



POLITÉCNICA

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES
UNIVERSIDAD POLITÉCNICA DE MADRID

José Gutiérrez Abascal, 2. 28006 Madrid
Tel.: 91 336 3060
info.industriales@upm.es

www.industriales.upm.es



Javier Martínez Ciria

TRABAJO FIN DE GRADO

INDUSTRIALES

TRABAJO FIN DE GRADO

Diseño de un módulo de comunicación implantable para enfermedades cerebrales

AGOSTO 2024

Javier Martínez Ciria

DIRECTORES DEL TRABAJO FIN DE GRADO:

Miguel Jiménez Carrizosa

Jorge Portilla Berruco



POLITÉCNICA

*Dedicado a
mis padres*

Agradecimientos

Quiero expresar mi más sincero agradecimiento a todas las personas e instituciones que han hecho posible la realización de este Trabajo de Fin de Grado. En primer lugar, agradezco a mis tutores, Jorge y Miguel, por su orientación y apoyo continuo a lo largo de este proceso, por su paciencia y por compartir su conocimiento y experiencia conmigo. Ha sido un privilegio formar parte de este gran proyecto, y su confianza en mí ha sido una fuente de inspiración y motivación constante.

Asimismo, deseo expresar mi agradecimiento al Instituto de Salud Carlos III, al Hospital de la Princesa, al Hospital Niño Jesús y a la empresa INSYTE por su participación y respaldo en este proyecto. Quisiera destacar especialmente la colaboración concreta de IMDEA Materiales, cuya contribución ha sido fundamental para la realización y éxito de este Trabajo de Fin de Grado.

Por último, quiero agradecer a mi familia y amigos, cuyo apoyo incondicional ha sido una fuente constante de motivación y fortaleza. Su paciencia, comprensión y ánimo han sido fundamentales para superar los desafíos que encontré en este camino.

A todos, muchas gracias.

Resumen

El presente Trabajo Fin de Grado se encuentra comprendido dentro de un proyecto de investigación más amplio, el cual tiene como objetivo el diseño de un dispositivo médico implantable para el tratamiento del glioblastoma multiforme (GBM).

Este proyecto se llama Dispositivo Implantable para el Tratamiento de Tumores cerebrales mediante Campos eléctricos (DITTce). Está financiado por el Ministerio de Ciencia e Innovación, la Agencia Estatal de Investigación y por la Unión Europea gracias a los fondos “Next Generation”. Los participantes de este proyecto son: Instituto de Salud Carlos III, IMDEA Materiales, Hospital de la Princesa, Hospital Niño Jesús, la empresa INSYTE y la UPM. Esta última organización es la coordinadora del proyecto. En este caso, el Trabajo Fin de Grado ha sido propuesto por el Centro de Electrónica Industrial (CEI) de la Escuela Técnica Superior de Ingenieros Industriales (ETSII) de la UPM.

Dentro de este ambicioso proyecto, este Trabajo Fin de Grado encaja dentro del diseño del módulo de comunicación del dispositivo, lo cual le dota de gran transversalidad al TFG, al poderse aplicar a multitud de enfermedades cerebrales. Esto permite su aplicación real, no solo en la colaboración con este proyecto en concreto, sino también en el análisis y monitorización de otras enfermedades distintas a la mencionada, como podría ser la enfermedad de Parkinson u otro tipo de enfermedades neurodegenerativas.

Según la Organización Mundial de la Salud (OMS), las enfermedades neurológicas son una de las principales causas de discapacidad y muerte a nivel mundial. Se estima además que alrededor de mil millones de personas en el mundo, aproximadamente un 15% de la población mundial, sufren este tipo de enfermedades [1]. Respecto a los tumores cerebrales, los estudios indican que afectan al 0,21% de la población en Estados Unidos [2]. Además, se sabe que la tasa de incidencia global de los tumores cerebrales malignos es de aproximadamente 3 a 5 casos por cada 100.000 habitantes [3]. Ver Figura 1.

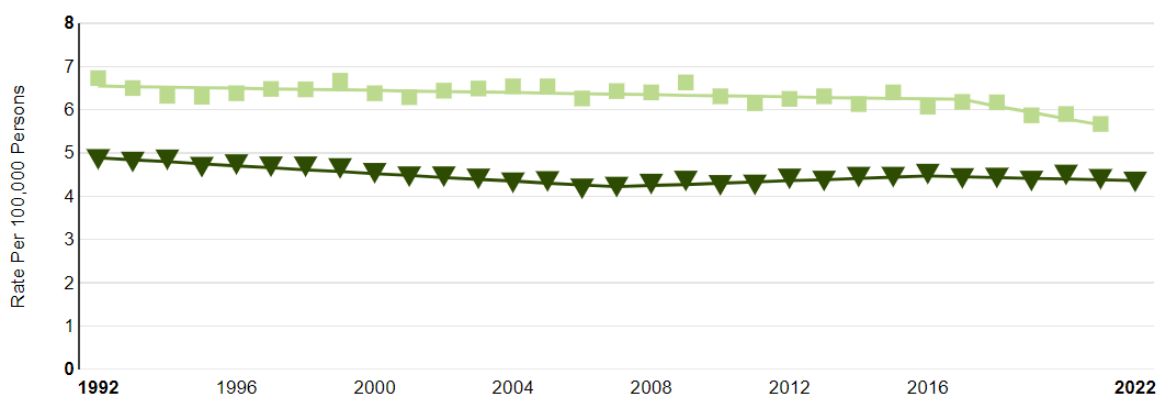


Figura 1 Ratio de tumores cerebrales por cada 100.000 habitantes en Estados Unidos

El proyecto propone un innovador tratamiento para el glioblastoma multiforme, un tipo de cáncer que, aunque de momento no tiene cura, podría beneficiarse de esta nueva tecnología para mejorar notablemente la esperanza de vida de los pacientes, actualmente de alrededor de 2 años. El enfoque se basa en la emisión de campos eléctricos alternos con una intensidad y frecuencia precisas, dirigidas de manera efectiva al tumor para optimizar los resultados.

Este proyecto cuenta con distintas restricciones.

- **Energía.** Debido a la complicada localización del dispositivo implantado, se tratará de evitar las recargas, por lo que una gestión óptima de la batería es primordial.
- **Medio de propagación.** Los distintos medios por los que se propaga la señal de comunicación, la frecuencia elegida debe ser capaz de atravesarlos sin tener grandes pérdidas en el tránsito.
- **Tamaño.** Al ser un dispositivo implantado en el cerebro, éste debe ser compacto para que el impacto sea mínimo en el paciente.
- **Temperatura.** El dispositivo debe mantener siempre una temperatura determinada para no afectar negativamente al paciente.

Como se puede observar, los objetivos de este TFG son muy similares a los objetivos que se tendrían para los estudios de otras enfermedades neurológicas. Al ser un implante en el propio cerebro, se podrá realizar un estudio de las señales propias del cerebro de una mejor manera que con sensores sobre el cráneo. Esto permitirá avances en el conocimiento que se tiene sobre dichas enfermedades y tal vez, en un futuro, poder detectarlas con antelación, tratarlas o prevenirlas.

Palabras Clave: LoRaWAN, Implante cerebral, Glioblastoma Multiforme

Códigos Unesco:

3307.12 Transmisores radio

3311.10 Instrumentos médicos

3325.05 Radiocomunicaciones

Índice

Agradecimientos.....	3
Resumen	5
Códigos Unesco:.....	6
Índice	7
Índice de figuras	10
Índice de tablas	12
Índice de ecuaciones.....	13
1 Objetivos.....	15
2 Introducción	17
2.1 Contexto.....	17
2.2 Metodología.....	17
3 Estado del arte.....	19
3.1 Glioblastoma Multiforme	19
3.2 Tratamiento de tumores con campos eléctricos.....	20
3.3 El cerebro.....	20
3.3.1 Fisiología.....	21
3.3.2 Funciones	22
3.3.3 Ondas de actividad cerebral	25
3.4 Comunicación de dispositivos médicos.....	27
3.5 Métricas	28
3.5.1 RSSI	28
3.5.2 SNR.....	29
4 Análisis y Selección del Protocolo de Comunicación.	31
4.1 Características.....	31
4.1.1 Alcance.....	31
4.1.2 Bajo Consumo	32
4.1.3 Límite de Absorción.....	32
4.1.4 Seguridad.....	33
4.1.5 Tasa de envío.....	33
4.1.6 Transmisibilidad.....	34
4.2 Elección del protocolo	35
4.3 Dash7	35

4.4	LoRaWAN.....	36
4.5	Energía consumida	37
4.5.1	Transmisión de datos requerida.....	37
5	Elección y Configuración del Hardware.....	39
5.1	Elección General del Hardware	39
5.2	Dash7.....	39
5.2.1	Cortex-M4.....	39
5.2.2	Cortex-M3.....	40
5.3	LoRaWAN.....	41
5.3.1	Cortex-M0.....	41
6	Implementación del protocolo de comunicación.	43
6.1	Dash 7.....	43
6.2	LoRaWAN.....	44
7	Experimentos	51
7.1	Planteamiento del problema	51
7.1.1	Herramientas software.....	51
7.1.2	Análisis estadístico.....	55
7.2	Ensayos.....	55
7.2.1	Ensayo en vacío.....	55
7.2.2	Ensayo con cerebro simulado.....	57
7.2.3	Ensayo con cerebro real	58
8	Resultados	61
8.1	Ensayo en vacío	61
8.2	Ensayo en cerebro simulado	63
8.3	Ensayo en cerebro real.....	64
8.3.1	Ensayo a 1 metro de distancia	65
8.3.2	Ensayo a 2 metros de distancia	66
8.3.3	Ensayo a 90°	67
9	Conclusiones y líneas futuras.....	69
9.1	Conclusiones.....	69
9.2	Líneas Futuras.....	70
9.3	Valoración de impactos y de aspectos relacionados con la responsabilidad legal, ética y profesional.....	70
9.3.1	Impactos Legales.....	70
9.3.2	Impactos Éticos	71

9.3.3 Responsabilidad Profesional	71
10 Planificación temporal y presupuesto	73
10.1 Estructura de Descomposición del Proyecto	73
10.2 Diagrama de Gantt.....	74
10.3 Presupuesto del Proyecto.....	77
11 Bibliografía	79
Anexo I: Códigos empleados.....	81
Prueba LoRa en Arduino	81
Código IDE Cube	82
NS1 Main.c	82
AT_Master Main.c	86
LoRa app_master.c	90
LoRa master_app.c	93
Anexo II: Otros cambios realizados en la Implementación.....	101

Índice de figuras

Figura 1 Ratio de tumores cerebrales por cada 100.000 habitantes en Estados Unidos	5
Figura 2: Glioblastoma Multiforme	19
Figura 3: Resonancia Magnética con gadolinio	19
Figura 4: Partes del cerebro.....	21
Figura 5: Absorción de los distintos tejidos del cerebro ante diferentes frecuencias ..	34
Figura 6: Imagen IDE Cube	36
Figura 7: Placa STM32-F427-429VG.....	39
Figura 8: Placa STM-Nucleo-F103RB.....	40
Figura 9: Mapeo PlacaF103RB con Arduino	41
Figura 10: Placas de Desarrollo P-NUCLEO-LRWAN3.....	42
Figura 11: Ejemplo del uso de comandos HAL	43
Figura 12: Consola LoRaWAN STMMicroelectronics por defecto	45
Figura 13: Web Lorient. Selección de la Región	46
Figura 14: Consola LoRaWAN. Cambio de parámetros.....	46
Figura 15: Permitir uso compartido de la Red	47
Figura 16: Interfaz Lorient con el dispositivo Gateway conectado	47
Figura 17: Consola LoRaWAN STMMicroelectronics con parámetros modificados.....	47
Figura 18: Selección carpeta en IDE Cube	48
Figura 19: Consola LoRaWAN Node	48
Figura 20: Interfaz Lorient. Enroll a new device	49
Figura 21: Interfaz Lorient. Selección de versión.....	50
Figura 22: Extracto del Código proporcionado por STMMicroelectronics destacando las diferentes nomenclaturas por versiones	50
Figura 23: Interfaz Lorient. Creación de una nueva aplicación.....	52
Figura 24: Interfaz Lorient. Configuración de una nueva aplicación	52
Figura 25: Interfaz Lorient. Enganche de un nuevo Nodo.....	52
Figura 26: Interfaz Lorient. Configuración de un nuevo Nodo	53
Figura 27: Interfaz Lorient. Información del Nodo	53
Figura 28: Interfaz Lorient. Menú websocket applications	54
Figura 29: Interfaz Lorient. Muestreo de los datos recibidos en la comunicación entre dispositivos.	54
Figura 30: Nodo en bolsa de plástico	55
Figura 31: Nodo en bolsa de plástico y recipiente de vidrio.....	56
Figura 32: Ensayo en vacío	56
Figura 33: Ensayo en cerebro simulado. Antena sumergida en medio acuoso con las propiedades del cerebro	58
Figura 34: Ensayo en cerebro real.....	58
Figura 35: Ensayo en cerebro real. A 90° respecto el ensayo inicial	59
Figura 36: Ensayo en cerebro real. Dos metros de distancia	59
Figura 37: Ensayo en cerebro real. Un metro de distancia.....	59
Figura 38: Resultados RSSI en Ensayo en vacío	61
Figura 39: Resultados SNR en Ensayo en vacío	62
Figura 40: Resultado RSSI en Ensayo en cerebro simulado.....	63

Figura 41: Resultados SNR en Ensayo en cerebro simulado	64
Figura 42: Resultados RSSI en Ensayo en cerebro real a 1 metro de distancia.....	65
Figura 43: Resultados SNR en Ensayo en cerebro real a 1 metro de distancia.....	65
Figura 44: Resultados RSSI. Ensayo en cerebro real a 2 metros de distancia	66
Figura 45: Resultados SNR. Ensayo en cerebro real a 2 metros de distancia.....	67
Figura 46: Resultados RSSI. Ensayo en cerebro real a 2 metros y 90°.....	67
Figura 47: Resultados SNR. Ensayo en cerebro real a 2 metros y 90°.....	68
Figura 48: EDP del Trabajo de Fin de Grado.....	74
Figura 49: Gantt	76
Figura 50: Cambio LoRa_Driver	101
Figura 51: Break en lrwan_rs1_atcmd.c	101
Figura 52: Error de GDP	102
Figura 53: Cambio TCP Original	102
Figura 54: Cambio TCP Alterado	102
Figura 55: Resultado de los cambios realizados	103

Índice de tablas

Tabla 1: Límites de Absorción (SAR) Cuerpo Humano	33
Tabla 2: Comparativa entre protocolos Sub-GHz	35
Tabla 3: Resultados Ensayos	68
Tabla 4: EDP del Trabajo de Fin de Grado	73
Tabla 5 Presupuesto del Proyecto	77

Índice de ecuaciones

Ecuación 1: Teorema del muestro de Nyquist:.....	33
Ecuación 2: Cálculo de la Tasa de Envío de Datos Necesaria	34

1 Objetivos

Este Trabajo Fin de Grado se encuentra enmarcado en el proyecto del diseño de un dispositivo implantable en el cerebro para el tratamiento del glioblastoma multiforme. Este trabajo se realiza en colaboración con Instituto de Salud Carlos III, IMDEA Materiales, Hospital de la Princesa, Hospital Niño Jesús y la empresa INSYTE.

El objetivo del Trabajo Fin de Grado se engloba en el módulo de comunicación del proyecto, concretamente en el diseño de una antena implantable, miniaturizada y compacta para el tratamiento y monitorización en pacientes con enfermedades cerebrales.

Los objetivos particulares del Trabajo Fin de Grado quedan en:

- Investigación, análisis y selección de protocolos de comunicación.
- Elección y configuración del Hardware.
- Implementación del protocolo de comunicación.
- Realización de ensayos y evaluación del desempeño.
- Análisis, documentación y presentación de resultados.

2 Introducción

2.1 Contexto

Tal y como se ha comentado, este Trabajo Fin de Grado se encuentra enmarcado en el proyecto del diseño de un dispositivo implantable en el cerebro para el tratamiento del glioblastoma multiforme. El proyecto está siendo llevado a cabo en colaboración con Instituto de Salud Carlos III, IMDEA Materiales, Hospital de la Princesa, Hospital Niño Jesús y la empresa INSYTE.

El proyecto persigue que, mediante el uso de campos eléctricos alternos de determinada frecuencia e intensidad, sea posible tratar el glioblastoma multiforme y de esta forma, conseguir frenar su expansión. Así, se está logrando según los primeros estudios casi triplicar la esperanza de vida del paciente la cual, sin el tratamiento, apenas llega al año [4].

Sin embargo, y en lo que respecta a este Trabajo de Fin de Grado, los usos de un módulo de comunicación implantado en el cerebro pueden ser múltiples y muy variados. Desde el tratamiento del tumor previamente mencionado, a la recopilación y análisis de información cerebral para el estudio de distintas enfermedades neurodegenerativas.

Esto dota a este Trabajo de Fin de Grado de una gran versatilidad en cuanto a su uso ya que este mismo módulo puede ser conectado a distintos sensores o emisores de ondas.

2.2 Metodología

Para llevar a cabo este ambicioso proyecto, es fundamental desglosarlo en componentes más sencillos, como el que se aborda en este TFG: el módulo de comunicación del dispositivo.

Dentro de este módulo, para la realización del Trabajo de Fin de Grado, se decide subdividirlo, además, en 4 partes principales:

1. El estudio de la información actual y la elección del protocolo idóneo para el proyecto.
2. La elección del hardware necesario para implementar el protocolo elegido y cumplir con los requisitos del proyecto.
3. La implementación del protocolo en el hardware adquirido.
4. Los distintos ensayos necesarios para testear este pequeño módulo.

3 Estado del arte

3.1 Glioblastoma Multiforme

El glioblastoma multiforme (GBM) es uno de los tipos de cáncer cerebral más agresivos y mortales que afectan a los adultos. Este tumor maligno se origina en las células gliales del cerebro, específicamente en los astrocitos, que son células en forma de estrella responsables de proporcionar soporte y nutrientes a las neuronas. Aparece a menudo en el frontal del cerebro del paciente. El GBM representa aproximadamente el 15% de todos los tumores cerebrales y entre el 60% y el 75% de los astrocitomas, una categoría de tumores que se desarrollan a partir de los astrocitos.

El glioblastoma multiforme se caracteriza por su rápido crecimiento y su capacidad para infiltrarse en el tejido cerebral circundante, lo que lo hace extremadamente difícil de tratar. A diferencia de otros tumores cerebrales que pueden estar encapsulados y ser más fáciles de extirpar quirúrgicamente, el GBM tiende a diseminarse de manera difusa a través del cerebro, lo que complica su tratamiento y contribuye a su mal pronóstico como se observa en la Figura 2 [5].

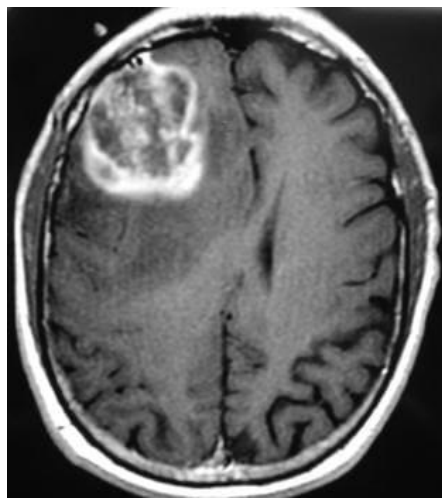


Figura 2: Glioblastoma Multiforme

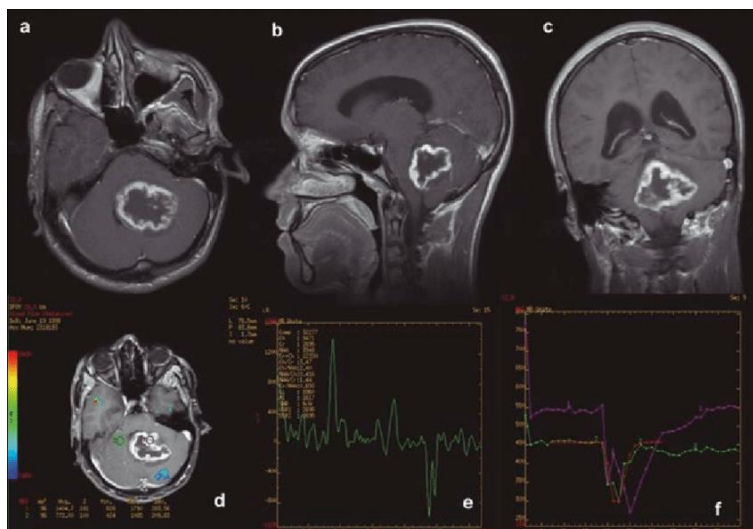


Figura 3: Resonancia Magnética con gadolinio

Los síntomas del GBM pueden variar ampliamente dependiendo de la ubicación del tumor en el cerebro. Los pacientes pueden experimentar dolores de cabeza persistentes, convulsiones, pérdida de memoria, cambios en la personalidad o el comportamiento, y déficits neurológicos focales, como debilidad en un lado del cuerpo. Estos síntomas a menudo llevan a los pacientes a buscar atención médica, donde el diagnóstico se realiza mediante técnicas de imagen como la resonancia magnética (RM) con gadolinio (Figura 3), que revela características típicas como un anillo de realce con necrosis central. La confirmación diagnóstica se obtiene a través de una biopsia estereotáctica o una resección quirúrgica, seguida de un análisis histopatológico.

El tratamiento del glioblastoma multiforme es multimodal e incluye una combinación de cirugía, radioterapia y quimioterapia. La cirugía se utiliza para extirpar la mayor cantidad posible del tumor, seguida de radioterapia y quimioterapia para destruir las células tumorales restantes. A pesar de estos esfuerzos, el pronóstico para los pacientes con GBM sigue siendo desfavorable, con una mediana de supervivencia de aproximadamente 12 a 15 meses. La recurrencia del tumor es común, y la mayoría de los pacientes experimentan una progresión de la enfermedad a pesar del tratamiento agresivo [6]. El proyecto en el que se enmarca este Trabajo de Fin de Grado se enfoca en el tratamiento por continua emisión de campos eléctricos alternos hacia el glioblastoma para ralentizar la expansión del tumor mediante el dispositivo implantado en el cerebro del paciente.

La investigación en el campo del glioblastoma multiforme está en constante evolución, con un enfoque en el desarrollo de nuevas terapias dirigidas, inmunoterapia, terapias génicas y ensayos clínicos para evaluar la eficacia de nuevos tratamientos. A pesar de los desafíos, los avances en la comprensión de la biología molecular del GBM y el desarrollo de nuevas estrategias terapéuticas ofrecen esperanza para mejorar los resultados y la calidad de vida de los pacientes afectados por esta devastadora enfermedad [7].

3.2 Tratamiento de tumores con campos eléctricos

Los tratamientos de tumores con campos electromagnéticos alternos representan una innovadora aproximación en la lucha contra el cáncer. Esta técnica utiliza ondas electromagnéticas de frecuencia y potencia específicas para interferir con la división celular del tumor, impidiendo su crecimiento y promoviendo su destrucción. A diferencia de los tratamientos tradicionales como la quimioterapia o la radiación, este enfoque es menos invasivo y puede focalizarse directamente en las células cancerígenas, reduciendo los efectos secundarios en los tejidos sanos. Los estudios actuales muestran resultados prometedores, especialmente en tumores resistentes a otras formas de tratamiento, abriendo nuevas posibilidades para mejorar la supervivencia y calidad de vida de los pacientes. [8]

3.3 El cerebro

El cerebro es el órgano principal del sistema nervioso central y desempeña un papel fundamental en el control y coordinación de todas las funciones del cuerpo humano. Es responsable de procesar la información sensorial, regular las funciones corporales, y permitir el pensamiento, la memoria y las emociones. En los seres humanos y en general vertebrados, el cerebro se encuentra protegido por el cráneo y está compuesto por varias estructuras que trabajan en conjunto para garantizar un funcionamiento adecuado [9].

3.3.1 Fisiología

Las partes del cerebro más importantes son las siguientes:

Corteza Cerebral: Es la capa externa del cerebro y está involucrada en funciones como el pensamiento, la percepción, y el control voluntario del movimiento. Se divide en cuatro lóbulos:

- **Lóbulo Frontal:** Asociado con funciones ejecutivas, como el razonamiento, la planificación y el autocontrol. Corresponde a la parte azul de la Figura 4.
- **Lóbulo Parietal:** Procesa la información sensorial relacionada con el tacto, la temperatura y el dolor. Corresponde a la parte morada de la Figura 4.
- **Lóbulo Occipital:** Responsable de la visión. Corresponde a la parte verde de la Figura 4.
- **Lóbulo Temporal:** Involucrado en la audición y la memoria. Corresponde a la parte amarilla de la Figura 4.

Cerebelo: Situado debajo de la corteza cerebral, el cerebelo coordina los movimientos voluntarios y el equilibrio.

Tronco Encefálico: Conecta el cerebro con la médula espinal y controla funciones vitales como la respiración, el ritmo cardíaco y la presión arterial.

Sistema Límbico: Incluye estructuras como el hipocampo y la amígdala, que están involucradas en la memoria y las emociones.



Figura 4: Partes del cerebro

3.3.2 Funciones

El cerebro tiene diversas funciones, y conocerlas es fundamental para saber cuáles se pueden llegar a ver afectadas por el glioblastoma multiforme o por la implantación del dispositivo que alberga el módulo de comunicación.

Las principales funciones son:

3.3.2.1 Funciones Cognitivas

Las funciones cognitivas del cerebro abarcan una amplia gama de procesos mentales que incluyen la percepción, el aprendizaje, la memoria, el lenguaje, la toma de decisiones y la creatividad.

3.3.2.1.1 Percepción y Procesamiento Sensorial

El cerebro recibe y procesa información sensorial a través de los sentidos: vista, oído, olfato, gusto y tacto. Cada modalidad sensorial tiene áreas específicas en el cerebro encargadas de su procesamiento:

Visión: La información visual se procesa principalmente en el lóbulo occipital, donde la corteza visual primaria interpreta los estímulos recibidos desde los ojos. Este procesamiento incluye la detección de formas, colores, movimientos y la integración de la información para formar imágenes coherentes.

Audición: La corteza auditiva en el lóbulo temporal procesa las ondas sonoras detectadas por los oídos, descomponiéndolas en componentes como tono, volumen y timbre para reconocer patrones auditivos como palabras y música.

Tacto y Percepción Somatosensorial: La corteza somatosensorial, ubicada en el lóbulo parietal, procesa estímulos táctiles como la presión, la temperatura y el dolor, así como la información sobre la posición del cuerpo en el espacio (propiocepción).

Olfato y Gusto: Los estímulos olfativos son procesados por el bulbo olfatorio y las áreas límbicas, mientras que el gusto se procesa en la corteza gustativa del lóbulo parietal.

3.3.2.1.2 Memoria y Aprendizaje

El cerebro almacena y recupera información a través de complejos procesos de memoria que se pueden categorizar en diferentes tipos:

Memoria a Corto Plazo (MCP): También conocida como memoria de trabajo, permite retener y manipular información durante breves periodos, como recordar un número de teléfono momentáneamente.

Memoria a Largo Plazo (MLP): Se refiere a la capacidad de almacenar información durante periodos prolongados, que puede durar desde días hasta toda la vida. Se divide en:

Memoria Declarativa: Incluye hechos y eventos (memoria episódica) y conocimientos generales (memoria semántica).

Memoria No Declarativa: Incluye habilidades motoras y hábitos (memoria procedimental), y condicionamientos.

El hipocampo es fundamental para la formación de nuevos recuerdos, especialmente en la conversión de la memoria a corto plazo en memoria a largo plazo. El deterioro del hipocampo, como se observa en la enfermedad de Alzheimer, conduce a la pérdida de memoria y dificultades para formar nuevos recuerdos.

3.3.2.1.3 Funciones Ejecutivas y Toma de Decisiones

Las funciones ejecutivas son procesos mentales de alto nivel que incluyen la planificación, la organización, la toma de decisiones, el control de impulsos y la adaptación a nuevas situaciones. Estas funciones son mediadas principalmente por la corteza prefrontal:

Planificación y Organización: Implica la capacidad de prever futuros eventos, establecer metas y diseñar estrategias para alcanzarlas.

Toma de Decisiones: La corteza prefrontal evalúa opciones, sopesando los pros y contras de cada alternativa, y selecciona la más adecuada basándose en la lógica, la experiencia y las emociones.

Control de Impulsos y Flexibilidad Cognitiva: El cerebro regula las respuestas impulsivas y permite ajustar el comportamiento en función de nuevas reglas o situaciones cambiantes.

Las disfunciones en las funciones ejecutivas pueden manifestarse en trastornos como el TDAH (Trastorno por Déficit de Atención e Hiperactividad) y el TOC (Trastorno Obsesivo-Compulsivo).

3.3.2.1.4 Lenguaje

El lenguaje es una de las funciones más complejas del cerebro, involucrando varias áreas específicas:

Área de Broca: Localizada en el lóbulo frontal, es crucial para la producción del habla y la articulación.

Área de Wernicke: Ubicada en el lóbulo temporal, es responsable de la comprensión del lenguaje.

El procesamiento del lenguaje también involucra la corteza motora para el control de los músculos del habla, así como la corteza auditiva para la comprensión de la lengua hablada.

3.3.2.2 Funciones Motoras

El control del movimiento es una de las funciones más fundamentales del cerebro, involucrando una coordinación precisa entre diferentes regiones:

3.3.2.2.1 Control Voluntario del Movimiento

El sistema motor del cerebro incluye:

Corteza Motora Primaria: Localizada en el lóbulo frontal, es la principal región responsable de la generación de impulsos que controlan los movimientos voluntarios.

Corteza Premotora y Motora Suplementaria: Planifican y coordinan movimientos complejos, como los que requieren la integración de múltiples grupos musculares.

3.3.2.2.2 Coordinación y Equilibrio

El cerebelo desempeña un papel clave en la coordinación del movimiento, asegurando que los movimientos sean suaves y precisos. También es esencial para mantener el equilibrio y la postura.

3.3.2.2.3 Movimientos Automáticos y Reflejos

Los ganglios basales y el tronco cerebral intervienen en el control de movimientos automáticos, como caminar, y en la regulación de reflejos involuntarios.

Los trastornos del sistema motor, como el Parkinson y la esclerosis múltiple, resultan en dificultades para iniciar, coordinar y controlar los movimientos.

3.3.2.3 Regulación Emocional y Comportamental

El cerebro no solo procesa información cognitiva y motora, sino que también es el centro de nuestras emociones y comportamientos:

3.3.2.3.1 Procesamiento Emocional

El sistema límbico, en particular la amígdala, es fundamental para el procesamiento emocional. Las emociones básicas como el miedo, la ira, la alegría y la tristeza son gestionadas y moduladas por esta estructura.

3.3.2.3.2 Regulación del Estrés

El hipotálamo regula la respuesta al estrés a través del eje hipotalámico-hipofisario-adrenal (HHA), que controla la liberación de hormonas como el cortisol, modulando la respuesta del cuerpo ante situaciones de estrés.

3.3.2.3.3 Motivación y Recompensa

El núcleo accumbens y el sistema de dopamina están involucrados en la motivación y la percepción de recompensa. Estas regiones influyen en la toma de decisiones y el comportamiento dirigido a objetivos.

El desequilibrio en la regulación emocional puede dar lugar a trastornos como la depresión, la ansiedad y el trastorno de estrés postraumático (TEPT).

3.3.2.4 Funciones Autónomas y Homeostasis

El cerebro también regula funciones autónomas, esenciales para la supervivencia:

3.3.2.4.1 Regulación del Ritmo Cardíaco y la Respiración

El tronco cerebral regula funciones vitales automáticas como la respiración, el ritmo cardíaco y la presión arterial. Estas funciones son controladas por centros específicos en el bulbo raquídeo y la protuberancia.

3.3.2.4.2 Regulación del Ciclo Sueño-Vigilia

El hipotálamo y el tronco cerebral también intervienen en la regulación del ciclo circadiano, gestionando la alternancia entre estados de sueño y vigilia a través de la liberación de hormonas como la melatonina.

3.3.2.4.3 Regulación de la Temperatura Corporal y el Balance Hídrico

El hipotálamo es el centro regulador de la temperatura corporal y del equilibrio de líquidos en el cuerpo, ajustando procesos como la sudoración y la sed en respuesta a cambios en el entorno o en la actividad física.

Para ampliar esta información, se puede acudir a “Kandel, E. R., Schwartz, J. H., & Jessell, T. M. (2000). *Principles of Neural Science* (4th ed.). McGraw-Hill” [10].

3.3.3 Ondas de actividad cerebral

Como se ha mencionado, este Trabajo de Fin de Grado no solo se puede emplear en el tratamiento frente al glioblastoma multiforme, sino también sirve como parte de un dispositivo de investigación y análisis frente a distintas enfermedades neurológicas. Es por ello, que el estudio de las ondas de actividad cerebral juega un importante papel en la elección del protocolo a emplear [11].

Ondas Delta

Frecuencia: 0.5 a 4 Hz

Características: Las ondas delta son las de frecuencia más baja y suelen tener la amplitud más alta. Son características del sueño profundo sin sueños, también conocido como sueño de ondas lentas (SWS).

Función: Están relacionadas con los procesos de restauración y recuperación del cerebro y el cuerpo. Son predominantes durante la fase de sueño no REM, donde el cuerpo se recupera y se consolida la memoria.

Ubicación: Principalmente en la corteza cerebral, más pronunciadas en los lóbulos frontales durante el sueño profundo.

Ondas Theta

Frecuencia: 4 a 8 Hz

Características: Las ondas theta son más rápidas que las delta, pero aún se consideran de baja frecuencia. Son prominentes en estados de sueño ligero, meditación profunda y relajación extrema.

Función: Asociadas con la creatividad, la intuición y la relajación profunda. Las ondas theta también aparecen durante las primeras etapas del sueño y están involucradas en la memoria y el aprendizaje.

Ubicación: Se encuentran comúnmente en la corteza temporal y en las regiones límbicas durante estados de relajación o en las primeras etapas del sueño.

Ondas Alfa

Frecuencia: 8 a 13 Hz

Características: Las ondas alfa representan un estado de calma y relajación mental sin estar dormido. Son típicamente observadas cuando los ojos están cerrados y la mente está en reposo, pero no completamente en un estado de sueño.

Función: Relacionadas con la meditación, la relajación, la reducción del estrés y la coordinación mente-cuerpo. También son esenciales para la transición entre los estados de alerta y sueño.

Ubicación: Mayormente en el lóbulo occipital, aunque pueden extenderse a otras áreas corticales durante la meditación o el reposo.

Ondas Beta

Frecuencia: 13 a 30 Hz

Características: Las ondas beta son de frecuencia más alta y están asociadas con la actividad mental activa, la concentración y la resolución de problemas.

Función: Representan un estado de alerta, pensamiento crítico y lógica. Estas ondas son predominantes cuando estamos despiertos y realizando tareas cognitivas activas, como hablar, escribir o resolver problemas.

Ubicación: Principalmente en la corteza frontal, aunque pueden ser observadas en otras partes del cerebro durante la actividad mental intensa.

Ondas Gamma

Frecuencia: 30 a 100 Hz

Características: Las ondas gamma son las de frecuencia más alta y están relacionadas con el procesamiento cognitivo de alto nivel, como la percepción y la conciencia.

Función: Asociadas con la integración de la información sensorial y la consciencia plena. Las ondas gamma están implicadas en la función cognitiva avanzada, como el aprendizaje y la memoria de trabajo.

Ubicación: Se generan en diversas áreas del cerebro, especialmente durante tareas que requieren procesamiento simultáneo de múltiples tipos de información.

Estas últimas tienen una especial importancia ya que determinarán la frecuencia mínima de muestreo y por ende, la tasa de envío mínima a la que el protocolo debe ser capaz de enviar información.

3.4 Comunicación de dispositivos médicos

Como se ha mencionado previamente, la comunicación de dispositivos médicos implantables, especialmente aquellos ubicados en el cerebro, es un campo crítico que involucra la transmisión segura, eficiente y confiable de datos entre el dispositivo implantado y sistemas externos. Estos sistemas de comunicación deben cumplir con estrictos requisitos en cuanto a seguridad, consumo energético, integridad de datos, y compatibilidad electromagnética, todo ello dentro de un entorno extremadamente sensible como el cerebro humano.

En la actualidad se tiene una extensa literatura sobre cómo afectan las frecuencias a través del cuerpo humano [12].

Banda MICS (Medical Implant Communication Service): 402-405 MHz

- Especialmente designada para comunicaciones con dispositivos implantables, ofrece un buen equilibrio entre penetración en el tejido y ancho de banda.
- Limitada en potencia de transmisión para minimizar el calentamiento del tejido (máximo 25 μ W EIRP).

Banda ISM (Industrial, Scientific, and Medical): 2.4-2.4835 GHz.

- Utilizada también en dispositivos implantables, aunque con más limitaciones debido a su mayor absorción por los tejidos corporales, lo que puede reducir la fiabilidad de la comunicación.
- Proporciona un mayor ancho de banda, pero a costa de un mayor consumo de energía y mayor atenuación en el tejido.

Banda UHF: ~433 MHz, la cual finalmente emplearemos.

- Ofrece una penetración de tejido superior a la banda de 2.4 GHz, con un consumo de energía moderado.

3.5 Métricas

En la evaluación de la calidad e intensidad de señales en sistemas de comunicación, es crucial utilizar métricas precisas que permitan valorar el rendimiento del sistema. Dos de las medidas más comúnmente empleadas son el **RSSI** (*Received Signal Strength Indicator*) y el **SNR** (*Signal-to-Noise Ratio*). Estas métricas proporcionan información valiosa sobre la potencia de la señal recibida y la calidad relativa de dicha señal frente al ruido de fondo, lo que es esencial para garantizar la fiabilidad en la comunicación una vez la señal ha atravesado el cerebro del paciente [13].

3.5.1 RSSI

El RSSI es una medida de la potencia de la señal recibida en un dispositivo de comunicación inalámbrica. Esta métrica es expresada en decibelios-milivoltios (dBm) y proporciona un valor que refleja la intensidad de la señal en el punto de recepción. Cuanto mayor sea el valor absoluto del RSSI (más negativo), menor será la potencia de la señal.

El RSSI es ampliamente utilizado en la evaluación del rendimiento de redes inalámbricas, tales como Wi-Fi, Bluetooth y sistemas de comunicación celular. En el contexto de sistemas implantados o en entornos biomédicos, el RSSI es crucial para asegurar que la señal transmitida sea suficientemente fuerte para atravesar tejidos y alcanzar los receptores deseados sin pérdidas significativas.

Respecto al rango de esta métrica se tiene:

- **< -50 dBm:** Señal excelente, ideal para la mayoría de las aplicaciones.
- **-51 dBm a -70 dBm:** Señal buena, generalmente fiable para la mayoría de las tareas.
- **-71 dBm a -90 dBm:** Señal débil, puede sufrir de interferencias y degradación.
- **-91 dBm a -120 dBm:** Señal muy débil, con alta probabilidad de caídas de conexión o pérdida de datos.

Una limitación importante del RSSI es que no proporciona información directa sobre la calidad de la señal, ya que solo mide la intensidad sin considerar el nivel de ruido en el entorno. Esto hace que el RSSI sea una medida insuficiente por sí sola para determinar la fiabilidad de

una conexión inalámbrica, especialmente en entornos ruidosos. Por eso mismo, para complementar esta métrica, se empleará el SNR.

3.5.2 SNR

El **SNR** (*Signal-to-Noise Ratio*) es una medida de la calidad de una señal en relación con el ruido de fondo presente en el entorno. Se calcula como la relación entre la potencia de la señal y la potencia del ruido, generalmente expresada en decibelios (dB). Un SNR más alto indica que la señal es más fuerte en comparación con el ruido, lo que resulta en una mayor calidad de la comunicación.

El SNR es una métrica fundamental en la mayoría de los sistemas de comunicación, incluyendo aquellos utilizados en entornos médicos o de laboratorio. Un alto SNR es esencial para garantizar la integridad de los datos transmitidos, ya que un SNR bajo puede llevar a errores en la transmisión y recepción de la señal, lo cual es particularmente crítico en aplicaciones donde la precisión es vital.

El rango de valores en el que se mueve esta métrica es el siguiente:

- **SNR > 20 dB:** Señal de alta calidad, ideal para la mayoría de las aplicaciones.
- **SNR entre 10 y 20 dB:** Señal aceptable, pero podría ser susceptible a interferencias.
- **SNR < 10 dB:** Señal pobre, con alta probabilidad de errores de transmisión.

A pesar de ser una medida excelente de la calidad de la señal, el SNR también tiene limitaciones. Por ejemplo, en situaciones donde tanto la señal como el ruido son bajos, el SNR puede ser alto, pero la calidad general de la señal puede no ser adecuada para una comunicación fiable.

4 Análisis y Selección del Protocolo de Comunicación.

4.1 Características

Para poder alcanzar el objetivo estipulado, y de acuerdo con las restricciones impuestas por el propio proyecto y a la información recogida en el estudio previo del cerebro, se puede concluir que la elección del protocolo de comunicación es una de las actividades críticas para el éxito del proyecto. Es por esto por lo que dicha actividad supone gran parte del tiempo destinado al Trabajo Fin de Grado, ya que una elección incorrecta implica automáticamente la invalidez del trabajo.

Las características requeridas por el protocolo a emplear son las siguientes:

- Alcance
- Bajo Consumo
- Límite de absorción
- Seguridad
- Tasa de envío
- Transmisibilidad

4.1.1 Alcance

El alcance hace referencia a la distancia entre el emisor y el receptor del módulo de comunicación. Un mayor alcance ofrece mayor libertad, pero puede limitar el resto de las características.

El proyecto no requiere de un alcance determinado, por lo que se plantea la solución de enviar los datos recogidos a un dispositivo receptor localizado en el propio teléfono del paciente o incluso en una pulsera o “smartwatch” que podría llegar a diseñarse si el paciente no cuenta con ella.

De esta forma, se está eliminando una de las restricciones, el alcance ya no sería un factor determinante a la hora de escoger el protocolo a utilizar.

4.1.2 Bajo Consumo

El consumo sí es un factor crítico para el proyecto debido a la dificultad que puede llegar a suponer realizar cargas de batería constantes. Si bien es cierto que el aspecto de la recarga de batería se está considerando en otros trabajos, se debe tratar de consumir lo mínimo posible, ya que esta característica afecta también al tamaño de la batería, y por tanto al del dispositivo, y al correcto funcionamiento del mismo. Ya que la función principal es el tratamiento del glioblastoma multiforme, el módulo de comunicación no debería consumir una parte importante de la energía disponible.

Con el objetivo de emplear la menor cantidad de energía posible, se tratará de encontrar protocolos asíncronos [14] para poder evitar el derroche de energía en tiempos muertos. Además se buscarán protocolos eficientes capaces de emitir los datos necesarios empleando la menor energía posible.

El consumo de energía se localiza principalmente en los cambios de modo de dormido a activo, por lo que se tratará de reducir el número de cambios realizados acumulando en un buffer los datos a enviar. Es por ello por lo que el protocolo debe poder adaptarse a este requerimiento.

4.1.3 Límite de Absorción

El límite de absorción, entendida como la absorción máxima permitida por el paciente, es una condición impuesta por el entorno en el que se trabaja, en este caso dentro de la cabeza del paciente. Por ello, este límite de absorción es un límite fisiológico y que trata de asegurar la salud del paciente. Para esta característica se emplean los datos de SAR.

Los límites de Tasa de Absorción Específica (SAR, Specific Absorption Rate) se refieren a la cantidad máxima de energía de radiofrecuencia (RF) que puede ser absorbida por el cuerpo humano cuando está expuesto a un campo electromagnético. La SAR se mide en vatios por kilogramo (W/kg) y se utiliza para garantizar que los dispositivos electrónicos, como teléfonos móviles y equipos de resonancia magnética (MRI), no excedan niveles seguros de exposición a la radiación RF [15]

En la Tabla 1 se puede observar los valores máximos de absorción en la cabeza. Además, se debe tener en cuenta, como indica la propia tabla, la temperatura de operación, la cual se sitúa entre los 36 y 41 grados centígrados [16]

Operating Mode	Volume Transmit Coils			Local Transmit Coils		
	Whole body	Partial body	Head	Head	Trunk	Extremities
Normal	2	2-10 ^a	3.2	10	10	20
1 st Level	4	4-10 ^b	3.2	20	20	40
2 nd Level	<IRB limit	<IRB limit	<IRB limit	<IRB limit	<IRB limit	<IRB limit

IEC SAR Limits (W/kg). All values are based on the environmental temperature being $\leq 25^{\circ}\text{C}$. For higher temperatures, the First Level whole body limit (4 W/kg) only is derated by 0.25 W/kg for each 1°C rise until the Normal operating level limit (2 W/kg) is reached. Partial body limits are scaled according to Mass Ratio (R) = RF-exposed patient mass \div total patient mass, giving ranges: a = $[10 - 8 \times R]$ W/kg and b = $[10 - 6 \times R]$ W/kg. Transmit/receive knee and wrist coils are covered under the "partial body" limits.

Tabla 1: Límites de Absorción (SAR) Cuerpo Humano

4.1.4 Seguridad

Al tratarse de datos personales y de salud, la seguridad debe ser muy elevada, cualquier tipo de filtración o robo de información puede repercutir muy negativamente tanto en la vida del paciente al conocerse sus datos de salud, como a nivel reputacional por parte de la organización responsable. Esta responsabilidad puede llegar a ser económica e incluso penal.

Otro aspecto de crucial importancia es que el dispositivo debe ser capaz de emitir distintas potencias y frecuencias en función de los requisitos médicos, en caso de ser vulnerable y recibir un ataque, podría llegarse a emitir campos eléctricos perjudiciales para la salud del paciente, lo que supone un grave peligro.

De cara a la realización del Trabajo Fin de Grado, se pospone la capa de seguridad necesaria para este tipo de información. Teniendo en cuenta exclusivamente que el protocolo sea capaz de adoptar esta seguridad cuando fuese necesario a modo de encriptación y de forma previa al envío de los datos.

4.1.5 Tasa de envío

La tasa de envío de los datos, entendida como los datos necesarios a enviar por segundo para tener una comunicación satisfactoria, dependerá fundamentalmente de los datos a enviar. Para este proyecto, y como se pretende enviar información de batería, temperatura y datos relacionados con el estado del paciente y del dispositivo no supondría una fuerte restricción.

Sin embargo, como el Trabajo Fin de Grado puede ser empleado para el estudio de otras enfermedades neurológicas, se tratará de enviar datos suficientes para el muestreo y análisis de las ondas cerebrales, ya que, en muchas ocasiones, estas no pueden analizarse al no tener suficiente transmisibilidad a través del cráneo y otro tipo de tejidos.

Ecuación 1: Teorema del muestro de Nyquist:

$$f_{\text{muestreo}} \geq 2 * f_{\text{max}} \quad (1)$$

Es por esto, que acorde al teorema de Nyquist [17] de la Ecuación 1 vamos a establecer una tasa de envío mínima requerida de 100kbps acorde a la Ecuación 2, partiendo de una $f_{max} = 100Hz$ y suponiendo un ADC de 64 bits y 8 canales [18].

Ecuación 2: Cálculo de la Tasa de Envío de Datos Necesaria

$$Tasa\ de\ Envío\ de\ Datos \geq \frac{Bits\ ADC}{T_{muestro}} = Bits\ ADC * f_{muestreo} \quad (2)$$

4.1.6 Transmisibilidad

La transmisibilidad, entendida como la capacidad de la onda electromagnética para atravesar distintos medios sin perder información, es otra característica clave de este proyecto. La elección de una frecuencia correcta, capaz de atravesar todas las capas de las que se compone la cabeza es fundamental para el correcto envío de la información a través del protocolo elegido.

La frecuencia elegida para este tipo de comunicación debe ser Sub-GHz como se expuso previamente. Y dentro de este tipo de frecuencias la recomendada es la de 400MHz [19]

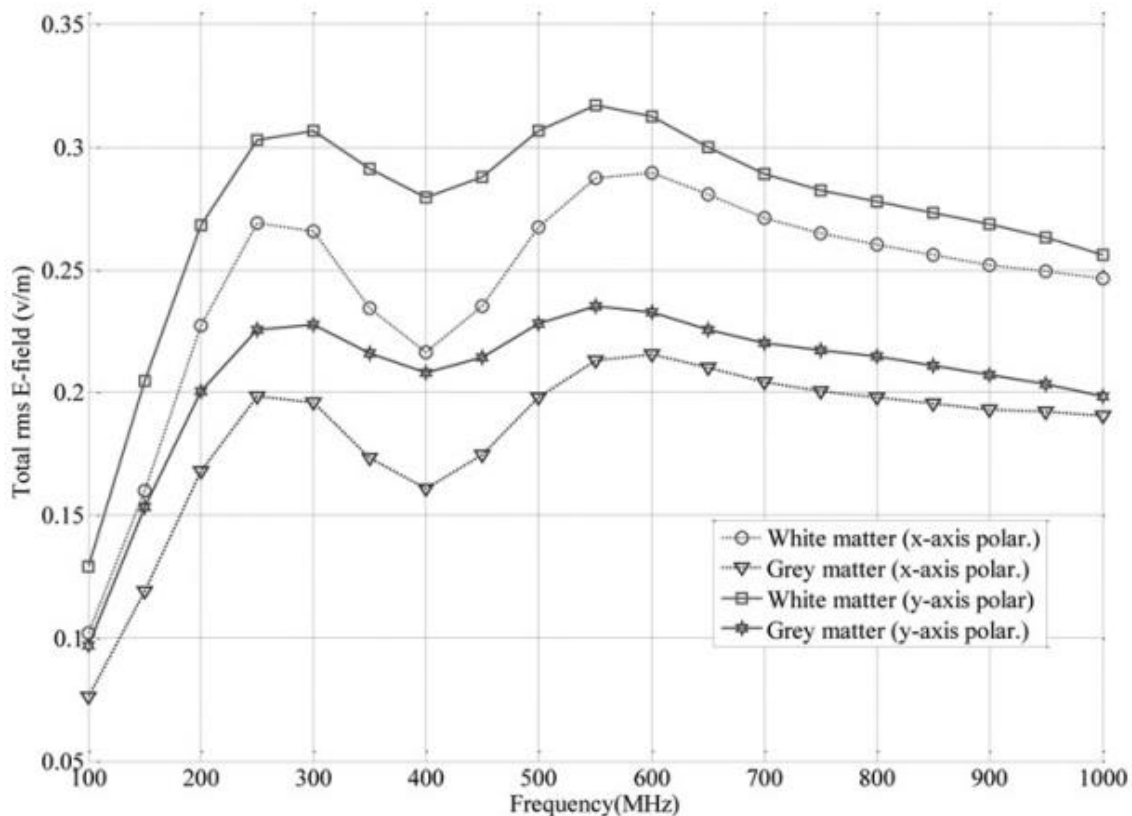


Figura 5: Absorción de los distintos tejidos del cerebro ante diferentes frecuencias

Como se puede observar en la Figura 5, la frecuencia de 400Hz da el mejor resultado ya que absorbe menos que otras frecuencias.

Los 100 MHz quedan descartado acorde a lo visto previamente respecto al criterio de Nyquist, en el que se establece que el mínimo deberá ser de 200MHz.

4.2 Elección del protocolo

En la Tabla 2 se observa una recopilación de distintos protocolos y sus características principales.

Protocolo	Frecuencia	Alcance	Velocidad de Datos	Consumo de Energía
SigFox	868 MHz (EU), 915 MHz (US)	30-50 km	Hasta 100 bps	Muy bajo
LoRa	433 MHz, 868 MHz, 915 MHz	10-15 km	0.3-50 kbps	Bajo
Wi-Fi HaLow	900 MHz	1 km	Hasta 347 Mbps	Moderado
Z-Wave	868 MHz (EU), 908 MHz (US)	30-100 m	Hasta 100 kbps	Bajo
Zigbee	868 MHz, 915 MHz	10-100 m	20-250 kbps	Muy bajo
Wireless M-Bus	868 MHz (EU), 915 MHz (US)	1-2 km	Hasta 100 kbps	Bajo
Dash7	433 MHz, 868 MHz, 915 MHz	Hasta 2 km	Hasta 167 kbps	Muy bajo

Tabla 2: Comparativa entre protocolos Sub-GHz.

Analizando la información de la tabla se encuentra que el protocolo a emplear debería ser el de Dash7 ya que:

- ✓ Permite la frecuencia de 433MHz, la más cercana a los 400MHz óptimos.
- ✓ Tiene un alcance suficiente como para poder dejar el dispositivo receptor en un lugar de la vivienda del paciente y que éste pueda realizar vida normal sin tener que preocuparse por el alcance.
- ✓ La Tasa de datos es superior a la requerida.
- ✓ El consumo de energía por parte de este protocolo es muy bajo

Se observa que el protocolo de LoRa, también es válido aunque tenga una tasa de datos menor y un consumo un poco mayor que el primero.

4.3 Dash7

Una vez elegido este protocolo, se comienza con su implementación en la placa “STM32 Nucleo-64-F103RB” que contiene el microprocesador Cortex-M3 compatible con esta placa, como se comentará más adelante. También se estudió el posible uso del Cortex-M4 el cual será descartado.

Para ello se emplea el entorno de “IDE Cube” junto al firmware proporcionado por el fabricante. Y se prueba a usar la antena C1101 [20] acorde a la información del fabricante.

Una vez probada la placa con un código “Blink” y probada la antena, se prepara el código para el uso de Dash7. Para ello se emplea información obtenida por estudios y ensayos pasados con este protocolo, como el que se puede encontrar en el punto 4 de este informe publicado por Sage en el año 2013 [21].

Lamentablemente, tras varios meses de intentos fallidos y aprendizaje del uso del entorno IDE junto de los comandos HAL compatibles con este tipo de placas de prueba, se notará que aparte de la poca información existente sobre este protocolo este se encuentra en desuso, se dejó de actualizar y ya prácticamente no se emplea más que en determinadas aplicaciones de tipo militar. Los ensayos con este fueron probados en la placa Cortex M4 y M3.

4.4 LoRaWAN

Tras el intento fallido de emplear Dash7, el cual parecía mejor como se pudo ver en el apartado 4.2, se decide emplear el protocolo de LoRa, el cual es igualmente viable para el proyecto teniendo en cuenta que tiene desventajas como que tanto la tasa de envío como el consumo eran algo peor pero con grandes ventajas en el uso efectivo del protocolo. Al ser más reconocido y empleado globalmente es posible encontrar más información acerca de su uso en distintos proyectos. Además y como veremos en el apartado del Hardware, este protocolo está implementado parcialmente en placas de desarrollo.

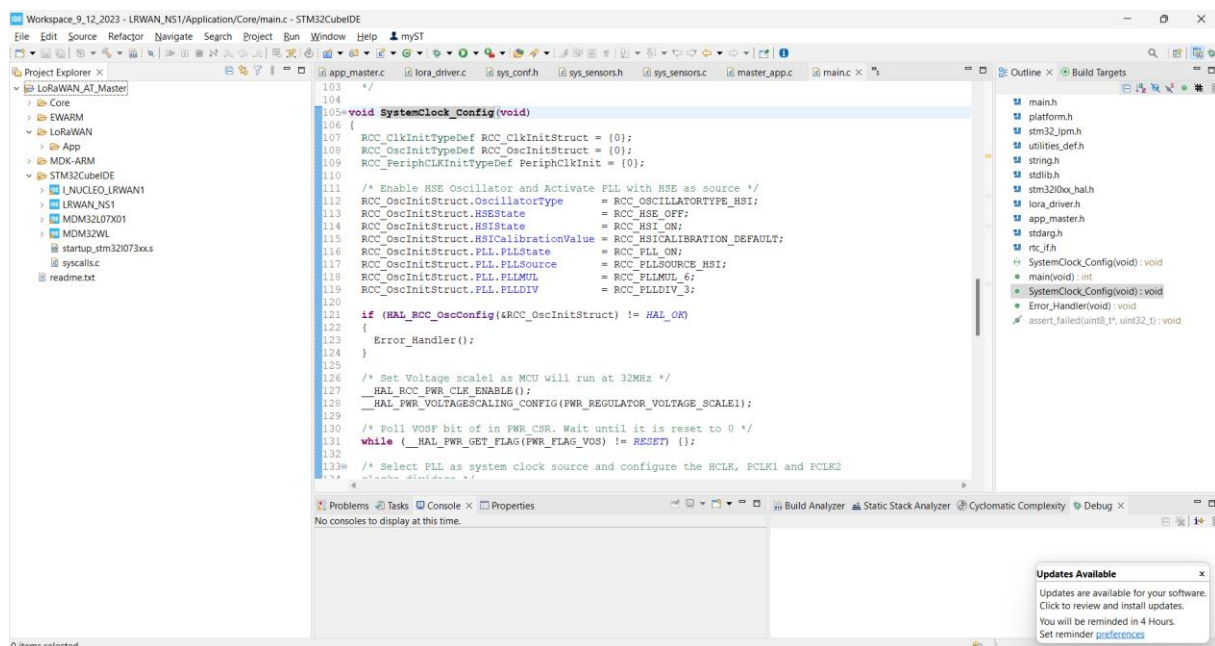


Figura 6: Imagen IDE Cube

En la Figura 6 se puede ver una imagen del entorno con el código proporcionado por el fabricante del modelo de desarrollo para permitir ensayos.

Este será por tanto el protocolo seleccionado para realizar los ensayos y ser empleado en el proyecto general de ser considerados validos como verificaremos en el punto 8.

4.5 Energía consumida

Para calcular el consumo de energía de un protocolo LoRa (o Dash7 ya que serán realmente similares) en una situación específica, se debe considerar varios factores, como la tasa de transmisión de datos, la frecuencia de envío, el tamaño de los datos a enviar, y las características del ADC (convertidor analógico-digital).

4.5.1 Transmisión de datos requerida

- Frecuencia de envío: 200Hz
- ADC: 64 bits x 8 canales
- Tasa de transmisión: 100kbps

Para calcularlo realizamos las siguientes operaciones:

$$\text{Bits de muestreo} = 64\text{bits} * 8\text{canales} = 512 \text{ bits}$$

$$\text{Tasa de bits requerida} = \frac{512\text{bits}}{\text{muestreo}} * 200 \frac{\text{muestras}}{\text{segundo}} = 102400 \frac{\text{bits}}{\text{segundo}}$$

Se puede ver que la tasa empleada de 100kbps prácticamente iguala la tasa requerida.

Por ello el consumo de energía se estaría realizando de forma permanente, sin llegar a estar en reposo. Con un consumo típico de LoRa de unos 100mW se estaría consumiendo 100mJ por segundo.

5 Elección y Configuración del Hardware

5.1 Elección General del Hardware

La elección del Hardware se ve fuertemente influenciada por la elección previa del protocolo como se ha podido comprobar en el Punto 4. Al ser un modelo de desarrollo el que se emplea en esta primera fase del proyecto, parámetros como el tamaño o la biocompatibilidad de los componentes no se están teniendo en cuenta para dicha elección.

Es por esto por lo que se diferencia el Hardware en función del protocolo elegido para los ensayos de transmisibilidad: Dash7 y LoRaWAN. Aunque finalmente solo el LoRaWAN es válido, se comenta la investigación realizada para implementar también el protocolo Dash7.

5.2 Dash7

5.2.1 Cortex-M4

Para el empleo de Dash7 es necesario emplear una placa de desarrollo que contase con 2 puntos fundamentales:

- Un microprocesador Cortex-M4
- Una antena Texas Instrument C1101

La elección de estos requisitos viene condicionada por el estudio previo realizado por Sage [21], en el que en el apartado 5 podemos encontrar claramente la siguiente información: *“OSS-7 is embedded stack implementation for D7A which currently supports CC430, CC1011, and ARM Cortex-M4.”*

Tras esto se preseleccionó una de las placas núcleo de desarrollo de ST, concretamente la placa STM32-F427-429VG que podemos ver en la Figura 7.

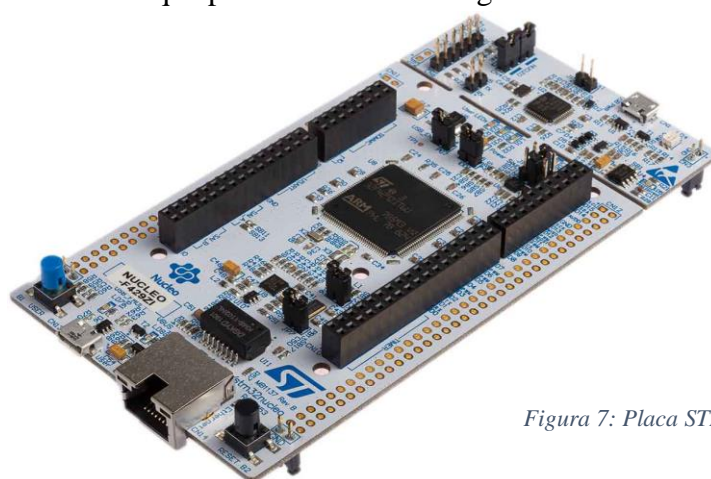


Figura 7: Placa STM32-F427-429VG

Sin embargo, previo a la compra definitiva de la placa se encuentra la siguiente información acerca de la incompatibilidad del protocolo Dash7 con este tipo de microprocesadores 8 bits

“the OpenTag DASH7 folk were saying that their code was “unlikely to be usable on 8-bit microcontrollers.”

<http://www.sourceforge.net/projects/opentag>” [22]

El problema que se encuentra aquí es que el enlace empleado como referencia en este foro ha sido eliminado por parte de la web, la cual es la oficial de OpenTag, quién es dueña del protocolo Dash7.

Esta fue la primera vez que se puso en duda la continuación de Dash7 como el protocolo elegido, ya que mucha información ha podido ser borrada por los propios propietarios del protocolo.

5.2.2 Cortex-M3

Una vez visto este problema, se continua con la búsqueda de un microprocesador capaz de soportar el protocolo, se eligió el Cortex-M3, el cual es perfectamente capaz de emplear este protocolo.

Concretamente se elige pasar al modelo de desarrollo: “STM-Nucleo-F103RB”, el cual podemos ver en la Figura 8.

Este modelo también emplea la antena CC1101 que ya sabíamos que era la más empleada para este protocolo y esta frecuencia (433MHz).

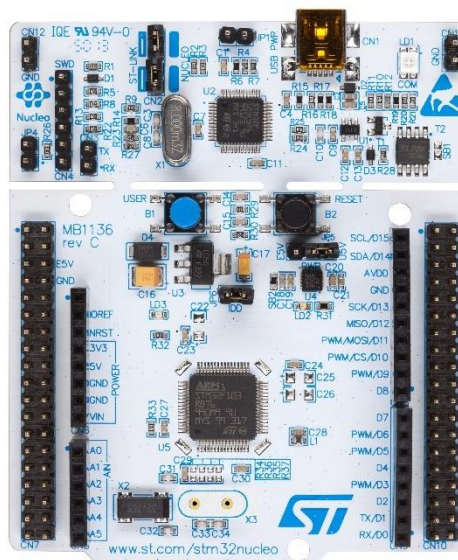


Figura 8: Placa STM-Nucleo-F103RB

Otra de las ventajas de esta placa era la similitud con el código Arduino, el cual es más rápido de programar a la hora de realizar pequeños ensayos de control de la placa. En la Figura 9 se puede ver el esquemático de la placa “STM-Nucleo-F103RB” junto al mapeo de los pines de la misma con los del Arduino.

Como se puede ver en el Punto 6 de implementación del código, el protocolo será de muy difícil implementación por lo que se acabará eligiendo LoRaWAN, es por ello que esta placa también será descartada a pesar de haber sido comprada y testeada tanto con código Arduino como con los comandos HAL en el entorno de IDE Cube.

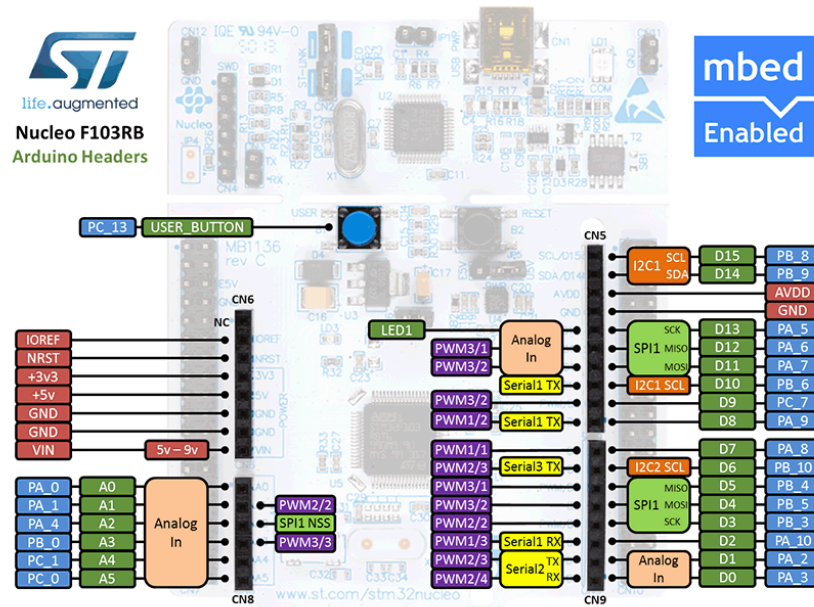


Figura 9: Mapeo PlacaF103RB con Arduino

5.3 LoRaWAN

5.3.1 Cortex-M0

Tras los intentos de emplear el protocolo Dash7, se tuvo que dejarlo de lado en favor del protocolo LoRaWAN, por los motivos explicados en el Punto 4.4.

Al tratarse de un nuevo protocolo, los requisitos Hardware varían, y es por eso que la búsqueda de una nueva placa de desarrollo era necesaria. En este caso, se tratará de buscar como principal característica la predisposición de la placa al uso de este nuevo protocolo, ya que al ser uno más empleado los propios fabricantes producen placas de desarrollo específicas para el LoRaWAN.

Otro de los requerimientos del Hardware era la posibilidad de tener 2 módulos, uno de emisión y otro de recepción de los mensajes y es por eso que el modelo “P-NUCLEO-LRWAN3” era perfecto para este propósito como vemos en la Figura 10.

Requisitos del nuevo Hardware

- Predisposición al uso del protocolo LoRa
- A ser posible preparado para realizar ensayos de emisión y recepción de mensajes
- Uso de la frecuencia de 433MHz

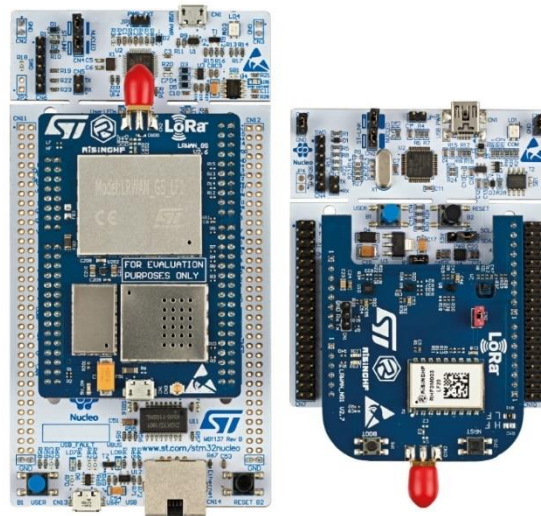


Figura 10: Placas de Desarrollo P-NUCLEO-LRWAN3

Finalmente, esta sería la placa elegida para la realización de los ensayos debido a su relativa facilidad de programación junto al cumplimiento de los requisitos enumerados previamente.

6 Implementación del protocolo de comunicación.

6.1 Dash 7

La implementación del protocolo Dash7 se realiza, principalmente, mediante el entorno de IDE Cube.

El primer desafío encontrado fue el manejo del entorno al igual que el empleo de los comandos HAL para controlar la placa de desarrollo del Cortex-M3 junto a la antena CC1101.

Los comandos HAL (Hardware Abstraction Layer) son una capa de abstracción que facilita la interacción con el hardware en el contexto de la programación de microcontroladores. Esta capa actúa como un intermediario entre el software de aplicación y los periféricos del microcontrolador, proporcionando una interfaz estandarizada que simplifica el desarrollo y mejora la portabilidad del código.

Además, los comandos HAL resultan fundamentales para abordar la complejidad del hardware. Al utilizar HAL, es posible escribir código que sea independiente del hardware específico, lo que permite adaptar el software a diferentes microcontroladores con cambios mínimos. Esta ventaja es crucial en proyectos donde la flexibilidad y la eficiencia son esenciales, como es este Trabajo Fin de Grado, ya que a priori no sabemos si el micro final a emplear será este exactamente u otro de características similares que cumpla los requisitos de tamaño, biocompatibilidad y consumo de energía además de seguir siendo viable para el envío de datos mediante el protocolo escogido.

Un último aspecto importante sobre los comandos HAL es el soporte que fabricantes como STMicroelectronics pueden aportar. Ver Figura 11.

Figura 11: Ejemplo del uso de comandos HAL

```

126@void SystemClock_Config(void)
127 {
128     RCC_OscInitTypeDef RCC_OscInitStruct = {0};
129     RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
130
131     /** Configure the main internal regulator output voltage
132     */
133     HAL_RCC_PWR_CLK_ENABLE();
134     __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE1);
135
136     /** Initializes the RCC Oscillators according to the specified parameters
137     * in the RCC_OscInitTypeDef structure.
138     */
139     RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
140     RCC_OscInitStruct.HSEState = RCC_HSE_BYPASS;
141     RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
142     RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
143     RCC_OscInitStruct.PLL.PLLM = 4;
144     RCC_OscInitStruct.PLL.PLLN = 168;
145     RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
146     RCC_OscInitStruct.PLL.PLLQ = 7;
147     if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
148     {
149         Error_Handler();
150     }
151
152     /** Initializes the CPU, AHB and APB buses clocks
153     */
154     RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
155         |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
156     RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
157     RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
158     RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV4;
159     RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV2;
160
161     if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_5) != HAL_OK)
162     {
163         Error_Handler();

```

Tras varias iteraciones tratando de hacer uso de la antena CC1101 junto al protocolo se decidió emplear Arduino para hacer ensayos con la antena de forma más rápida, al ser un entorno y un código más sencillo.

En la Figura 9 se puede ver el mapeo de los pines de la placa de desarrollo con los de la placa Arduino, además de las funciones como puede ser el SPI o los serial Rx y Tx entre otras.

Haciendo uso de la librería de "ELECHOUSE_CC1101_SRC_DRV.h" se realizaron las pruebas de la antena, las cuales funcionaron correctamente. Sin embargo, a la hora de tratar de emplear el protocolo de Dash7 para la transmisión de datos se pudo comprobar la enorme dificultad de implementarlo, debido principalmente a la falta de información sobre el protocolo al haberse convertido en uno privado y para utilizado para fines militares.

En el Anexo 1 se puede encontrar los códigos empleados.

6.2 LoRaWAN

Para la implementación de LoRaWAN se siguió una ruta de trabajo similar a la empleada con el Dash7.

Al ser una placa de desarrollo nueva, en este caso con un Cortex-M0, en primer lugar se trató de emplear el código Arduino para el control de la antena CC1101 como paso previo al uso del protocolo en sí mismo.

Una vez verificado el funcionamiento se continua con la implementación del protocolo LoRaWAN para la transmisión de datos entre puntos.

Antes de analizar la implementación del código cabe destacar la terminología que se empleará:

- **Nodo:** se trata del dispositivo emisor, el que se encontraría dentro del paciente.
- **Gateway:** se trata del dispositivo receptor, este se encontraría en un dispositivo móvil como el teléfono móvil del paciente o una pulsera inteligente/smartwatch. Este dispositivo, además de recibir información, está conectado a internet, siendo capaz de subir los datos a un servidor. En este caso se empleará un servidor en la nube propuesto por el fabricante de la placa de desarrollo, sin embargo, es posible mandarlo a un servidor privado respetando así las normativas en materia de seguridad de datos personales.

Partiendo desde el principio y suponiendo que no se tiene ningún programa previamente instalado se comienza con el procedimiento de implantación del protocolo:

- 1) Descargar I-CUBE_LRWAN de la página: <https://www.st.com/en/embedded-software/i-cube-lrwan.html#get-software> Se trata de la expansión para el IDE Cube de STMicroelectronics del protocolo LoRa.

- 6) Tras crearla debe añadirse un “nuevo Gateway”, en este caso elegimos un dispositivo “ST P-NUCLEO-LRWAN 2 and 3” e introducimos la dirección MAC que teníamos en la consola (correspondiente a “MACADDR”)
- 7) Una vez creada la conexión, aparecerá como Disconnected, primero se tendrá que seleccionar la región: en este caso EU y la frecuencia, 433. Ver Figura 13

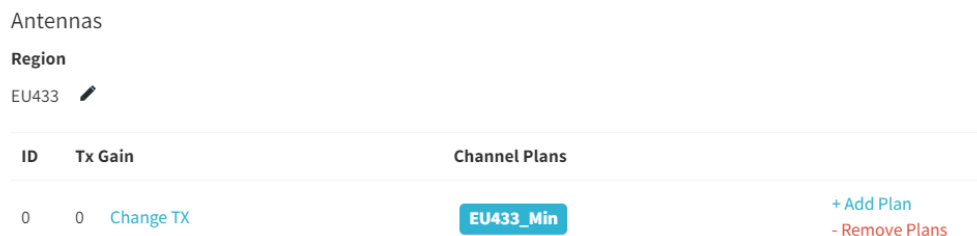


Figura 13: Web Lorient. Selección de la Región

- 8) Ahora, entro de la consola, se cambian estos dos parámetros. Ver Figura 14

```
AT+CH=EU433
AT+PKTFWD=eu3.loriot.io, puerto (1780)
```

```
AT+CH=EU433
+CH: EU433
+CH: 0, 433175000, A, SF7/SF12, BW125KHz (LORA_MULTI_SF)
+CH: 1, 433375000, A, SF7/SF12, BW125KHz (LORA_MULTI_SF)
+CH: 2, 433575000, A, SF7/SF12, BW125KHz (LORA_MULTI_SF)
+CH: 3, 433775000, A, SF7/SF12, BW125KHz (LORA_MULTI_SF)
+CH: 4, 433975000, B, SF7/SF12, BW125KHz (LORA_MULTI_SF)
+CH: 5, 434175000, B, SF7/SF12, BW125KHz (LORA_MULTI_SF)
+CH: 6, 434375000, B, SF7/SF12, BW125KHz (LORA_MULTI_SF)
+CH: 7, 434575000, B, SF7/SF12, BW125KHz (LORA_MULTI_SF)
+CH: 8, OFF (LORA_STANDARD)
+CH: 9, OFF (FSK)
AT+PKTFWD=eu3.loriot.io, 1780
+PKTFWD: eu3.loriot.io, 1780, 1780
[]
```

Figura 14: Consola LoRaWAN. Cambio de parámetros

- 9) Por último, para conectar el Gateway a la red desplegada, se tiene que habilitar el uso de ethernet desde el ordenador a la placa. Para ello se selecciona el menú ejecutar de Windows y se introduce “NCPA.CPL”, lo cual llevará al menú de “conexiones de red”. Allí se selecciona click derecho en “Wifi” → “Propiedades” → “Uso compartido” y se habilita la opción de “Permitir que los usuarios de otras redes se conecten a través de la conexión a Internet de este equipo”. Ver Figura 15

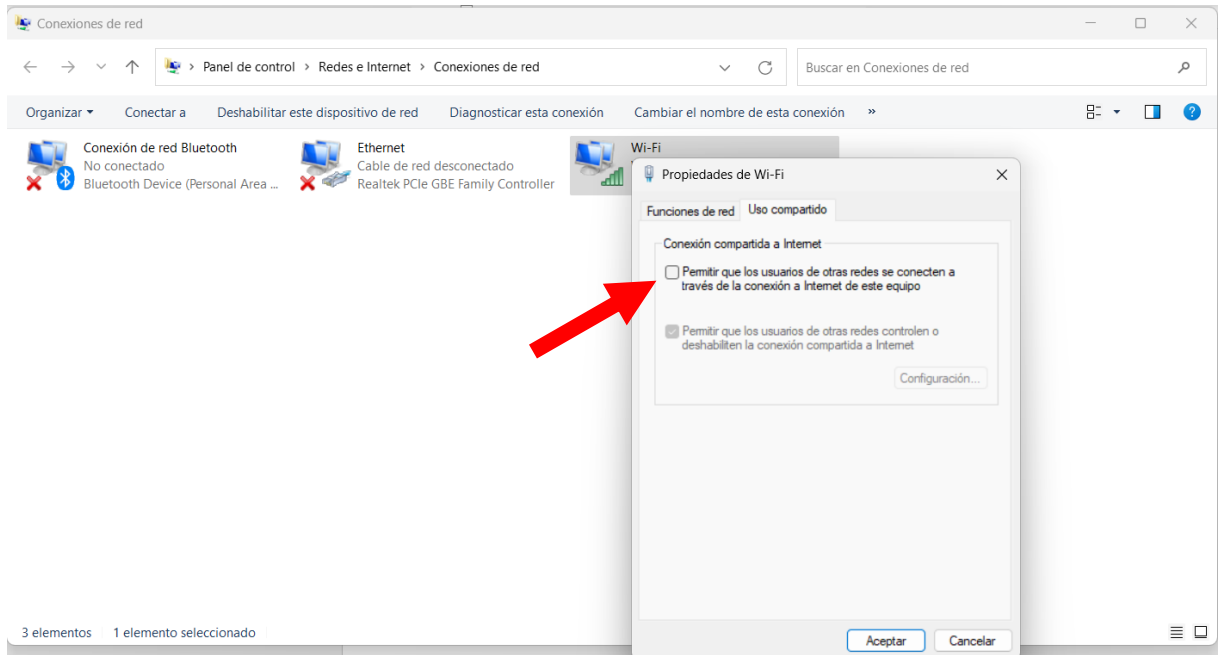


Figura 15: Permitir uso compartido de la Red

10) Al reiniciar la placa aparecerá por consola el mensaje del paso 4) añadiendo al final la IP y la UPD. Ver Figura 16
 También se puede ver como el LoRaWAN Server ha cambiado de dirección. Al refrescar la página de Lorient aparecerá el dispositivo conectado como puede observarse en la Figura 17.

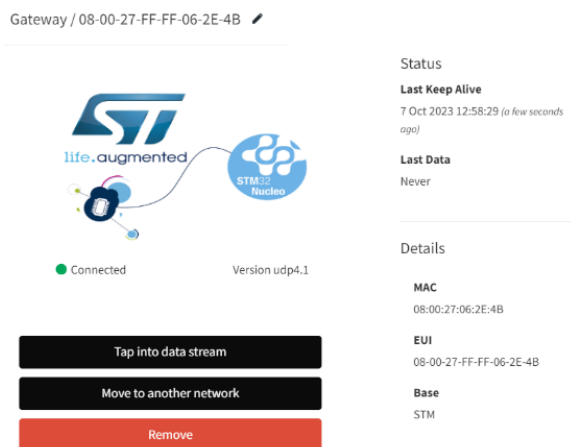


Figura 16: Interfaz Lorient con el dispositivo Gateway conectado

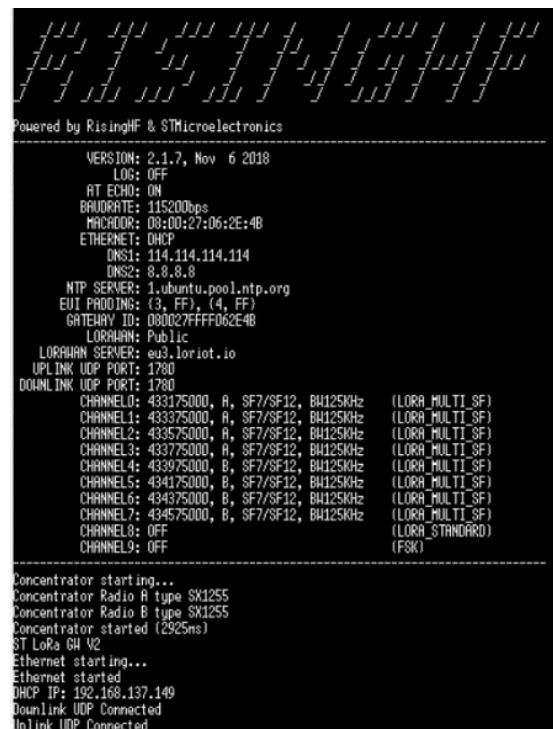


Figura 17: Consola LoRaWAN STMMicroelectronics con parámetros modificados

11) A partir de aquí se pasa a la configuración del Nodo. En primer lugar habrá que descargar el STM32CubeMX y una vez abierto importar la placa a emplear (en este caso L073RZ) y en la pestaña de “STMicroelectronics” instalar también el “X-Cube-MEMS1”

12) Dentro del IDE Cube, seleccionar :

I_CUBE\STM32CubeExpansion_LRWAN_V2.1.0\Projects\NUCLEO-L073RZ\Applications\LoRaWAN\LoRaWAN_End_Node\STM32CubeIDE

13) Dentro, se seleccionará la carpeta STM32CubeID y se hace click en finalizar como aparece en la Figura 18

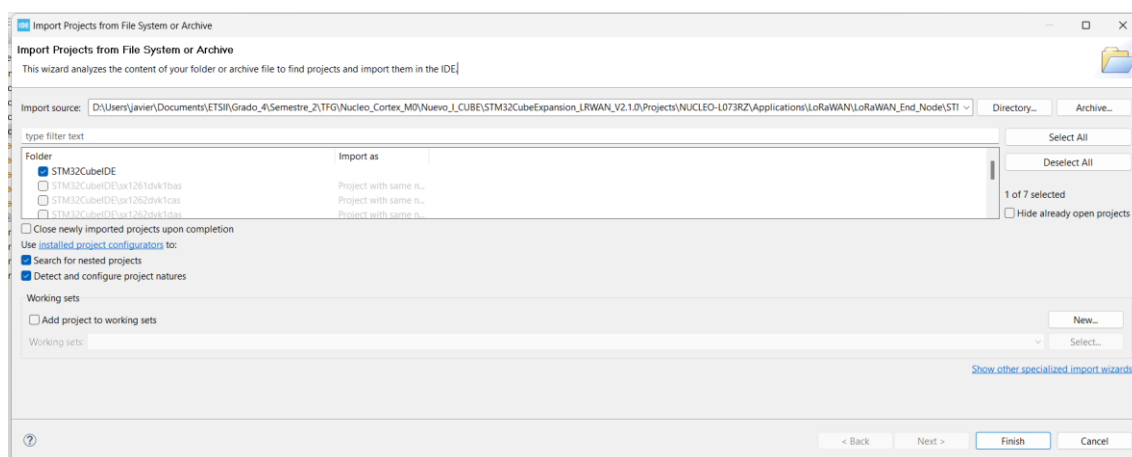


Figura 18: Selección carpeta en IDE Cube

14) Una vez hecho, hacer click en “build” y correr el código de sx1276mb1mas. Se debe tener en cuenta que puede saltar un error de GDB, se trata de un error que indica que el puerto no es accesible. Si esto sucede es posible desactivar el uso compartido del internet (realizado en el set up del Gateway) ya que interfiere con el puerto.

15) Una vez hecho con éxito, y de nuevo la terminal Tera Term con el puerto de la L073RF y 115200 baudios, al dar a Reset la placa nos dará su información como se observa en la Figura 19.

```
APP VERSION: 41.1.0
MH LoRaWAN VERSION: 42.3.0
MH RADIO VERSION: 41.1.0
##### OTRA #####
##### AppKey: 2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C
##### NukKey: 2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C
##### ABP #####
##### AppSKey: 2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C
##### NukSKey: 2B:7E:15:16:28:AE:D2:A6:AB:F7:15:88:09:CF:4F:3C
##### DevEui: 38:37:34:34:56:38:6D:01
##### AppEui: 01:01:01:01:01:01:01:01
##### DevAddr: 35:55:0F:2E
Ds061:TX on freq 868100000 Hz at DR 0
```

Figura 19: Consola LoRaWAN Node

16) Ahora se tendrá que cambiar los parámetros ya que, por ejemplo, es posible observar que la frecuencia preestablecida es de 868MHz. Habrá que cambiar:

- a. Lorawan_conf.h → Region EU433
- b. Lorawan_conf.h → HYBRID_ENABLE tiene que estar activado (1)
- c. Lora_app.h → Activate Region LORAMAC_REGION_EU433
- d. Lora_app.h → Línea 68 Lorawan_Default_class: CLASS_A
- e. Lora_app.h → Línea 90 Lorawan_Default_Activation_Type: Ativation_type_OTAA
- f. Sys_conf.h → Activate Debugger ENABLE
- g. Commissioning.h → Línea 73 Lorawan_Public_Network: TRUE
- h. Commissioning.h → Línea 78 Lorawan_Network_ID: (uint32_t)0
- i. Se-identity.h → Comprobar que tenga el valor correcto Join EUI y App Key

17) En la página de Lorient, se debe hacer click en aplicaciones → Enroll a new device y después añadir la información que aparece al iniciarse el End-node tal y como se observa en la Figura 20. Cabe destacar que el DevEUI, AppEUI y AppSKey se ponen sin los dos puntos ni espacios ya que de otro modo sale erróneo.

The screenshot shows the 'Enroll a new device' page in the Lorient web interface. The page has a dark sidebar on the left with navigation options like 'Back to Applications', 'SAMPLEAPP', 'Enroll Device', 'Bulk Import', 'Devices', 'Devices Map', 'Output', 'Output Data Format', 'Websocket Applications', 'Statistics', 'Join Server', 'Access Tokens', and 'Log'. The main content area is titled 'Enroll a new device' and contains the following form fields:

- LoRaWAN[®] Version:** A dropdown menu with 'LoRaWAN[®] 1.0.x' selected.
- Enrollment Process:** A dropdown menu with 'OTAA' selected.
- Location:** A toggle switch currently set to 'DISABLED'. Below it, a note says: 'You can define coordinates for static devices enabling this option.'
- Details:**
 - Title:** Text input field containing 'STM32L0'.
 - Device EUI:** Text input field containing '3837343456386D01'.
 - Join EUI:** Text input field containing '0101010101010101'.
 - Description:** A large text area for additional information.
 - Application Key:** Text input field containing '2B7E151628AED2A6ABF7159809CF4F3C'.
 - Device Template:** An empty text input field.
- Buttons:** At the bottom right, there is a checkbox for 'Create Another', an 'Enroll' button, and a 'Reset' button.

Figura 20: Interfaz Lorient. Enroll a new device

- 18) Si sigue fallando, revisar la versión de LoRaWAN, en este caso la versión es la 1.1 como se observa al inicio del mensaje que aparece por terminal al conectar el End-Node. Ver Figura 19, Figura 21 y Figura 22

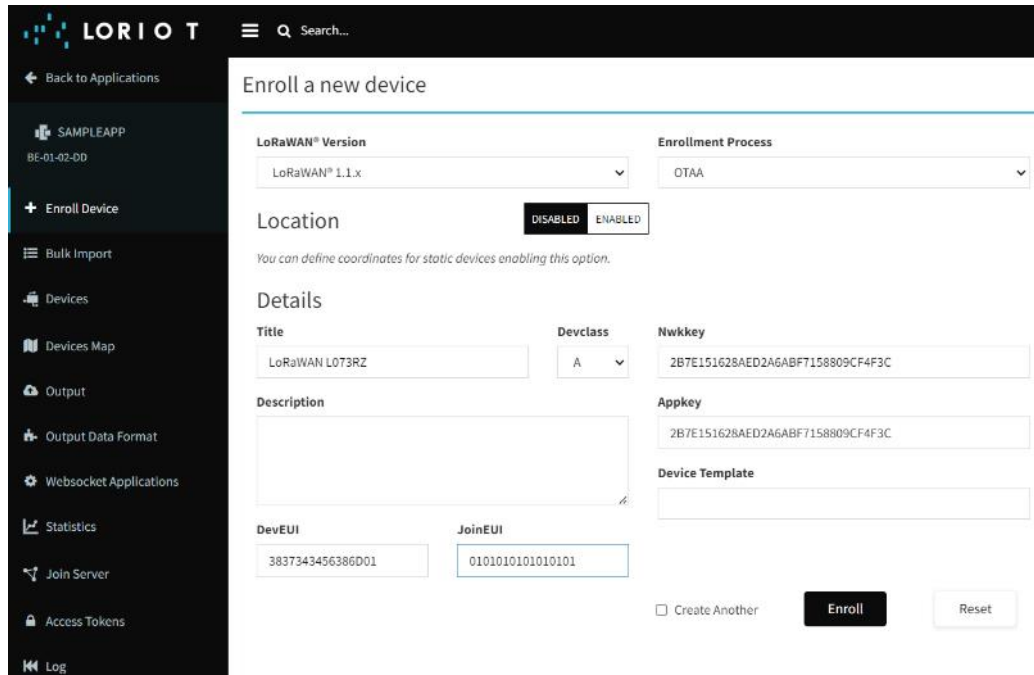


Figura 21: Interfaz Loriot. Selección de versión

```

)/*!
*****
***** WARNING *****
*****
The secure-element implementation supports both 1.0.x and 1.1.x LoRaWAN
versions of the specification.
Thus it has been decided to use the 1.1.x keys and EUI name definitions.
The below table shows the names equivalence between versions:
+-----+-----+
| 1.0.x | 1.1.x |
+-----+-----+
| LORAWAN_DEVICE_EUI | LORAWAN_DEVICE_EUI |
+-----+-----+
| LORAWAN_APP_EUI | LORAWAN_JOIN_EUI |
+-----+-----+
| LORAWAN_GEN_APP_KEY | LORAWAN_APP_KEY |
+-----+-----+
| LORAWAN_APP_KEY | LORAWAN_NWK_KEY |
+-----+-----+
| LORAWAN_NWK_S_KEY | LORAWAN_F_NWK_S_INT_KEY |
+-----+-----+
| LORAWAN_NWK_S_KEY | LORAWAN_S_NWK_S_INT_KEY |
+-----+-----+
| LORAWAN_NWK_S_KEY | LORAWAN_NWK_S_ENC_KEY |
+-----+-----+
| LORAWAN_APP_S_KEY | LORAWAN_APP_S_KEY |
+-----+-----+
*****
*****
*****

```

Figura 22: Extracto del Código proporcionado por STMicroelectronics destacando las diferentes nomenclaturas por versiones

En el Anexo 2 puede observarse otros cambios importantes realizados.

7 Experimentos

7.1 Planteamiento del problema

Una vez escogido el protocolo a emplear, el hardware y haber conseguido implementarlo, se debe realizar los experimentos necesarios para verificar, o no, el posible uso de dicho protocolo y frecuencia para el proyecto de la antena implantada en el cerebro.

Para ello, se debería poder medir la calidad de la señal recibida por la antena externa emitiendo mensajes desde la que debería estar implantada.

Con ese objetivo se plantean 3 tipos de ensayos para comprobar dicha calidad:

1. Un ensayo en vacío en el que se mida la calidad de la señal a una distancia determinada sin poner más que el aire de por medio.
2. Un segundo ensayo con cerebro simulado. Aquí se hace referencia a cualquier medio que tenga las mismas propiedades eléctricas que el cerebro ante la señal de 433MHz. En este caso emplearemos un líquido que mantiene dichas propiedades durante el tiempo necesario para realizar iteraciones. La importancia de este segundo ensayo radica en que permitiría iterar sin necesidad de comprar nuevos cerebros animales, los cuales duran poco tiempo en buen estado.
3. El tercer y último ensayo se realizará con cerebros reales, en este caso se ha escogido el cerebro de cerdo debido a la similitud con el humano en cuanto a propiedades resistivas [23].

Hay que destacar también que las condiciones atmosféricas deben ser lo más similares posible y las distancias definidas, en este caso a 1 metro para poder comparar entre ensayos aunque en el caso del tercer ensayo se estudiará el comportamiento de la calidad a distancias de 1 y 2 metros. Estas distancias fueron escogidas ya que en caso de estar implantada en el paciente, solo se transmitirá desde la cabeza hasta el brazo o teléfono móvil del mismo, lo que no es más de un metro de distancia. El uso de la distancia de 2 metros se realizará para comprobar la escalabilidad a otras distancias siempre y cuando sean relativamente pequeñas.

7.1.1 Herramientas software

Para captar los datos se ha empleado el portal de Lorient, en el que es posible crear una cuenta de forma gratuita capaz de desplegar una aplicación. Se entiende aplicación como el uso de un Gateway subiendo información a la nube de Lorient a través de internet.

Una vez creada la cuenta puede generarse la nueva aplicación dentro de la ventana de Dashboard como se puede observar en la Figura 23.

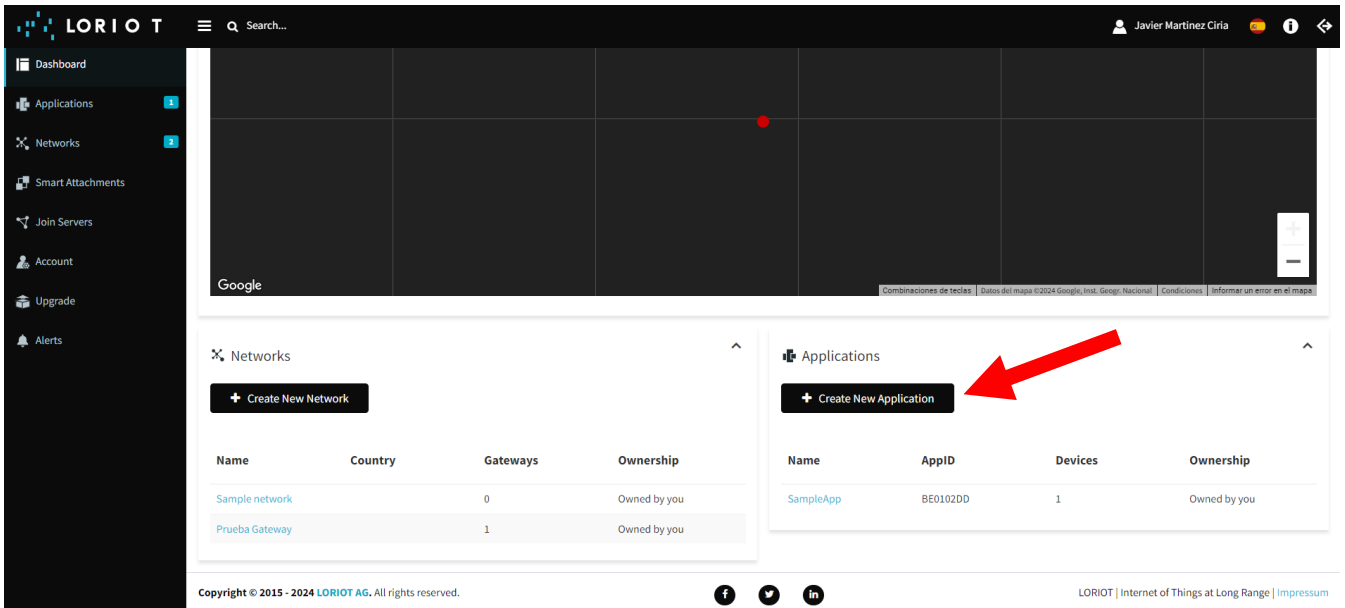


Figura 23: Interfaz Lorient. Creación de una nueva aplicación

Indicando el nombre de la aplicación y el número de dispositivos enganchados al Gateway (en la versión gratuita disponemos de hasta 10 nodos) es posible crear la aplicación. Ver Figura 24.

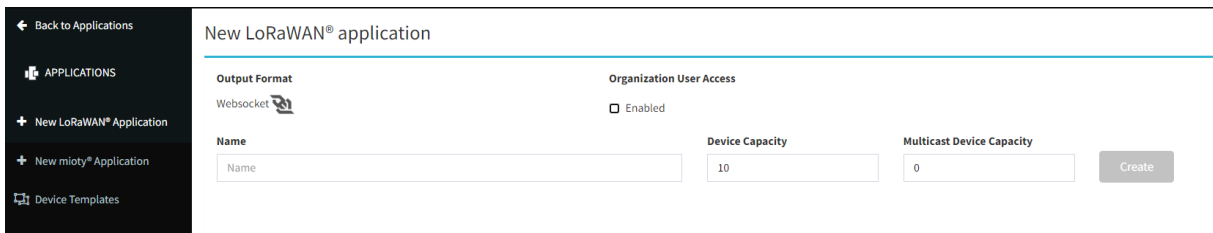


Figura 24: Interfaz Lorient. Configuración de una nueva aplicación

Una vez creada la aplicación, es posible enganchar los nodos haciendo click en “Enroll Device”, en el menú de la izquierda. Ver Figura 25.



En el menú de “Enroll Device” se observa que permite introducir en primer lugar la versión. Es importante conocer la versión de LoRa que emplea la placa de desarrollo, ya que de indicar otra versión no funcionará. En el Punto 6.2, en el apartado 18) aparece como conocer la versión y la relevancia de esta.

Además de la versión, se debe introducir el resto de parámetros que localizan e identifican el nodo como se observa en la Figura 26.

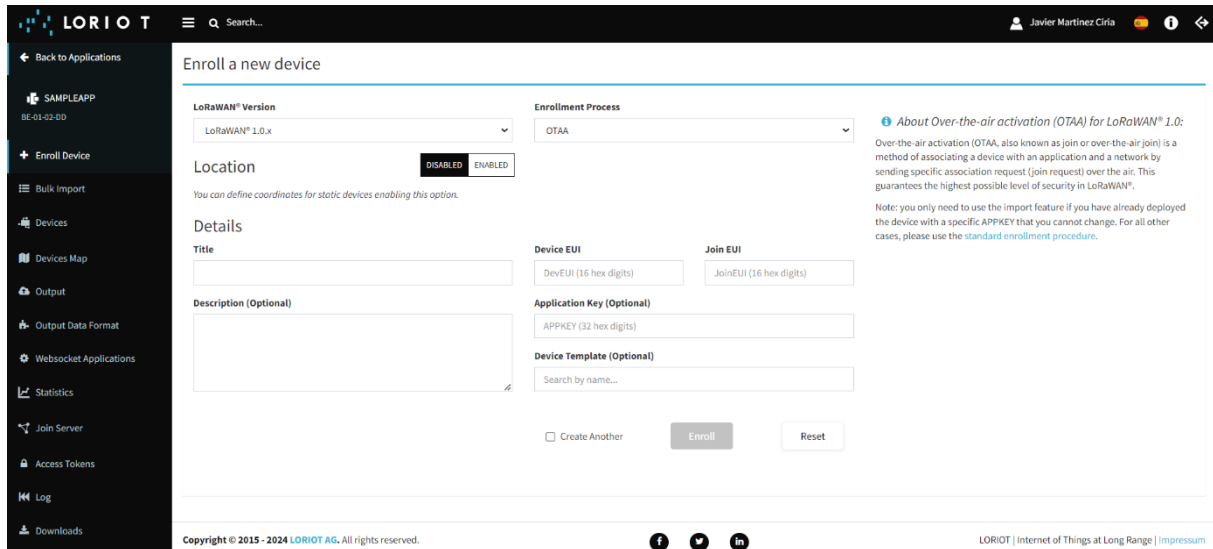


Figura 26: Interfaz Lorient. Configuración de un nuevo Nodo

Una vez añadido el nuevo nodo, es posible acceder a él en el apartado “Devices” del menú de la izquierda de la Figura 26. Ahí podemos visualizar distinta información como los paquetes enviados, estadísticas de estos envíos o localización entre otros. Ver Figura 27.

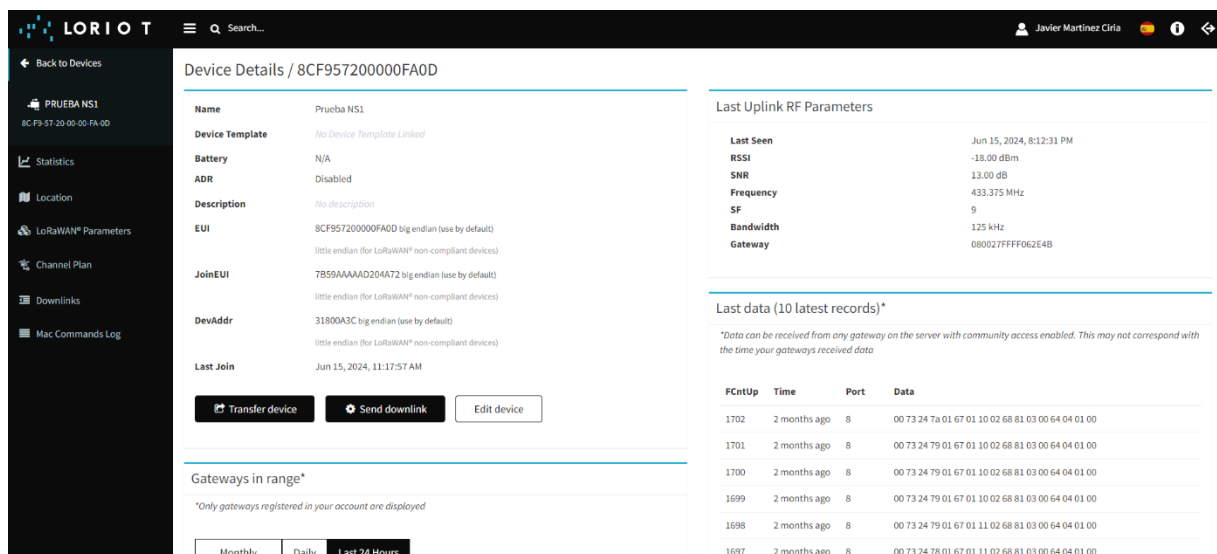


Figura 27: Interfaz Lorient. Información del Nodo

Ya por último, para poder observar los paquetes recibidos por el Gateway, debemos acceder al apartado de “WebSocket Applications” dentro del menú izquierdo de la aplicación, el cual aparece en la Figura 26. Una vez ahí, se debe acceder al WebSocket sample by Lorient como se observa en la Figura 28.

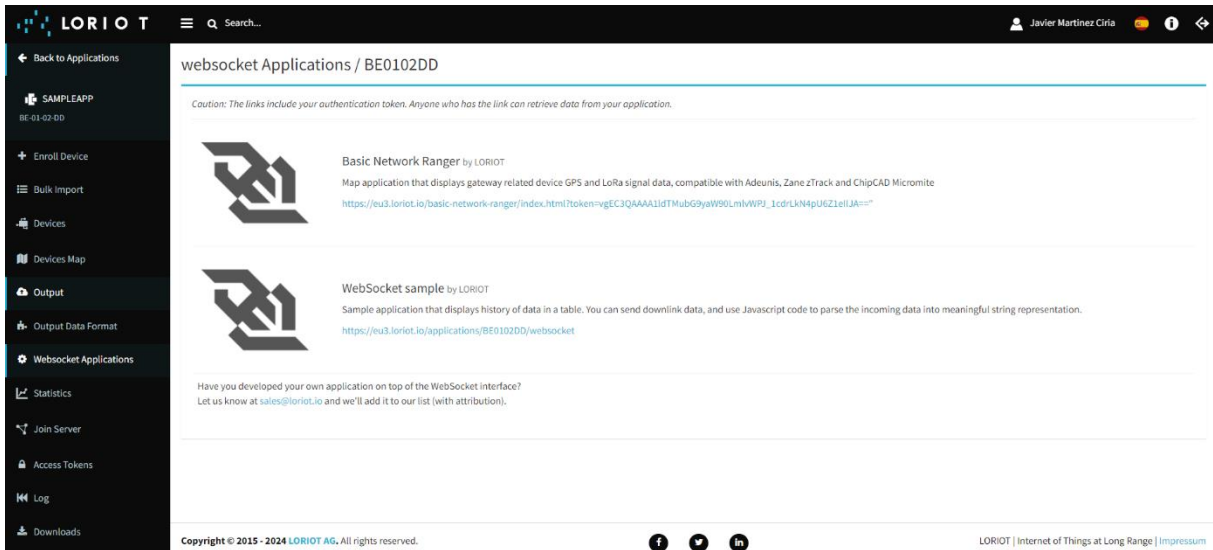


Figura 28: Interfaz Lorient. Menú websocket applications

Una vez ahí, es posible acceder a las muestras que obtengamos en los ensayos. Ver Figura 29.

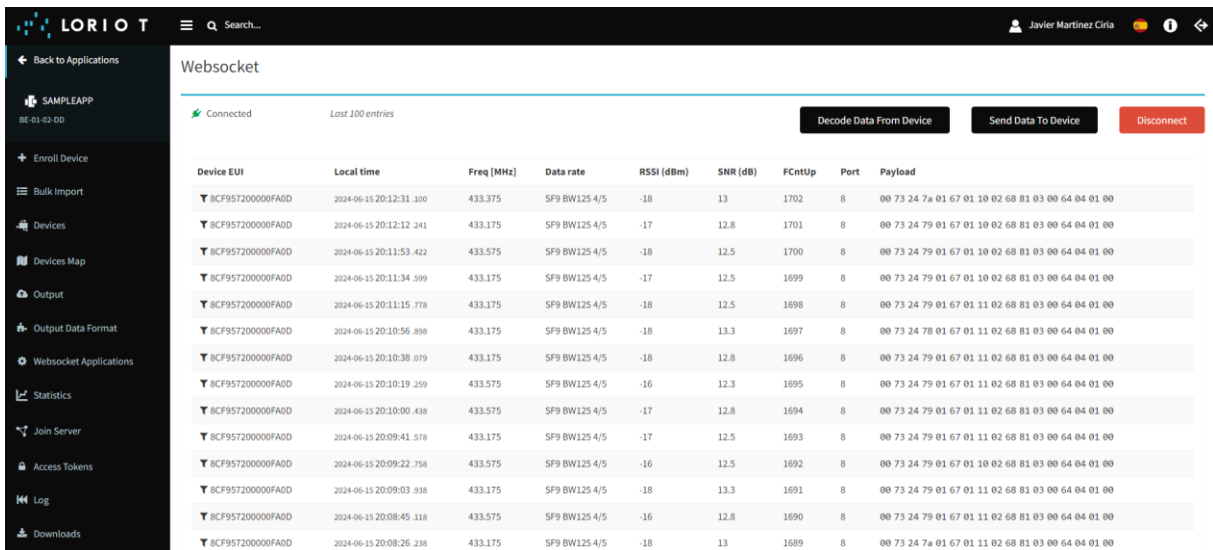


Figura 29: Interfaz Lorient. Muestreo de los datos recibidos en la comunicación entre dispositivos.

Es posible extraer el contenido de los datos en otro apartado de la interfaz. Sin embargo, para este tipo de ensayos, lo que necesitamos obtener es la calidad de la señal recibida ya que es el principal motivo de este trabajo, ser capaces de enviar datos de un punto a otro sin perder señal.

Cabe destacar que los datos en la versión gratuita se almacenan en grupos de 100, por lo que si no se copian los datos a tiempo es posible que se sobrescriban. Es por esto por lo que se almacenarán los datos en una hoja de cálculo Excel.

7.1.2 Análisis estadístico

El análisis estadístico realizado con los datos ha sido exclusivamente el cálculo de media y varianza. Otro tipo de análisis estadísticos como la extrapolación de las distancias de la antena, no son de gran utilidad para esta parte del proyecto, ya que en los ensayos se ha podido medir los resultados a distancias más que suficientes para el empleo de esta tecnología el caso deseado de antena implantada en el cráneo del paciente y receptor en un teléfono móvil o pulsera inteligente/smartwatch.

7.2 Ensayos

A continuación se mostrará el montaje de los distintos ensayos realizados.

7.2.1 Ensayo en vacío

Para realizar este primer ensayo emplearemos ambas antenas, nodo y Gateway, un recipiente de vidrio y una bolsa de plástico. Es importante realizarlo con recipiente de vidrio y bolsa de plástico ya que el propósito de este ensayo es comprobar la transmisibilidad por el cerebro, al tener que ponerlo en los cerebros en un futuro y también en el medio acuoso debemos proteger la antena en una bolsa y por eso mismo el ensayo en vacío debe considerar esas posibles pérdidas. Ver Figura 30 y Figura 31.



Figura 30: Nodo en bolsa de plástico



Figura 31: Nodo en bolsa de plástico y recipiente de vidrio

Ahora se debe medir la distancia a la que colocar la segunda antena, en este caso el Gateway tal y como se observa en la Figura 32. También mencionar que las conexiones necesarias son desde las antenas a la red y desde el Gateway al router mediante un cable Ethernet.



Figura 32: Ensayo en vacío

7.2.2 Ensayo con cerebro simulado

El montaje de este ensayo es muy similar al primero, ya que únicamente cuenta con el medio acuoso que simula las propiedades del cerebro.

Este líquido fue proporcionado por el IMDEA teniendo en cuenta distintas características relevantes:

- El líquido cefalorraquídeo contiene distintos iones que desempeñan un papel crucial en la conductividad del cerebro. Es importante ser capaces de sintetizar dichos iones mediante sales para poder reproducir estas interacciones.
- El pH del líquido también debe ser lo más parecido posible al que se encuentra en el cerebro humano. Esto es entre un 7,33 y un 7,45.
- La disolución debe ser estéril para impedir la aparición de bacterias en ella ya que podría llegar a alterar los resultados.

Es por ello por lo que los reactivos a emplear en la preparación de la disolución son los siguientes:

- ~ Agua destilada o desionizada: 1 litro
- ~ Cloruro de sodio (NaCl): 124 mM (aproximadamente 7.3 g/L)
- ~ Cloruro de potasio (KCl): 2.5 mM (aproximadamente 0.2 g/L)
- ~ Cloruro de calcio (CaCl₂): 2.4 mM (aproximadamente 0.2 g/L)
- ~ Glucosa: 3.5 mM (aproximadamente 0.2 g/L)
- ~ Bicarbonato de sodio (NaHCO₃): 25 mM (aproximadamente 1.5 g/L)
- ~ Proteínas (BSA - Albumina Serica Bovina): 0.6-1.0 g/L

Y además, para controlar el pH de la disolución se debe añadir la cantidad necesaria de ácido clorhídrico (HCl) o hidróxido de sodio (NaOH).

En la Figura 33 se puede observar cómo se sumerge la antena en el interior del líquido.



Figura 33: Ensayo en cerebro simulado. Antena sumergida en medio acuoso con las propiedades del cerebro

7.2.3 Ensayo con cerebro real

Para este último y definitivo ensayo se ha empleado, como se mencionó previamente, cerebros de cerdo. El montaje vuelve a ser similar a los anteriores ya que de ello se trata, diferenciándose en el uso de cerebros en vez de líquido acuoso como se observa en la Figura 34.

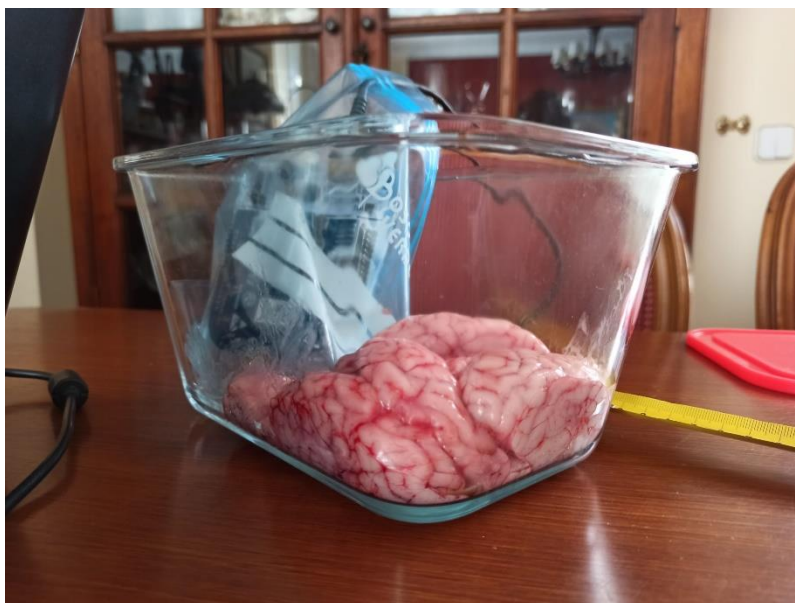


Figura 34: Ensayo en cerebro real

Para este ensayo, como ya se comentó, se emplearán dos distancias distintas de uno y dos metros y dos orientaciones, 0° y 90° . Ver Figura 35, Figura 36 y Figura 37.



Figura 37: Ensayo en cerebro real. Un metro de distancia



Figura 36: Ensayo en cerebro real. Dos metros de distancia



Figura 35: Ensayo en cerebro real. A 90° respecto el ensayo inicial

8 Resultados

Para evaluar los resultados se deberá prestar atención a los datos obtenidos de RSSI y SNR.

- **RSSI:** Received Signal Strength Indicator, es un indicador de la intensidad de la señal recibida. Se utiliza para medir el nivel de potencia de las señales recibidas por un dispositivo en redes inalámbricas. Es por eso que es una medida de la intensidad de la señal [24].
- **SNR:** Signal-to-Noise Ratio, es una medida que compara el nivel de una señal deseada con el nivel del ruido de fondo. Se expresa en decibelios (dB) y se utiliza para evaluar la calidad de una señal en diversas aplicaciones, como audio, comunicaciones y medicina.
Un SNR alto indica que la señal es mucho más fuerte que el ruido, lo que resulta en una mejor calidad de la señal. Por el contrario, un SNR bajo sugiere que el ruido es comparable o incluso mayor que la señal, lo que puede dificultar la interpretación o el uso de la señal [25].

A continuación, se detallan los resultados obtenidos en cada uno de los ensayos

8.1 Ensayo en vacío

En primer lugar se mostrarán los resultados del RSSI en el Figura 38.

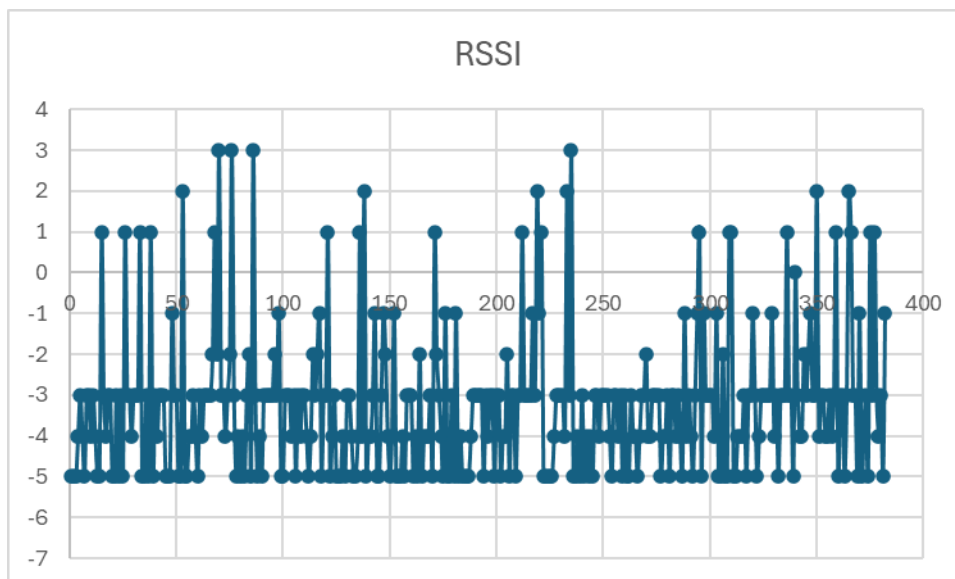


Figura 38: Resultados RSSI en Ensayo en vacío

Media	-3,2513089
Varianza	2,98919986

Con estos resultados es posible observar que la intensidad es excelente según los rangos del apartado 3.4.1 aunque es cierto que es inestable. Se puede deber a distintas causas, y se tendrá en cuenta tanto para la posterior comparativa con los otros dos ensayos como a la hora de continuar con el proyecto.

A continuación se muestran los resultados del SNR en la Figura 39.

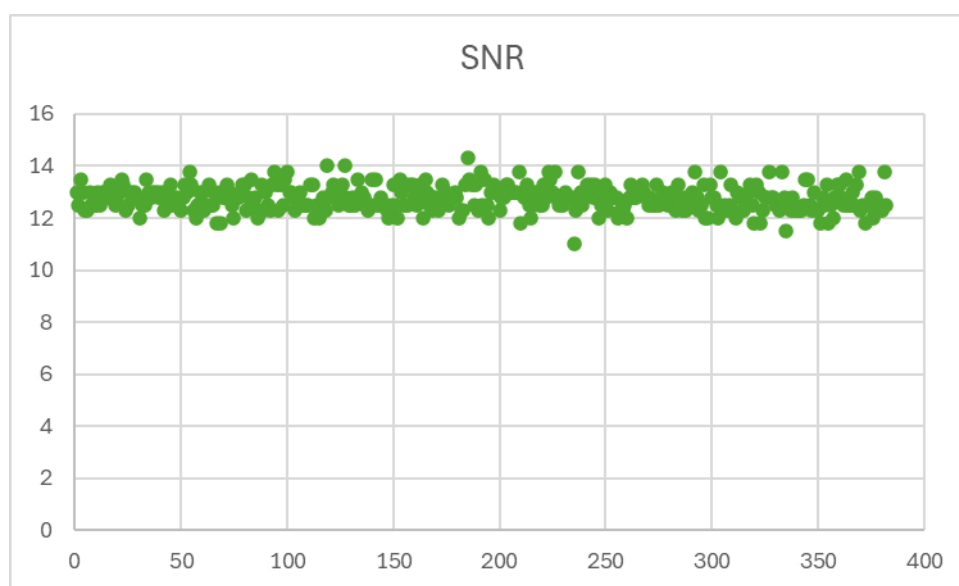


Figura 39: Resultados SNR en Ensayo en vacío

Media	12,7704188
Varianza	0,22823027

En este caso es posible observar que el ruido que aparece es relativamente elevado, esto puede deberse también a múltiples factores a evaluar en un futuro. Como en este caso el objetivo es establecer una base contra la que comparar los datos, puede darse por válido. Respecto a la varianza hay que destacar su gran estabilidad.

8.2 Ensayo en cerebro simulado

En primer lugar se mostrarán los resultados del RSSI en el Figura 40.

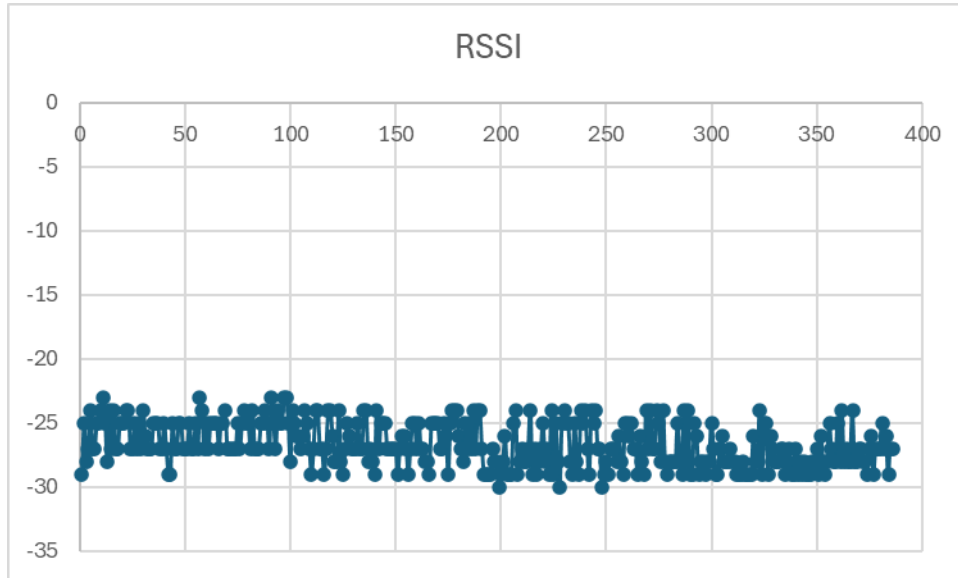


Figura 40: Resultado RSSI en Ensayo en cerebro simulado

Media	-26,6632124
Varianza	2,88139279

Se observa que la intensidad ha empeorado ligeramente, mejorando de forma considerable la varianza.

Dentro de lo que cabe, se trata de un resultado muy bueno, ya que la diferencia entre el ensayo en vacío y este es de tan solo 14 dBm y sigue entrando en el rango de señal excelente.

Añadir que esto no quiere decir que el anterior ensayo fuese peor, simplemente que no ha tenido esas pequeñas discontinuidades que provocaron la excesiva varianza.

A continuación se muestran los resultados del SNR en la Figura 41.

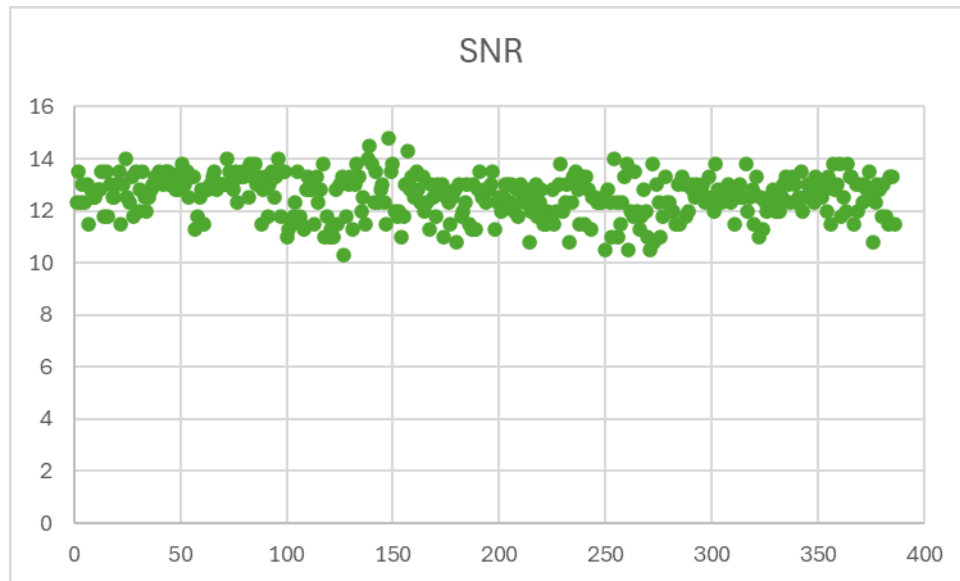


Figura 41: Resultados SNR en Ensayo en cerebro simulado

Media	12,5569948
Varianza	0,62115874

Se observa unos resultados de ruido y calidad muy similares a los encontrados en el caso de vacío, por lo que de comportarse como en el caso del cerebro real estos resultados serían considerados muy positivamente.

8.3 Ensayo en cerebro real

A continuación se mostrarán los resultados de cada variación con la antena introducida en los cerebros animales.

Las conclusiones de estos ensayos resultan ser muy positivas como se observa a continuación:

8.3.1 Ensayo a 1 metro de distancia

En primer lugar se muestra en la Figura 42 los resultados de RSSI.

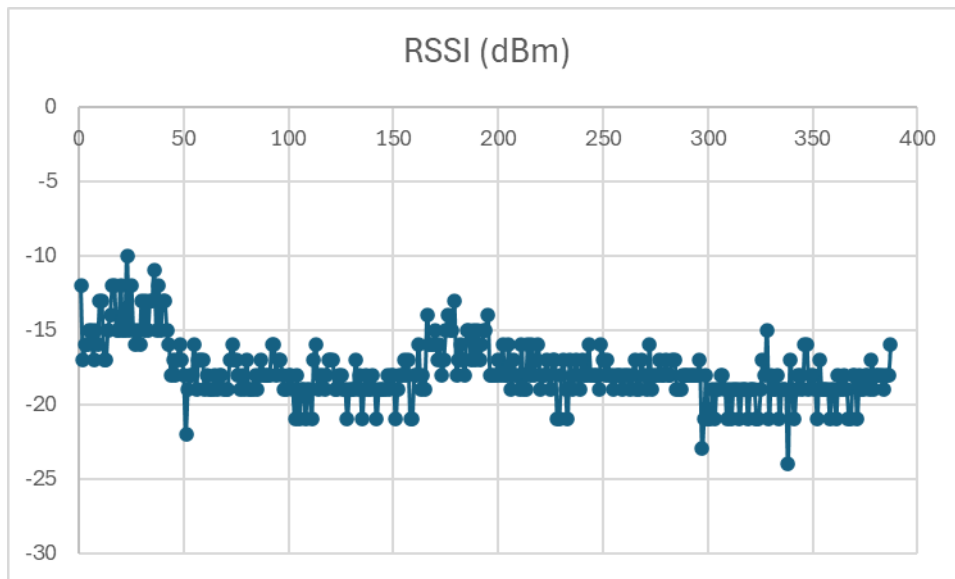


Figura 42: Resultados RSSI en Ensayo en cerebro real a 1 metro de distancia

Media	-17,7571059
Varianza	3,8376433

Se puede ver de nuevo como los resultados, aunque con cierta varianza, son muy positivos al contar con una señal de intensidad Excelente, ya que es mayor a -40 dBm.

A continuación se muestran los resultados del SNR en la Figura 43.

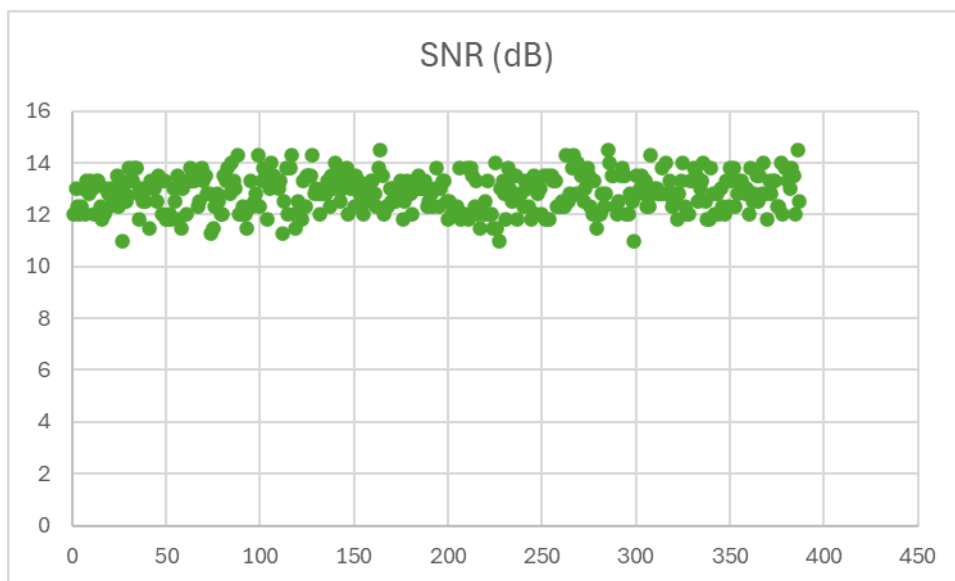


Figura 43: Resultados SNR en Ensayo en cerebro real a 1 metro de distancia

Media	12,8330749
Varianza	0,50553534

Se observa cómo, al igual que en el caso anterior, el efecto del ruido y la calidad es muy similar al ensayo en vacío.

8.3.2 Ensayo a 2 metros de distancia

En primer lugar se muestra en la Figura 44 los resultados de RSSI.

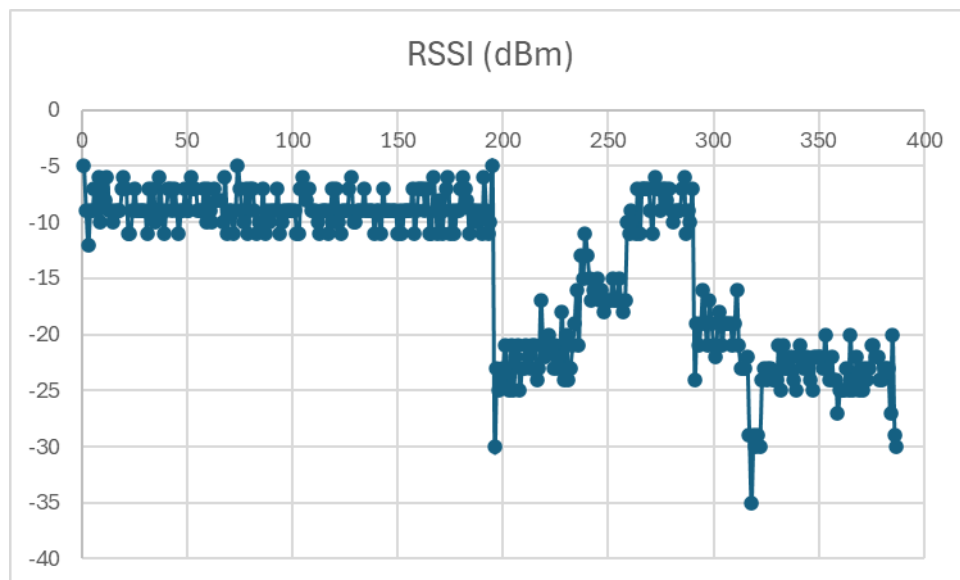


Figura 44: Resultados RSSI. Ensayo en cerebro real a 2 metros de distancia

Media	-14,0878553
Varianza	47,9561057

Se observa como la media de la intensidad de señal sigue siendo excelente a pesar de contar con una varianza muy elevada.

A continuación se muestran los resultados del SNR en la Figura 45.

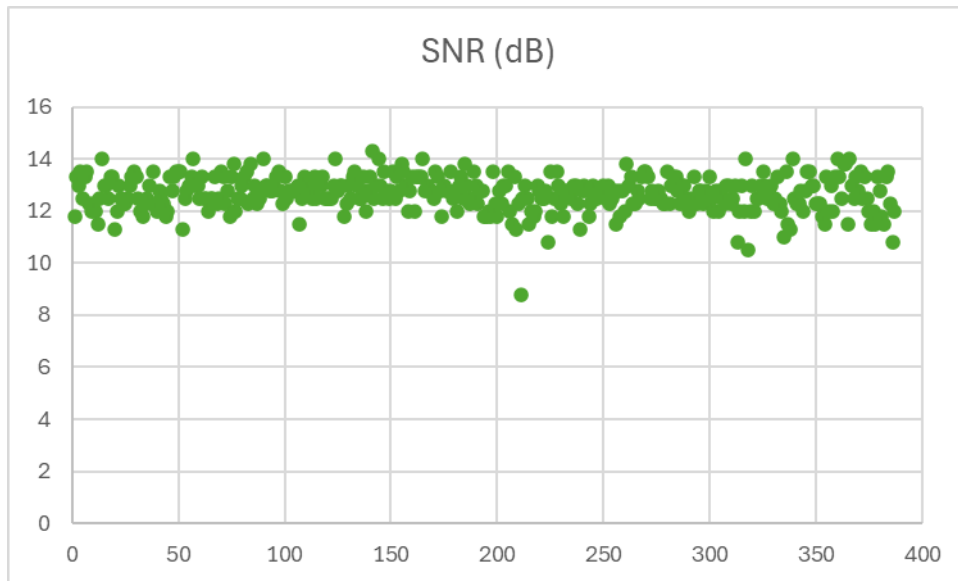


Figura 45: Resultados SNR. Ensayo en cerebro real a 2 metros de distancia

Media	12,6883721
Varianza	0,43222195

Se observa como los resultados de SNR permanecen igual al resto de ensayos anteriores

8.3.3 Ensayo a 90°

Se exponen los resultados de RSSI del último ensayo realizado. Ver Figura 46.

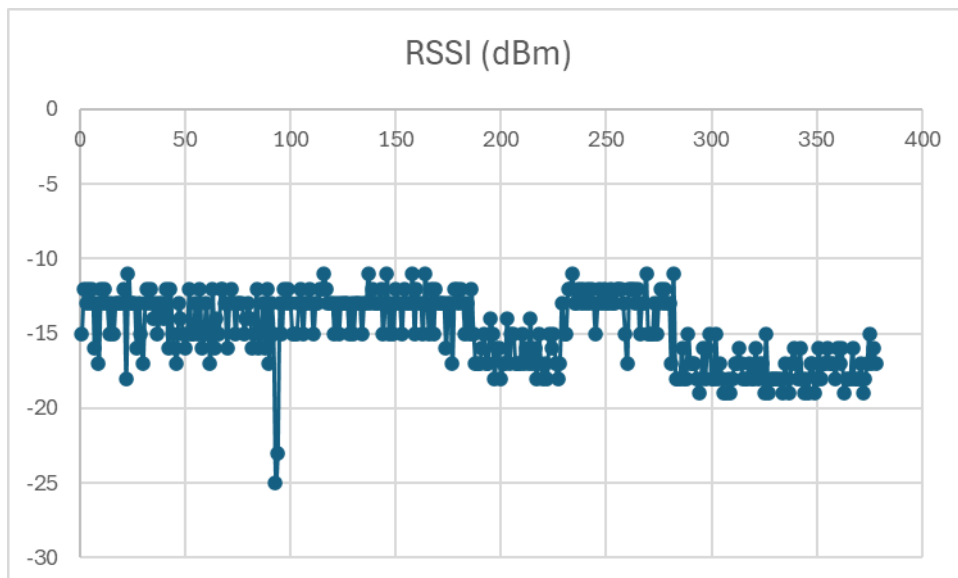


Figura 46: Resultados RSSI. Ensayo en cerebro real a 2 metros y 90°

Media	-14,8386243
Varianza	5,43692086

En este último ensayo se observa una media muy similar al caso de orientación 0° pero con una mucha menor varianza. La varianza del caso previo lo más probable es que fuese alguna alteración del entorno de ensayo.

A continuación se muestran los resultados del SNR en la Figura 47.

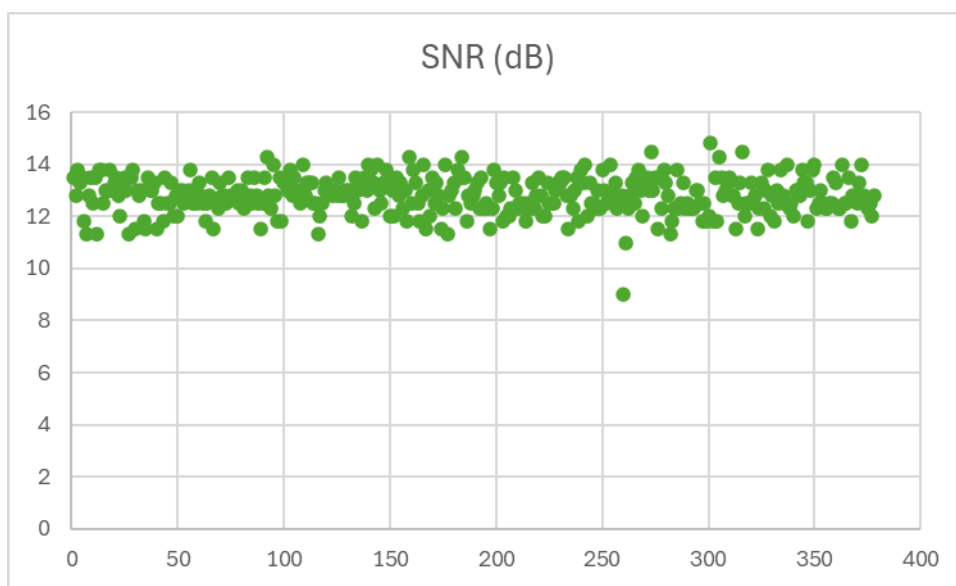


Figura 47: Resultados SNR. Ensayo en cerebro real a 2 metros y 90°

Media	12,805291
Varianza	0,49838042

De nuevo observamos que los resultados son prácticamente iguales a los ensayos anteriores. Ver Tabla 3.

	Vacío	Medio acuoso	Cerebro Real
RSSI	-3,251308901	-26,66321244	-17,75710594
SNR	12,77041885	12,55699482	12,83307494

Tabla 3: Resultados Ensayos

9 Conclusiones y líneas futuras

9.1 Conclusiones

En este apartado se pretende comentar los resultados obtenidos y revisar los objetivos planteados para este Trabajo de Fin de Grado y el proyecto en el que se engloba.

De entre las gráficas obtenidas se quiere remarcar en primer lugar el caso de ensayo en vacío y el caso de ensayo en cerebro real a una distancia de un metro.

En ellos se observa como la intensidad de la señal no se ve prácticamente afectada por el paso a través de los cerebros, lo que es todo un éxito ya que se cumple con el objetivo principal de este Trabajo de Fin de Grado. Aunque es cierto que se aprecia una pequeña pérdida de intensidad, esta no es lo suficientemente relevante ya que sigue en el rango de señal excelente como se comentó en el Punto 3.4.1.

Respecto al análisis de la distancia y orientación, comparando los ensayos con el cerebro real a 1 metro de distancia y a 2, es posible confirmar que la distancia apenas tiene relevancia en la intensidad más allá de una mayor variación que se observa como desaparece en el ensayo a 90° y 2 metros de distancia. Esto no indica que se pueda alejar indefinidamente, solamente que a distancias cortas, como las que se darían en un uso real del módulo de comunicación implantado en el cerebro del paciente, la intensidad de la señal no se ve afectada prácticamente.

Si analizamos la orientación es fácil ver como esta no afecta prácticamente a la intensidad.

Respecto al SNR, se observa como sí aparece un ruido más o menos elevado que puede afectar negativamente a la comunicación. Este ruido se observa en todo tipo de ensayos por lo que no ha sido posible determinar que se deba a algo en particular como puede ser el ambiente, la bolsa de plástico, el recipiente o incluso la propia antena.

Por tanto, a raíz de los resultados obtenidos en los ensayos, es posible concluir que:

- El análisis previo de características fisiológicas y comportamiento frente a ondas electromagnéticas del cerebro ha sido el adecuado para poder tomar una elección correcta desde el inicio del proyecto.
- El hardware empleado ha permitido realizar los ensayos e implementar el protocolo correctamente.
- El protocolo y frecuencia empleada son los correctos para poder atravesar el cerebro del paciente sin perder la intensidad de la misma como se ha podido comprobar mediante los experimentos realizados.

9.2 Líneas Futuras

De cara a continuar este trabajo, los puntos principales a tratar serían los siguientes:

- Diseño de una placa miniaturizada, la cual permita ser implantable en el cerebro del paciente de acuerdo con las dimensiones de diseño.
- Análisis de la biocompatibilidad del módulo de comunicación, otro de los puntos clave de los dispositivos implantados.
- Futuros ensayos en cráneos completos, considerar el ensayo en un cuerpo humano.
- Emisión de señales de forma más personalizada, incluir información relevante del paciente en cuestión y adaptar la frecuencia de estas emisiones de datos.
- Aplicación de una capa de seguridad que permita encriptar los datos emitidos por el módulo de comunicación.
- Análisis del consumo de batería del módulo de comunicación.

9.3 Valoración de impactos y de aspectos relacionados con la responsabilidad legal, ética y profesional.

En el desarrollo de un Trabajo Fin de Grado (TFG) que implica la investigación y experimentación con tecnologías implantables en el cerebro, es fundamental considerar los impactos y las responsabilidades desde varias perspectivas: legal, ética y profesional. Este análisis es crucial para asegurar la conformidad con las normativas y principios vigentes.

9.3.1 Impactos Legales

El diseño e implementación de un módulo de comunicación implantado en el cerebro implica una serie de desafíos legales. El uso de dispositivos electrónicos dentro del cuerpo humano está sujeto a una estricta regulación, especialmente cuando se trata de sistemas implantables que interactúan con órganos tan sensibles como el cerebro. En este contexto, es esencial asegurar que todas las fases del proyecto, desde la compra de las placas de desarrollo hasta los ensayos en animales, cumplan con las normativas vigentes en materia de dispositivos médicos y

experimentación animal. En este caso, no ha sido necesario aplicar normativas ya que se trataban de cerebros comprados en comercio.

En la Unión Europea, por ejemplo, se regula según el Reglamento (UE) 2017/745 sobre productos sanitarios establece los requisitos para la comercialización y puesta en servicio de dispositivos médicos. Aunque este TFG no busca la comercialización inmediata del módulo desarrollado, es importante que el diseño del dispositivo y la documentación asociada sigan las mejores prácticas descritas en dicha regulación para facilitar futuras certificaciones si el proyecto avanzara hacia aplicaciones clínicas.

Además, y de cara a un futuro más o menos cercano, los ensayos realizados con cerebros de animales vivos, aunque no involucren seres humanos, deben cumplir con la legislación sobre bienestar animal, que exige que los experimentos en animales se realicen bajo condiciones que minimicen el sufrimiento y estén justificados por la relevancia científica.

9.3.2 Impactos Éticos

El trabajo con tecnología implantable en el cerebro presenta consideraciones éticas significativas. La posibilidad de intervenir directamente en el cerebro humano abre debates en torno a la autonomía, el consentimiento informado y los posibles efectos a largo plazo de tales intervenciones.

En el caso específico de este TFG, aunque los experimentos se han realizado en un entorno controlado y con modelos animales, es imperativo considerar las implicaciones éticas futuras si esta tecnología se trasladara a aplicaciones humanas. Debe asegurarse que cualquier implementación futura en humanos cuente con un consentimiento informado completo y que los riesgos sean comprendidos y aceptados por los participantes.

La simulación del cerebro mediante un medio acuoso y los ensayos en cerebros de cerdo son enfoques válidos para minimizar el riesgo a los seres humanos. Sin embargo, es fundamental que este enfoque ético siga siendo parte integral del proyecto en todas sus fases. La reflexión ética debe ser continua, con la consulta a comités de ética y la incorporación de principios de bioética como la beneficencia, la no maleficencia, la autonomía y la justicia.

9.3.3 Responsabilidad Profesional

Como ingeniero involucrado en el desarrollo de tecnologías que pueden tener un impacto significativo en la salud y el bienestar de los individuos, se tiene una responsabilidad profesional de adherirse a los más altos estándares de la profesión. Esto incluye la precisión y la transparencia en la presentación de resultados, la gestión responsable de los datos, y la consideración de las implicaciones a largo plazo del trabajo realizado.

En este contexto, la comunicación clara de los resultados, tanto en términos de éxitos como de limitaciones, es crucial. La sobrestimación de los beneficios o la subestimación de los riesgos podría llevar a un uso inapropiado de la tecnología o a expectativas no realistas. La honestidad y la integridad deben guiar todas las acciones, asegurando que el trabajo se realice con el máximo rigor científico y respeto por la vida.

En resumen, la valoración de los impactos legales, éticos y profesionales en este TFG destaca la necesidad de un enfoque integral que contemple no solo la viabilidad técnica del módulo de comunicación implantado en el cerebro, sino también las consecuencias y responsabilidades que su desarrollo conlleva. Este enfoque garantiza que la investigación avance de manera responsable y respetuosa, sentando las bases para futuras innovaciones que puedan mejorar la vida de las personas, siempre dentro de un marco legal y ético sólido.

10 Planificación temporal y presupuesto

En este apartado se describe la planificación del Trabajo de fin de Grado mediante el diagrama de Estructura de Descomposición del Proyecto o EDP y el diagrama Gantt. A continuación, se incluye el presupuesto con el coste de los diferentes conceptos identificados.

10.1 Estructura de Descomposición del Proyecto

La Estructura de Descomposición del Proyecto (EDP) es una herramienta fundamental en la gestión de proyectos que permite organizar y descomponer las actividades en unidades manejables. En el contexto de este Trabajo Fin de Grado, la EDP se ha desarrollado para proporcionar una visión clara y detallada de todas las tareas necesarias para la realización del proyecto, desde la investigación preliminar hasta la implementación y validación final del módulo de comunicación cerebral. Esta descomposición en componentes más pequeños facilita la asignación de recursos, la gestión del tiempo y el seguimiento del progreso, garantizando que cada etapa del proyecto esté claramente definida y pueda ser ejecutada de manera eficiente. La EDP no solo sirve como una guía para el desarrollo del proyecto, sino que también es crucial para identificar los hitos y los entregables, permitiendo un control riguroso sobre el avance del TFG. En la Tabla 4 y Figura 48 se puede ver la EDP del Trabajo de Fin de Grado

T0 - Planificación y supervisión	WP0.1 – Definición de Objetivos
T1 - Estudios previos	WP1.1 – Estado del Arte de Fisiología del Cerebro
	WP1.2 – Estado del Arte de Protocolos
	WP1.3 – Estado del Arte de Medidas de Señales
T2 – Elección del protocolo	WP2.1 – Estudio de Requisitos Proyecto
	WP2.2 – Recopilación de Opciones
	WP2.3 – Elección final del protocolo
T3 – Elección del Hardware	WP3.1 – Estudio de Requisitos Hardware
	WP3.2 – Recopilación de Opciones
	WP3.3 – Elección final del Hardware
T4 – Implementación del Protocolo	WP4.1 – Búsqueda del Software Necesario
	WP4.2 – Implementación del Protocolo
T5 – Ensayos	WP5.1 – Ensayos en vacío
	WP5.2 – Ensayos en cerebro simulado
	WP5.3 – Ensayos en cerebro real
T6 - Documentación	WP6.1 – Redacción de la memoria
	WP6.2 - Presentación

Tabla 4: EDP del Trabajo de Fin de Grado

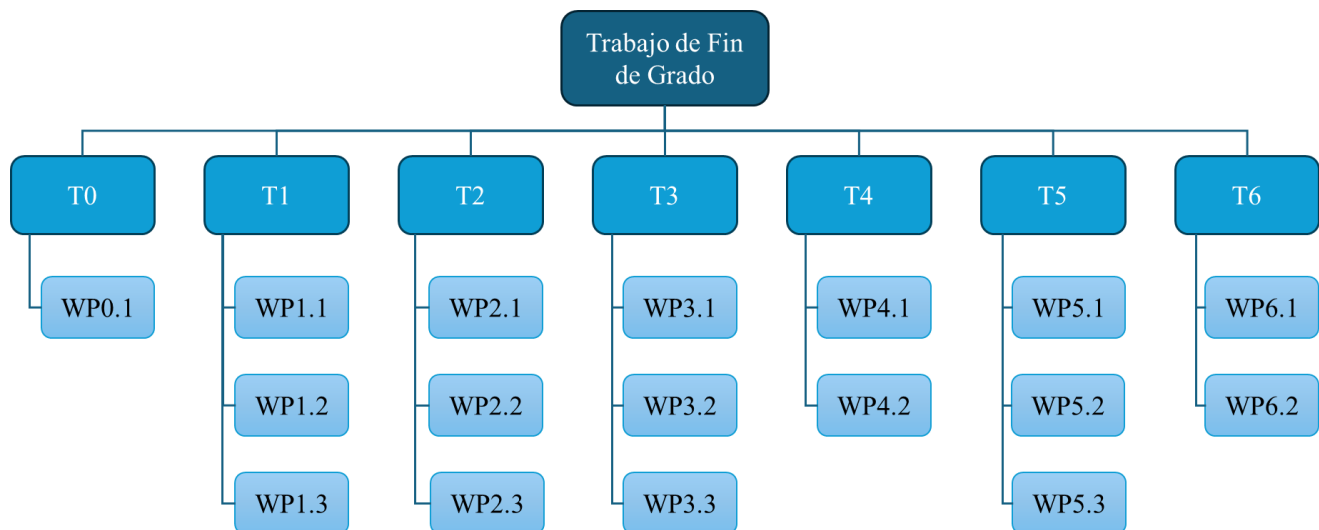
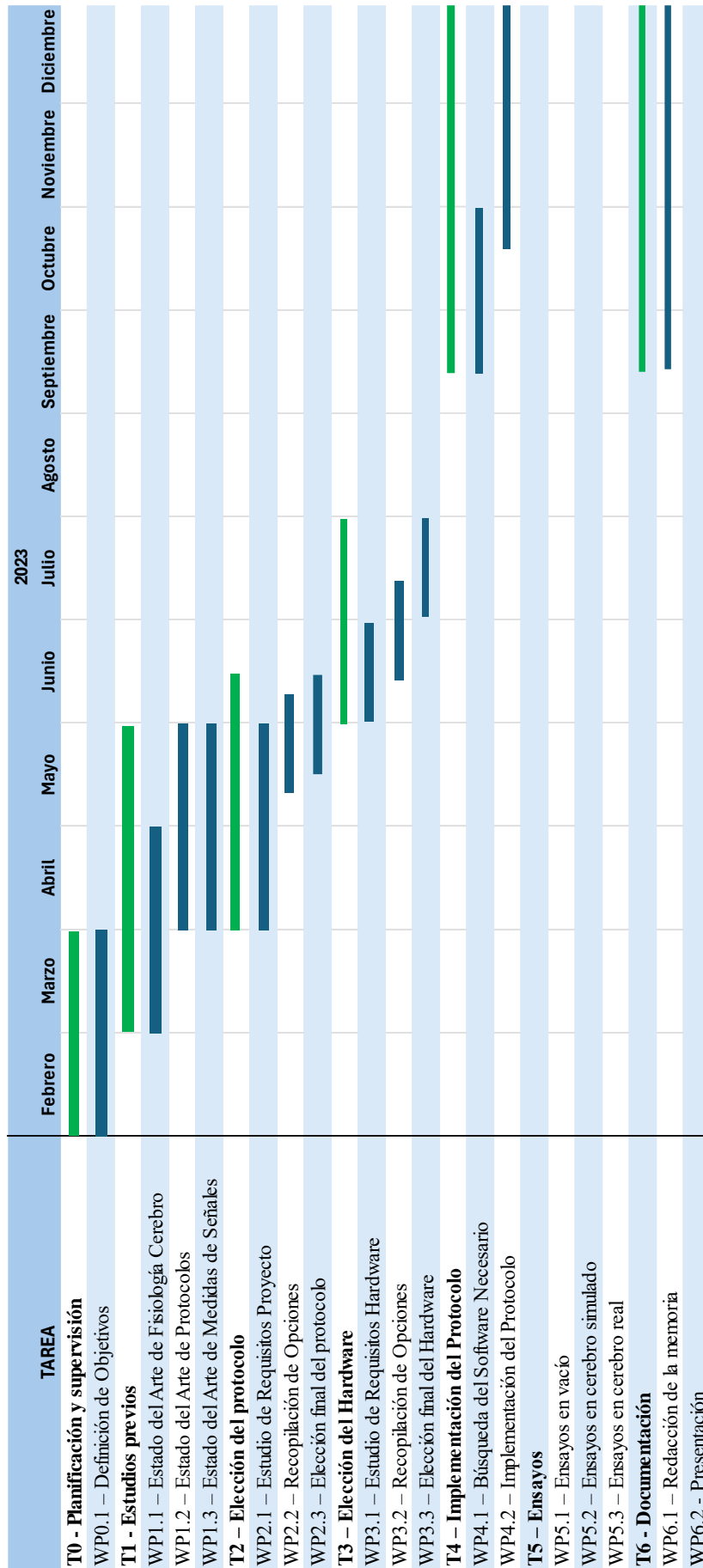


Figura 48: EDP del Trabajo de Fin de Grado

10.2 Diagrama de Gantt

El Diagrama de Gantt es una herramienta esencial para la planificación y gestión del tiempo en proyectos complejos. En este Trabajo Fin de Grado, se ha utilizado el Diagrama de Gantt para estructurar y visualizar el cronograma de todas las actividades del proyecto, asegurando que cada tarea se realice en el tiempo previsto y que se mantenga una secuencia lógica en la ejecución. Este diagrama permite identificar dependencias entre tareas, establecer fechas de inicio y fin, y monitorizar el progreso en tiempo real. La utilización del Gantt es clave para asegurar que el proyecto se complete dentro del plazo establecido, facilitando la coordinación de las actividades y permitiendo realizar ajustes si surgieran desviaciones en el plan inicial. Gracias a esta herramienta, se puede tener una visión global del proyecto y tomar decisiones informadas para mantener el ritmo de trabajo adecuado.

En la Figura 49 se muestra el diagrama de Gantt del Trabajo Fin de Grado.



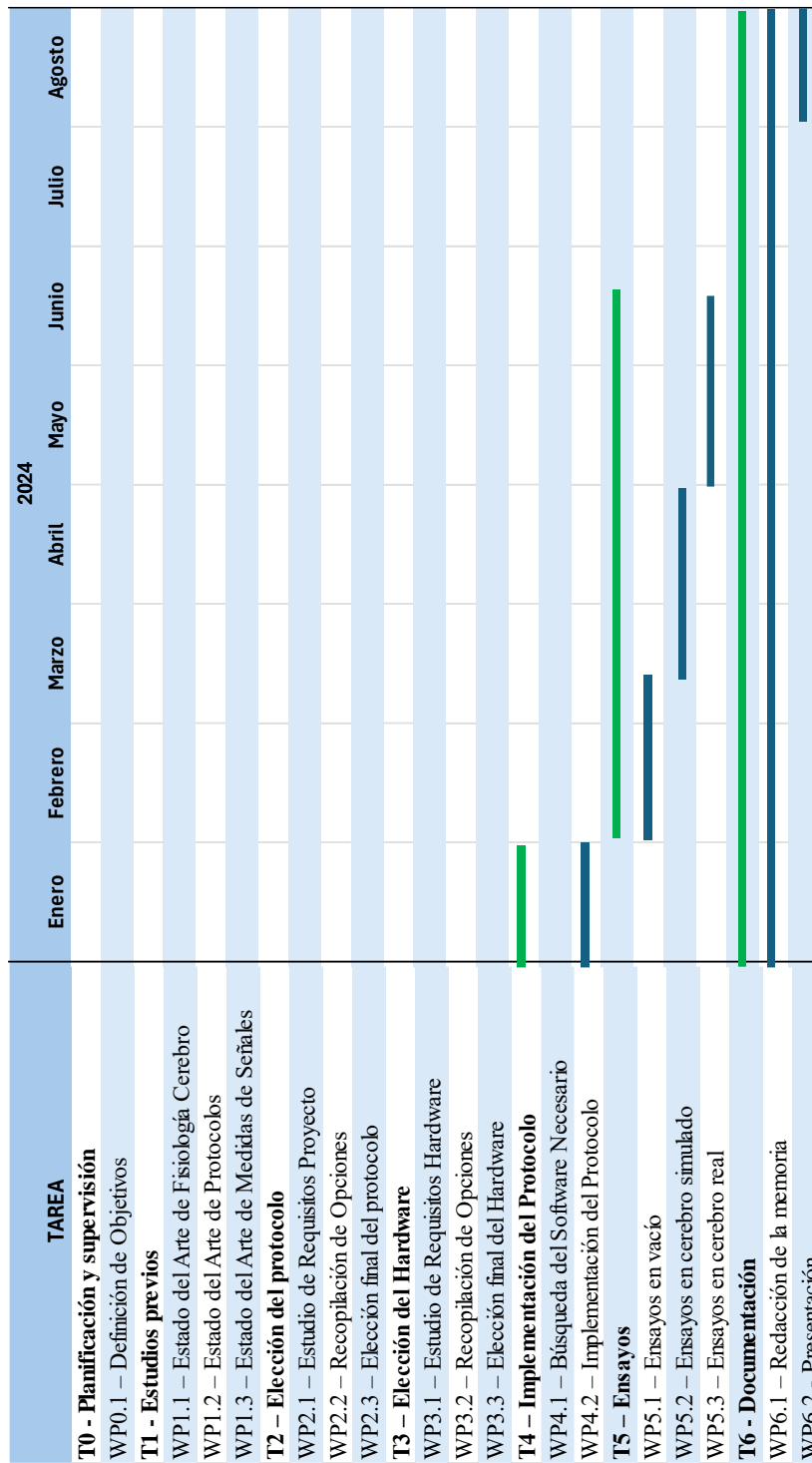


Figura 49: Gantt

10.3 Presupuesto del Proyecto

La elaboración del presupuesto es una etapa crucial en la gestión de cualquier proyecto, ya que permite estimar y asignar los recursos financieros necesarios para su ejecución. En el contexto de este Trabajo Fin de Grado, se ha desarrollado un presupuesto detallado que incluye los costos asociados a todos los elementos identificados en la Estructura Desglosada del Proyecto (EDP). Esto abarca desde la adquisición de materiales y equipos, hasta los costos indirectos relacionados con el desarrollo y la implementación del módulo de comunicación cerebral. La correcta estimación y control del presupuesto asegura que el proyecto pueda completarse dentro de los límites financieros establecidos, evitando sobrecostos y garantizando una asignación eficiente de los recursos disponibles. Se puede observar el desglose de costes en la Tabla 5.

Nombre	Cantidad	Precio
Materiales		
Ordenador Lenovo ideapad 330	1	1.000,00 €
Licencia Office365	1	25,00 €
STM-Nucleo-F103RB	2	15,00 €
P-Nucleo-LRWAN3	2	120,00 €
Cerebro animal	3	7,00 €
Total Material		1.316,00 €
Personal		
Estudiante	320 horas	12,00 €
Total Personal		3.840,00 €
Total Proyecto		5.156,00 €

Tabla 5 Presupuesto del Proyecto

11 Bibliografía

- [1] (OMS), Organización Mundial de la Salud, Atlas: Country Resources for Neurological Disorders, 2004.
- [2] American Cancer Society, «Key Statistics for Brain and Spinal Cord Tumors,» 2023.
- [3] national Cancer Institute, «Cancer Stat Facts: Brain and Other Nervous System Cancer».
- [4] M. Aaron Cohen-Gadol, «aaroncohen-gadol.com,» 2024. [En línea]. Available: <https://www.aaroncohen-gadol.com/es/pacientes/glioma/supervivencia/descripcion-general>.
- [5] M. P. P. M. P. F. C. P. M. F. Jigisha P Thakkar, «American Association of Neurological Surgeons,» 15 Abril 2024. [En línea]. Available: <https://www.aans.org/patients/conditions-treatments/glioblastoma-multiforme/>.
- [6] N. B. Ferrara, «Mayo Clinic,» 21 Diciembre 2021. [En línea]. Available: <https://cancerblog.mayoclinic.org/2021/12/21/glioblastoma-in-older-adults-improving-survival-and-quality-of-life>.
- [7] S. M. V. G. Tejaswi Kanderi, «PubMed,» National Library of Medicine, 2024. [En línea]. Available: <https://pubmed.ncbi.nlm.nih.gov/32644380/>.
- [8] R. e. a. Stupp, EF-14 Trial: Tumor Treating Fields plus Temozolomide for Glioblastoma., JAMA, 2012.
- [9] «Lifeder,» [En línea]. Available: <https://www.lifeder.com/funciones-partes-del-cerebro-humano/>.
- [10] E. R. S. J. H. & J. T. M. Kandel, Principles of Neural Science (4th ed.), McGraw-Hill, 2000.
- [11] E. & d. S. F. L. Niedermeyer, Electroencephalography: Basic Principles, Clinical Applications, and Related Fields, Lippincott Williams & Wilkins, 2004.
- [12] IEEE, «IEEE 802.15.6 - Wireless Body Area Networks,» 2012.
- [13] T. S. Rappaport, Wireless Communications: Principles and Practice, Prentice Hall, 2002.
- [14] «Dusuniot,» [En línea]. Available: <https://www.dusuniot.com/es/blog/sub-ghz-communication/>.

- [15] «mriquestions,» [En línea]. Available: <https://mriquestions.com/sar-limits.html>. [Último acceso: 7 julio 2023].
- [16] G. L. Lluch, «Univerisdad Pablo de Olavide,» 6 julio 2022. [En línea]. Available: <https://www.upo.es/diario/theconversation/2022/07/el-cerebro-estas-caliente-de-lo-que-se-pensaba>.
- [17] «National Instruments,» [En línea]. Available: <https://www.ni.com/es/shop/data-acquisition/measurement-fundamentals/analog-fundamentals/acquiring-an-analog-signal--bandwidth--nyquist-sampling-theorem-.html>. [Último acceso: 2023].
- [18] E. P. Puig, «Diseño y Evaluación de un Módulo de Comunicación para Implantes Cerebrales,» 2022.
- [19] M. E. S. R. C.-S. F. M. a. I. B. A. Khaleghi, «Exposure of the human brain to an electromagnetic plane wave in the 100-1000MHz frequency range for potential treatment of neurodegenerative diseases,» 2011.
- [20] T. Instrument, «Texas Instrument,» [En línea]. Available: [https://www.ti.com/lit/ds/symlink/cc1101.pdf?\\$ts=1615220724186\\$](https://www.ti.com/lit/ds/symlink/cc1101.pdf?$ts=1615220724186$).
- [21] G. E. P. H. Maarten Weyn, «Survey of the DASH7 Alliance Protocol for 433 MHz Wireless Sensor Communication,» *SageJournals*, 2013.
- [22] «Arduino Forum,» 2011. [En línea]. Available: <https://forum.arduino.cc/t/dash7-possible-with-arduino/51206>. [Último acceso: 2023].
- [23] A. Dalmau, «IRTA,» 2023. [En línea]. Available: <https://www.irta.cat/es/estudiar-el-microbioma-intestinal-dels-porcs-per-curar-les-malalties-mentals-humanes/>. [Último acceso: 2023].
- [24] «Teltonika,» marzo 2024. [En línea]. Available: <https://teltonika-networks.com/es/newsroom/what-do-rssi-rsrp-mean-in-your-network-setup>. [Último acceso: abril 2024].
- [25] «Clínica Universidad de Navarra,» [En línea]. Available: <https://www.cun.es/diccionario-medico/terminos/senal-ruido>. [Último acceso: 2024].

Anexo I: Códigos empleados

Prueba LoRa en Arduino

```
#include <Arduino.h>
#include "LoRa.h"
#include "HardwareSerial.h"
#include "SPI.h"

HardwareSerial serial(PB_7, PB_6);
bool initialized = false;

void setup() {

  pinMode(LED_BUILTIN, OUTPUT);
  Serial.begin(9600);

  delay(5000);
  Serial.println("starting lora\n");
  delay(1000);

  SPI.setMISO(PA_6);
  SPI.setMOSI(PA_7);
  SPI.setSCLK(PA_5);
  SPI.setSSSEL(PA_4);

  LoRa.setSPIFrequency(1E6);

  Serial.println("frequency set\n");
  delay(1000);

  LoRa.setPins(PA_4, -1, -1);

  Serial.println("pins set\n");
  delay(1000);

  LoRa.begin(433E6);

  initialized = LoRa.begin(433E6);

  Serial.println("started\n");
}

void loop() {
  digitalWrite(LED_BUILTIN, HIGH);
  delay(500);
  digitalWrite(LED_BUILTIN, LOW);
  delay(500);

  if (!initialized) {
    Serial.println("init failed\n");
    initialized = LoRa.begin(433E6);
  } else {
```

```

        Serial.println("success init\n");
    }
}

```

Código IDE Cube

NS1 Main.c

```

/**
*****
***
* @file    main.c
* @author  MCD Application Team
* @brief   Main program body
*****
***
* @attention
*
* <h2><center>&copy; Copyright (c) 2018 STMicroelectronics.
* All rights reserved.</center></h2>
*
* This software component is licensed by ST under Ultimate Liberty
license
* SLA0044, the "License"; You may not use this file except in compliance
with
* the License. You may obtain a copy of the License at:
*
*                                     www.st.com/SLA0044
*
*****
***
*/

/* Includes -----
---*/
#include "main.h"
#include "platform.h"
#include "stm32_lpm.h"
#include "utilities_def.h"
#include <string.h>
#include <stdlib.h>

#include "stm32l0xx_hal.h"
#include "lora_driver.h"
#include "app_master.h"
#include <stdarg.h>
#include "rtc_if.h"

#include ATCMD_MODEM          /* preprocessing definition in sys_conf.h*/

```

```

/* Private includes -----
---*/

/* Private typedef -----
---*/

/* Private define -----
---*/

/* Private macro -----
---*/

/* Private variables -----
---*/

/* Private function prototypes -----
---*/

void SystemClock_Config(void);

/**
 * @brief Entry point program
 * @param None
 * @retval int
 */
int main(void)
{

    /* MCU Configuration-----
    -----*/

    /* Reset of all peripherals, Initializes the Flash interface and the
    SysTick. */
    HAL_Init();

    /* Configure the system clock */
    SystemClock_Config();

    /* Initialize all configured peripherals */
    MX_Master_Init();

    /* DBG_PRINTF("before idle \n"); */ /*could be a semi-hosting printf */

    /* Infinite loop */
    while (1)
    {
        MX_Master_Process();
    }
}

/**

```

```

* @brief System Clock Configuration
* The system Clock is configured as follow :
* System Clock source = PLL (HSI)
* SYSCLK(Hz) = 32000000
* HCLK(Hz) = 32000000
* AHB Prescaler = 1
* APB1 Prescaler = 1
* APB2 Prescaler = 1
* HSI Frequency(Hz) = 16000000
* PLLMUL = 6
* PLLDIV = 3
* Flash Latency(WS) = 1
* @retval None
*/

void SystemClock_Config(void)
{
    RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
    RCC_OscInitTypeDef RCC_OscInitStruct = {0};
    RCC_PeriphCLKInitTypeDef PeriphClkInit = {0};

    /* Enable HSE Oscillator and Activate PLL with HSE as source */
    RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
    RCC_OscInitStruct.HSEState = RCC_HSE_OFF;
    RCC_OscInitStruct.HSISState = RCC_HSI_ON;
    RCC_OscInitStruct.HSICALibrationValue = RCC_HSICALIBRATION_DEFAULT;
    RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
    RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI;
    RCC_OscInitStruct.PLL.PLLMUL = RCC_PLLMUL_6;
    RCC_OscInitStruct.PLL.PLLDIV = RCC_PLLDIV_3;

    if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
    {
        Error_Handler();
    }

    /* Set Voltage scale1 as MCU will run at 32MHz */
    __HAL_RCC_PWR_CLK_ENABLE();
    __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE1);

    /* Poll VOSF bit of in PWR_CSR. Wait until it is reset to 0 */
    while ( __HAL_PWR_GET_FLAG(PWR_FLAG_VOS) != RESET) {};

    /* Select PLL as system clock source and configure the HCLK, PCLK1 and
    PCLK2
    clocks dividers */
    RCC_ClkInitStruct.ClockType = (RCC_CLOCKTYPE_SYSCLK | RCC_CLOCKTYPE_HCLK
    | RCC_CLOCKTYPE_PCLK1 | RCC_CLOCKTYPE_PCLK2);
    RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
    RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
    RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV1;
    RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;
    if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_1) != HAL_OK)
    {
        Error_Handler();
    }

#ifdef USE_USART2
    PeriphClkInit.PeriphClockSelection = RCC_PERIPHCLK_USART2;
    PeriphClkInit.Usart2ClockSelection = RCC_USART2CLKSOURCE_HSI;
#endif
}

```

```

#else
    PeriphClkInit.PeriphClockSelection = RCC_PERIPHCLK_LPUART1;
    PeriphClkInit.Usart2ClockSelection = RCC_LPUART1CLKSOURCE_HSI;
#endif
    HAL_RCCEx_PeriphCLKConfig(&PeriphClkInit);

#ifdef USE_LRWAN_NS1
    PeriphClkInit.PeriphClockSelection = RCC_PERIPHCLK_USART1;
    PeriphClkInit.Usart1ClockSelection = RCC_USART1CLKSOURCE_HSI;

    HAL_RCCEx_PeriphCLKConfig(&PeriphClkInit);
#endif
    HAL_SYSTICK_Config(HAL_RCC_GetHCLKFreq() / 1000);

    HAL_SYSTICK_CLKSourceConfig(SYSTICK_CLKSOURCE_HCLK);

    /* SysTick_IRQn interrupt configuration */
    HAL_NVIC_SetPriority(SysTick_IRQn, 1, 0);
}

/**
 * @brief This function is executed in case of error occurrence.
 * @retval None
 */
void Error_Handler(void)
{
    /* USER CODE BEGIN Error_Handler_Debug */
    /* User can add his own implementation to report the HAL error return
state */
    while (1)
    {
    }
    /* USER CODE END Error_Handler_Debug */
}

#ifdef USE_FULL_ASSERT
/**
 * @brief Reports the name of the source file and the source line number
 * where the assert_param error has occurred.
 * @param file: pointer to the source file name
 * @param line: assert_param error line source number
 * @retval None
 */
void assert_failed(uint8_t *file, uint32_t line)
{
    /* USER CODE BEGIN 6 */
    /* User can add his own implementation to report the file name and line
number,
ex: printf("Wrong parameters value: file %s on line %d\r\n", file,
line) */
    while (1)
    {
    }
    /* USER CODE END 6 */
}
#endif /* USE_FULL_ASSERT */

/***** (C) COPYRIGHT STMicroelectronics *****/
FILE*****/

```

AT_Master Main.c

```
/**
*****
***
* @file    main.c
* @author  MCD Application Team
* @brief   Main program body
*****
***
* @attention
*
* <h2><center>&copy; Copyright (c) 2018 STMicroelectronics.
* All rights reserved.</center></h2>
*
* This software component is licensed by ST under Ultimate Liberty
license
* SLA0044, the "License"; You may not use this file except in compliance
with
* the License. You may obtain a copy of the License at:
*                                     www.st.com/SLA0044
*
*****
***
*/

/* Includes -----
---*/
#include "main.h"
#include "platform.h"
#include "stm32_lpm.h"
#include "utilities_def.h"
#include <string.h>
#include <stdlib.h>

#include "stm32l0xx_hal.h"
#include "lora_driver.h"
#include "app_master.h"
#include <stdarg.h>
#include "rtc_if.h"

#include ATCMD_MODEM          /* preprocessing definition in sys_conf.h*/

/* Private includes -----
---*/

/* Private typedef -----
---*/

/* Private define -----
---*/
```

```

/* Private macro -----
---*/

/* Private variables -----
---*/

/* Private function prototypes -----
---*/

void SystemClock_Config(void);

/**
 * @brief Entry point program
 * @param None
 * @retval int
 */
int main(void)
{

    /* MCU Configuration-----
    -----*/

    /* Reset of all peripherals, Initializes the Flash interface and the
    SysTick. */
    HAL_Init();

    /* Configure the system clock */
    SystemClock_Config();

    /* Initialize all configured peripherals */
    MX_Master_Init();

    /* DBG_PRINTF("before idle \n"); */ /*could be a semi-hosting printf */

    /* Infinite loop */
    while (1)
    {
        MX_Master_Process();
    }
}

/**
 * @brief System Clock Configuration
 * The system Clock is configured as follow :
 * System Clock source = PLL (HSI)
 * SYSCLK (Hz) = 32000000
 * HCLK (Hz) = 32000000
 * AHB Prescaler = 1
 * APB1 Prescaler = 1
 * APB2 Prescaler = 1
 * HSI Frequency (Hz) = 16000000
 * PLLMUL = 6
 * PLLDIV = 3
 * Flash Latency (WS) = 1

```

```

* @retval None
*/

void SystemClock_Config(void)
{
    RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
    RCC_OscInitTypeDef RCC_OscInitStruct = {0};
    RCC_PeriphCLKInitTypeDef PeriphClkInit = {0};

    /* Enable HSE Oscillator and Activate PLL with HSE as source */
    RCC_OscInitStruct.OscillatorType      = RCC_OSCILLATORTYPE_HSI;
    RCC_OscInitStruct.HSEState            = RCC_HSE_OFF;
    RCC_OscInitStruct.HSIState            = RCC_HSI_ON;
    RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
    RCC_OscInitStruct.PLL.PLLState        = RCC_PLL_ON;
    RCC_OscInitStruct.PLL.PLLSource       = RCC_PLLSOURCE_HSI;
    RCC_OscInitStruct.PLL.PLLMUL          = RCC_PLLMUL_6;
    RCC_OscInitStruct.PLL.PLLDIV          = RCC_PLLDIV_3;

    if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
    {
        Error_Handler();
    }

    /* Set Voltage scale1 as MCU will run at 32MHz */
    __HAL_RCC_PWR_CLK_ENABLE();
    __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE1);

    /* Poll VOSF bit of in PWR_CSR. Wait until it is reset to 0 */
    while ( __HAL_PWR_GET_FLAG(PWR_FLAG_VOS) != RESET) {};

    /* Select PLL as system clock source and configure the HCLK, PCLK1 and
    PCLK2
    clocks dividers */
    RCC_ClkInitStruct.ClockType = (RCC_CLOCKTYPE_SYSCLK | RCC_CLOCKTYPE_HCLK
    | RCC_CLOCKTYPE_PCLK1 | RCC_CLOCKTYPE_PCLK2);
    RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
    RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
    RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV1;
    RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;
    if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_1) != HAL_OK)
    {
        Error_Handler();
    }

#ifdef USE_USART2
    PeriphClkInit.PeriphClockSelection = RCC_PERIPHCLK_USART2;
    PeriphClkInit.Usart2ClockSelection = RCC_USART2CLKSOURCE_HSI;
#else
    PeriphClkInit.PeriphClockSelection = RCC_PERIPHCLK_LPUART1;
    PeriphClkInit.Usart2ClockSelection = RCC_LPUART1CLKSOURCE_HSI;
#endif
    HAL_RCCEx_PeriphCLKConfig(&PeriphClkInit);

#ifdef USE_LRWAN_NS1
    PeriphClkInit.PeriphClockSelection = RCC_PERIPHCLK_USART1;
    PeriphClkInit.Usart1ClockSelection = RCC_USART1CLKSOURCE_HSI;

    HAL_RCCEx_PeriphCLKConfig(&PeriphClkInit);
#endif
}

```

```

HAL_SYSTICK_Config(HAL_RCC_GetHCLKFreq() / 1000);

HAL_SYSTICK_CLKSourceConfig(SYSTICK_CLKSOURCE_HCLK);

/* SysTick_IRQn interrupt configuration */
HAL_NVIC_SetPriority(SysTick_IRQn, 1, 0);
}

/**
 * @brief This function is executed in case of error occurrence.
 * @retval None
 */
void Error_Handler(void)
{
    /* USER CODE BEGIN Error_Handler_Debug */
    /* User can add his own implementation to report the HAL error return
    state */
    while (1)
    {
    }
    /* USER CODE END Error_Handler_Debug */
}

#ifdef USE_FULL_ASSERT
/**
 * @brief Reports the name of the source file and the source line number
 * where the assert_param error has occurred.
 * @param file: pointer to the source file name
 * @param line: assert_param error line source number
 * @retval None
 */
void assert_failed(uint8_t *file, uint32_t line)
{
    /* USER CODE BEGIN 6 */
    /* User can add his own implementation to report the file name and line
    number,
    ex: printf("Wrong parameters value: file %s on line %d\r\n", file,
    line) */
    while (1)
    {
    }
    /* USER CODE END 6 */
}
#endif /* USE_FULL_ASSERT */

/***** (C) COPYRIGHT STMicroelectronics *****/
FILE*****/

```

LoRa app_master.c

```

/**
*****
***
* @file    app_master.c
* @author  MCD Application Team
* @brief   Entry point AT master application
*****
***
* @attention
*
* <h2><center>&copy; Copyright (c) 2020 STMicroelectronics.
* All rights reserved.</center></h2>
*
* This software component is licensed by ST under Ultimate Liberty
license
* SLA0044, the "License"; You may not use this file except in compliance
with
* the License. You may obtain a copy of the License at:
*                                     www.st.com/SLA0044
*
*****
***
*/

/* Includes -----
---*/

#include "master_app.h"
#include "stm32_lpm.h"
#include "sys_app.h"
#include "lora_driver.h"
/*#include "stm32_seq.h"*/ /* if using sequencer uncomment */
#include "usart.h"
#include "app_master.h"
#include "sys_conf.h"
/* USER CODE BEGIN Includes */

/* USER CODE END Includes */

/* External variables -----
----*/
/* USER CODE BEGIN EV */

/* USER CODE END EV */

/* Private typedef -----
---*/
/* USER CODE BEGIN PTD */

/* USER CODE END PTD */

/* Private define -----
---*/

```

```

/* USER CODE BEGIN PD */

/* USER CODE END PD */

/* Private macro -----
---*/
/* USER CODE BEGIN PM */

/* USER CODE END PM */

/* Private variables -----
---*/

/* USER CODE BEGIN PV */

/* USER CODE END PV */

/* Private function prototypes -----
---*/

/* USER CODE BEGIN PFP */

/* USER CODE END PFP */

/* Exported functions -----
---*/

#define LEDn_MODEM                1

/* GPIO configuration for I_NUCLEO_LRWAN1 modem's led*/
GPIO_TypeDef *LED_MODEM_PORT[LEDn_MODEM] = {GPIOA};

const uint16_t LED_MODEM_PIN[LEDn_MODEM] = {GPIO_PIN_0};

void MX_Master_Init(void)
{
    /* USER CODE BEGIN MX_LoRaWAN_Init_1 */

    /* USER CODE END MX_LoRaWAN_Init_1 */
    SystemApp_Init();
    /* USER CODE BEGIN MX_LoRaWAN_Init_2 */

    /* USER CODE END MX_LoRaWAN_Init_2 */

    MasterApp_Init();
    /* USER CODE BEGIN MX_LoRaWAN_Init_3 */

    /* USER CODE END MX_LoRaWAN_Init_3 */
}

void MX_Master_Process(void)
{
    /*if using sequencer comment the following code*/
    /* run the LoRa Modem state machine*/
    Lora_fsm();
    DISABLE_IRQ();
    /* if an interrupt has occurred after DISABLE_IRQ, it is kept pending
    * and cortex will not enter low power anyway */
}

```

```

    if ((lora_getDeviceState() == DEVICE_SLEEP) &&
(HW_UART_Modem_IsNewCharReceived() == RESET))
    {

#ifdef (LOW_POWER_DISABLE) && (LOW_POWER_DISABLE == 1)
        UTIL_LPM_EnterLowPower();
#elif !defined (LOW_POWER_DISABLE)
#endif
    }
    ENABLE_IRQ();
    /* USER CODE BEGIN MX_LoRaWAN_Process_1 */

    /* USER CODE END MX_LoRaWAN_Process_1 */
    /* if using sequencer uncomment the call to the sequencer*/
    /*UTIL_SEQ_Run(UTIL_SEQ_DEFAULT);*/
    /* USER CODE BEGIN MX_LoRaWAN_Process_2 */

    /* USER CODE END MX_LoRaWAN_Process_2 */
}

/* USER CODE BEGIN EF */

/* USER CODE END EF */

/* Private Functions Definition -----
-----*/
/* USER CODE BEGIN PrFD */

/* USER CODE END PrFD */

/**
 * @brief Configures LED GPIO for external modem.
 * @param Led: Led to be configured.
 * @retval None
 */
void Master_LED_Modem_Init(Led_TypeDef Led)
{
    GPIO_InitTypeDef  gpioinitstruct;

    /* Enable the GPIO_LED Clock */
    LEDx_GPIO_CLK_ENABLE(Led);

    /* Configure the GPIO_LED pin */
    gpioinitstruct.Pin = LED_MODEM_PIN[Led];
    gpioinitstruct.Mode = GPIO_MODE_OUTPUT_PP;
    gpioinitstruct.Pull = GPIO_NOPULL;
    gpioinitstruct.Speed = GPIO_SPEED_FREQ_VERY_HIGH;

    HAL_GPIO_Init(LED_MODEM_PORT[Led], &gpioinitstruct);

    /* Reset PIN to switch off the LED */
    HAL_GPIO_WritePin(LED_MODEM_PORT[Led], LED_MODEM_PIN[Led],
GPIO_PIN_RESET);
}

/**
 * @brief DeInit LEDs GPIO for external modem.
 * @param Led: LED to be de-init.
 * @note Led DeInit does not disable the GPIO clock nor disable the Mfx

```

```

    * @retval None
    */
void Master_LED_Modem_DeInit(Led_TypeDef Led)
{
    GPIO_InitTypeDef  gpio_init_structure;

    /* Turn off LED */
    HAL_GPIO_WritePin(LED_MODEM_PORT[Led], LED_MODEM_PIN[Led],
GPIO_PIN_RESET);
    /* DeInit the GPIO_LED pin */
    gpio_init_structure.Pin = LED_MODEM_PIN[Led];
    HAL_GPIO_DeInit(LED_MODEM_PORT[Led], gpio_init_structure.Pin);
}

/**
 * @brief Turns selected LED Modem On.
 * @param Led: Specifies the Led to be set on.
 * @retval None
 */
void Master_LED_Modem_On(Led_TypeDef Led)
{
    HAL_GPIO_WritePin(LED_MODEM_PORT[Led], LED_MODEM_PIN[Led], GPIO_PIN_SET);
}

/**
 * @brief Turns selected LED Modem Off.
 * @param Led: Specifies the Led to be set off.
 * @retval None
 */
void Master_LED_Modem_Off(Led_TypeDef Led)
{
    HAL_GPIO_WritePin(LED_MODEM_PORT[Led], LED_MODEM_PIN[Led],
GPIO_PIN_RESET);
}

/***** (C) COPYRIGHT STMicroelectronics *****/
FILE*****/

```

LoRa master_app.c

```

/**
*****
***
 * @file    master_app.c
 * @author  MCD Application Team
 * @brief   Application of the AT Master
*****
***
 * @attention
 *
 * <h2><center>&copy; Copyright (c) 2020 STMicroelectronics.
 * All rights reserved.</center></h2>
 *
 * This software component is licensed by ST under Ultimate Liberty
license

```

```
 * SLA0044, the "License"; You may not use this file except in compliance
with
 * the License. You may obtain a copy of the License at:
 *
 *                               www.st.com/SLA0044
 *

*****
***
 */

/* Includes -----
---*/
#include "platform.h"
#include "sys_app.h"
#include "master_app.h"
#include "lora_driver.h"
#include "sys_sensors.h"
#include "rtc_if.h"
#include "stdio.h"

#include "lrwan_ns1_atcmd.h"          /* preprocessing definition in
sys_conf.h*/

/* USER CODE BEGIN Includes */

/* USER CODE END Includes */

/* External variables -----
---*/
/* USER CODE BEGIN EV */

/* USER CODE END EV */

/* Private typedef -----
---*/

/* USER CODE BEGIN PTD */

/* USER CODE END PTD */

/* Private define -----
---*/
/* CAYENNE_LLDP is myDevices Application server*/
#define CAYENNE_LPP
#define LPP_DATATYPE_DIGITAL_INPUT 0x0
#define LPP_DATATYPE_DIGITAL_OUTPUT 0x1
#define LPP_DATATYPE_HUMIDITY 0x68
#define LPP_DATATYPE_TEMPERATURE 0x67
#define LPP_DATATYPE_BAROMETER 0x73

/* Private variables -----
---*/

static sensor_t Sensor;              /* struct for data
sensor*/
```

```

#ifdef USE_LRWAN_NS1
                                //MODIFICADO POR JAVIER
static bool user_button_flag = true;
#endif

/* Private function prototypes -----
---*/

static void SensorMeasureData(sSendDataBinary_t *SendDataBinary);

/* load call backs*/
static LoRaDriverCallback_t LoRaDriverCallbacks = { SensorMeasureData,
                                                    NULL
                                                    };

#ifdef USE_LRWAN_NS1
                                //MODIFICADO POR JAVIER
/**
 * @brief EXTI line detection callbacks.
 * @param GPIO_Pin: Specifies the pins connected to the EXTI line.
 * @retval None
 */
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    /*HW_RTC_DelayMs(10);*/
    RTC_IF_DelayMs(10);
    if (!HAL_GPIO_ReadPin(USER_BUTTON_GPIO_PORT, USER_BUTTON_PIN))
    {
        if (user_button_flag)
        {
            Modem_IO_DeInit();
            BSP_LED_DeInit(LED2);
            user_button_flag = false;
            while (1);
        }
        else
        {
            Modem_IO_Init();
            NVIC_SystemReset();
        }
    }
}
#endif

/* Private macro -----
---*/

#ifdef CAYENNE_LPP
#define LORAWAN_APP_PORT          99;          /*LoRaWAN application
port*/
#else
#define LORAWAN_APP_PORT          2;          /*LoRaWAN application
port*/
#endif

#define LORAWAN_CONFIRMED_MSG    ENABLE        /*LoRaWAN confirmed
messages*/

#define SENSORS_MEASURE_CYCLE    15000        /*Periode to do sensors
measurement*/

```

```

#define JOIN_MODE                                OTAA_JOIN_MODE    /*ABP_JOIN_MODE */
/*LoRaWan join methode*/

#ifdef USE_LRWAN_NS1
#define FREQ_BAND                                EU433 /*CN470PREQUEL*/
#endif

/* Init LoRa Driver modem parameters*/
#ifdef USE_LRWAN_NS1
static LoRaDriverParam_t LoRaDriverParam = {  SENSORS_MEASURE_CYCLE,
JOIN_MODE, FREQ_BAND};
#else
static LoRaDriverParam_t LoRaDriverParam = {  SENSORS_MEASURE_CYCLE,
JOIN_MODE};
#endif

/*!
 * Master context initialization following the LoRa modem used
 *
 */
void MasterApp_Init(void)
{

    /* if using sequencer uncomment the task creation */
    /*UTIL_SEQ_RegTask((1 << CFG_SEQ_Task_Lora_fsm), UTIL_SEQ_RFU, Lora_fsm);
*/

    Lora_Ctx_Init(&LoRaDriverCallbacks, &LoRaDriverParam);
}

/* USER CODE BEGIN PFP */

/* USER CODE END PFP */

/***** LoRa Part Application
*****/

/*****
****
 * @brief SensorMeasureData
 * @param none
 * @return none
****
*****/
static void SensorMeasureData(sSendDataBinary_t *SendDataBinary)
{
    uint8_t LedState = 0;                /*just for padding*/
    uint16_t pressure = 0;
    int16_t temperature = 0;
    uint16_t humidity = 0;
    uint32_t BatLevel = 0;                /*end device connected to external
power source*/
    uint8_t index = 0;
    ATEerror_t LoraCmdRetCode;

```

```

    /*read pressure, Humidity and Temperature in order to be send on
    LoRaWAN*/
    // BSP_sensor_Read(&Sensor);
    EnvSensors_Read(&Sensor);

#ifdef CAYENNE_LPP
    uint8_t cchannel = 0;

    temperature = (int16_t)(Sensor.temperature * 10);          /* in °C * 10
    */
    pressure     = (uint16_t)(Sensor.pressure * 100 / 10);     /* in hPa / 10
    */
    humidity     = (uint16_t)(Sensor.humidity * 2);           /* in %*2
    */

#ifdef USE_LRWAN_NS1
    uint8_t str_buff[100];
    memset(str_buff, 0, sizeof(str_buff));
    sprintf((char *)str_buff, "Pressure: %.1f hPa   Temperature: %.1f degreeC
    Humidity: %.1f\r\n", Sensor.pressure, \
        Sensor.temperature, \
        Sensor.humidity);
#endif defined CMD_DEBUG
    at_printf_send(str_buff, strlen((const char *)str_buff));
#endif

#endif

    SendDataBinary->Buffer[index++] = cchannel++;
    SendDataBinary->Buffer[index++] = LPP_DATATYPE_BAROMETER;
    SendDataBinary->Buffer[index++] = (pressure >> 8) & 0xFF;
    SendDataBinary->Buffer[index++] = pressure & 0xFF;
    SendDataBinary->Buffer[index++] = cchannel++;
    SendDataBinary->Buffer[index++] = LPP_DATATYPE_TEMPERATURE;
    SendDataBinary->Buffer[index++] = (temperature >> 8) & 0xFF;
    SendDataBinary->Buffer[index++] = temperature & 0xFF;
    SendDataBinary->Buffer[index++] = cchannel++;
    SendDataBinary->Buffer[index++] = LPP_DATATYPE_HUMIDITY;
    SendDataBinary->Buffer[index++] = humidity & 0xFF;
    SendDataBinary->Buffer[index++] = cchannel++;
    SendDataBinary->Buffer[index++] = LPP_DATATYPE_DIGITAL_INPUT;

    /*get battery level of the modem (slave)*/
    LoraCmdRetCode = Lora_GetBatLevel(&BatLevel);
#ifdef USE_LRWAN_NS1
    if (LoraCmdRetCode == ATCTL_RET_CMD_VDD)
#else
    if (LoraCmdRetCode == AT_OK)
#endif
    {
        DBG_PRINTF("Msg status = %d --> OK\n", (uint16_t)BatLevel);
    }
    else
    {
        DBG_PRINTF("Msg status = %d --> KO\n", (uint16_t)BatLevel);
    }

    SendDataBinary->Buffer[index++] = BatLevel * 100 / 254;
    SendDataBinary->Buffer[index++] = cchannel++;
    SendDataBinary->Buffer[index++] = LPP_DATATYPE_DIGITAL_OUTPUT;
    SendDataBinary->Buffer[index++] = LedState;

```

```

#else

    int32_t latitude, longitude = 0;      /*just for padding*/
    uint16_t altitudeGps = 0;           /*just for padding*/

    temperature = (int16_t)(Sensor.temperature * 100);      /* in °C * 100
*/
    pressure     = (uint16_t)(Sensor.pressure * 100 / 10);  /* in hPa / 10
*/
    humidity     = (uint16_t)(Sensor.humidity * 10);       /* in %*10
*/

    latitude = Sensor.latitude;    /* not relevant*/
    longitude = Sensor.longitude;  /* not relevant*/

    /*fill up the send buffer*/
    SendDataBinary->Buffer[index++] = LedState;
    SendDataBinary->Buffer[index++] = (pressure >> 8) & 0xFF;
    SendDataBinary->Buffer[index++] = pressure & 0xFF;
    SendDataBinary->Buffer[index++] = (temperature >> 8) & 0xFF;
    SendDataBinary->Buffer[index++] = temperature & 0xFF;
    SendDataBinary->Buffer[index++] = (humidity >> 8) & 0xFF;
    SendDataBinary->Buffer[index++] = humidity & 0xFF;

    /*get battery level of the modem (slave)*/
    LoraCmdRetCode = Lora_GetBatLevel(&BatLevel);
    if (LoraCmdRetCode == AT_OK)
    {
        DBG_PRINTF("Msg status = %d --> OK\n", BatLevel);
    }
    else
    {
        DBG_PRINTF("Msg status = %d --> KO\n", BatLevel);
    }
    SendDataBinary->Buffer[index++] = (uint8_t)BatLevel;

    /*remaining data just for padding*/
    SendDataBinary->Buffer[index++] = (latitude >> 16) & 0xFF;
    SendDataBinary->Buffer[index++] = (latitude >> 8) & 0xFF;
    SendDataBinary->Buffer[index++] = latitude & 0xFF;
    SendDataBinary->Buffer[index++] = (longitude >> 16) & 0xFF;
    SendDataBinary->Buffer[index++] = (longitude >> 8) & 0xFF;
    SendDataBinary->Buffer[index++] = longitude & 0xFF;
    SendDataBinary->Buffer[index++] = (altitudeGps >> 8) & 0xFF;
    SendDataBinary->Buffer[index++] = altitudeGps & 0xFF;

#endif /*CAYENNE_LPP*/

    SendDataBinary->DataSize = index;
#ifdef USE_LRWAN_NS1
    SendDataBinary->Port = LORAWAN_APP_PORT;
#endif

#ifdef USE_I_NUCLEO_LRWAN1
    SendDataBinary->Ack = !LORAWAN_CONFIRMED_MSG;
#endif
}

#ifdef USE_FULL_ASSERT

```

```
/**
 * @brief Reports the name of the source file and the source line number
 * where the assert_param error has occurred.
 * @param file: pointer to the source file name
 * @param line: assert_param error line source number
 * @retval None
 */
void assert_failed(uint8_t *file, uint32_t line)
{

    /* User can add his own implementation to report the file name and line
    number,
    ex: printf("Wrong parameters value: file %s on line %d\r\n", file,
    line) */

}

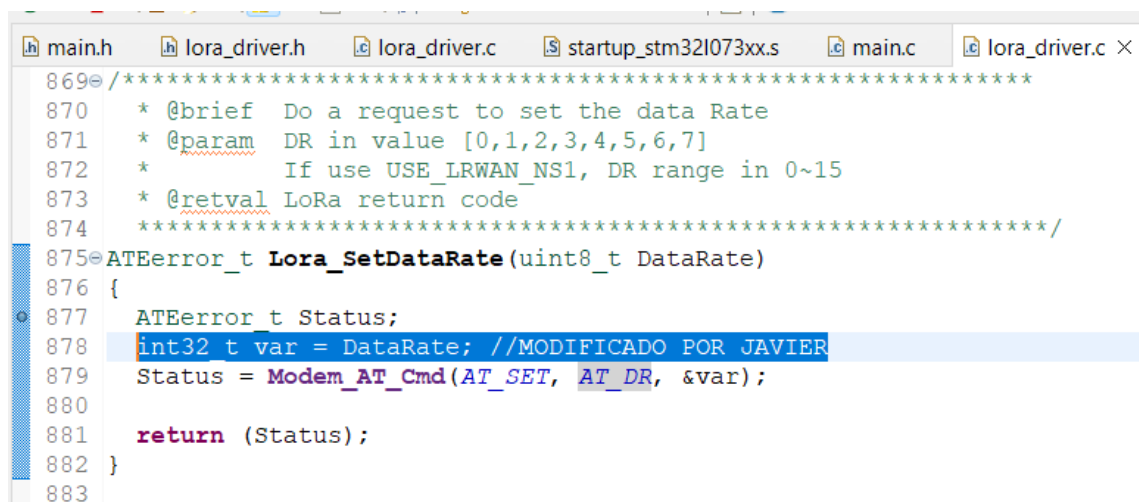
#endif

/***** (C) COPYRIGHT STMicroelectronics *****/
FILE*****/
```


Anexo II: Otros cambios realizados en la Implementación

Dentro del código recomendado por el fabricante, se encuentran algunos fallos que pueden llegar a impedir la correcta compilación o ejecución del programa. Es por esto que se debe tener en consideración los siguientes cambios:

- Cambio de DataRate en LoRa_Driver.c. **Figura XX.**



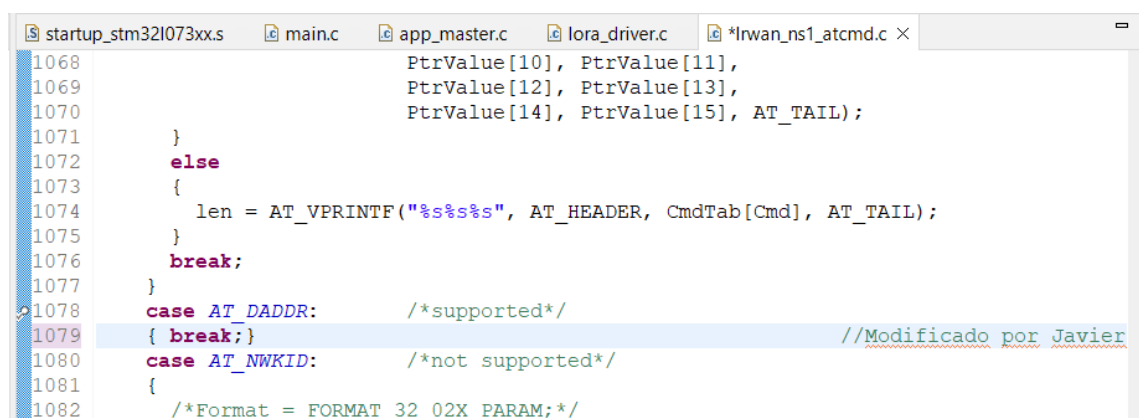
```

869 /*****
870  * @brief Do a request to set the data Rate
871  * @param DR in value [0,1,2,3,4,5,6,7]
872  *       If use USE_LRWAN_NS1, DR range in 0~15
873  * @retval LoRa return code
874  *****/
875 ATEError_t Lora_SetDataRate(uint8_t DataRate)
876 {
877     ATEError_t Status;
878     int32_t var = DataRate; //MODIFICADO POR JAVIER
879     Status = Modem_AT_Cmd(AT_SET, AT_DR, &var);
880
881     return (Status);
882 }
883

```

Figura 50: Cambio LoRa_Driver

- En caso de saltar un error en el DR, añadir “break” en lrwan_rs1_atcmd.c. **Figura XX.**



```

1068         PtrValue[10], PtrValue[11],
1069         PtrValue[12], PtrValue[13],
1070         PtrValue[14], PtrValue[15], AT_TAIL);
1071     }
1072     else
1073     {
1074         len = AT_VPRINTF("%s%s", AT_HEADER, CmdTab[Cmd], AT_TAIL);
1075     }
1076     break;
1077 }
1078 case AT_DADDR: /*supported*/
1079 { break; } //Modificado por Javier
1080 case AT_NWKID: /*not supported*/
1081 {
1082     /*Format = FORMAT_32_02X_PARAM;*/

```

Figura 51: Break en lrwan_rs1_atcmd.c

- Si apareciese error de GDP Server como el de la **Figura XX** al subir un archivo a la placa, se tiene que cambiar el TCP como se observa en la **Figura XX** y **Figura XX**.

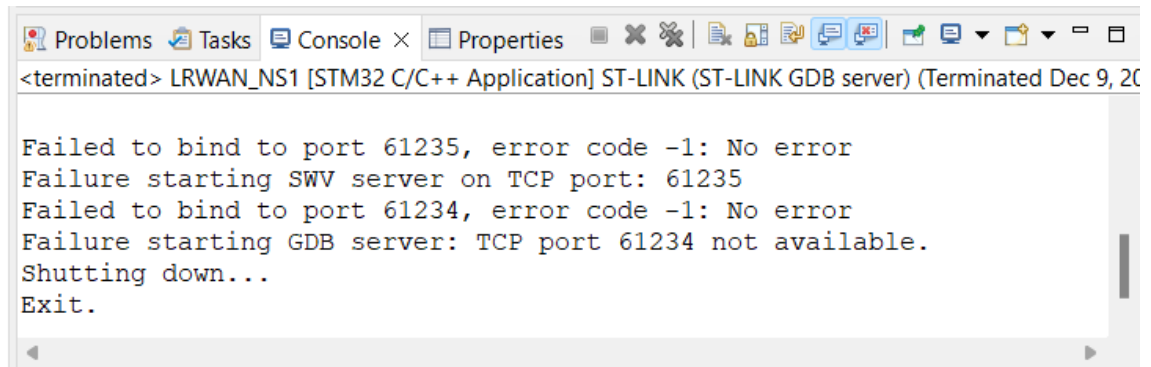


Figura 52: Error de GDP

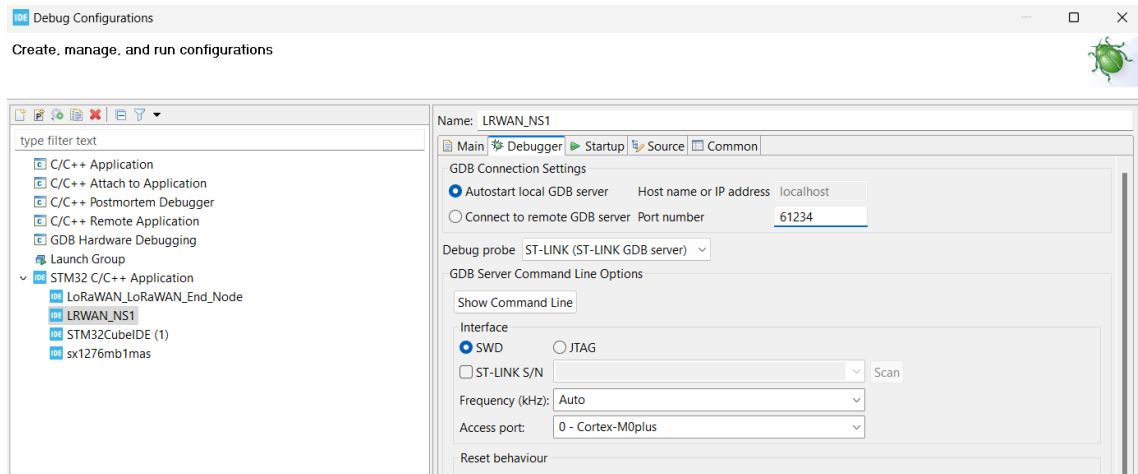


Figura 53: Cambio TCP Original

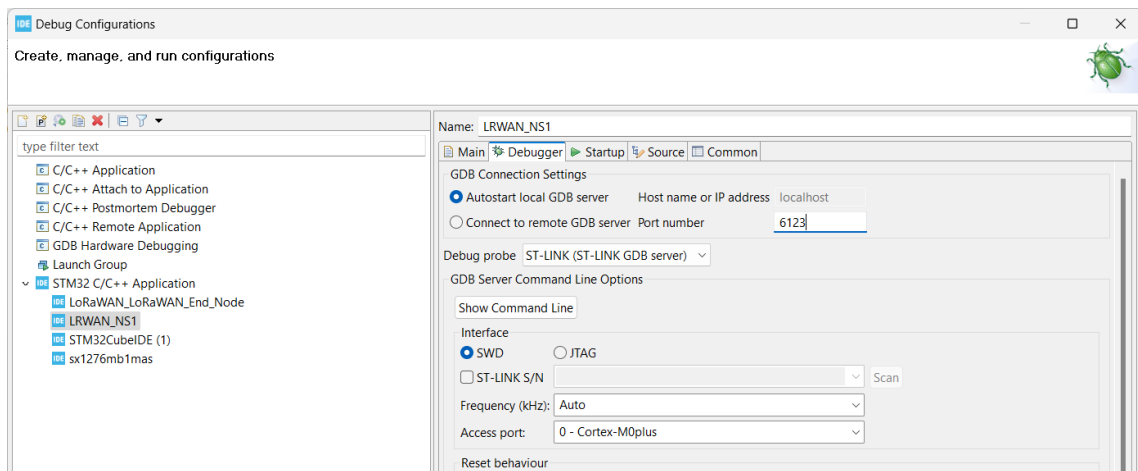
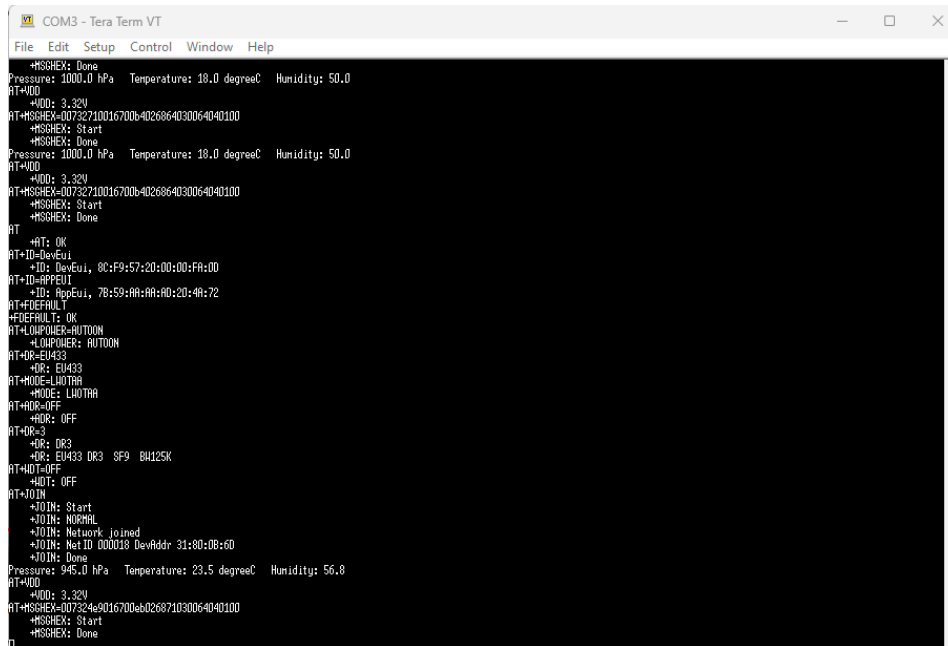


Figura 54: Cambio TCP Alterado

- En sys_conf.h se debe cambiar en el define SENSOR_ENABLED 0 → por un 1
- Una vez cambiados los archivos sys_sensors se observa como cambian los valores por defecto a los reales El cambio ha sido total como se observa en la **Figura XX**.



```
COM3 - Tera Term VT
File Edit Setup Control Window Help
+MSHEX: Done
Pressure: 1000.0 hPa Temperature: 18.0 degreeC Humidity: 50.0
AT+MOD
+MOD: 3.32V
AT+MSHEX=00732710016700b4026864030064040100
+MSHEX: Start
+MSHEX: Done
Pressure: 1000.0 hPa Temperature: 18.0 degreeC Humidity: 50.0
AT+MOD
+MOD: 3.32V
AT+MSHEX=00732710016700b4026864030064040100
+MSHEX: Start
+MSHEX: Done
AT
+AT: OK
AT+ID=DevEui
+ID: DevEui, 80:F9:57:20:00:00:FA:00
AT+ID=AppEUI
+ID: AppEui, 78:59:8A:8A:80:20:48:72
AT+DEFAULT
+DEFAULT: OK
AT+LOWPOWER=AUTON
+LOWPOWER: AUTON
AT+DR=EUI433
+DR: EUI433
AT+MODE=L40TAA
+MODE: L40TAA
AT+ADR=OFF
+ADR: OFF
AT+DR=3
+DR: DR3
+SP: EUI433 DR3 SF9 BU125K
AT+MOD=OFF
+MOD: OFF
AT+JOIN
+JOIN: Start
+JOIN: NORMAL
+JOIN: Network joined
+JOIN: NetID 000018 DevAddr 31:80:08:60
+JOIN: Done
Pressure: 945.0 hPa Temperature: 23.5 degreeC Humidity: 56.8
AT+MOD
+MOD: 3.32V
AT+MSHEX=007324e9016700eb026871030064040100
+MSHEX: Start
+MSHEX: Done
```

Figura 55: Resultado de los cambios realizados

