.

El mezclador, eleva las componentes en fase y cuadratura a la frecuencia de las portadoras generadas por el PLL. Tiene ganancia configurable a través del registro RegTxGain:

$$G = -38 + 2 \cdot TxMixerGain$$

En dB. Debe utilizarse este registro para limitar la potencia de salida.

A continuación, las señales pasan por un amplificador, que saca del encapsulado la señal modulada en forma diferencial (RF_OP y RF_ON, una desfasada 180 grados respecto a la otra). Esta señal también se utiliza como señal interna para calibrar la ganancia del receptor. La razón de usar dos señales diferenciales reside en la cancelación de ruido: si el ruido EMI se acoplase por igual en las dos líneas, al estar desfasadas, se cancelaría. Ofrece también una menor radiación interna y eficiencia en alta frecuencia.

La potencia de salida pico es de 5 dBm (3.16 mW), pero es recomendable dejar un margen suficiente para prevenirse de posibles picos.

El amplificador tiene una impedancia de 100 ohmios de salida diferencial, lo que equivale a 50 ohmios por cada rama.

4.3.3 Bloque de recepción

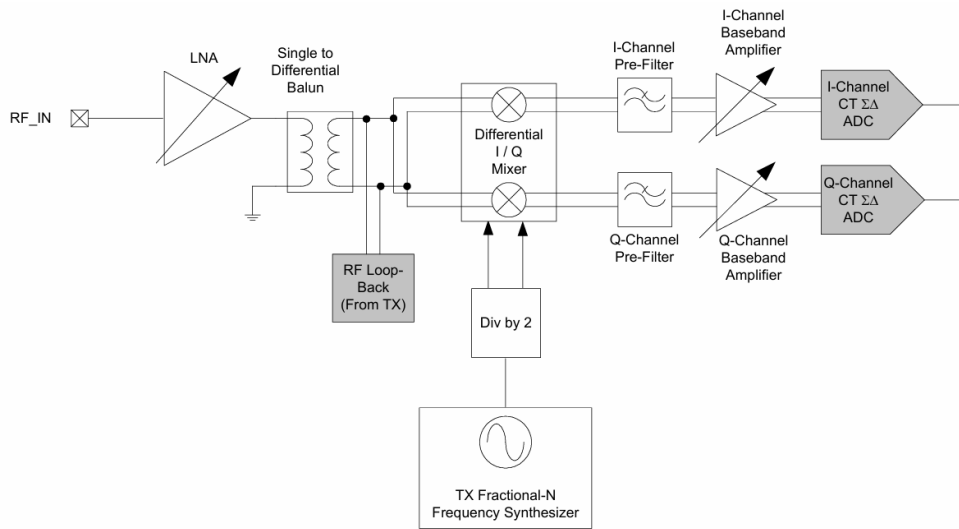


Figura 55: SX1257. *Front-end* del receptor [44].

La señal recibida entra en el encapsulado por el puerto RF_IN, con una amplitud muy atenuada (del orden de nV). El LNA (Low Noise Amplifier), con ganancia e impedancia configurables a través de RegRxAnaGain, la amplifica sin añadir apenas ruido.

A continuación, la señal de alta frecuencia se convierte a modo diferencial mediante un balún y un mezclador devuelve ambas señales a banda base, a la par que las descompone en fase y en cuadratura. Un prefiltro analógico, con ancho de banda configurable entre 250 y 750 kHz definirá la banda de recepción. Como puede intuirse, este valor debe situarse algo por encima de la banda especificada en la transmisión, de forma que esta última quede totalmente confinada en el ancho de banda de la recepción con cierto margen de error.

Las señales pasan ahora por una nueva etapa de amplificación (amplificador de banda base), que se rige por la siguiente expresión:

$$G = -24 + 2 \cdot RxBasebandGain$$

En dB.

Por último, los ADCs convierten la señal en digital mediante una modulación Sigma-Delta, que transmiten por I_OUT y Q_OUT a una velocidad de 32 o 36 M de muestras por segundo, según si se utiliza CLK_OUT o CLK_IN como señal de reloj. El ancho de banda de los ADCs puede configurarse hasta un máximo de 500 KHz.

4.3.4 Registros

La configuración del encapsulado se hace a través de registros, tales como los que se han comentado en las secciones anteriores. Para reescribir o leer los registros, se utilizan los puertos NSS, SCK, MOSI y MISO, pertenecientes al protocolo SPI. Para hacer comprobaciones, estas señales son accesibles a través de cuatro *test points*.

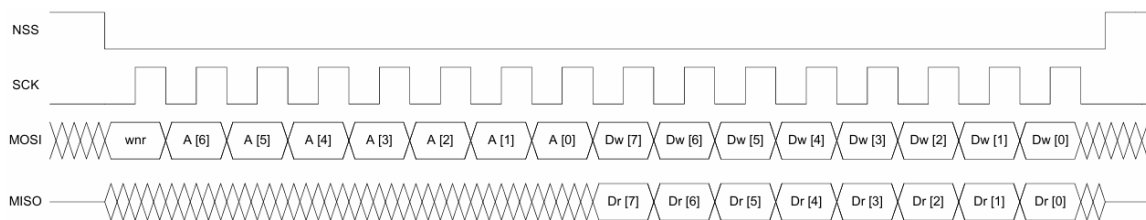


Figura 56: SX1257. Protocolo SPI [44].

Una vez que el NSS está a nivel bajo, se espera el bit de lectura (0) o escritura (1), junto a la dirección de 7 bits del registro a modificar. A continuación, se envía el byte de datos que se desea guardar en memoria o un *dummy* byte (0x00 ó 0xFF) para leer la información que hay almacenada en el registro. Al finalizar, el NSS vuelve a nivel alto.

El modo de comunicación del reloj SPI queda definido por su CPOL (Clock Polarity) y su CPHA (Clock Phase), ambos a 0. Esto indica que la señal de reloj estará inactiva a nivel bajo y se capturarán los datos en el flanco de subida (el método más habitual).

Los registros disponibles y su dirección de acceso quedan expuestos en las siguientes figuras.

| Address | Register Name | Reset Value | Default Value | Description |
|---------|----------------|-------------|---------------|---|
| 0x00 | RegMode | 0x00 | | Operating modes of the SX1257 |
| 0x01 | RegFrRxMsb | 0xCB | | RX carrier frequency MSB |
| 0x02 | RegFrRxMid | 0x55 | | RX carrier frequency intermediate bits |
| 0x03 | RegFrRxLsb | 0x55 | | RX carrier frequency LSB |
| 0x04 | RegFrTxMsb | 0x00 | | RX carrier frequency MSB |
| 0x05 | RegFrTxMid | 0xCB | | RX carrier frequency intermediate bits |
| 0x06 | RegFrTxLsb | 0x55 | | RX carrier frequency LSB |
| 0x07 | RegVersion | 0x01 | | Semtech ID relating to the silicon revision |
| 0x08 | RegTxGain | 0x2E | | TX DAC and mixer gain setting |
| 0x0A | RegTxBw | 0x30 | | TX FE PLL and analog filter bandwidths |
| 0x0B | RegTxDacBw | 0x02 | | TX DAC bandwidth |
| 0x0C | RegRxAnaGain | 0x3F | | RX FE LNA and baseband amplifier gain |
| 0x0D | RegRxBw | 0xFD | | RX FE ADC and analog filter bandwidths |
| 0x0E | RegRxPLLBw | 0x06 | | RX FE PLL bandwidth |
| 0x0F | RegDioMapping | 0x00 | | Mapping of DIO pins |
| 0x10 | RegClkSelect | 0x02 | | Sampling clock configuration |
| 0x11 | RegMode Status | 0x00 | | SX1257 mode status |
| 0x1A | RegLowBatThres | 0x02 | | Low battery threshold |

Figura 57: SX1257. Registros [44].

| Name (Address) | Bits | Variable Name | Mode | Default Value | Description |
|----------------|------|----------------|------|---------------|---|
| RegMode (0x00) | 7-4 | - | r | 0000 | unused |
| | 3 | PADriverEnable | rw | 0 | Enables the TX PA driver 0 = Disabled 1 = Enabled |
| | 2 | TxEnable | rw | 0 | Enables the TX front-end and PLL (except the TX PA driver) 0 = Disabled 1 = Enabled |
| | 1 | RxEnable | rw | 0 | Enables the RX front-end and PLL 0 = Disabled 1 = Enabled |
| | 0 | StandbyEnable | rw | 1 | Enables the PDS and the oscillator in Standby Mode 0 = Disabled 1 = Enabled |

Figura 58: SX1257. Registros de modo de funcionamiento⁵³ [44].

| Name (Address) | Bits | Variable Name | Mode | Default Value | Description |
|--------------------|------|---------------|------|---------------|--|
| RegFrRxMsb (0x01) | 7-0 | FrRx(23:16) | rw | 0xCB | MSB of the RX carrier frequency |
| ReffFrRxMid (0x02) | 7-0 | FrRx(15:8) | rw | 0x55 | Middle byte of the RX carrier frequency |
| ReffFrRxLsb (0x03) | 7-0 | FrRx(7:0) | rw | 0x55 | LSB of the RX carrier frequency $F_{RF} = F_{STEP} \times Frfxx(23, 0)$ With a 36 MHz XO default value 0xCB5555 FrRx = 915 MHz and frequency resolution = 68.66455 Hz The RX RF frequency is updated only under the following conditions:Hz (36 MHz XO) - RefFrRxLsb is written - When exiting SLEEP mode |
| RegFrTxMsb (0x04) | 7-0 | FrTx(23:16) | rw | 0xCB | MSB of the TX carrier frequency |
| ReffFrTxMid (0x05) | 7-0 | FrTx(15:8) | rw | 0x55 | Middle byte of the TX carrier frequency |
| ReffFrTxLsb (0x06) | 7-0 | FrTx(7:0) | rw | 0x55 | LSB of the TX carrier frequency $F_{RF} = F_{STEP} \times Frfxx(23, 0)$ With a 36 MHz XO default value 0xCB5555 FrTx = 915 MHz and frequency resolution = 68.66455 Hz The TX RF frequency is updated only under the following conditions:Hz (36 MHz XO) - RefFrTxLsb is written - When exiting SLEEP mode |

Figura 59: SX1257. Registros del PLL [44].

⁵³ Hay una errata: el valor por defecto del registro 0x00 es 0x00, no 0x01.

| Name (Address) | Bits | Variable Name | Mode | Default Value | Description |
|--------------------|------|---------------|------|---------------|--|
| RegRevision (0x07) | 7-0 | Version | r | 0x21 | IC version code (Current revision V2A) |

Figura 60: SX1257. Registros de revisión [44].

| Name (Address) | Bits | Variable Name | Mode | Default Value | Description |
|-------------------|------|---------------|------|---------------|--|
| RegTxGain (0x08) | 7 | - | r | 0 | Unused |
| | 6-4 | TxDacGain | rw | 010 | DAC gain, programmable in 3 dB steps: 000 = maximum gain - 9 dB 001 = maximum gain - 6 dB 010 = maximum gain - 3 dB 011 = maximum gain (0 dB full scale) |
| | 3-0 | TXMixerGain | rw | 1110 | Mixer gain, programmable in 2 dB steps: $Gain \sim -38 + 2 \times Int(TxMixerGain(3, 0))dB$ |
| RegTxBw (0x0A) | 7 | - | r | 0 | unused |
| | 6-5 | TxPllBw | rw | 11 | TX PLL bandwidth, programmable: 00 = 75 kHz 01 = 150 kHz 10 = 225 kHz 11 = 300 kHz |
| | 4-0 | TxAnaBw | rw | 0000 | TX analog filter bandwidth, programmable: $Bandwidth = \frac{17.5}{2 \times (41 - Int(TxBw(4, 0)))} MHz$ |
| RegTxDacBw (0x0B) | 7-3 | - | r | 00000 | unused |
| | 2-0 | TxDacBw | rw | 010 | Number of taps of TX FIR-DAC, programmable: $N = 24 + 8 \times Int(TxDacBw(2, 0))$ where $N_{MAX} = 64$ |

Figura 61: SX1257. Registros del transmisor⁵⁴ [44].

⁵⁴ En la descripción del registro RegTxBw hay una errata. La fórmula para calcular el ancho de banda debe tomar de numerador 17.15, en lugar de 17.5.

| Name (Address) | Bits | Variable Name | Mode | Default Value | Description |
|---------------------|------|----------------|------|---------------|--|
| RegRxAnaGain (0x0C) | 7-5 | RxLnaGain | rw | 001 | RX LNA gain setting: 000 = not used 001 = G1 = highest gain power - 0 dB 010 = G2 = highest gain power - 6 dB 011 = G3 = highest gain power - 12 dB 100 = G4 = highest gain power - 24 dB 101 = G5 = highest gain power - 36 dB 110 = G6 = highest gain power - 48 dB 111 = not used |
| | 4-1 | RxBasebandGain | rw | 1111 | RX Baseband amplifier gain, programmable: $Gain = Gain + 2 \times Int(RxBasebandGain(4,1))dB$ |
| | 1 | LnaZin | rw | 1 | LNA input impedance 0 = 50 Ω 1 = 200 Ω |
| RegRxBw (0x0D) | 7-5 | RxAdcBw | rw | 111 | RX ADC BW, programmable: 010 = 100 kHz < RxAdcBw < 200 kHz 101 = 200 kHz < RxAdcBw < 400 kHz 111 = 400kHz < RxAdcBw |
| | 4-2 | RxAdcTrim | rw | 111 | RX ADC trim 32 MHz reference crystal: RxAdcTrim = 110 36 MHz reference crystal: RxAdcTrim = 101 |
| | 1-0 | RxBasebandBw | rw | 01 | Bandwidth of RX analog roofing filter, programmable: 00 = 750 kHz 01 = 500 kHz 10 = 375 kHz 11 = 250 kHz |
| RegRxPLLBw (0x0E) | 7-3 | - | r | 0000 | unused |
| | 2-1 | RxPllBw | rw | 11 | RX PLL bandwidth, programmable: 00 = 75 kHz 01 = 150 kHz 10 = 225 kHz 11 = 300 kHz |
| | 0 | RxAdcTemp | rw | 0 | RX ADC temperature measurement mode 0 = Disabled 1 = Enabled |

Figura 62: SX1257. Registros del receptor [44].

| Name (Address) | Bits | Variable Name | Mode | Default Value | Description |
|----------------------|------|---------------|------|---------------|--|
| RegDioMapping (0x0F) | 7-6 | Dio0Mapping | rw | 00 | Mapping of pins DIO0 to DIO3. See Table 11 for further information. |
| | 5-4 | Dio1Mapping | rw | 00 | |
| | 3-2 | Dio2Mapping | rw | 00 | |
| | 1-0 | Dio3Mapping | rw | 00 | |

Figura 63: SX1257. Registros de los puertos DIO (Digital Input/Output) [44].

| Name (Address) | Bits | Variable Name | Mode | Default Value | Description |
|----------------------|------|-----------------|------|---------------|--|
| RegClkSelect (0x10) | 7-4 | - | r | 0000 | unused |
| | 3 | DigitalLoopBack | rw | 0 | Digital loop back enable 0 = Disabled 1 = Enabled |
| | 2 | RfLoopBack | rw | 0 | RF Loop back enable for RX analog gain calibration 0 = Disabled 1 = Enabled |
| | 1 | Clk_out | rw | 1 | CLK_OUT enable 0 = Disabled 1 = Enabled |
| | 0 | TxDacClkSelect | rw | 0 | Clock select for TX DAC 0 = Internal clock (XTAL) 1 = External clock (CLK_IN) |
| RegModeStatus (0x11) | 7-3 | - | r | 00000 | unused |
| | 2 | LowBatEnable | rw | 0 | LowBat detect flag 0 = VBAT > LowBat threshold 1 = VBAT < LowBat threshold (low battery) |
| | 1 | PllLockRx | r | 0 | Asserted when RX PLL locked |
| | 0 | PllLockTx | r | 0 | Asserted when TX PLL locked |

| Name (Address) | Bits | Variable Name | Mode | Default Value | Description |
|-----------------------|------|---------------|------|---------------|--|
| RegLowBatThres (0x1A) | 7-3 | - | r | 00000 | unused |
| | 2-0 | LowBatThres | rw | 010 | LowBat Threshold 000 = 2.516 V 001 = 2.619 V 010 = 2.724 V 011 = 2.829 V 100 = 2.935 V 101 = 3.037 V 110 = 3.143 V 111 = 3.245 V |

Figura 64: SX1257. Registros adicionales [44].

Todos los registros vuelven a su valor de *reset* al conectar la alimentación. Adicionalmente, puede forzarse manualmente un *reset* a través del pin RESET.

4.3.5 Otras funciones

Existe la posibilidad de habilitar el ADC del receptor como un sensor de medición de temperatura del transceptor, en cuyo caso debe encontrarse en modo Standby. La lectura se devuelve por Q_OUT.

El sistema cuenta también con la posibilidad de habilitar *loop-backs* a nivel de radiofrecuencia y a nivel digital, es decir, la señal se traslada directamente al receptor, obviando las etapas posteriores. Esta funcionalidad es muy útil, pues puede corroborarse el funcionamiento del encapsulado sin necesidad de antenas ni redes de adaptación.

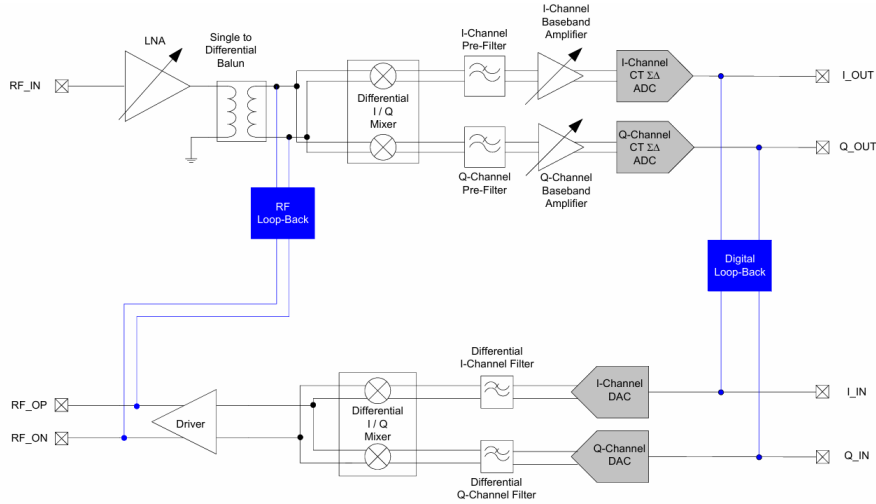


Figura 65: SX1257. Loop-back [44].

Los DIOs también pueden utilizarse para hacer comprobaciones, como el estado de la batería, el funcionamiento del oscilador local o si los PLLs se han estabilizado en la frecuencia deseada.

| Mode | Diox Mapping | DIO3 | DIO2 | DIO1 | DIO0 |
|---------|--------------|-------------|------------|-------------|-------------|
| Sleep | 00 | - | - | - | - |
| | 01 | - | - | - | - |
| | 10 | - | - | - | - |
| | 11 | - | - | - | - |
| Standby | 00 | - | xosc_ready | - | - |
| | 01 | - | - | - | - |
| | 10 | - | - | - | - |
| | 11 | - | - | - | - |
| RX | 00 | pll_lock_rx | - | - | pll_lock_rx |
| | 01 | - | - | - | pll_lock_rx |
| | 10 | - | - | - | pll_lock_rx |
| | 11 | - | - | - | Low Bat |
| TX | 00 | pll_lock_tx | - | pll_lock_tx | - |
| | 01 | - | - | - | - |
| | 10 | - | - | - | - |
| | 11 | - | - | - | - |

Figura 66: SX1257. Mapa de DIOs [44].

En concreto, se han conectado dos LEDs a los DIOs 0 y 1, para comprobar el estado de los PLLs de los bloques de transmisión y recepción (pues en los modos de funcionamiento TX y RX, muestran esa información por defecto), de forma que se encienden cuando enganchan la fase. Se añaden también dos resistencias de 2 K Ω , para limitar la corriente que circula por los diodos.

Los DIOs 2 y 3 también son comprobables, a partir de dos *test points*.

4.4 SC18IS602B: El puente I²C-SPI

Aunque el SX1257 hace uso del protocolo SPI, este no es el más interesante, pues necesita de al menos tres líneas (MOSI, MISO y SCLK) y tantos cables extra como esclavos se desean conectar. Por lo tanto, es preferible utilizar el protocolo I²C, que solo requiere de dos cables, pues cada esclavo es accesible por su dirección. Para convertir la información, transmitida desde la FPGA en formato I²C, a SPI, de forma que los datos sean legibles por el transceptor, se implementa el encapsulado SC18IS602B, que actúa como puente entre ambos protocolos [45].

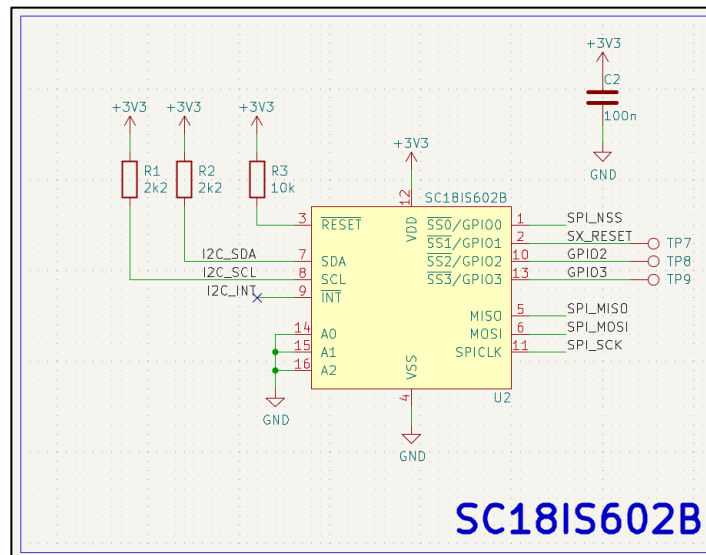


Figura 67: Revisión D. SC18IS602B.

En primer lugar y como viene siendo frecuente, se conecta un condensador de desacoplo de 100 nF junto al pin de alimentación, indicado como VDD (Voltage at the Drain). Además, se observa en el esquema que al desconectar la alimentación, también se hace *reset* en el dispositivo (RESET es un puerto con activación a nivel bajo, de ahí la línea horizontal sobre el nombre). Se utiliza también una resistencia de 10 K Ω para proteger el encapsulado de posibles picos de corriente. La tierra se conecta al puerto VSS (Voltage at the Source)⁵⁵.

La información llega a través de SDA y SCL, y la tarjeta integra ya las resistencias de pull-up de 2.2 K Ω , por lo que no es necesario sintetizarlas en la FPGA. La dirección de esclavo del encapsulado la conforman 7 bits, a los que se le añade un octavo para indicar si se va a leer (1) o escribir (0).

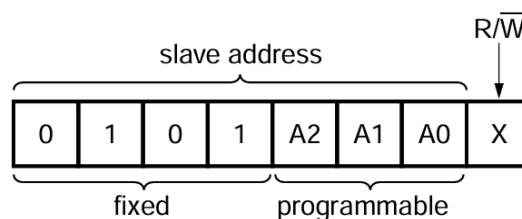


Figura 68: SC18IS602B. Dirección de esclavo [45].

⁵⁵ Aquí es necesaria realizar una aclaración. Antes de los MOSFETs, la tecnología estaba basada mayoritariamente en otros transistores: Los BJT (Bipolar Junction Transistor), formados por una base, un colector y un emisor, análogos a la puerta, el drenador y la fuente en el MOSFET. Las tecnologías basadas en MOSFETs, usan VDD – VSS como referencias de tensión, mientras que las basadas en BJT, lo especifican mediante VCC – VEE. No obstante, muchos sistemas utilizan esta última terminología por tradición, a pesar de que la electrónica basada en BJTs está cada vez más en desuso.

Los bits programables de la dirección, A1, A2 y A3, se han cortocircuitado a tierra, obteniendo una dirección fija: 0101000 + R/W (0x50 para escribir, 0x51 para leer).

Toda la comunicación con el SC18IS602B se realiza a través de un *buffer* (búfer) de datos, con dos posibles acciones: escribir en él (W) o leer de él (R). La acción escogida la define el octavo bit. Si se desea escribir, será necesario enviar primero el código de la operación a realizar (FunctionID), seguido de un número concreto de bytes de datos, que varía de 0 a 200. En cambio, si se desea leer, tan solo se debe enviar un *dummy* byte por cada byte que se va a recibir. Tras cada byte de datos, se envía un ACK (o NACK si se desea finalizar la lectura), como es común en este protocolo.

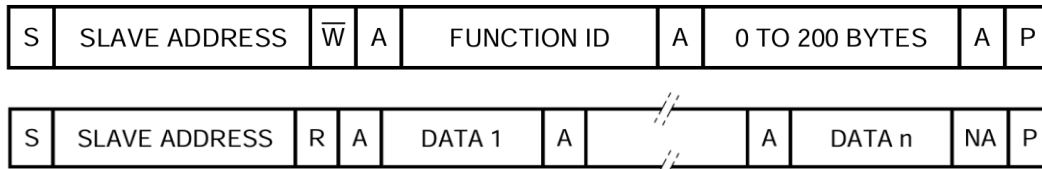


Figura 69: SC18IS602B. Escritura y lectura en el *buffer* [45].

La función más importante que ofrece el dispositivo es la comunicación por SPI, que sigue el siguiente esquema:

| | | | | | | | |
|---|---|---|---|-----|-----|-----|-----|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | SS3 | SS2 | SS1 | SS0 |

Figura 70: SC18IS602B. FunctionID para comunicación SPI [45].

Por lo tanto, basta con indicar qué esclavo se va a utilizar con un 1 en el bit correspondiente. En este caso, el transceptor está conectado al esclavo 0 (SS0), por lo que la función quedaría: 00000001 (0x01 en hexadecimal). A través de los pines MISO, MOSI y SPICLK se transmite la información en el nuevo formato. El siguiente byte, correspondería a la dirección del registro del SX1257 a modificar, siendo el primer bit el indicador de lectura/escritura. Por último, se envían los bytes de datos (o *dummy* bytes para leer).

El proceso de escritura en los registros del transceptor quedaría así:

- ✓ Inicio.
- ✓ Enviar dirección del esclavo + W.
- ✓ Enviar FunctionID para comunicación SPI, habilitando el esclavo adecuado.
- ✓ Enviar W + dirección del registro del SX1257.
- ✓ Enviar byte de datos.
- ✓ Parada.

Mientras que el de lectura:

- ✓ Inicio.
- ✓ Enviar dirección del esclavo + W.

- ✓ Enviar FunctionID para comunicación SPI, habilitando el esclavo adecuado.
- ✓ Enviar R + dirección del registro del SX1257.
- ✓ Enviar *dummy* byte.
- ✓ Parada.
- ✓ Inicio.
- ✓ Enviar dirección del esclavo + R.
- ✓ Enviar *dummy* byte.
- ✓ Parada.

Nótese que primero hay que escribir en el puente para poder leer del SX1257. Una vez que los datos están en el *buffer* del SC18IS602B, se produce una nueva comunicación con el puente, esta vez en modo de lectura.

El reloj SPI puede configurarse mediante la función 0x0F. Tras su envío, se transmite un byte de datos que indica la configuración, organizada de la forma que se muestra en la Figura 71.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|--------|---|---|-------|---|-------|-------|----|----|
| Symbol | X | X | ORDER | X | MODE1 | MODE0 | F1 | F0 |
| Reset | X | X | 0 | X | 0 | 0 | 0 | 0 |

| Bit | Symbol | Description |
|-----|-------------|--|
| 7:6 | - | reserved |
| 5 | ORDER | When logic 0, the MSB of the data word is transmitted first. If logic 1, the LSB of the data word is transmitted first. |
| 4 | - | reserved |
| 3:2 | MODE1:MODE0 | Mode selection 00 - SPICLK LOW when idle; data clocked in on leading edge (CPOL = 0, CPHA = 0) 01 - SPICLK LOW when idle; data clocked in on trailing edge (CPOL = 0, CPHA = 1) 10 - SPICLK HIGH when idle; data clocked in on trailing edge (CPOL = 1, CPHA = 0) 11 - SPICLK HIGH when idle; data clocked in on leading edge (CPOL = 1, CPHA = 1) |
| 1:0 | F1:F0 | SPI clock rate 00 - 1843 kHz 01 - 461 kHz 10 - 115 kHz 11 - 58 kHz |

Figura 71: SC18IS602B. Byte de configuración para reloj SPI [45].

El encapsulado generará una interrupción después de cualquier transmisión SPI, poniendo el pin INT a nivel bajo. Para limpiarla, debe utilizarse la función 0xF1. Para evitar molestias, este pin se deja en alta impedancia (circuito abierto). También de esta forma se ahorran pines en el conector del PMOD.

Por otra parte, los pines correspondientes a los SS también pueden configurarse de forma alternativa como GPIOs, a través del comando 0xF6. El byte de datos enviado a continuación define dicha disposición, significando un 1 que se va a usar como GPIO y un 0 que actuarán como SS.

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|-----|-----|-----|-----|
| X | X | X | X | SS3 | SS2 | SS1 | SS0 |

Figura 72: SC18IS602B. Byte de habilitación de GPIOs [45].

Aprovechando esta mecánica, se conecta el puerto RESET del SX1257 al GPIO1 del puente. De esta forma, puede programarse el reinicio del dispositivo. Como un añadido, se colocan *test points* en los GPIOs 1, 2 y 3 expandiendo la posibilidad de hacer comprobaciones.

Para leer o escribir en los GPIOs, se utilizan las funciones 0xF5 y 0xF4 respectivamente. El byte de datos siguiente sigue la misma estructura que la Figura 72.

Los GPIOs pueden configurarse también de manera más avanzada, en términos de disposición del hardware interno, mediante el comando 0xF7. Aunque no se profundizará en la electrónica, las distintas configuraciones quedan descritas en la Tabla 5.

| Nombre | Descripción |
|---------------------|---|
| Input-Only | Solo sirve para leer. No se puede forzar 0 o 1. |
| Push-Pull | Solo sirve para escribir. Permite forzar un 1 fuerte ⁵⁶ o un 0 fuerte. Se puede leer, pero como la salida siempre está activamente en 1 o en 0, la lectura siempre reflejará el valor que uno mismo fuerza, no el externo. |
| Open-Drain | Solo puede forzarse un 0 fuerte. Si se intenta forzar un 1, el pin queda en alta impedancia, por lo que su valor de salida dependerá de lo que se esté leyendo. Al leer ocurre igual, solo se está leyendo si se fuerza un 1, ya que, si se fuerza un 0, siempre se leerá un 0. |
| Quasi-Bidireccional | Se puede forzar un 1 débil o un 0 fuerte. Esto es lo que permite leer, ya que, al ser un 1 débil, <i>perderá</i> contra lo que está leyendo el pin. |

Tabla 5: Configuración de los GPIOs.

Puesto que solo se utiliza el GPIO1 para escribir, la configuración de mayor interés en este caso es la Push-Pull. El byte de datos se distribuye de forma acorde a la Figura 73.

⁵⁶ En electrónica digital más avanzada, los niveles lógicos pueden ser fuertes si hay poca resistencia (se entrega más corriente), o débiles si hay mucha. Esto da pie a una serie de mecánicas como las que se explican en esta página.

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|-------|--------|---|-------|-------|-------|-------|-------|--|
| SS3.1 | SS3.0 | SS2.1 | SS2.0 | SS1.1 | SS1.0 | SS0.1 | SS0.0 | |
| Bit | Symbol | Description | | | | | | |
| 7 | SS3.1 | SS3[1:0] = 00: quasi-bidirectional | | | | | | |
| 6 | SS3.0 | SS3[1:0] = 01: push-pull SS3[1:0] = 10: input-only (high-impedance) SS3[1:0] = 11: open-drain | | | | | | |
| 5 | SS2.1 | SS2[1:0] = 00: quasi-bidirectional | | | | | | |
| 4 | SS2.0 | SS2[1:0] = 01: push-pull SS2[1:0] = 10: input-only (high-impedance) SS2[1:0] = 11: open-drain | | | | | | |
| 3 | SS1.1 | SS1[1:0] = 00: quasi-bidirectional | | | | | | |
| 2 | SS1.0 | SS1[1:0] = 01: push-pull SS1[1:0] = 10: input-only (high-impedance) SS1[1:0] = 11: open-drain | | | | | | |
| 1 | SS0.1 | SS0[1:0] = 00: quasi-bidirectional | | | | | | |
| 0 | SS0.0 | SS0[1:0] = 01: push-pull SS0[1:0] = 10: input-only (high-impedance) SS0[1:0] = 11: open-drain | | | | | | |

Figura 73: SC18IS602B. Byte de configuración de GPIOs [45].

Por último, es posible seleccionar un modo de bajo consumo, accesible a través del comando 0xF2.

4.5 Circuitería de las antenas

Esta última parte de la PCB es donde las señales se radian al espacio cercano para transmitir la información. Sigue la distribución recomendada por el *datasheet* del SX1257 [44].

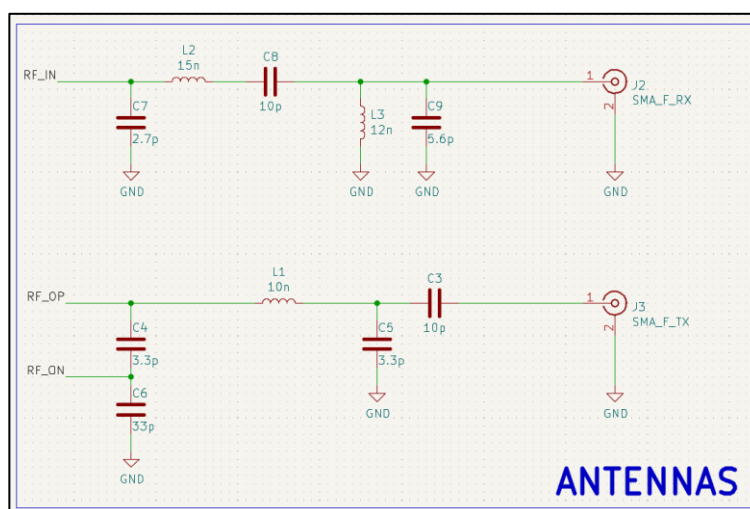


Figura 74: Revisión D. Circuitería de las antenas.

En la etapa de transmisión, la salida diferencial del SX1257 presenta una impedancia característica de aproximadamente 100Ω , mientras que la antena es de 50Ω ⁵⁷ en su frecuencia de resonancia. Al no coincidir

⁵⁷ En transmisiones inalámbricas, se ha estandarizado que las antenas tengan siempre este valor. Es decir, el voltaje entre sus terminales debe ser 50 veces más grande que la corriente que circula. Si todo se diseña de esta manera, idealmente se podría radiar toda la potencia de salida al espacio, aunque diversas pérdidas hacen que esto no sea posible.

estos valores, se pierde buena parte de la potencia de la señal. Además, parte de la onda de radiofrecuencia que llegase al punto de unión se reflejaría, enviando la potencia de nuevo al transceptor, que podría resultar dañado. Es por esto que, antes de transmitir, se deben casar estas impedancias mediante una red de adaptación, práctica conocida como *matching*.

Así pues, las señales diferenciales provenientes del transceptor se suman en una sola, cancelando la gran mayoría de los ruidos (aunque se añade un condensador de desacoplo de 33 pF para acabar de refinar la calidad de la información). A continuación, pasa por un filtro π (llamado así por la forma que tiene), compuesto por dos condensadores de 3.3 pF y una bobina⁵⁸ de 10 nH. Se han escogido estos valores de forma que se cumplan dos funciones: actuar como un filtro paso bajo (con frecuencia de corte en torno a los 1.32 GHz), a la par que se consiguen 50 Ω de impedancia equivalente.

Justo antes de la antena se encuentra un condensador de acoplo de 10 pF, llamado así porque retiene la componente continua, dejando pasar solo la señal de radiofrecuencia.

En la etapa de recepción, el principio es similar. La señal captada por la antena, que se encuentra en el rango de 862–960 MHz, debe acoplarse correctamente a la entrada del LNA integrado en el SX1257. Aunque este amplificador tiene una impedancia de entrada configurable a 50 o 200 Ω , debe utilizarse solo la primera opción, pues para este valor está diseñada la red de adaptación. Los condensadores de 2.7 y 5.6 pF junto con la bobina de 15 nH forman un filtro π , que se complementa con un filtro LC constituido por un condensador de 10 pF y una bobina de 12 nH. En conjunto, se crea un filtro paso bajo muy preciso, que consigue casar las impedancias de entrada y salida correctamente.

4.6 Componentes escogidos y presupuesto

Como se comentó en el capítulo dos, KiCad permite exportar los archivos de fabricación de la PCB de forma muy sencilla. Realmente solo son necesarios los ficheros Gerber, pero el tamaño de los componentes es demasiado pequeño como para soldar cada elemento del circuito manualmente, por lo que se le encomendó también al fabricante. Es por esto que se utiliza también el BOM y el CPL a la hora de realizar el pedido.

El fabricante escogido es JLCPCB, una empresa china asombrosamente económica, a pesar de la larga distancia que debían recorrer las tarjetas. El resultado obtenido fue más que satisfactorio, con una calidad y profesionalidad inigualables.

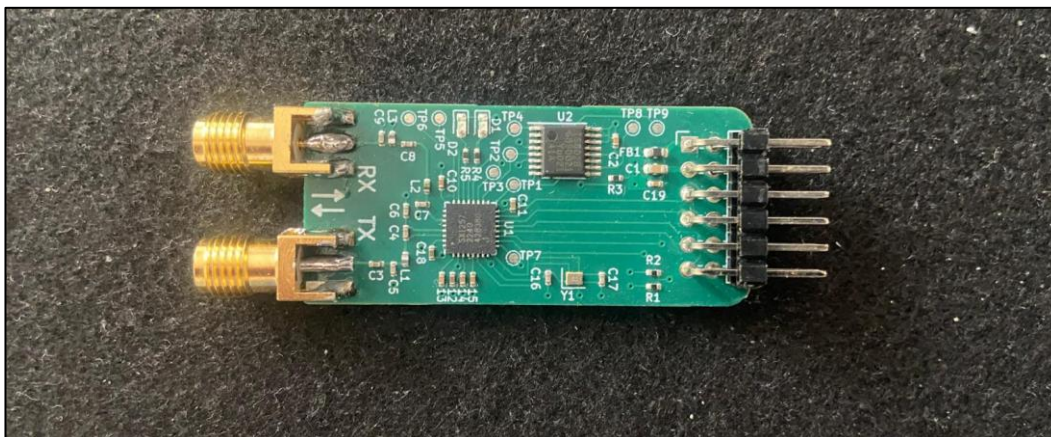


Figura 75: Revisión D del PMOD.

⁵⁸ A diferencia del condensador, que almacena energía en forma de campo eléctrico, la bobina la almacena en forma de campo magnético. A bajas frecuencias se comporta como un cortocircuito, mientras que a partir de su frecuencia de resonancia se comporta como un circuito abierto.

Se fabricaron y ensamblaron un total de cinco PCBs para hacer pruebas y corroborar el funcionamiento, de forma que, si ofrecen la funcionalidad esperada, en el futuro se replique el pedido a mayor escala (lo ideal sería que cada alumno dispusiese de su propia tarjeta). El pedido lo gestionó la tienda de electrónica Mleon, bajo las instrucciones que indicaba el alumno por correo electrónico.

Las PCBs eran de cuatro capas, 20.32 mm de ancho por 45.47 mm de largo y un espesor de 1.6 mm, aunque esta última característica no es acertada, como se explicará más adelante. En futuros pedidos, podría ser necesario especificar un espesor de 0.8 mm. El resto de las características, más avanzadas, se dejaron por defecto.

La hoja de componentes puede observarse en la Tabla 6. En ocasiones es necesario encargar una cantidad mínima de algunos componentes para poder hacer el pedido, aunque luego no se les de utilidad.

| Identificador | Comentario | Referencia | Cantidad | Precio (€) |
|------------------------|--------------|----------------------------|----------|------------|
| C1 | 10 uF | CL10A106KP8NNNC | 5 | 0.026 |
| C2, C10, C14, C15, C19 | 100 nF | 0402X104K100NT | 35 | 0.034 |
| C3, C8 | 10 pF | 0402CG100J500NT | 20 | 0.017 |
| C4, C5 | 3.3 pF | 0402CG3R3C500NT | 20 | 0.017 |
| C6 | 33 pF | 0402CG330J500NT | 5 | 0.0085 |
| C7 | 2.7 pF | 0402CG2R7C500NT | 20 | 0.017 |
| C9 | 5.6 pF | 0402CG5R6C500NT | 20 | 0.017 |
| C11, C12, C18 | 1 uF | CL05A105KP5NNNC | 25 | 0.034 |
| C13 | 10 nF | 0402B103K500NT | 20 | 0.017 |
| C16, C17 | 16 pF | 0402CG160J500NT | 20 | 0.017 |
| D1, D2 | LED | KT-0603R | 10 | 0.051 |
| FB1 | Ferrita | GZ1608D121TF | 20 | 0.12 |
| J1 | Conector 2x6 | 2.54-2*6PWZ | 5 | 0.26 |
| J2, J3 | Conector SMA | EMPCB.SMAFSTJ.B.HT | 10 | 45.19 |
| L1 | 10 nH | MHQ1005P10NJT000 | 20 | 0.43 |
| L2 | 15 nH | LQG15WZ15NJ02D | 9 | 0.29 |
| L3 | 12 nH | MHQ1005P12NJT000 | 20 | 0.34 |
| R1, R2 | 2.2 kΩ | 0402WGF2201TCE | 10 | 0.0085 |
| R3 | 10 kΩ | 0402WJ0103TCE | 20 | 0.0085 |
| R4, R5 | 2 kΩ | 0402WGF2001TCE | 10 | 0.0085 |
| U1 | SX1257 | SX1257IWLTRT | 5 | 77.23 |
| U2 | SC18IS602B | SC18IS602BIPW/S8HP | 5 | 10.96 |
| Y1 | 36MHz 10ppm | ABM12W-36.0000MHZ-6-D1X-T3 | 5 | 27.12 |

Tabla 6: Hoja de componentes.

Como puede observarse, la mayoría del coste que suponen los componentes lo asumen los dos encapsulados, el cristal de cuarzo y los conectores. La Tabla 7 refleja el total de gastos del proyecto.

| Concepto | Coste (€) |
|-------------------------|---------------|
| Fabricación de las PCBs | 1.71 |
| Componentes | 162.22 |
| Ensamblado | 45.26 |
| Envío | 31.36 |
| Total | 240.55 |

Tabla 7: Hoja de gastos.

A este total, habría que aplicarle un porcentaje de comisión para Mleon y el IVA, pero no se tendrán en cuenta estos gastos en las conclusiones de este trabajo, porque dependen exclusivamente del contexto en el que se haga el pedido, no del proyecto en sí. Por lo tanto, se establece un coste de 48.11 € por cada PCB (incluyendo el envío), significando la opción más económica de entre todas las mostradas en el estado del arte, aunque se reservarán estas reflexiones para más adelante.

Para hacer pruebas con el proyecto, también se han utilizado dos antenas de 50 Ω , con un ancho de banda que abarca el rango de operación (862-960 MHz), dos cables coaxiales con terminaciones SMA⁵⁹ hembra, dos adaptadores (para poder conectar dos terminaciones hembra), un osciloscopio, una placa de pruebas (*proto-board*) y algunos componentes menores como resistencias y condensadores. No obstante, todos estos materiales ya estaban en disposición del alumno o fueron prestados por la universidad, por lo que no han supuesto un gasto extra.

Se cometió una equivocación en el pedido con un componente: los conectores SMA encargados se diseñaron para un espesor de PCB de 0.8 mm, mientras que el espesor especificado en el pedido era de 1.6 mm. En futuros envíos, se debe o bien sustituir el componente⁶⁰, o bien reducir el espesor de la tarjeta a 0.8 mm. En el segundo caso, el precio total de la PCB se incrementa en 4.16 € para un pedido de cinco placas. También es de esperar que, en un encargo más grande, el precio unitario quede muy mermado.

⁵⁹ El conector de la antena.

⁶⁰ En ese caso, una buena opción sería CON-SMA-EDGE de RF Solutions, diseñados para 1.6 mm de espesor de PCB [46].

5 PRUEBAS CON EL PMOD

*El hombre que mueve montañas empieza apartando
piedras pequeñas.*

- Confucio -

Este capítulo hace un seguimiento de toda la preparación previa y la experimentación realizada con la tarjeta, para aprender a manejarla y corroborar su funcionamiento. Se comienza hablando de cuestiones indirectamente relacionadas, como lo es la programación de la iCEBreaker, que darán paso a tareas de mayor importancia como la detección de errores de la revisión B, las modificaciones realizadas a la revisión C y la puesta a punto de la revisión D para su uso como sistema de radiofrecuencia.

5.1 Búsqueda de herramientas de software

El proyecto partió con la entrega, por parte de D. Hipólito, de un ejemplar de la revisión B, un par de antenas y dos cables coaxiales con terminaciones hembra. El ejemplar del PMOD había sido provisto por Konrad Beckmann algunos años atrás, sin solicitar ninguna clase de remuneración. No obstante, no se llegó a experimentar con él, por lo que el primer paso era corroborar la falta de prestaciones de la revisión B.

Para configurar el PMOD y compartir información a través de sus puertos de entrada y salida, debe utilizarse una FPGA. En un principio, se intentó trabajar con la Basys3 Artix-7 de Xilinx, pues contaba con un entorno de desarrollo muy intuitivo con el que el alumno ya se familiarizó en la asignatura de Sistemas Electrónicos. No obstante, se alejaba del principio de *aprendizaje de por vida*. Finalmente, se descartó esta opción, y se optó por una tarjeta basada en herramientas libres: la iCEBreaker. Además, el departamento de Ingeniería Electrónica ya dispone de varias decenas de réplicas, por lo que podría facilitarse una tarjeta por alumno.

El alumno también había trabajado con la iCEBreaker en Sistemas Electrónicos para Automatización y conocía la secuencia *síntesis-enrutado-bitstream*. El PMOD, se configuraba a través del protocolo I²C, por lo que se implementó también la NEORV32, de forma que pudiesen programarse con lógica secuencial todas las instrucciones necesarias para configurar los registros del SX1257 y el SC18IS602B. Sin embargo, esto trajo consigo una serie de problemas.

En Sistemas Electrónicos para Automatización, se había trabajado con una versión de la NEORV32 que tenía varios años de antigüedad, por lo que no estaba actualizada con las últimas mejoras y parches de errores. Tal era este atraso, que el módulo de I²C no funcionaba correctamente, sin que esto se indicase claramente en ningún mensaje de error ni foro del repositorio. Esto supuso un gran esfuerzo en términos de tiempo, tratando de

encontrar un error en los códigos que el alumno no había cometido.

Finalmente, se dedujo la causa de esta traba, y se actualizó el procesador a la última versión, lo que trajo consigo nuevos retos, pues ahora eran las versiones de Yosys y GHDL, además de otros, las que estaban anticuadas. Se actualizaron todas las herramientas a su última versión (GHDL 6.0.0-dev, Yosys 0.54+37...) y esta vez sí se pudo trabajar con normalidad. Puede consultarse el repositorio del proyecto para más detalles [26].

Aunque más adelante surgirían más complicaciones, la NEORV32 tenía la ventaja de estar muy bien documentada y contar con varias plantillas muy fáciles de modificar. Se escribieron las líneas de código pertenecientes al *BoardTop* (*neorv32_iCEBreaker_BoardTop_MinimalBoot.vhd*) de la FPGA, es decir, el módulo VHDL con el nivel de jerarquía más alto. Es aquí donde se declaran todos los submódulos o componentes que se van a utilizar y se relacionan entre ellos mediante señales. También cuenta con la plantilla de configuración del PLL y el oscilador de alta frecuencia (hasta 48 MHz) de la FPGA, no obstante, por ahora bastará con el reloj de 12 MHz del cristal de cuarzo.

A partir de este fichero, se instancian una serie de submódulos que continuarán añadiendo funcionalidades a la FPGA. En esta primera versión, tan solo se declara el procesador (*neorv32_ProcessorTop_MinimalBoot*), pues esta entidad ya llama al núcleo de la CPU (*neorv32_top.vhd*), que a su vez instancia todas las prestaciones mínimas para garantizar el funcionamiento de la NEORV32. Tan solo es necesario habilitar los periféricos que van a utilizarse: UART, TWI y GPIO (este último de alguna manera se utiliza en la lógica interna y da error si no se declara).

Las señales que se adentran o salen de un módulo a otro también quedan especificadas en estos archivos. A la hora de asignar las señales del protocolo TWI, es necesario recordar que, cuando se quiere permitir que el esclavo escriba en la línea, la salida debe dejarse en alta impedancia (Z).

```
-- TWI
-- Como salida
twi_sda_io <= '0' when (twi_sda_o = '0') else 'Z';
twi_scl_io <= '0' when (twi_scl_o = '0') else 'Z';

-- Como entrada
twi_sda_i <= twi_sda_io;
twi_scl_i <= twi_scl_io;
```

Figura 76: Declaración de las señales del TWI.

En cuanto a las señales de entrada y salida del *BoardTop*, deben relacionarse con los pines del PMOD por medio de un fichero de restricciones (*constraints.pcf*), donde también se define el botón de *reset*, la señal de reloj y los puertos UART.

Los códigos desarrollados para el *BoardTop*, el *ProcessorTop* y las restricciones pueden visualizarse en el repositorio del proyecto [26], así como las modificaciones de los *makefiles* pertinentes para sintetizarlos correctamente. Debe clonarse el repositorio de la NEORV32 [25] y colocar los ficheros mencionados en su carpeta correspondiente. Todos los pasos de programación de la tarjeta quedan también expuestos en [26].

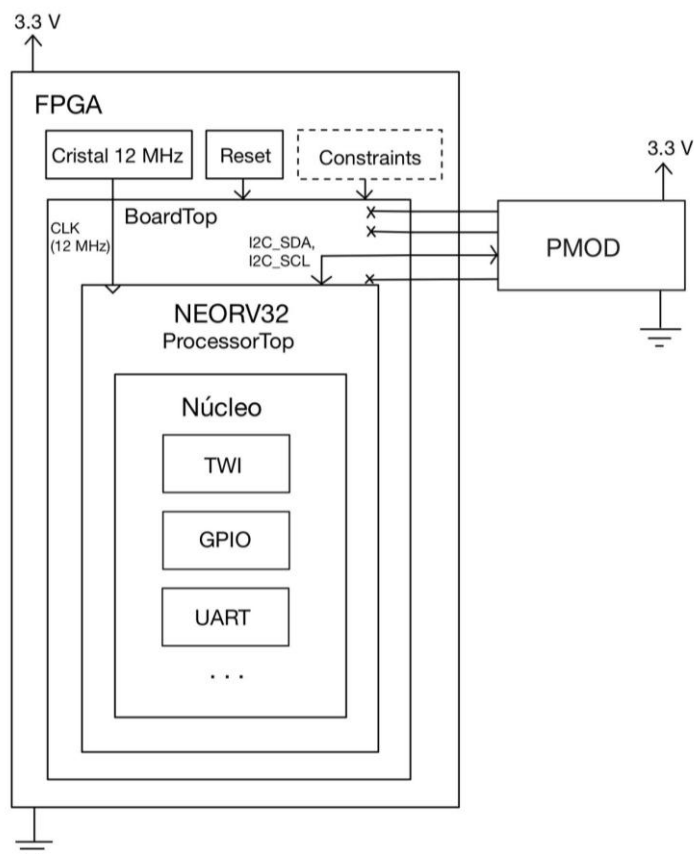


Figura 77: Esquema de códigos y señales I.

Como puede observarse, de momento solo se enrutan las señales pertenecientes al I²C por razones que se comentarán a continuación. Nótese que el fichero de constraints se ha incluido en el esquema para una mayor claridad, aunque este no es sintetizable (no se implementa en el hardware).

5.2 Pruebas con la revisión B

La primera tarea que debía realizarse era corroborar la falta de prestaciones de la revisión B. Tras leer detenidamente los *datasheets* de los dos chips, se llegó a la conclusión de que en primer lugar, se debía configurar el PMOD a medida con la función que iba a realizar, porque no llevar el control de los registros podría invalidar las pruebas que se hiciesen (no estar en el modo de operación correcto...) o incluso poner en peligro la tarjeta. Por ejemplo, La impedancia del LNA del receptor está a 200 Ω por defecto, mientras que la red de adaptación es de 50 Ω . Parte de la señal rebotaría y se perdería la mayor parte de la potencia. El *datasheet*, aunque era una referencia muy útil, tampoco podía tomarse al pie de la letra toda la información que incluía, pues se habían localizado numerosas erratas durante su lectura.

El código para programar los dos encapsulados, desarrollado en C, dado que lo va a ejecutar la NEORV32, es demasiado extenso como para incluirse aquí, pero puede visualizarse en el repositorio del proyecto [26] junto a los *makefiles* modificados y las instrucciones de compilación (carpeta *I2CAutomatico*).

El SC18IS602B queda configurado de la siguiente forma:

| FunctionID | Explicación | Byte | Explicación |
|------------|--|---------------|---|
| 0x01 | Indica R/W en el esclavo 0. | No procede | - |
| 0xF0 | Indica que se quiere configurar el Puente. | 0x00 | Se envía primero el MSB, flanco de subida en SPI y frecuencia del reloj de SPI de 1843 KHz. |
| 0xF1 | Limpia la interrupción que se genera tras una transmisión SPI. | No procede | - |
| 0xF2 | Inicia el modo de bajo consumo. | No procede | - |
| 0xF4 | Indica que los pines GPIO seleccionados van a escribir. | 0x00 | Todos los GPIO escriben. |
| 0xF5 | Indica que los pines GPIO seleccionados van a leer. | No se utiliza | - |
| 0xF6 | Decide si los pines son SS o GPIO. | 0x0E | SS3, SS2 y SS1 pasan a ser GPIO. |
| 0xF7 | Decide la configuración de los GPIO. | 0x55 | Todos los GPIO pasan a estar en modo Push-Pull. |

Tabla 8: Configuración del SC18IS602B.

No obstante, para hacer pruebas con el osciloscopio, se redujo la frecuencia del reloj SPI a 58 kHz. De esta forma, el aparato captaba las señales correctamente. Una vez configurado el SC18IS602B, se procedió a modificar los registros del SX1257. En la Figura 78, se muestra como se sobrescribe el registro 0x00 (por defecto a 0x00) a 0x0F. Si se observa detenidamente la señal del osciloscopio, se apreciará como se envían los bytes siguiendo la secuencia explicada en el apartado 4.4. La señal amarilla corresponde al SCL y la verde al SDA.

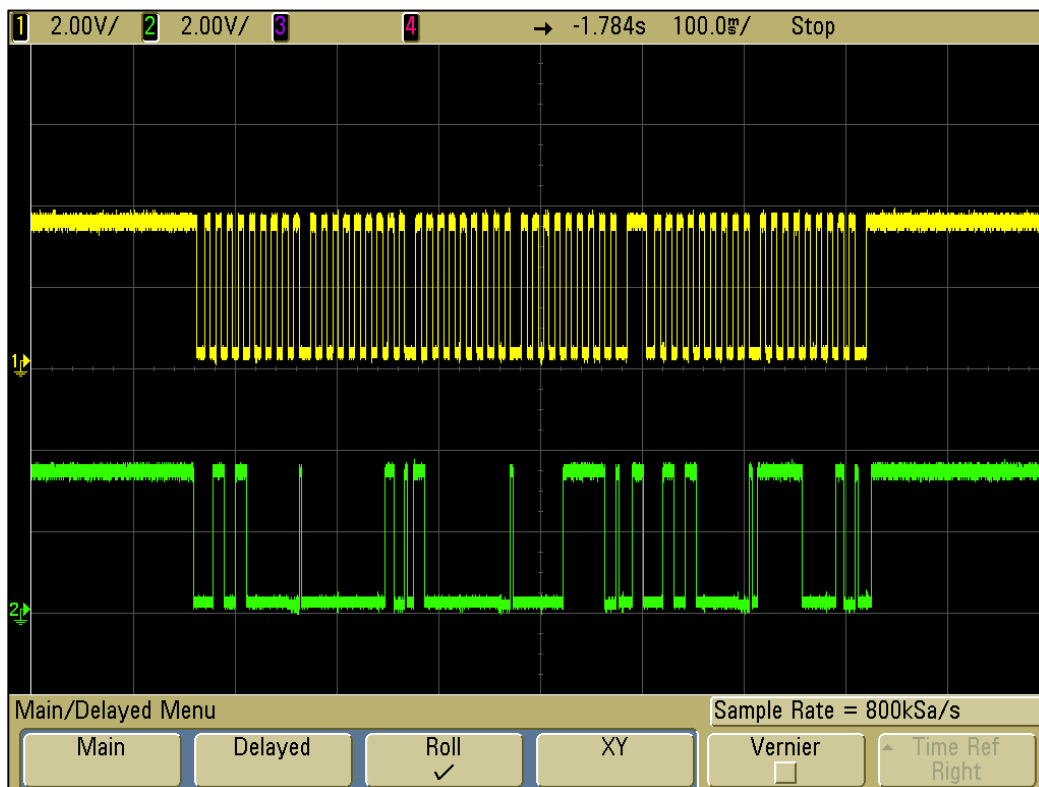


Figura 78: Sobreescribiendo un registro del SX1257.

Del mismo modo, se sobreescriben el resto de los registros, de forma que se conoce con exactitud como se comportará el chip. Pese a que la mayoría tiene el valor correcto para el trabajo por defecto, no supuso ningún esfuerzo asegurarse de la veracidad del *datasheet*, que cuenta con una decena de errores e incongruencias, al sobreescribirlos todos mediante un bucle en el código. Algunos registros son de solo lectura (marcados con 'r'). Para más información, véase las figuras de la sección 4.3.

| Registro | Explicación | Valor | Explicación |
|----------|---|-------|---|
| 0x00 | Modos de operación. | 0x0F | Se activa el PDS, el oscilador, la electrónica del transmisor (y su amplificador) y del receptor. |
| 0x01 | RX: Frecuencia de la portadora (MSB, Most Significant Bits). | 0xC0 | 868 MHz en la recepción. |
| 0x02 | RX: Frecuencia de la portadora (Bits del medio). | 0xE3 | |
| 0x03 | RX: Frecuencia de la portadora (LSB, Least Significant Bits). | 0x8F | |

| | | | |
|------|--|------|---|
| 0x04 | TX: Frecuencia de la portadora (MSB). | 0xC0 | 868 MHz en la transmisión. |
| 0xF5 | TX: Frecuencia de la portadora (Bits del medio). | 0xE3 | |
| 0x06 | TX: Frecuencia de la portadora (LSB). | 0x8F | |
| 0x07 | Versión (r). | 0x21 | Solo lectura. |
| 0x08 | TX: Ganancias del DAC y el mezclador (atenúan la señal). | 0x2E | DAC: -3 dB. Mezclador: -10 dB. |
| 0x0A | TX: Ancho de banda del PLL y el filtro analógico. | 0x76 | PLL: 300 kHz. Filtro: 451 kHz (banda ancha). |
| 0x0B | TX: Ancho de banda del DAC. | 0x00 | 24 coeficientes, que equivalen a más de 450 kHz. |
| 0x0C | RX: Ganancia del LNA y los amplificadores de banda base. Impedancia del LNA. | 0x3E | LNA: -0 dB (Ganancia máxima). Amplificadores: 6 dB. Impedancia: 50 Ω. |
| 0x0D | RX: Ancho de banda del ADC y filtro analógico. Selector del reloj del ADC (32 ó 36 MHz). | 0xF5 | ADC: >400 kHz. Filtro: 500 kHz. Reloj: 36 MHz. |
| 0x0E | RX: Ancho de banda del PLL y modo de medición de temperatura. | 0x06 | PLL: 300 kHz. Modo de medición de temperatura desactivado. |
| 0x0F | Mapeado de DIOs. | 0x00 | Por defecto. Ver Figura 66. |
| 0x10 | Habilitación de <i>loop-backs</i> y CLK_OUT. Selector del reloj del DAC (32 ó 36 MHz). | 0x02 | Reloj: 36 MHz. Se habilita CLK_OUT. |
| 0x11 | Testigos de baja batería y fase fijada en los PLLs (RX y TX) (r). | 0x00 | Solo lectura. |
| 0x1A | Umbral de baja batería. | 0x02 | Umbral: 2,724 V. |

Tabla 9: Configuración del SX1257.

Puesto que se va a transmitir a muy corta distancia, conviene una banda ancha. No obstante, el rendimiento del chip decae a partir de los 500 kHz, por lo que ha tratado de mantenerse en torno a los 450 kHz. Por otra parte, la frecuencia de transmisión debe estar contenida en la banda ISM europea, que abarca de los 863 a los 870 MHz. Se han escogido los 868 MHz. Operar fuera de la banda transmitida puede traer consigo implicaciones legales, pues se podría estar provocando interferencias en otras transmisiones (policía, hospitales, servicios privados...).

Una vez que los dos encapsulados estaban correctamente configurados según la funcionalidad que se le iba a dar, podía comenzar a realizarse toda clase de pruebas. Sin embargo, no fue necesario hacer muchas comprobaciones:

- En primer lugar, los LEDs de la PCB nunca llegaban a emitir luz, aunque se cambiase a los modos de transmisión y recepción. Estos estaban conectados a los DIOs 0 y 1, lo que significaba que los PLLs nunca conseguían enganchar la fase.
- Por otra parte, la señal *xosc_ready*, proveniente del DIO 2, siempre estaba a nivel bajo. Esto quería decir que no se estaba obteniendo una oscilación estable en el oscilador local. En consecuencia, CLK_OUT también estaba siempre deshabilitado. Tan solo en una ocasión se consiguió visualizar en el osciloscopio estas dos señales activadas.

La falta de estabilidad se atribuyó en un principio al cristal de cuarzo, suponiéndose que estaba estropeado. Más tarde, se descubrió la verdadera causa: la revisión B tenía graves erratas en el *layout*.

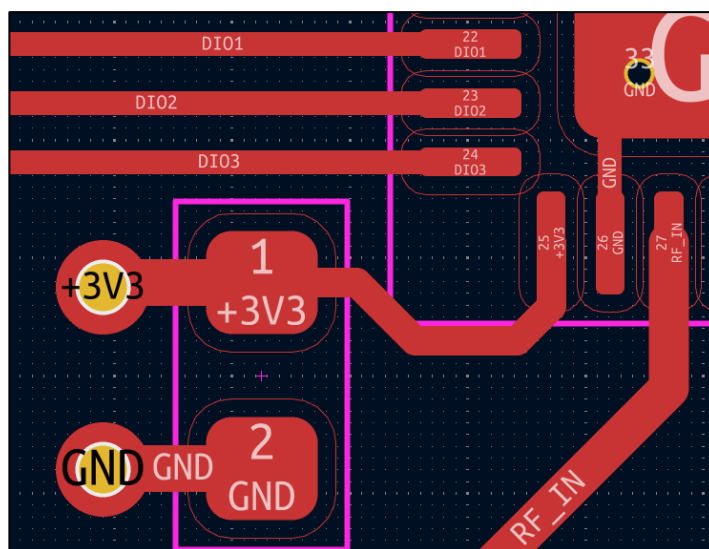


Figura 79: Errores de la revisión B [16].

Konrad Beckmann había conectado a la alimentación los puertos VR_ANA1, VR_ANA2 (que es el *pad* 25 que se observa en la Figura 79), VR_DIG y VR_PA por error, cuando solo se esperaban condensadores de desacoplo para cortocircuitar los ruidos a tierra. La inestabilidad de la que se hablaba la causaban con bastante seguridad estos fallos de conexionado.

Además, la distribución de la placa no era la óptima, pues las pistas que unían *pads* y componentes eran a veces relativamente largas, aumentando la sensibilidad de las señales al ruido.

Tras encontrar estas erratas, se corroboró la baja fidelidad de la revisión B y se descartó su uso. No se probó a transmitir con ella, pues cuando Konrad lo intentó en su día no se obtuvieron resultados satisfactorios y no había ninguna razón por la que esta premisa fuese a cambiar esta vez. En su lugar, comenzó a estudiarse la versión más actualizada de la tarjeta, la revisión C.

5.3 Análisis y modificaciones de la revisión C

En esta versión, Konrad Beckmann llegó a las mismas conclusiones que el alumno y había redistribuido toda la tarjeta, solucionando los errores comentados en la sección anterior. De esta manera, se solucionaban los problemas de inestabilidad y pérdida de potencia. No obstante, el proyecto quedó pausado de forma indefinida en 2019. Tras intercambiar algunos correos con el creador, se llegó a la conclusión de que habría que verificar todo el material disponible antes de trabajarlo, pues la información podía estar desactualizada. Todo apuntaba a que la tarjeta nunca fue fabricada ni probada, aunque esta afirmación puede ser errónea.

Comenzó entonces una etapa exhaustiva de comprensión de la electrónica del proyecto, para lo que también fue necesario aprender a utilizar el software KiCad, con el que el alumno no estaba familiarizado. La electrónica ya se explicó en el capítulo cuatro, por lo que aquí solo se expondrán las modificaciones.

En primer lugar, esta nueva revisión contaba con un oscilador programable a partir del protocolo I²C: el Si514. En principio, este tercer encapsulado serviría para conseguir otras frecuencias de oscilación diferentes a 36 MHz.

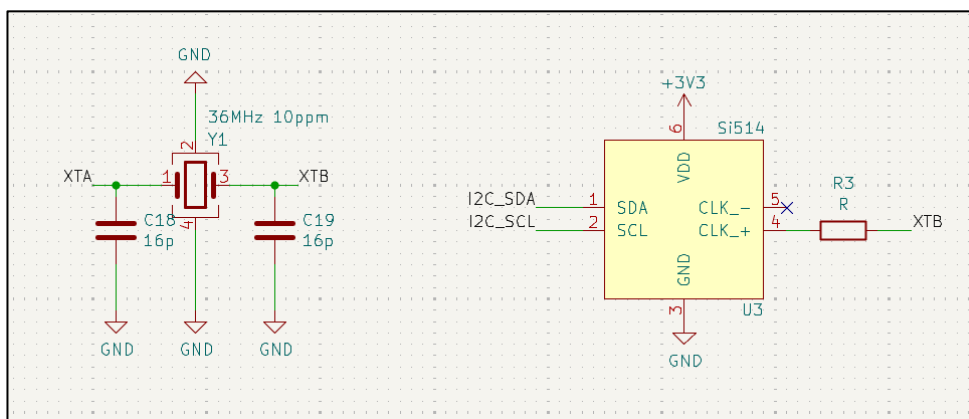


Figura 80: Si514 [16].

No obstante, se eliminó este componente por varias razones:

- Se consideró innecesario poder elegir la frecuencia de oscilación, pues el DAC y el ADC trabajan obligatoriamente con un reloj de 32 o 36 MHz. Sí es cierto daría mayores libertades para elegir entre ambas, pero siempre se puede usar la segunda opción a través de CLK_IN.
- Incluir otro encapsulado aumentaría el precio de la tarjeta.
- La distribución de la Figura 80 no tenía sentido, si se va a utilizar el Si514 debe eliminarse el cristal de cuarzo. Además, la resistencia sin valor asignado daba a pensar que el diseño estaba a medio hacer.

Por otra parte, Konrad Beckmann había seguido de forma casi idéntica el ejemplo proporcionado por el *datasheet* del SX1257 para construir las redes de adaptación de la tarjeta. No obstante, la baja calidad de la imagen daba pie a confusión.

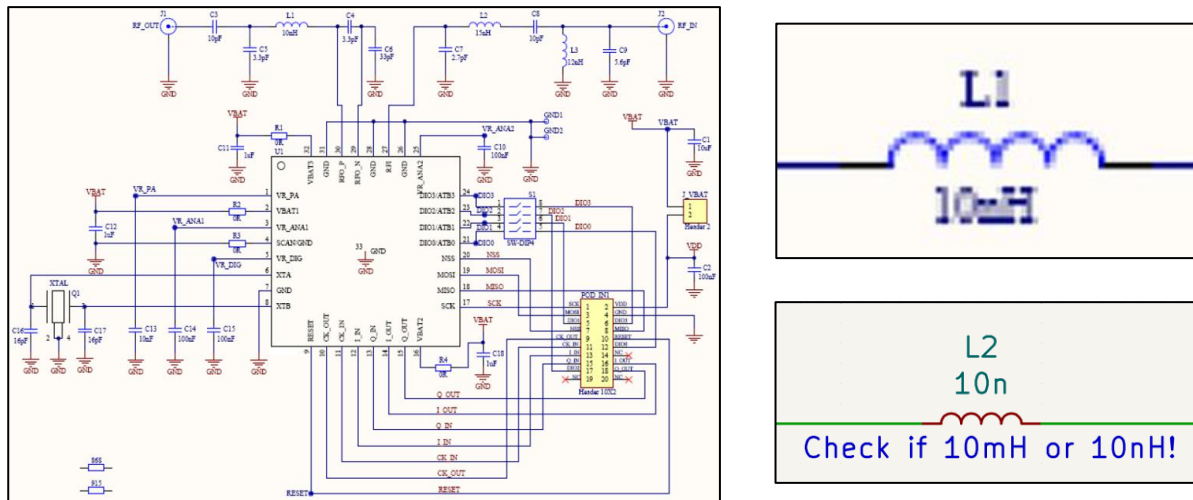


Figura 81: Dudas en el esquemático [44] [16].

Para comprobar el valor de esta bobina, se utilizó Micro-Cap 12, simulando el circuito del transmisor. Si se usaban 10 mH, la simulación posicionaba la frecuencia de corte en torno a 600 kHz, mientras que 10 nH la ubicaba en 1 GHz. Se demostró entonces que este último era el valor adecuado para nuestra aplicación.

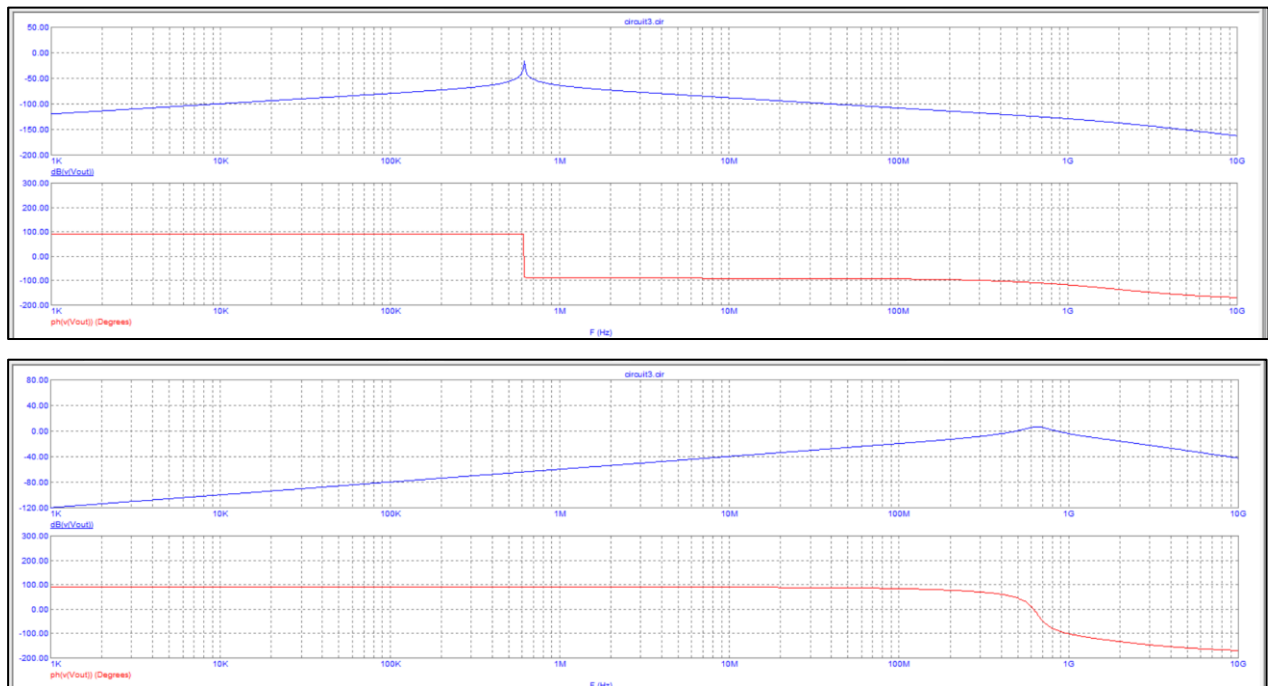


Figura 82: Simulación en Micro-Cap 12. Arriba: 10 mH. Abajo: 10 nH.

Por supuesto, esta falta de calidad de imagen provocó desconfianza hacia el resto de los componentes, por lo que se estudió uno a uno si el valor asignado tenía sentido. No obstante, no se encontró ningún fallo.

En cuanto al *layout*, en general, trató de mantenerse en la medida de lo posible el diseño original. Se efectuaron algunas reubicaciones en componentes y serigrafía para que la PCB diese una imagen más ordenada y se comprobó que la huella de todos los elementos era la adecuada. Puesto que el fabricante escogido fue JLCPCB, la tarjeta debía cumplir sus reglas de diseño. Debieron modificarse una serie de pormenores, resumidos en

diámetros de taladro, separación entre pistas y anchura de las mismas.

La realización del pedido tampoco fue sencilla, dado que en radiofrecuencia la elección de componentes no es para nada trivial. En general, se optó por buena calidad de funcionamiento y se comprobó en repetidas ocasiones que los componentes soportaban 3,3 V de tensión y eran válidos en el rango de frecuencias 862-960 MHz. En el capítulo cuatro, puede visualizarse la referencia exacta para cada componente.

Como ya se ha comentado, se realizó un primer pedido incompleto (sin que el alumno tuviese conocimiento de ello), retrasando la entrega del trabajo un mes. El segundo pedido, que fue gestionado en su totalidad por el alumno, sí que se efectuó de manera correcta. Se dio el nombre de revisión D a esta nueva trajeta.

5.4 Puesta a punto de la revisión D

La revisión D tenía un problema: los conectores de las antenas pedidos estaban diseñados para un espesor de PCB de 0,8 mm, mientras que la tarjeta fabricada era de 1,6 mm. JLCPCB pegó con cinta adhesiva estos componentes a la placa, consiguiendo con esta estrategia que pasara la aduana sin problema. Después, el alumno los soldó manualmente solo por dos de sus patas, dejando las otras dos al aire. Esta estrategia permitió continuar con el trabajo sin derrochar el dinero invertido y sin volver a perder tiempo en esperar a recibir nuevos componentes. Sin embargo, en próximos pedidos este error debe corregirse, como se especifica en el capítulo cuatro.

Puesto que ya se iban a realizar transmisiones, era muy importante tener siempre las antenas conectadas para evitar la reflexión de la potencia ya comentada (pues no conectar la antena equivaldría a una impedancia de infinitos ohmios). No obstante, las antenas de las que se disponía no podían colocarse a la vez (eran demasiado anchas). Aunque se valoró hacerse con un par de diámetro más pequeño, se acabó por utilizar cables que permitiesen alejar una antena de la otra. Por supuesto estos cables también atenuaban la señal. El analizador de espectros reveló que las pérdidas estaban entre los 2 y 3 dB en los 900 MHz, por lo que la potencia se reduciría a la mitad. No obstante, el objetivo de este TFG consistía en corroborar que el funcionamiento de la tarjeta era el adecuado, lo que podía comprobarse sin conseguir la máxima transferencia de potencia. Esta tarea se reserva para trabajos futuros.

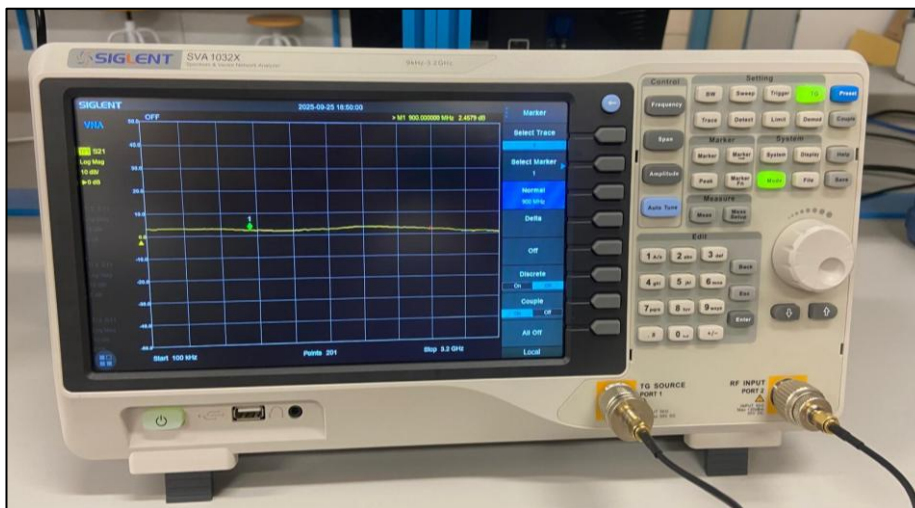


Figura 83: Analizador de espectros.

Por otra parte, alejar las antenas era una práctica muy beneficiosa, pues su diagrama de radiación es de tal forma (se crea un espacio en medio en el que no se radia nada), que se podría perder información si el transmisor y el receptor se encuentran a muy poca distancia.

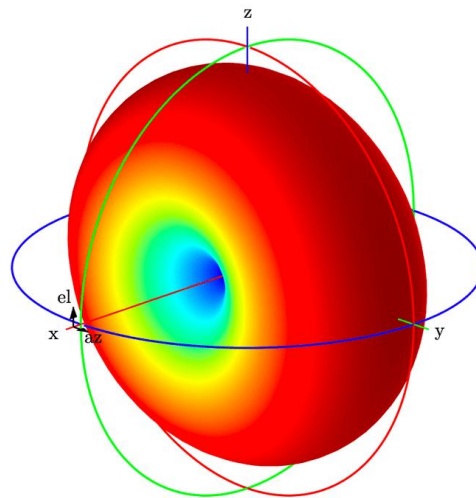


Figura 84: Diagrama de radiación de las antenas [47].

Se disponía ahora de todos los elementos para empezar a trabajar, aunque fuese solo en local (es decir, se transmite y recibe desde la misma PCB). Se programaron los registros del SX1257 y se obtuvo un resultado satisfactorio:

- La señal *xosc_ready* se puso a nivel alto.
- Se consiguió ver por el osciloscopio la señal de reloj de 36 MHz proveniente de CLK_OUT.
- Los LEDs que indicaban el enganche de fase de los PLLs se encendieron.

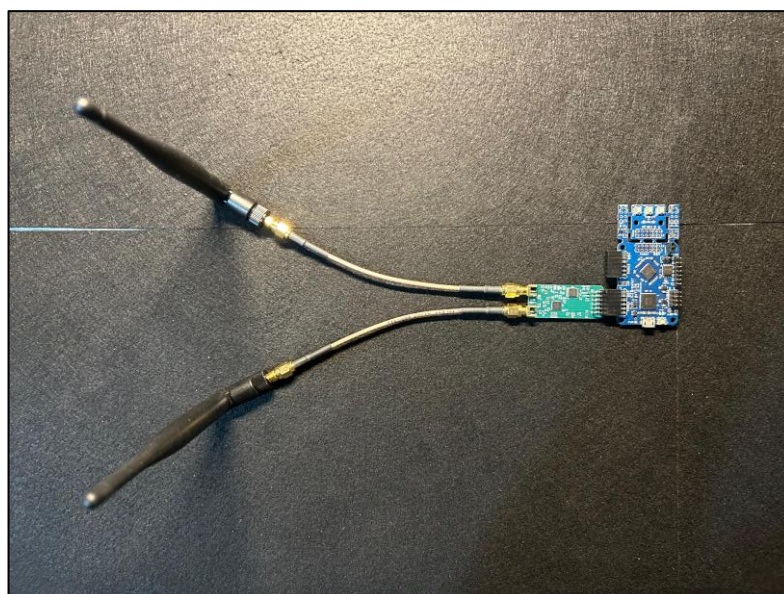


Figura 85: Sistema completo.

Todo apuntaba a que la PCB funcionaría, por lo que podía empezar a prepararse una prueba de concepto.

6 DEMOSTRACIÓN PRÁCTICA CON EL PMOD

La teoría te lleva del punto A al punto B. La experimentación te lleva a todas partes.

- Albert Einstein -

Ahora que se han explicado las motivaciones del proyecto, las herramientas para trabajarlo, los fundamentos teóricos que lo envuelven, la electrónica diseñada y las pruebas realizadas con ella, en este capítulo se unen todos estos conceptos para realizar una primera demostración del funcionamiento de la tarjeta. Se desarrollarán los códigos necesarios para ello y se incluirán los resultados obtenidos para su análisis.

6.1 Preparaciones previas

Preparar la prueba de concepto tampoco era tarea sencilla. En primer lugar, debía construirse un modulador Sigma-Delta, pues así estaba diseñado el SX1257: tanto I_IN como Q_IN eran señales digitales, pensadas para transportar las componentes en fase y en cuadratura de la señal de información en forma de 1s y 0s alternando a altas frecuencias. Por suerte, el *datasheet* daba indicaciones muy precisas de como construirlo:

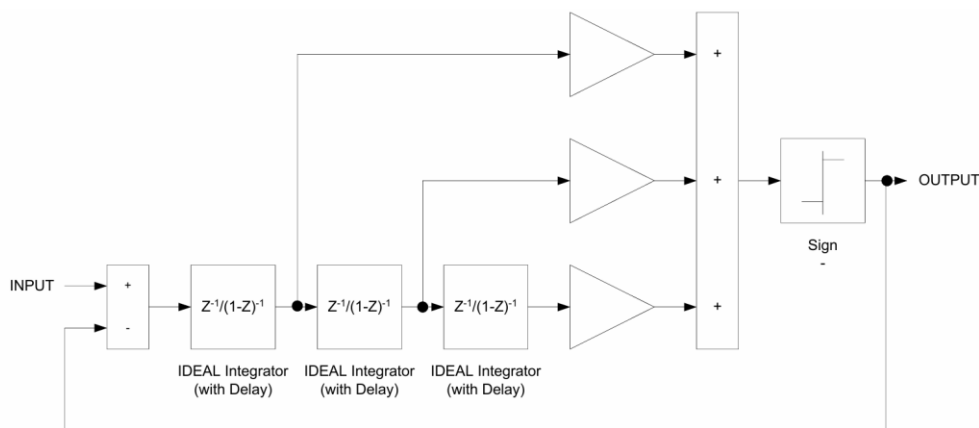


Figura 86: SX1257. Propuesta del modulador Sigma-Delta [44].

La figura muestra un modulador de tercer orden (tres integradores), es decir, los ruidos en la frecuencia de la banda base son prácticamente nulos, pues hay una fuerte corrección a la tendencia del error. Las salidas de los integradores deben hacerse pasar por una ganancia de lazo, para depurar el funcionamiento (los triángulos de la Figura 86). Los valores escogidos fueron:

$$G_1 = 0.25$$

$$G_2 = 0.25$$

$$G_3 = 0.5$$

De esta forma, se evita que el modulador sature a la par que se le da más peso al tercer integrador, que es el que regula la tendencia lenta. Tras sumar las salidas de los tres integradores, un comparador enviará un 1 si este valor es mayor que 0, o un 0 si no lo es. El resultado se realimenta para calcular el error y el bucle se repite. Además, tal y como lo especificaba el *datasheet*, se ha sido muy cauteloso con los problemas de desbordamiento⁶¹, añadiendo saturación en todas las etapas.

Se diseñaron los bloques para el integrador (*Integrator.vhd*) y el comparador (*Comparator.vhd*) y se sintetizaron en el modulador Sigma-Delta (*SigmaDeltaModulator.vhd*) [26]. Para sincronizar el modulador con el SX1257, se utilizó la señal de reloj proveniente de CLK_OUT. Como especificaba el *datasheet*, el modulador Sigma-Delta enviaría los datos en el flanco de subida del reloj, mientras que el DAC interno del encapsulado los leería en el flanco de bajada.

Si se quiere usar la segunda opción de muestreo (32 M de muestras por segundo), se podía utilizar el PLL de la FPGA para conseguir dicho reloj, enviarlo por CLK_IN y cargarlo también en el modulador. No obstante, esta opción es menos favorable, ya que no pueden coexistir a la vez el reloj de 12 MHz del cristal de cuarzo y el de 32 MHz del PLL, por lo que sería necesario eliminar el primero. Dado que se daban dos opciones, se optó por trabajar siempre a 36 MHz, evitando esta complicación.

El proyecto se enfrentó entonces a un nuevo problema: la NEORV32 se sintetizaba de tal forma, que solo soportaba una frecuencia de reloj máxima de 25 MHz, por lo que no podría estar sincronizada con el transceptor (que recibe y envía datos a 36 MHz). Esto causaba nuevas molestias, pues ahora debían implementarse colas FIFO entre el modulador Sigma-Delta y el procesador, que recibiesen los datos a una frecuencia de reloj y los enviasen a otra, cuando la prueba de concepto debía consistir en algo poco complejo que permitiese hacer comprobaciones rápidamente. Finalmente, se decidió que las señales I y Q nunca entrarían en la NEORV32, la cual seguiría trabajando a 12 MHz.

Aunque existía la opción de pasar la información al ordenador por puerto serie y tratar los datos con herramientas más avanzadas, esto implicaría diseñar más bloques de código, de nuevo alejándose de la simplicidad original de la prueba. Al final, se abandonó definitivamente la idea de procesar los datos, que como ya se ha mencionado en otras ocasiones, se reservó para continuaciones del proyecto.

Ahora que la señal no se generaba en la NEORV32, se escribieron nuevos módulos que actuaran como generadores de señales. En concreto, se diseñaron señales triangulares (*TriangleWave.vhd*), piramidales con tres niveles (*Pyramid3Wave.vhd*) y en forma de sierra (*SawWave.vhd*) [26]. Todas periódicas cada 1 ms (1kHz). También estaban coordinadas con CLK_OUT.

Al recibir los datos, se trató de diseñar un “demodulador” Sigma-Delta, pero se descartó la idea, pues la señal recuperada no podría visualizarse de ninguna manera (la FPGA no tiene ningún pin analógico). Se optó por una solución mucho más sencilla: pasar I_OUT y Q_OUT por un filtro paso bajo, que eliminase el ruido de alta frecuencia provocado por la modulación Sigma-Delta y permitiese visualizar el resultado en el osciloscopio. Para diseñar el filtro solo fueron necesarios una *proto-board*, un condensador de 100 nF y un juego de resistencias, pues según la señal sería necesario ajustar la frecuencia de corte.

De esta forma la prueba de concepto estaba lista. Aunque no se modulase en QPSK, podía suponerse que las componentes en fase y en cuadratura resultantes de la modulación fuesen, por ejemplo, una señal triangular y

⁶¹ En VHDL, el tipo *signed* establece el MSB como el bit de signo, siendo 0 positivo y 1 negativo. Por ejemplo, si a 01111111 (+127) se le suma 1, pasaría a 10000000 (-128). Estos cambios bruscos inestabilizarían completamente el sistema, por lo que lo correcto en este caso sería saturar en 01111111.

una señal en forma de sierra⁶². Si en la recepción se visualizaban estas mismas señales, significa que al demodular en QPSK se habría recuperado la información correctamente. Como se mencionó en el capítulo cinco, la comunicación debe establecerse en la banda ISM europea (863-870 MHz), por lo que se transmitirá siempre a 868 MHz.

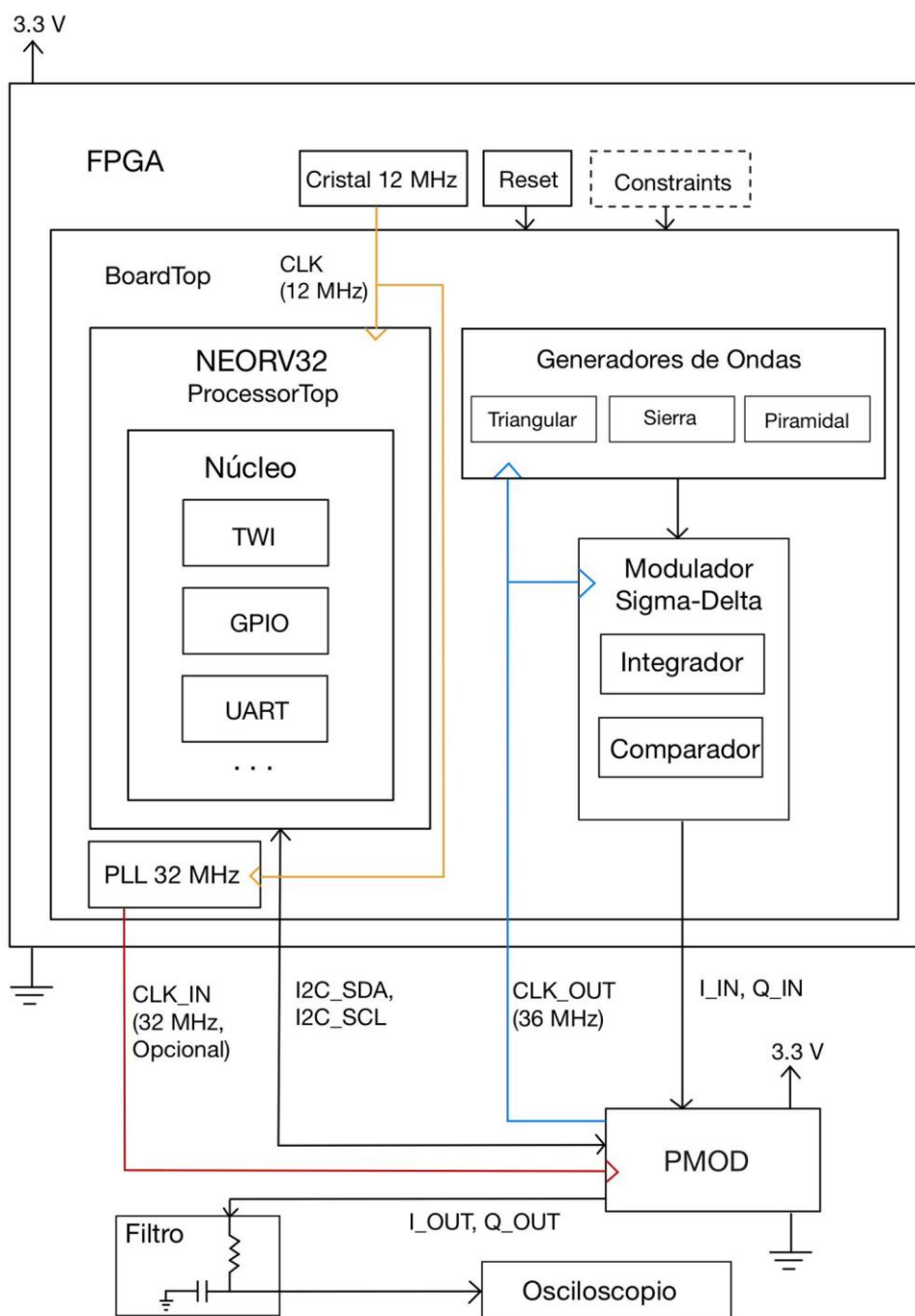


Figura 87: Esquema de códigos y señales II.

⁶² Como se comentará más adelante, esta suposición era errónea.

6.2 Demostración en montaje local

Procedió a montarse el sistema según la distribución explicada en la sección anterior. Para esta primera prueba, se pretendía transmitir y recibir desde la misma PCB.

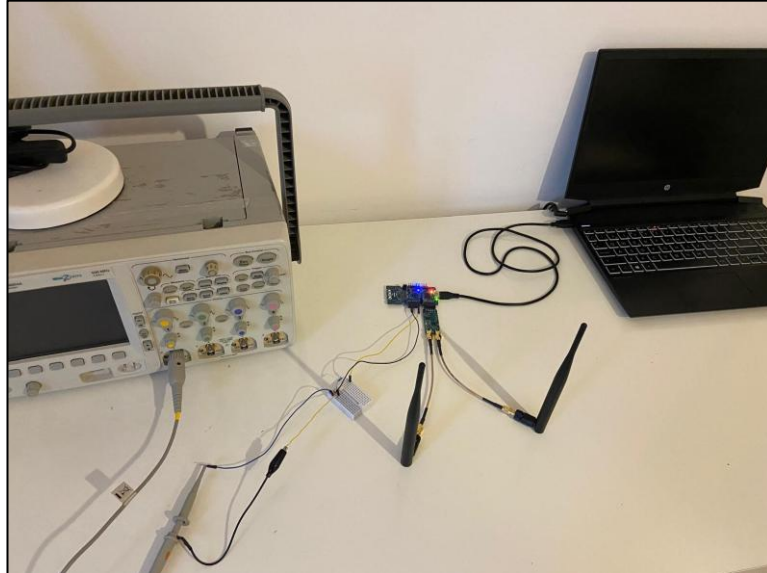
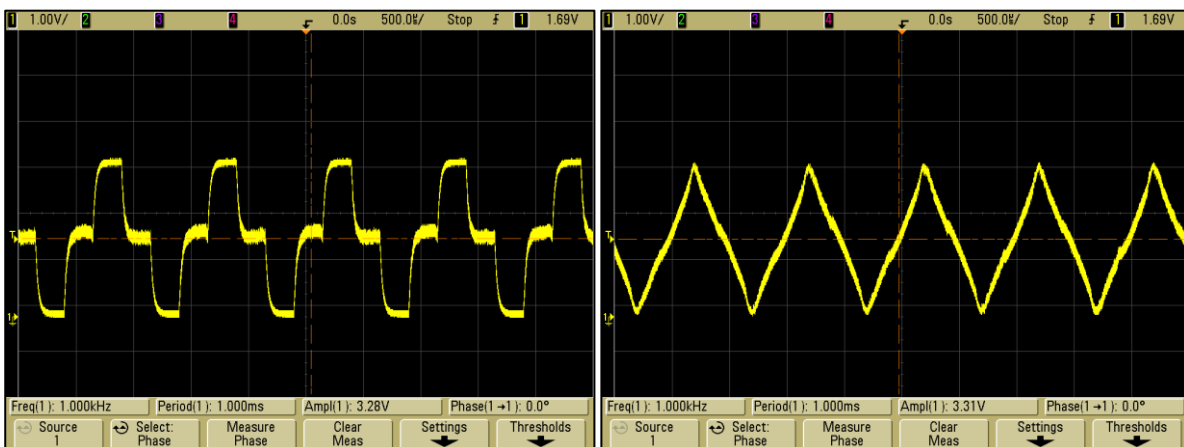


Figura 88: Montaje local.

Se utilizó una resistencia de $220\ \Omega$ para el filtro, que daba una frecuencia de corte de $7.23\ \text{kHz}$, la suficiente para dejar pasar todas las componentes de la banda útil de las señales a la par que se eliminaban los ruidos de alta frecuencia. El resultado fue satisfactorio, pues se consiguieron visualizar correctamente los tres tipos de señales. Además, si se modificaba la frecuencia de oscilación, este cambio también se apreciaba en el osciloscopio.



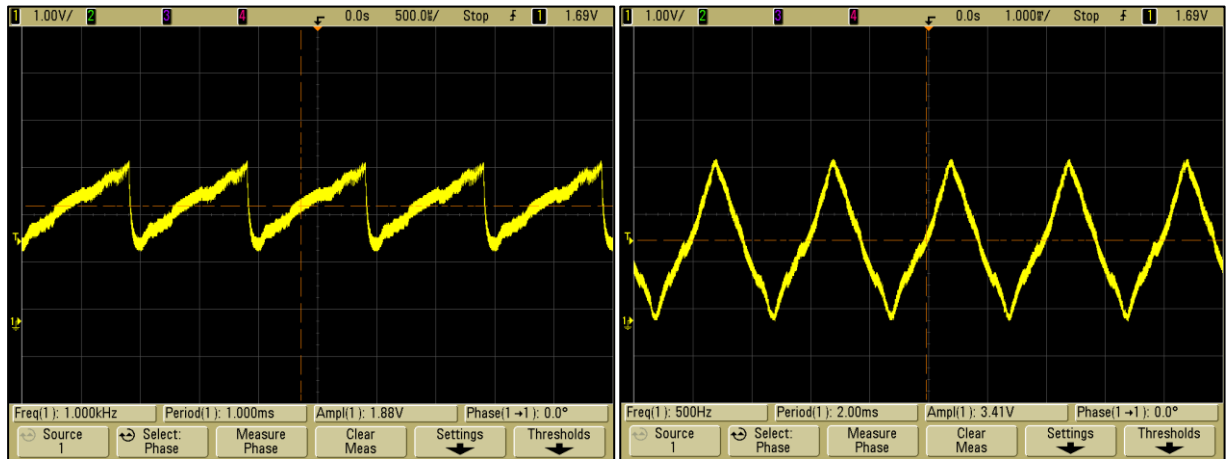


Figura 89: Demostración en montaje local.

De izquierda a derecha y de arriba abajo: señal piramidal, señal triangular (1 kHz), señal en forma de sierra y señal triangular (500 Hz).

No obstante, se descubrió la principal limitación del montaje local al desenroscar la antena de recepción: a pesar de que ya no había antena, las formas de onda se seguían visualizando.

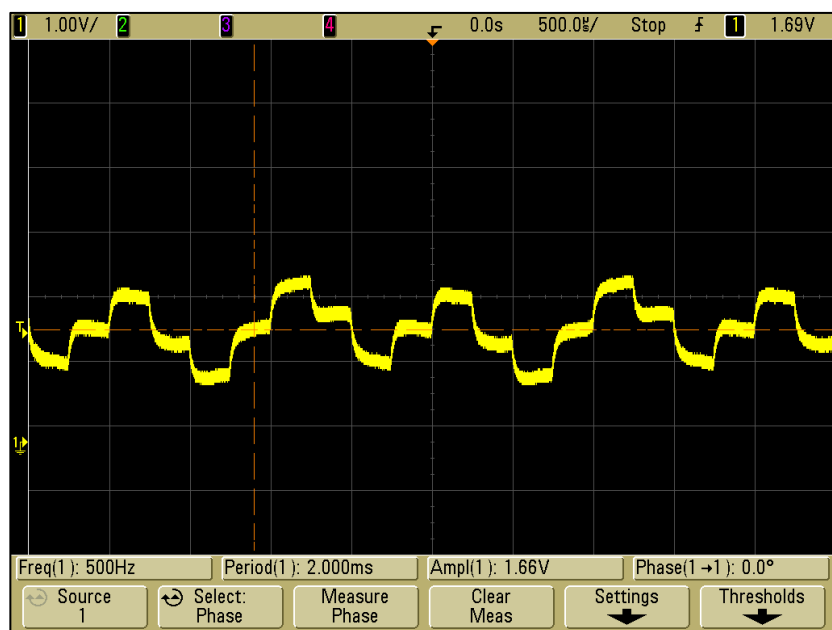


Figura 90: Señal piramidal atenuada y distorsionada, recibándose aún sin haber antena de recepción.

Tras comprobar que efectivamente los *loop-backs* estaban desactivados, se llegó a la conclusión de que este fenómeno debía causarlo el acoplamiento eléctrico: dos pistas paralelas actúan como las placas de un condensador. Si la pista de transmisión varía rápidamente su voltaje (como ocurre en las transmisiones RF), podría inducirse corriente en la pista del receptor a través de la capacitancia parásita entre ellas. Dado que entre las pistas del transmisor y el receptor de la PCB apenas hay unos milímetros de separación, esta es con certeza la causa de esta recepción involuntaria.

Por lo tanto, el montaje local no era el más adecuado para realizar pruebas, pues no se podía saber con seguridad si la señal se recibiría correctamente si no se produjese el acoplamiento. Era necesario transmitir de una PCB a otra para dar por concluida la prueba de concepto.

6.3 Demostración en montaje completo

Se procedió entonces a realizar un segundo experimento, esta vez diferenciando a la tarjeta transmisora de la receptora y separando ambas antenas medio metro. El sistema operó en primera instancia en *half-duplex* (no se transmitía y recibía a la vez).

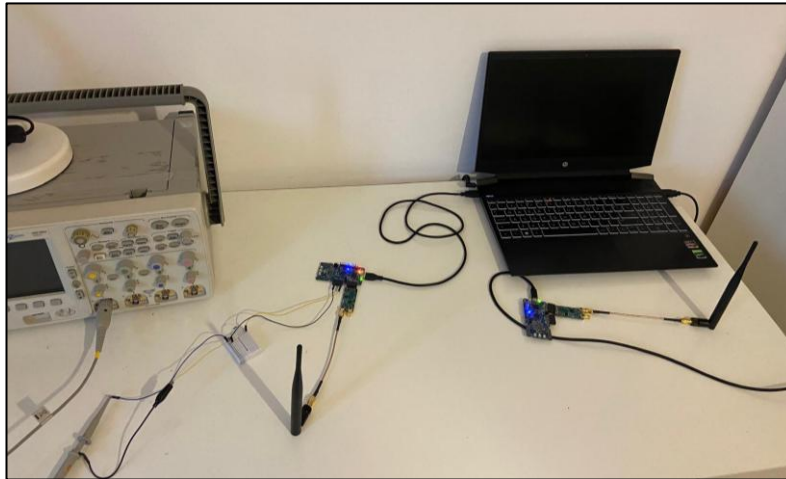


Figura 91: Montaje completo. *Half-duplex*.

Esta vez, no se obtuvieron los resultados esperados. En la Figura 92, se aprecia como las dos señales se solapan en una sola.

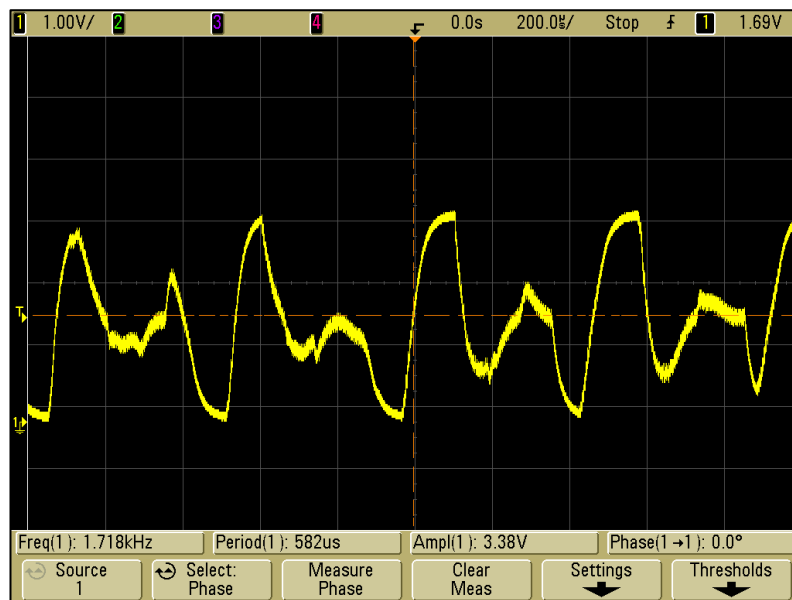


Figura 92: Señal triangular solapándose con una señal piramidal.

Tras comprobar que no estaba habiendo acoplamiento eléctrico de ninguna manera, el alumno comprendió el error que se estaba cometiendo: si se enviaba por I una señal triangular y por Q una señal piramidal, al no ser estas ondas ortogonales, era inevitable el acoplamiento. Al recibirse, no se podrían separar correctamente. Esto contrastaba con la prueba anterior, donde sí que se había logrado esta hazaña. Se dedujo que debía ser un efecto secundario del acoplamiento eléctrico.

Se probó a mandar una sola señal por I y nada por Q y, aunque la teoría decía que funcionaría, no fue así en la realidad.

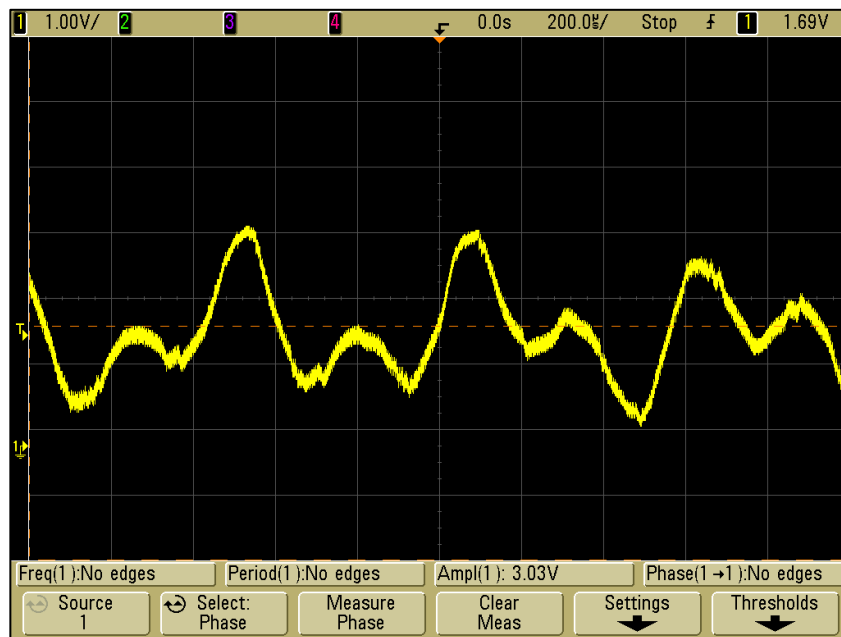


Figura 93: Transmisión de señal triangular distorsionada.

La única manera de conseguirlo era que las señales enviadas por I y Q fueran realmente ortogonales. Se optó por diseñar nuevos generadores de onda para un seno (*SinWave.vhd*) y un coseno (*CosWave.vhd*) [26], dos señales ortogonales por naturaleza. Dado que la frecuencia de oscilación era de 500 Hz, se utilizó una resistencia de 2 k Ω en el filtro, obteniendo una frecuencia de corte de 800 Hz. El resultado fue esta vez satisfactorio.

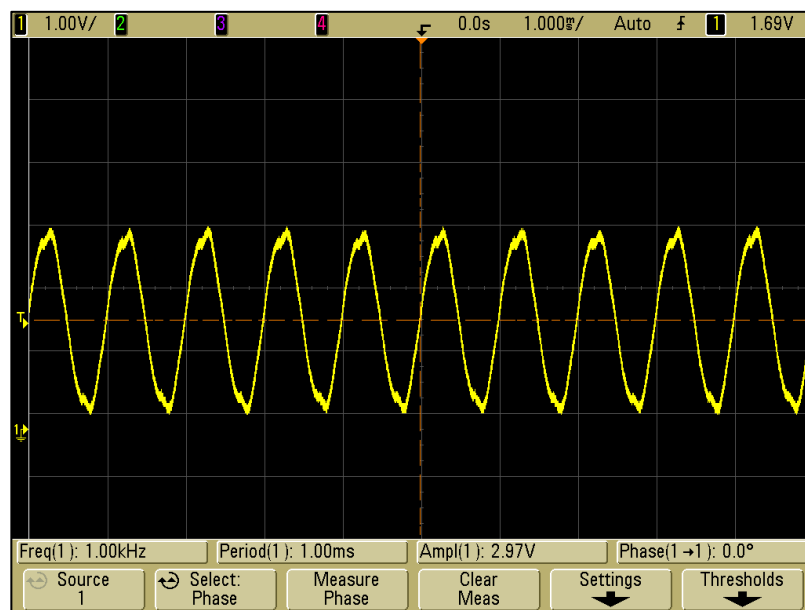


Figura 94: Demostración en montaje completo. *Half-duplex*.

El seno se generaba a partir de un número finito de valores, por lo que al no ser un tono puro sufría de pequeñas distorsiones. No obstante, es de esperar que con un seno totalmente analógico esto no ocurra. Además, al

desconectar la antena de recepción esta vez, no se visualizaba nada por el osciloscopio, por lo que se estaba evitando el acoplamiento eléctrico.

Por último, se realizó un tercer experimento donde el sistema operaría en *full-duplex*. No se disponía de más antenas, pero podía conseguirse este funcionamiento si una de las transmisiones era guiada, en lugar de inalámbrica.

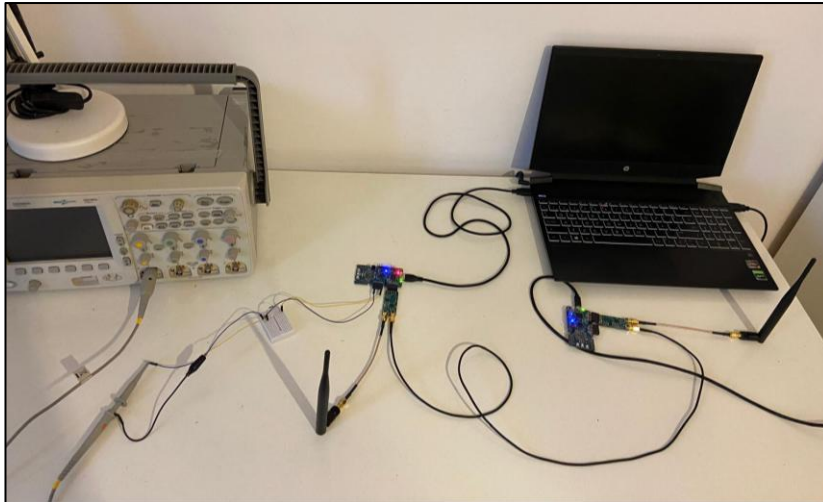


Figura 95: Montaje completo. *Full-duplex*.

Para comprobar ambas recepciones, bastaba con intercambiar el filtro y la sonda del osciloscopio a la otra tarjeta. Además, para evitar el acoplamiento eléctrico, se transmitió esta vez a 863 y 870 MHz. De nuevo, el resultado fue satisfactorio.

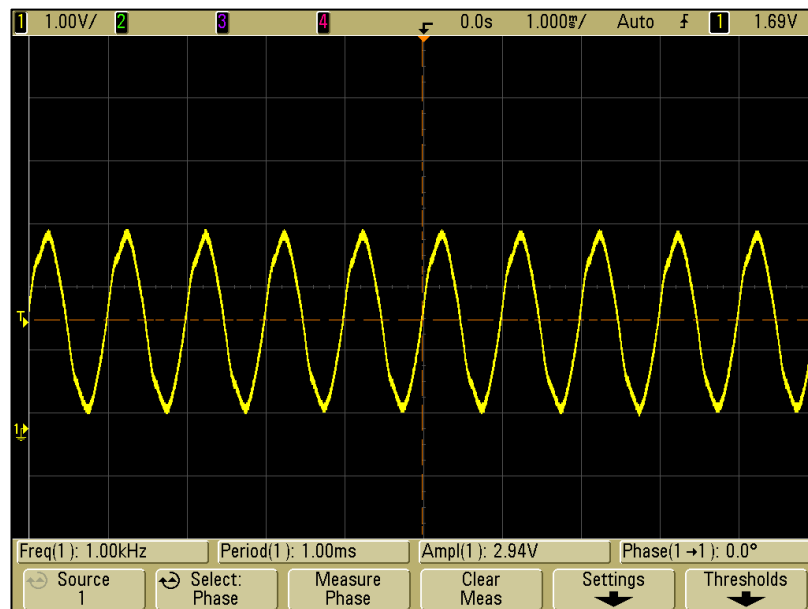


Figura 96: Demostración en montaje completo. *Full-duplex*.

Con esto, quedaba demostrado el correcto funcionamiento de la revisión D y podía plantearse su implementación en Electrónica Digital para Comunicaciones.

7 CONCLUSIONES Y TRABAJO FUTURO

No es el final, ni siquiera es el principio del final. Pero es, quizás, el final del principio.

- Winston Churchill -

Para dar por finalizado el TFG, se harán una serie de reflexiones de cara a la implementación del trabajo realizado en Electrónica Digital para Comunicaciones y la calidad de la solución propuesta, así como comentarios del alumno a nivel más personal.

7.1 Conclusiones

Como se demostró en el capítulo seis, con un coste unitario de tan solo 48.11 € (incluyendo los gastos de envío), la revisión D ofrece en una sola tarjeta toda la funcionalidad necesaria para la experimentación con señales de radiofrecuencia en entorno real. Quedan cubiertas, por lo tanto, las expectativas de Electrónica Digital para Comunicaciones, de forma que en un futuro pueda ofrecer a sus alumnos una enseñanza de calidad.

El rendimiento de la tarjeta es más que suficiente para el uso que se le dará en la asignatura y aún así supone la solución más económica de todo el estado del arte. Por otro lado, todas las herramientas para trabajarla son de software libre, por lo que el alumno puede replicar los conocimientos obtenidos fuera del ámbito académico con muy poco presupuesto, cosa que no ocurría con el resto de las tarjetas en el mercado.

Si bien queda mucho trabajo por hacer, como se comentará en la siguiente sección, esta tarjeta es la puerta hacia una infinidad de ideas y proyectos. En las manos adecuadas, la PCB puede llegar a:

- ✓ Transmitir modulaciones complejas, como OFDM.
- ✓ Integrarse en una red inalámbrica de sensores (los cuales se añadirían a la iCEBreaker).
- ✓ Intercambiar mensajes de texto de una tarjeta a otra, como si de un móvil se tratara.

La versatilidad del aparato, su bajo coste y su programación con el uso exclusivo de software libre de licencias, hacen que la solución propuesta sea cuanto menos elegante.

El siguiente párrafo deja por escrito los pensamientos del alumno acerca del trabajo realizado:

A pesar de todo el tiempo que he dedicado y la inmensa cantidad de veces que me he frustrado tratando de hacer que las cosas funcionen, este TFG ha sido una experiencia inolvidable. En el proceso, he aprendido muchísimas cosas. He afianzado mis conocimientos de electrónica y he descubierto el mundo de las telecomunicaciones, del cual no sabía absolutamente nada y me ha dejado fascinado. He aprendido a diseñar PCBs con herramientas libres, e incluso a gestionar el pedido para fabricarlas. Me he quebrado la cabeza tratando de dar respuesta a cientos de problemas, leyendo documentos, probando soluciones diferentes una y otra vez... Hasta que funcionó. En ese momento, toda la fatiga acumulada se desvaneció y el orgullo me llenó por dentro, pues por primera vez, sentí que era un ingeniero de verdad.

Y no solo es que funcionase, si no que el contexto que envuelve al proyecto es sencillamente genial. No puedo esperar a dentro de unos años, cuando vea esta pequeña plaquita que tantos problemas me ha causado en las manos de cientos de alumnos transmitiendo y recibiendo señales de un lado para otro.

Por supuesto, todo esto ha sido posible gracias al mejor tutor que podría haber escogido, D. Hipólito Guzmán Miranda. Él sabe muy de cerca la complejidad del trabajo, pues me ha respondido a decenas de correos, de lunes a domingo, a veces entrada la madrugada. Siempre con respuestas muy precisas de varios párrafos, que conseguían que saliese del atasco en el que estaba (para acabar metiéndome en otro). Me prestó muchísimas cosas sin las cuales no habría llegado hasta aquí, como un libro y un osciloscopio que mejoraba con creces a todos los de la escuela. Llegué a su despacho pidiéndole algo desafiante, y él me dio justo lo que buscaba: un proyecto digno de un TFG. Gracias de corazón.

7.2 Trabajo futuro

Si bien el trabajo realizado ha sido muy completo, hay muchos temas que han quedado pendientes y sería interesante probar antes de hacer un pedido a gran escala.

En primer lugar, no se llegó a completar el séptimo objetivo, en el que se finalizaba el sistema de telecomunicación digital implementando también los bloques para codificar, modular, demodular y decodificar las señales. Sería interesante probar varios tipos de modulaciones, comenzando por la QPSK, verificando que el sistema sigue actuando de la forma esperada. El procesamiento de las señales puede hacerse en la NEORV32, para lo que es necesario diseñar el sistema de colas FIFO mencionado en el capítulo seis, que permita leer y enviar desde el procesador las componentes en fase y cuadratura a otra frecuencia de reloj (o si se prefiere, puede transmitirse al ordenador por el puerto serie para tratarse con herramientas más potentes). Además, dado que ahora las señales recibidas pasarían por un procesador en lugar del osciloscopio, también debe construirse el “demodulador” Sigma-Delta.

Por otra parte, no se ha hecho ningún estudio acerca de las pérdidas de potencia que se producen durante la transmisión de las señales ni de otros ratios, como la relación SNR (Signal to Noise Ratio). Aunque todo apunta a que no son significativos, sería conveniente comprobar si estos parámetros están dentro de los límites admisibles de la asignatura.

Tras toda esta experimentación previa con la revisión D, podría efectuarse el pedido a gran escala para proveer a todo el alumnado de esta PCB. Debe tenerse en cuenta el error de diseño que se comentó en el capítulo cuatro y ponerle remedio, bien disminuyendo el espesor de la PCB o bien intercambiando el componente erróneo.

Una vez que llegasen las tarjetas y se corroborase su correcto funcionamiento, comenzaría una nueva etapa, en la que se estudiaría como integrarlas adecuadamente en la enseñanza impartida durante las clases de Electrónica Digital para Comunicaciones.

REFERENCIAS

- [1] National Geographic España, *Alexander Graham Bell y la polémica del teléfono*, [En línea]. Disponible en:
https://historia.nationalgeographic.com.es/a/alexander-graham-bell-y-polemica-telefono_15118
- [2] National Geographic España, *Marconi y la primera transmisión por radio*, [En línea]. Disponible en:
https://historia.nationalgeographic.com.es/a/marconi-y-primer-transmision-por-radio-historia_14204
- [3] Universidad de Sevilla, *Máster Universitario en Ingeniería de Telecomunicación*. [En línea]. Disponible en:
<https://www.us.es/estudiar/que-estudiar/oferta-de-masteres/master-universitario-en-ingenieria-de-telecomunicacion-0#edit-group-plan1>
- [4] Universidad de Sevilla, *Grado en Ingeniería de las Tecnologías de Telecomunicación*. [En línea]. Disponible en:
<https://www.us.es/estudiar/que-estudiar/oferta-de-grados/grado-en-ingenieria-de-las-tecnologias-de-telecomunicacion>
- [5] Guzmán Miranda, H., Programa de la asignatura: Electrónica Digital para Comunicaciones. Concurso de acceso a plazas de cuerpos docentes universitarios. Universidad de Sevilla. Noviembre 2019.
- [6] Universidad de Sevilla, *BOUS Anexo IV: Presupuesto del Departamento de Ingeniería Electrónica (2025)*. [En línea]. Disponible en:
https://www.us.es/sites/default/files/secretaria-general/bous/files/2025_10_5AnexoIIAcuerdo2.7CSUS2025.06.25_1Parte_firmado.pdf
- [7] D. Gisselquist, *SDR: Software Defined Radio Repository*, GitHub, [En línea]. Disponible en:
<https://github.com/ZipCPU/sdr>
- [8] Guzmán Miranda, H., *Planteamientos Docentes e Investigadores y Proyecto de Investigación*. Concurso de acceso a plazas de cuerpos docentes universitarios. Universidad de Sevilla. Noviembre 2019.
- [9] Avnet, *XRF16 Gen3 SoM*. [En línea]. Disponible en:
<https://www.avnet.com/americas/products/avnet-boards/avnet-board-families/xrf-soms/xrf16-gen3-som/>
- [10] Avnet, *Xilinx RFSoc Gen 3 Kit for mmWave*. [En línea]. Disponible en:
<https://www.avnet.com/americas/products/avnet-boards/avnet-board-families/xilinx-rfsoc-gen-3-kit-for-mmwave/>
- [11] Real Digital, *RFSoc 4x2 Hardware Platform*. [En línea]. Disponible en:
<https://www.realdigital.org/hardware/rfsoc-4x2>

- [12] HamGeek, *Zynq7010 AD9363 SDR Development Board*. [En línea]. Disponible en:
<https://www.hgeek.com/es/products/hamgeek-70mhz-6ghz-zynq7010-ad9363-sdr-software-defined-radio-development-board-for-pluto-sdr-matlab>
- [13] HamGeek, *HackRF One R10C V2.0 X with PortaPack H4M SDR Radio*. [En línea]. Disponible en:
<https://www.hgeek.com/es/products/hamgeek-1mhz-to-6ghz-hackrf-one-r10c-v2-0-x-with-portapack-h4m-sdr-radio-software-defined-radio-with-antennas>
- [14] Crowd Supply, *iCEBreaker v1.0e — 3D Rendering*. [En línea]. Disponible en:
https://www.crowdsupply.com/img/1c02/d7fb1c1a-83a8-4fa5-97d8-1b5e76b61c02/icebreaker-v10e-3d-rendering_png_gallery-lg.jpg
- [15] Crowd Supply, *iCEBreaker v1.0b with PMODs*. [En línea]. Disponible en:
https://www.crowdsupply.com/img/9650/c2cb169c-d69b-42ba-a855-b0027e1e9650/icebreaker-v1-0b-with-pmods_jpg_gallery-lg.jpg
- [16] K. Beckmann, *SX1257 PMOD Repository*. GitHub, [En línea]. Disponible en:
<https://github.com/xil-se/SX1257-PMOD>
- [17] Icebreaker FPGA Team, *iCEBreaker FPGA Repository*, GitHub, [En línea]. Disponible en:
<https://github.com/icebreaker-fpga/icebreaker>
- [18] “Dual 4-input AND gates.” *DelDsim*. [En línea]. Disponible en:
<https://deldsim.com/study/material/34/dual-4-input-and-gates>
- [19] *FPGAs Overview*. FPGAKKey, [En línea]. Disponible en:
<https://www.fpgakey.com/wiki/details/51?srsltid>
- [20] “SRAM Cell Inverter Loop.” *Wikipedia*, [En línea]. Disponible en:
https://es.m.wikipedia.org/wiki/Archivo:SRAM_Cell_Inverter_Loop.png
- [21] “Phase-Locked Loop Diagram.” *Wikipedia (UK)*, [En línea]. Disponible en:
https://uk.m.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Phase_locked_loop.svg
- [22] *Electrónica Avanzada*, “Oscilador en anillo.” [Video]. YouTube, [En línea]. Disponible en:
<https://www.youtube.com/watch?v=b56vAut10SQ>
- [23] *The Von Neumann Architecture*. ComputerScience.GCSE.guru, [En línea]. Disponible en:
<https://www.computerscience.gcse.guru/theory/von-neumann-architecture>
- [24] INCIBE, *Microcontroladores y sus principales medidas de protección*. [En línea]. Disponible en:
<https://www.incibe.es/incibe-cert/blog/microcontroladores-y-sus-principales-medidas-de-proteccion>
- [25] S. Nolting, *NEORV32 Open-Source Processor Repository*. GitHub, [En línea]. Disponible en:
<https://github.com/stnolting/neorv32>

- [26] C. Cabrera Maldonado, *PMOD GDR Concepts Repository*. GitLab, [En línea]. Disponible en:
https://gitlab.com/edcmit/pmod_gdr_concepts
- [27] C. Payán, *Lección 1. Introducción*, Documento interno, Universidad de Sevilla, 2025.
- [28] *Analog vs Digital Signals*. Unison Audio, [En línea]. Disponible en:
<https://unison.audio/analog-vs-digital/>
- [29] *Fourier Transforms Visualization*. DIBS Methods Meetings, [En línea]. Disponible en:
<https://dibsmethodsmeetings.github.io/fourier-transforms/>
- [30] V. Ruiz, *Apuntes de Teoría de Señales*, Universidad de Almería, [En línea]. Disponible en:
<https://w3.ual.es/~vruiiz/Docencia/Apuntes/Signals/Theory/index.html>
- [31] “Bandwidth Diagram.” *Wikipedia*, [En línea]. Disponible en:
<https://es.wikipedia.org/wiki/Archivo:Bandwidth.svg>
- [32] “How Do Electromagnetic Waves Wave?” *Physics Stack Exchange*, [En línea]. Disponible en:
<https://physics.stackexchange.com/questions/359696/how-do-electromagnetic-waves-wave>
- [33] *Antena Dipolo*. *Wikipedia*, [En línea]. Disponible en:
https://es.wikipedia.org/wiki/Dipolo_%28antena%29#/media/
- [34] *Amplitude Modulation*. TechnologyUK.net, [En línea]. Disponible en:
<https://www.technologyuk.net/telecommunications/telecom-principles/amplitude-modulation.shtml>
- [35] M. Lathi, *Modern Digital and Analog Communication Systems*, Springer, 2019. [En línea]. Disponible en:
https://link.springer.com/rwe/10.1007/978-1-4614-3447-4_44
- [36] *Diagrama de constelación QPSK*. *ResearchGate*, [En línea]. Disponible en:
https://www.researchgate.net/figure/Diagrama-de-constelacion-QPSK-5_fig1_368874232
- [37] *Modulación en amplitud cuadratura (QAM)*. Flu Project, [En línea]. Disponible en:
<https://www.flu-project.com/2019/11/teleco-in-nutshell-v87-modulacion-amplitud-cuadratura.html>
- [38] MathWorks, *Delta-Sigma Toolbox Blog Post*, [En línea]. Disponible en:
<https://blogs.mathworks.com/pick/2013/05/24/delta-sigma-toolbox/>
- [39] Texas Instruments, *Delta-Sigma ADC Basics: Understanding the Delta-Sigma Modulator*, [En línea]. Disponible en:
https://e2e.ti.com/blogs_/archives/b/precisionhub/posts/delta-sigma-adc-basics-understanding-the-delta-sigma-modulator
- [40] V. H. Adams, *UART Protocol Explanation*, [En línea]. Disponible en:
<https://vanhunteradams.com/Protocols/UART/UART.html>

- [41] V. H. Adams, *SPI Protocol Explanation*, [En línea]. Disponible en:
<https://vanhunteradams.com/Protocols/SPI/SPI.html>
- [42] ProgramarFácil, *Protocolo I²C — Mensaje completo*, [En línea]. Disponible en:
<https://programarfácil.com/wp-content/uploads/2021/01/13-mensaje-completo-protocolo-i2c.png>
- [43] J. A. Gómez, *Comunicación I²C con Arduino*. Programar Fácil, [En línea]. Disponible en:
<https://programarfácil.com/blog/arduino-blog/comunicacion-i2c-con-arduino/>
- [44] Semtech, *SX1257 Low Power Digital I/Q RF Transceiver, Data Sheet v1.2*, Mar. 2018.
- [45] NXP Semiconductors, *SC18IS602B I²C–SPI Bridge, Data Sheet*, 2014.
- [46] RF Solutions, *CON-SMA-EDGE-S SMA Connector*, Mouser Electronics, [En línea]. Disponible en:
<https://www.mouser.es/ProductDetail/RF-Solutions/CON-SMA-EDGE-S>
- [47] *Theoretical radiation pattern of a $\frac{1}{4}\lambda$ 868 MHz dipole antenna*. ResearchGate, [En línea]. Disponible en:
https://www.researchgate.net/figure/Theoretical-radiation-pattern-of-a-quarterwavelength-868-MHz-v-dipole-antenna_fig4_363708439