



## Trabajo de Fin de Grado

Grado de Ingeniería en Sistemas de Telecomunicación

---

Diseño de un  
Gateway WirelessHART con conexión a Internet  
mediante tecnología NB-IoT/CAT-M1 y protocolo MQTT

Imanol Rojas Pérez

---

Tutor: Dr. Pere Tuset-Peiró

Departamento de Telecomunicación e Ingeniería de Sistemas

Escola d'Enginyeria

Universitat Autònoma de Barcelona (UAB)

Septiembre de 2021

# Tabla de contenido

1.	Introducción .....	4
1.1.	Contexto .....	4
1.2.	Objetivos .....	5
1.3.	Requerimientos .....	5
1.4.	Diagrama de bloques .....	6
1.5.	Organización de la memoria .....	7
2.	Estado del arte .....	8
2.1.	Comunicaciones WPAN/LPWAN .....	8
2.1.1.	Tecnologías LPWAN.....	9
2.1.2.	Tecnologías WPAN .....	12
2.2.	Comunicaciones celulares .....	20
2.2.1.	NB-IoT.....	22
2.2.2.	LTE-M/CAT-M1 .....	24
2.3.	Protocolos de transporte .....	26
2.3.1.	CoAP (Constrain Application Protocol) .....	26
2.3.2.	MQTT (Message Queuing Telemetry Transport) .....	26
2.4.	Conclusiones.....	27
3.	Diseño y desarrollo del hardware .....	30
3.1.	Selección de componentes .....	30
3.2.	Diseño del subsistema nRF9160.....	31
3.3.	Diseño del subsistema NB-IoT/CAT-M1 y GPS .....	34
3.3.1.	Antenas NB-IoT/CAT-M1 y GPS.....	34
3.3.2.	Diseño del circuito para la SIM.....	36
3.4.	Diseño del subsistema IEEE802.15.4E.....	37
3.5.	Diseño del subsistema de alimentación.....	39
3.5.1.	Power Switch.....	40
3.5.2.	Power Backup.....	41
3.5.3.	Power Regulator.....	42
3.5.4.	Estimación de consumo energético y elección de la batería de emergencia .....	42
3.5.5.	Control de alimentación para el subsistema IEEE802.15.4E .....	46
3.6.	Diseño del subsistema de interfaz con el usuario.....	48
3.7.	Estimación de costes .....	49
4.	Validación del hardware.....	53
4.1.	Plan de validación.....	53

4.2.	Validación eléctrica .....	53
4.3.	Validación lógica.....	56
5.	Diseño y desarrollo del firmware .....	59
5.1.	Arquitectura de la solución .....	59
5.2.	Implementación .....	60
6.	Validación de la solución.....	62
7.	Conclusiones y trabajo futuro .....	65
8.	Bibliografía .....	67
8.1.	Referencias.....	67
8.2.	Figuras .....	69
9.	Anexo I: Esquemáticos .....	70
10.	Anexo II: Bill of Materials (BOM).....	81

## 1. Introducción

La presente introducción contextualiza el proyecto en el que se basa esta memoria y muestra un pequeño acercamiento a las tecnologías IoT (Internet of Things). Se describen los objetivos del proyecto respecto al estado actual de éste, se enumeran los requerimientos y se expone la solución mediante un diagrama de bloques.

### 1.1. Contexto

El proyecto descrito en esta memoria se desarrolla en el marco de las tecnologías IoT. Podríamos definir IoT como la agrupación e interconexión de dispositivos y objetos mediante una red, privada o Internet, donde todos los nodos podrían ser visibles e interaccionar entre ellos con el fin de automatizar operaciones, reducir costes y mejorar la calidad de vida de quienes usan estas tecnologías. Estos nodos podrían ser dispositivos mecánicos o sensores y actuadores como lámparas, interruptores, *wearables* (relojes o ropa), etc. El objetivo de IoT, por tanto, es una interacción máquina a máquina (M2M, Machine to Machine). En la actualidad estas tecnologías se están aplicando en diversos mercados como la domótica, las *Smart Cities*, el sector industrial, el sector de la ganadería y muchos más. Las tecnologías IoT son muy versátiles y esto hace que se puedan aplicar para resolver problemas o necesidades que se puedan plantear actualmente.

Un ejemplo de proyecto con aplicación de las tecnologías IoT sería el proyecto SmartMarina [1] de INRIA (Institut National de Recherche en Informatique et en Automatique) junto con la compañía FALCO [2]. INRIA se encarga de la investigación y desarrollo de la solución y FALCO se encarga de hacer el despliegue y la instalación de la tecnología desarrollada. El equipo de SmartMarina considera que en un futuro las marinas se van a convertir en vecindarios flotantes, y que las marinas de hoy serán las *Smart Cities* del mañana. Con la evolución de las marinas se plantean tres ventajas principales:

- Desde el punto de vista de la gestión de la marina, la gestión automática del amarre de las embarcaciones y el control del consumo de la electricidad y el agua, permiten al personal liberarse para recibir a los visitantes y centrarse por completo en su bienestar.
- Los propietarios de embarcaciones anuales y los visitantes ocasionales de la marina ahora pueden disfrutar de nuevos servicios, desde una mayor disponibilidad de amarres hasta el monitoreo remoto y alertas sobre el estado de su embarcación.
- Automatizar operaciones manuales para reducir costes operativos.

Esta memoria se basa en el diseño del *gateway* que se ha llevado a cabo dentro del grupo WiNe de la UOC (Universitat Oberta de Catalunya) junto con INRIA y pretende interconectar sensores de una marina a Internet con tal de ofrecer nuevos servicios a los propietarios de embarcaciones y a los visitantes.

Actualmente las marinas donde se va a desplegar el sistema completo tienen instalados los sensores con tecnología WirelessHART que miden la presencia de las embarcaciones, el consumo eléctrico y el consumo de agua. El sistema actualmente envía los datos al *backend* mediante un *gateway* conectado a internet por cable Ethernet o por WiFi (proporcionados por la marina).

En el diagrama de la Figura 1 se puede ver una representación simplificada del estado actual del proyecto.

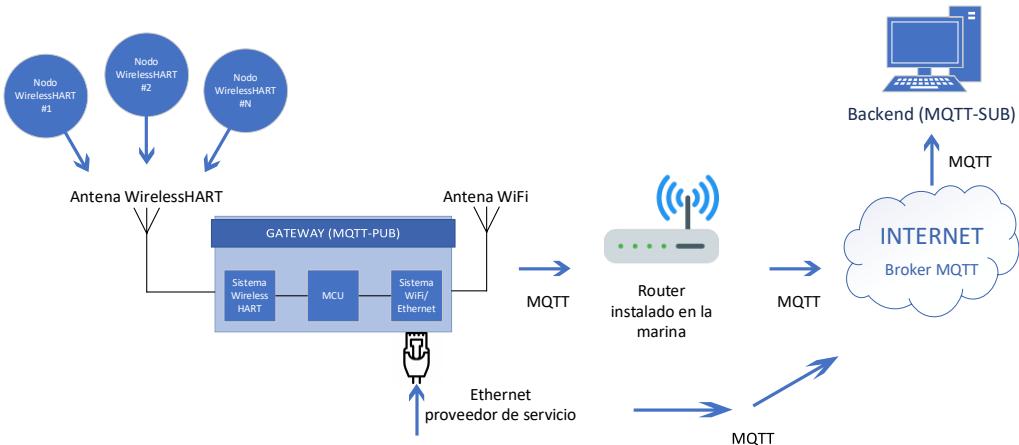


Figura 1 - Diagrama del estado actual del proyecto

Dado que, dependiendo de la marina, la instalación de un cable *Ethernet* es difícil o en algunos casos imposible, y que no siempre se dispone de un router WiFi cercano, en el proyecto del que trata esta memoria, se desarrolla una solución que mejora la actual corrigiendo estos problemas.

## 1.2. Objetivos

Teniendo en cuenta el estado actual del proyecto, el objetivo de este es diseñar, implementar y validar un *gateway* para lograr la conexión inalámbrica de una red de sensores con la tecnología WirelessHART a la nube mediante la red NB-IoT/CAT-M1 y el protocolo MQTT (Message Queuing Telemetry Transport). En concreto, la solución planteada es un *bridge* que recibe los paquetes de datos de los sensores a través del coordinador de la red WirelessHART (IEEE 802.15.4E), los procesa y, finalmente, los reenvía utilizando el protocolo MQTT utilizando una conexión NB-IoT o CAT-M1 dependiendo de la región donde opere el dispositivo.

## 1.3. Requerimientos

Considerando los objetivos del proyecto, descritos en el apartado anterior, los requerimientos funcionales del proyecto son los siguientes:

- Para la red WirelessHART se requiere que el *gateway* pueda soportar hasta 100 nodos.
- Para la red NB-IoT/CAT-M1 se pide soportar las bandas principales a nivel mundial (B1, B2, B3, B4, B5, B8, B12, B13, B14, B17, B18, B19, B20, B25, B26, B28 y B66).
- La tarjeta SIM deberá ser reemplazable, de modo que se requerirá un porta-SIM como alternativa a los sistemas eSIM, que van soldados directamente a la PCB y dificultan el cambio de tarjeta.
- El sistema incluirá una radio GPS para obtener una sincronización de tiempo global con un error pequeño (menor que 1ms). Este tiempo servirá para poner los *timestamp* de los paquetes que se envíen desde el *gateway* hacia el *backend*.
- Para el transporte de información se utilizará el protocolo MQTT, de modo que será necesario incorporar un MCU para recibir, procesar y reenviar hacia Internet a través de la red NB-IoT/CAT-M1 los datos de la red WirelessHART.

- El sistema irá alimentado a 12VDC. Para ello se utilizará una fuente de alimentación externa que convierta los 220VAC a 12VDC.
- La electrónica de la placa irá alimentada a 3.3 VDC. Para ello se usará un conversor de voltaje que regule el voltaje de 12VDC a 3.3 VDC.
- El sistema incluirá una batería de *backup* que deberá soportar el funcionamiento del sistema durante un mínimo de 48 horas, lo suficiente para soportar un fin de semana sin la alimentación principal. La batería se cargará mientras el sistema disponga de una fuente de alimentación principal.
- El regulador de voltaje debe mantener la tensión a 3.3 VDC cuando la fuente de alimentación sea la batería de *backup*.
- La electrónica (incluida la fuente de alimentación) irá encerrada en una carcasa IP-67 para protegerla de los elementos externos.
- Las antenas WirelessHART, GPS y NB-IoT irán instaladas en el exterior de la caja a través de un pasamuros. Por tanto, la PCB utilizará conectores de radiofrecuencia SMA o u.FL.
- A pesar de que el sistema irá conectado a la red eléctrica existe la posibilidad de desplegar el sistema en entornos donde el acceso a la red eléctrica esté limitado. Por tanto, a pesar de no ser un aspecto crítico, se tendrá en cuenta el bajo consumo a la hora de escoger y diseñar los componentes del sistema para facilitar su alimentación a través de fuentes de energía alternativas como paneles solares.
- Se incluirán LEDS y botones programables para facilitar el proceso de desarrollo y validación.

## 1.4. Diagrama de bloques

Teniendo en cuenta los requerimientos descritos en el apartado anterior, a continuación, se muestra el diagrama de bloques del proyecto:

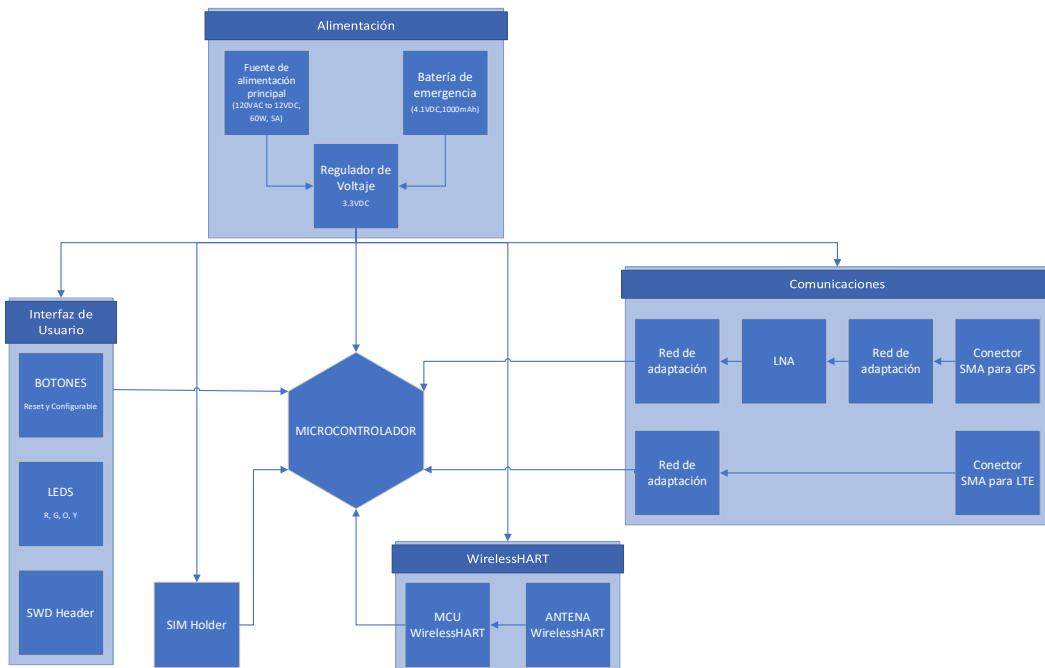


Figura 2 - Diagrama de bloques del proyecto

Como se puede observar en la Figura 2, la alimentación principal es de 12VDC. El circuito que se va a diseñar funciona a 3.3 VDC de modo que se ha de convertir la tensión de 12 VDC a 3.3 VDC. Para ello se usará un regulador de tensión; este regulador debe gestionar también el voltaje de la batería de emergencia y debe mantenerla cargada en todo momento. Si la fuente de alimentación principal se desconecta, el regulador debe informar al microcontrolador y alimentar el sistema con la batería.

El sistema GPS de la placa consta de una red de adaptación de impedancias, un LNA (Low Noise Amplifier) y un conector SMA donde se conectarán las antenas posteriormente. La red LTE consta de una red de adaptación de impedancias y un conector SMA para la antena.

Para tener conexión LTE es necesario insertar una tarjeta con contrato con un proveedor de servicios. Como se ha mencionado en el apartado anterior, se desea que esta SIM sea reemplazable. Por ello, se instalará un *SIM-Holder* donde se insertará la SIM y ésta podrá ser reemplazada por el usuario en cualquier momento.

Para la comunicación con los sensores con tecnología WirelessHART, hace falta una radio y un MCU (Microcontroller Unit) que implemente el protocolo WirelessHART. Se pretende optar por un SOM que tenga ambos componentes integrados. La comunicación del sistema WirelessHART con el microcontrolador se efectuará mediante una interfaz UART. También se pretende que el microcontrolador sea un SOM que incorpore tanto el MCU como la radio NB-IoT/CAT-M1.

La programación del microcontrolador y los cambios que se hagan en el firmware se subirán a la placa mediante un conector SWD (Serial Wire Debug). El usuario también dispondrá de LEDs programables. El propósito de los LEDs es que se pueda conocer el estado de la placa o que se programen para depurar el sistema. Se van a instalar dos botones, uno de ellos tendrá la función de reiniciar la placa de manera general y el otro será programable por el usuario.

## 1.5. Organización de la memoria

Tras este primer apartado introductorio, la memoria consta de otros seis capítulos principales que se describen a continuación:

- En el capítulo 2 se introduce el estado del arte del proyecto. Éste consta de una introducción a las tecnologías necesarias para el desarrollo y diseño de la solución final del proyecto.
- En el capítulo 3 se explica el diseño y el desarrollo del hardware de la solución del proyecto donde además se ha realizado una estimación del consumo eléctrico del sistema para demostrar que la batería escogida cumple con los requerimientos y una estimación del coste del producto final en función de la cantidad de unidades que se produzcan.
- En el capítulo 4 se explica el plan de validación de la parte electrónica/hardware del proyecto. Para ello se harán dos validaciones, la eléctrica y la lógica.
- En el capítulo 5 se describe el desarrollo del *firmware* que se cargará en el hardware para realizar la funcionalidad de reenviar los paquetes de la red WirelessHART al backend utilizando la conexión NB-IoT/CAT-M1 y el protocolo MQTT.
- En el capítulo 6 se explica cómo hacer la validación de la solución una vez esta se haya producido, donde se cargará el *firmware* para validar el correcto funcionamiento del programa y de los distintos módulos que conforman el sistema.
- En el capítulo 7 se exponen las conclusiones y se explica el trabajo futuro.

## 2. Estado del arte

A continuación, se presenta el diagrama que describe el entorno donde se desarrolla el proyecto.

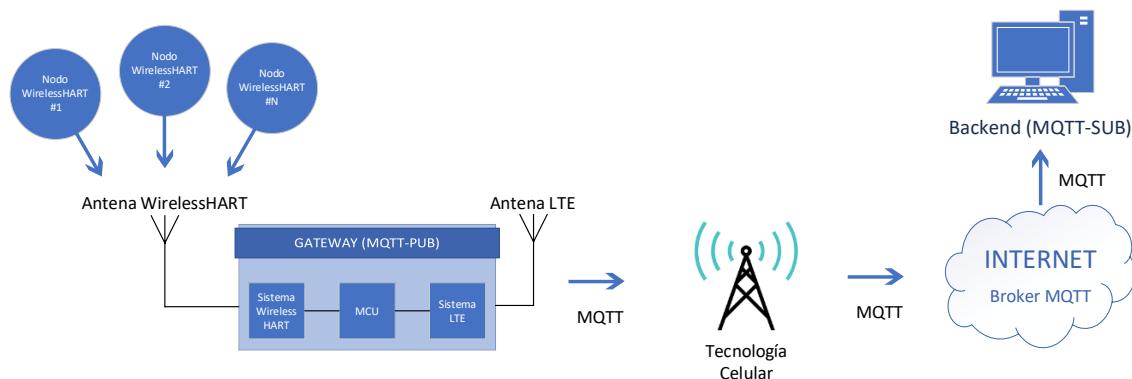


Figura 3 - Diagrama del entorno

Los sensores de la marina deben comunicarse con el *gateway* y este redirigir los paquetes a Internet. La tecnología que se está usando para desempeñar esta función es WirelessHART, una red mallada inalámbrica segura, de recuperación y optimización automáticas.

Para dotar de conectividad a Internet a la pasarela se pretende usar una tecnología celular. Estas tecnologías y sus alternativas serán vistas en el apartado siguiente y en las conclusiones de este capítulo se explicará qué tecnología se ha escogido y por qué.

Para el envío de los datos al *backend* se debe usar un protocolo de transporte adecuado. Se quiere acceder a los datos en todo momento y posiblemente poderse comunicar de vuelta con el *gateway* desde el *backend*.

### 2.1. Comunicaciones WPAN/LPWAN

Como se ha mencionado anteriormente, se necesita una tecnología que permita comunicar los sensores con el *gateway*. En el entorno de las IoT las tipologías de redes más comunes para comunicar dispositivos son las Wireless Personal Area Networks (WPAN) y las Low-Power Wide-Area Networks (LPWAN). En esta sección se va a introducir el funcionamiento y características de estas dos tipologías de red y posteriormente se darán ejemplos de tecnologías que se usan actualmente en el entorno IoT. Finalmente, se explicará la elección de las tecnologías aplicadas en el proyecto y por qué se han escogido.

Para describir las diferentes tecnologías y para elegir de entre ellas las que se pueden aplicar en el proyecto, se usarán las características principales que definen una red de comunicaciones inalámbrica, estas son:

- **Ancho de banda:** Se trata de la capacidad máxima y la cantidad de datos que se pueden transmitir a través de una conexión.
- **Distancia de enlace:** Distancia máxima a la que se puede producir un enlace entre los nodos de una red. Normalmente se mide en metros o kilómetros.
- **Velocidad de los datos:** Velocidad en bps (bits por segundo) a la que van los datos en los enlaces de la red.

- **Arquitectura:** La arquitectura de una red es el conjunto de protocolos que facilitan la comunicación entre dispositivo de la red.
- **Topología:** La topología de una red se define como un mapa físico o lógico de una red para intercambiar datos, es decir, es la forma en la que está diseñada la red a nivel del plano físico o lógico.
- **Regulación:** Regulación, normativas y legislación en cada país, que definen el marco en el que pueden funcionar estas tecnologías a nivel de tiempo de transmisión, ancho de banda, bandas de frecuencia, potencia, etc.

### 2.1.1. Tecnologías LPWAN

Las redes LPWAN [3] (Low-Power Wide-Area Networks), como indica su nombre, son redes con un gasto energético muy bajo. Los nodos LPWAN pueden llegar a durar 20 años activos con baterías pequeñas y de bajo coste. Estas redes tienen un ancho de banda pequeño, esto hace que tengan una velocidad de transmisión lenta (entre 0.3 kbit/s y 50 kbit/s) y por tanto que la latencia sea alta comparada con las redes celulares (5G, 4G, 3G, etc.). En cuanto a la distancia de la comunicación, son redes diseñadas para comunicaciones de largo alcance en las que los nodos pueden llegar a estar separados por varios kilómetros en áreas urbanas (en áreas rurales incluso decenas de kilómetros). Por último, los costes de estas redes suelen ser bajos dado que, al usar protocolos livianos, la complejidad en el diseño del hardware disminuye y esto baja los costes.

A continuación, se explican dos tecnologías LPWAN muy comunes en el sector IoT.

#### 2.1.1.1. *SigFox*

Sigfox [4] es una tecnología de red celular *one-hop* de largo alcance que utiliza Ultra-NarrowBand (UNB) con topología de estrella y que funciona en las frecuencias ISM (868 MHz en Europa y 902-928 MHz en Estados Unidos). A diferencia de otras tecnologías, SigFox tiene un límite de mensajes al día. Este límite es de 140 mensajes al día con un límite de 7 mensajes cada hora y 4 transmisiones entrantes por día. Esto es para que la red no se colapse si hay muchos nodos usando las instalaciones (antenas de SigFox). Cada mensaje puede tener hasta 12 bytes de longitud. La red tiene un *bit rate* muy bajo, de 100bps aproximadamente.

El modelo de negocio de SigFox es igual que el de cualquier operador de red, alquila su infraestructura a los usuarios. Para ello cada nodo debe tener el módulo de radio compatible y un plan de suscripción renovable en cada dispositivo IoT. SigFox gestiona la comunicación entre el dispositivo y el servidor de modo que la integración del módulo de radio es muy sencilla para los desarrolladores. La API para interactuar con el módulo de radio la proporciona la empresa y no es necesaria ninguna configuración.

Al tener un bitrate tan bajo su uso está destinado a soluciones que no necesiten velocidad en transmisión y que no necesiten transmitir mucha cantidad de datos o que no dependan de la latencia de la comunicación. También se usa en soluciones que necesiten comunicación one-hop de larga distancia ya que SigFox proporciona un rango de comunicación de varios kilómetros.

Si se analiza SigFox a nivel de arquitectura de capas, tenemos lo siguiente:

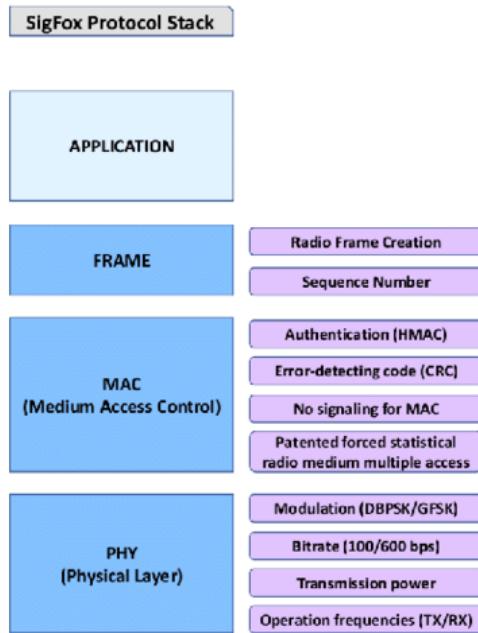


Figura 4 - Arquitectura de capas de SigFox

La arquitectura de capas de SigFox cubre de la capa 1 hasta la capa 4 del modelo OSI. SigFox está compuesto por las capas de *frame*, Medium Access Control (MAC) y la capa física (PHY). Todas estas capas están embebidas en los dispositivos con chips SigFox. A continuación, se explica la función de cada una de las capas que construyen esta tecnología.

- **PHY:** Esta capa es la encargada de decidir cómo se sintetiza la información proveniente de las capas anteriores (Aplicación, Frame y MAC). Se usa la modulación DBPSK en *uplink* y la modulación GFSK en *downlink*. El bitrate, como se ha mencionado anteriormente, es de 100 bps o 600 bps dependiendo de la región donde se esté operando el dispositivo. La potencia de transmisión es de alrededor de 14 dBm en Europa y de 22 dBm en América (tanto norte como sur). En esta capa también se gestionan las frecuencias en las que se opera, siendo estas 900 MHz en regiones FCC y 868 MHz en Europa.
- **MAC:** Esta capa añade campos para la autenticación del dispositivo (HMAC) y códigos para detectar errores (CRC). SigFox no implementa señalización para MAC, esto implica que los dispositivos no estén sincronizados con la red de modo que rara vez pueden acceder de forma aleatoria al medio físico.
- **Frame (Trama):** Una vez se recibe el *payload* desde la capa de Aplicación, se genera la trama radio o radio *frame*, donde se adjunta un numero de secuencia sistemáticamente.

#### 2.1.1.2. LoRa

LoRa [5] (Long Range) es una tecnología de red celular *one-hop* de largo alcance derivada de la tecnología Chirp Spread Spectrum (CSS). Opera en las frecuencias ISM (868 MHz en Europa y 902-928 MHz en Estados Unidos). Esta tecnología destaca por tener un buen ancho de banda, gran alcance, buen nivel de seguridad y resistencia a las interferencias. LoRa dispone de canales con anchos de banda de 125 kHz, 250 kHz o 500 kHz dependiendo de la región donde se esté usando el sistema. El tener estos anchos de banda permite transmitir a velocidades de unos 50 kbps, superiores a las de SigFox. Esta tecnología puede transmitir en rangos de hasta 10 Km en áreas rurales y 3 Km en áreas urbanas gracias al Chirp Spread Spectrum. El consumo energético

de los módulos de LoRa es algo superior al de los de SigFox, aun así, la duración de los dispositivos que implementan LoRaWAN con baterías pequeñas puede llegar a ser de 10 años. LoRaWAN asegura una comunicación segura entre el nodo final y el servidor ya que implementa encriptación AES-128. Hay dos maneras de usar LoRa: desplegar una red propia o utilizar la red de un operador de red LoRa.

A diferencia de SigFox, LoRa no tiene límite de mensajes, la red puede soportar millones de mensajes de miles de gateways. El modelo de negocio de lora es muy diferente el de SigFox, LoRa vende los chips de Semtech que son necesarios para crear los módulos de radio, y los fabricantes de hardware pueden vender estos módulos, pero necesitan una certificación para que estos funcionen dentro de la red. De este modo, LoRa Alliance (actuales administradores de la red), no se posiciona como proveedor de red y logra su objetivo, desarrollar un estándar (LoRaWAN) y vender chips.

Si se analiza LoRa a nivel de arquitectura de capas se observa lo siguiente:

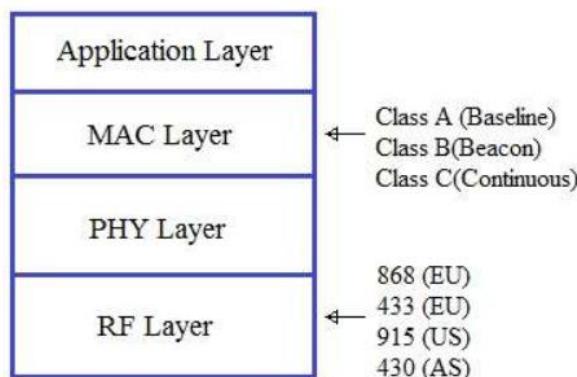


Figura 5 - Arquitectura de capas de LoRa

La arquitectura de LoRa se divide en las siguientes capas:

- **Capa RF:** Dependiendo de la región donde se esté operando el dispositivo LoRa las frecuencias de las bandas cambian. Para Europa existen dos, 868 MHz y 433 MHz, para Estados Unidos 915 MHz y, finalmente, para Asia 430 MHz.
- **Capa física (PHY):** La capa física de LoRa se encarga de construir la trama para transmitir el *payload* de la capa MAC mediante el enlace RF. También se encarga de insertar los preámbulos PHDR y PHDR\_CRC y el CRC para detectar errores en toda la trame. El campo CRC solo está disponible en las trames de mensajes *uplink*. También se usan como preámbulo, palabras de sincronización constante específicas dependiendo de la técnica de modulación usada en el momento, estas modulaciones pueden ser LORA, GFSK o FSK. Esta capa usa las frecuencias mencionadas anteriormente dependiendo de la región donde se esté operando.
- **Capa MAC:** La capa MAC de LoRa se encarga de establecer la conexión entre la capa MAC de pares, por ejemplo, entre un dispositivo final y un *gateway*. También se encarga de manejar la transmisión y recepción de comandos y datos MAC desde la capa de aplicación. La capa MAC agrega MHDR (encabezado MAC) y MIC (código de integridad del mensaje) al principio y al final del *payload*. El encabezado MAC tiene un tamaño de 1 octeto y MIC tiene un tamaño de 4 octetos. En una red LoRa hay tres clases de

dispositivo según los cuales la estructura de la trama de LoRa cambia. Estas clases de dispositivo son las siguientes:

- **Dispositivos finales de clase A:** A nivel temporal disponen de un slot *uplink* seguido de dos slots de *downlink*. El slot *uplink* se programa temporalmente basándose en las necesidades del dispositivo final, se programa aleatoriamente de forma similar al protocolo ALOHA. Es el dispositivo final que necesita menos potencia.
- **Dispositivos finales de clase B:** Esta clase de dispositivos disponen de ventanas de recepción extras a parte de los dos slots temporales de recepción que se han mencionado en los dispositivos de clase A. Los dispositivos de clase B reciben estas ventanas de recepción extras con una duración especificada en una trama *beacon* por el *gateway*. De este modo el sistema LoRa indica al servidor cuando el dispositivo final puede escuchar.
- **Dispositivos finales de clase C:** Este tipo de dispositivos pueden escuchar todo el tiempo a no ser que estén en modo transmisión. Son dispositivos ideales para usos donde se necesiten más transmisiones *downlink*. Es el tipo de dispositivo LoRa que más potencia consume dado que está permanentemente escuchando. Estos dispositivos logran tener una latencia más baja en la comunicación entre servidor y dispositivo final que la de las otras dos clases.

### 2.1.2. Tecnologías WPAN

Las redes WPAN [6] (Wireless Personal Area Network) son las que se encuentran dentro de la norma IEEE 802.15. Son redes de corto alcance (alrededor de los 10 a los 100 metros dependiendo de la tecnología). Hay algunas tecnologías que tienen más alcance que otras y por ello tienen aplicaciones específicas. Su uso más común es el de interconectar dispositivos personales y accesorios de estos dispositivos (mandos, móviles, altavoces, televisiones, consolas, etc.). La velocidad de transmisión depende de la tecnología que se use, por ejemplo, Z-Wave tiene una tasa de transmisión de 40 a 100 kbit/s y la de Bluetooth (IEEE 802.15.1) va desde los 1 Mbit/s (Bluetooth v1.2) a los 50 Mbit/s (Bluetooth v5.0). Las redes WPAN usan anchos de banda pequeños y esto hace que el consumo de energía sea bajo, lo cual hace que su uso en dispositivos IoT con baterías pequeñas sea muy adecuado. Hay tecnologías WPAN de corta distancia como Z-Wave (30 m) y de larga distancia como ZigBee (300 m). También pueden ser tecnologías que usen redes malladas como WirelessHART o *one-hop* como Bluetooth (aunque se ha desarrollado una versión *multi-hop* llamada Bluetooth Smart o “BLE”). A continuación, se procede a explicar algunas de las tecnologías WPAN más usadas en el sector IoT.

#### 2.1.2.1. WirelessHART

WirelessHART [7] es una red mallada inalámbrica segura, de recuperación y optimización automáticas. En 1986 la fundación HART estandarizó el protocolo HART, una tecnología de red que permitía comunicarse con nodos (normalmente sensores de temperatura, presión, caudal, etc.) usando lazos de corriente de 4 a 20 mA. Posteriormente, en 2007, 37 compañías de la fundación HART desarrollaron lo que hoy se conoce como WirelessHART. Este protocolo está basado en el estándar IEEE 802.15.4 e incorpora la tecnología de TSCH (Time Synchronized Channel Hopping), lo cual ha influido en la creación del estándar IEEE 802.15.4E enfocado dar soporte a redes inalámbricas de uso industrial. WirelessHART funciona en la banda de 2.4 GHz donde tiene 15 canales espaciados 5 MHz entre ellos y de 2 MHz de ancho de banda cada uno. Estos 15 canales pueden ser usados a la vez, esto significa que 15 dispositivos podrían estar

transmitiendo a la vez. Cada slot de tiempo en el que se puede transmitir es de 10 ms. La velocidad de transmisión de datos es de unos 250 kbps (4 bits por símbolo a 62'5 kBaud). Dado que se realizan transmisiones durante poco tiempo e intentando que los sensores solo transmitan cuando sea estrictamente necesario, el consumo energético de las redes WirelessHART es muy bajo y optimizado. Las baterías de los nodos WirelessHART pueden llegar a durar de 5 a 7 años.

La siguiente figura representa una red WirelessHART y los dispositivos que la componen.

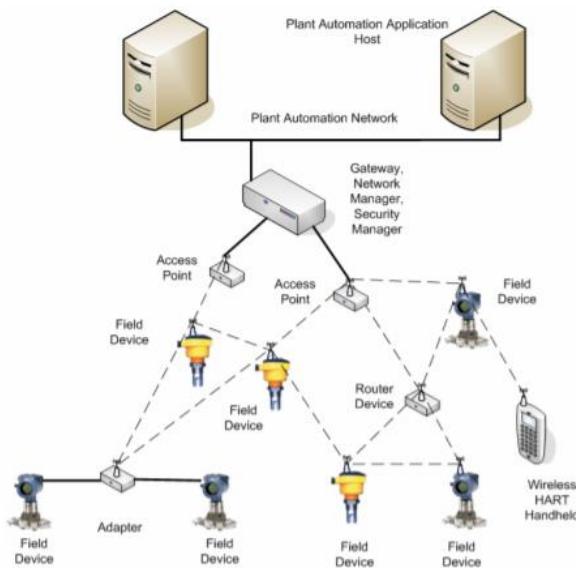


Figura 6 - Red WirelessHART

Las redes WirelessHART están compuestas de varios tipos de nodo, estos son:

- Sensores, que miden el valor deseado en el campo.
- Adaptadores, que permiten la conectividad inalámbrica con la red a sensores que no disponen de antena o que están conectados mediante cable.
- Puntos de acceso, que llevan los datos de la red hacia los *gateways*.
- *Routers*.
- *Gateways* que llevan los datos al *backend* (controlador de proceso de automatización o Aplicación en un Host).
- *Network Managers*, encargados de gestionar la red y decidir las rutas que sigue la información dentro de la red hasta los *gateways*.
- *Wireless Handhelds*, dispositivos de mano que sirven para poder configurar de manera segura los nodos de la red.

La siguiente figura es el diagrama de capas de WirelessHART. A continuación, se procede a analizar a nivel de capas el protocolo WirelessHART.

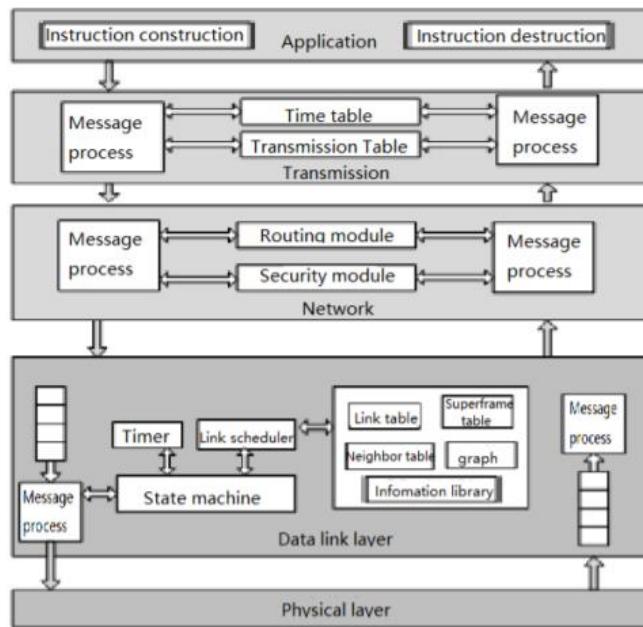


Figura 7 - Capas de WirelessHART

- **Capa física:** La capa física de WirelessHART está basada en el estándar IEEE 802.15.4 en su versión de 2006. Se trata de una capa física que funciona a 2.4 GHZ y que usa una modulación O-QPSK con DSSS (Direct Sequence Spread Spectrum). El tamaño máximo del *payload* es de 127 bytes. La potencia de transmisión es de 10 dBm y es ajustable en pasos discretos.
- **Capa de enlace de datos:** La capa de enlace de datos de WirelessHART está basada en una MAC del estándar IEEE 802.15.4-2016. WirelessHART mejora la MAC del estándar implementando slots temporales fijos de 10ms (100 time-slots por cada segundo), salto de frecuencias sincronizado y TDMA (Time Division Multiple Access). Añadiendo estas prestaciones se consiguen comunicaciones deterministas y sin colisiones. Para gestionar los slots de tiempo se usan “super tramas” (super frames) que agrupan una secuencia de ranuras temporales consecutivas. La super trama es periódica y su periodo es la longitud total de los slots que la componen. Estas supertramas se almacenan en la tabla de supertramas. En WirelessHART una transacción en un slot de tiempo está descrita por un vector con la forma siguiente: {frame id, index, type, src addr, dst addr, channel offset}, donde *frame id* identifica la supertrama, *index* es el índice del slot en la supertrama, *type* indica el tipo de slot (transmit/receive/idle), *src addr* y *dst addt* son las direcciones de los dispositivos de origen y destino respectivamente y *channel offset* es el canal lógico que se va a usar para la transacción. Si hay canales afectados por interferencias de manera consistente, WirelessHART tiene una lista negra (blacklist) donde poner estos canales para que posteriormente el administrador de red los deshabilite totalmente para la red. Para que el *channel-hopping* sea posible, cada dispositivo mantiene una tabla de canales activos actualizada. Estas tablas pueden tener menos de 16 canales debido al *blacklisting*.
- **Capa de red:** La capa de red de WirelessHART se encarga de varias tareas. Las más importantes son el enruteado de paquetes y la seguridad de la comunicación. La capa de

red, como se puede ver en la figura anterior, tiene tablas de *routing* y tablas de tiempo. Las tablas de enrutado sirven para dirigir las comunicaciones a lo largo de los gráficos de la capa de enlace de datos. Las tablas de tiempo sirven para asignar ancho de banda a servicios como la publicación de datos y la transferencia de bloques de datos. Los paquetes se encriptan usando encriptación AES de 128 bits.

- **Capa de transporte:** La capa de transporte se encarga de proveer un servicio de transporte inalámbrico y robusto hacia la capa de aplicación de WirelessHART. Desde la capa de aplicación se puede configurar el envío de ACKs por parte del receptor para que la red pueda reenviar paquetes perdidos a través de la capa de transporte.
- **Capa de aplicación:** La capa de aplicación de WirelessHART consiste en un *command builder* y un analizador (parser). Esta capa está definida en el estándar IEC\_PAS 62591-2009 [8].

#### 2.1.2.2. ZigBee

ZigBee [9] es una tecnología de red basada en el estándar IEEE 802.15.4 de redes inalámbricas de área personal (WPAN). Opera en las bandas de ISM, 2.4 GHz en todo el mundo, 915 MHz en Estados Unidos y 868 MHz en Europa. El objetivo de esta tecnología es su aplicación en entornos donde se necesiten comunicaciones seguras, con baja tasa de envío de datos y un consumo eléctrico bajo para alargar la vida útil de las baterías. El mercado está enfocando el uso de ZigBee en la domótica, ya que se trata de una tecnología con red mallada de recuperación automática y permite la interconexión de muchos más dispositivos que por ejemplo Bluetooth, lo que viene bien en el caso de querer tener muchos nodos como en una casa (bombillas, electrodomésticos, alarmas, sensores en las ventanas, etc.). Los nodos que usa esta tecnología tienen poca electrónica ya que la funcionalidad de ZigBee recae en el código del protocolo. Debido a que el código del protocolo es muy pesado los nodos acaban requiriendo de más memoria que los de otras soluciones, aunque el coste por nodo suele ser bajo. El bitrate de la red es de 250 kbps y los nodos pueden comunicarse entre sí a una distancia de unos 300 metros.

Las redes ZigBee tienen tres tipos de topología dependiendo de la disposición de los nodos. Estas tipologías son: malla, árbol o estrella. La más usada es la topología en estrella ya que permite que la red se reorganice en caso de que algún nodo falle. Los nodos de las redes de ZigBee pueden asumir tres tipos de roles como se muestra en la figura siguiente.

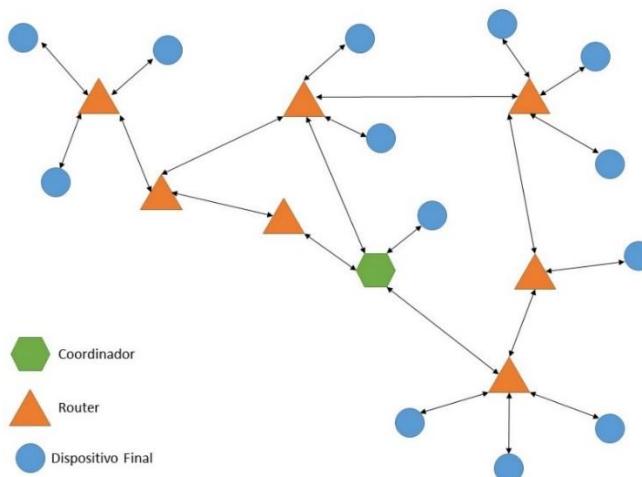


Figura 8 - Red ZigBee

La función de estos tipos de nodo es la siguiente:

- **Coordinador ZigBee, ZigBee Coordinator, ZC:** Es el dispositivo más completo y sólo existe uno por red. Se encarga de controlar la red y los caminos que deben seguir los dispositivos para conectarse entre ellos.
- **Router ZigBee, ZigBee Router, ZR:** El trabajo del router es interconectar dispositivos separados en la topología de red. Los routers de ZigBee también ofrecen un nivel de aplicación para dar la posibilidad de ejecutar código de usuario.
- **Dispositivo Final ZigBee, ZigBee End Device, ZED:** Tiene la funcionalidad necesaria para comunicarse con un nodo coordinador o con un nodo *router* pero no puede transmitir información a otros dispositivos. Estos nodos están durmiendo la mayor parte del tiempo, de modo que la vida útil de sus baterías es muy larga. Un ZED tiene muy poco requerimiento a nivel de memoria y por ello suele ser el tipo de dispositivo más barato de la red.

Si nos basamos en la funcionalidad de los dispositivos ZigBee tenemos la siguiente clasificación:

- **Dispositivos de funcionalidad completa (FFD):** Se conocen como nodos activos. Son capaces de recibir mensajes en formato IEEE802.15.4. Al tener memoria adicional pueden funcionar como Coordinadores o como Routers de la red ZigBee. También pueden ser usados en dispositivos de red que sirvan de interfaz para los usuarios.
- **Dispositivos de funcionalidad reducida (RFD):** Se conocen como nodos pasivos. Tienen la capacidad y la funcionalidad limitadas con el objetivo de conseguir un bajo coste y una gran simplicidad. Son los sensores y actuadores de la red.

A continuación, se procede a hacer un análisis de la arquitectura de capas de ZigBee. La figura siguiente muestra cómo se dividen las capas en ZigBee.

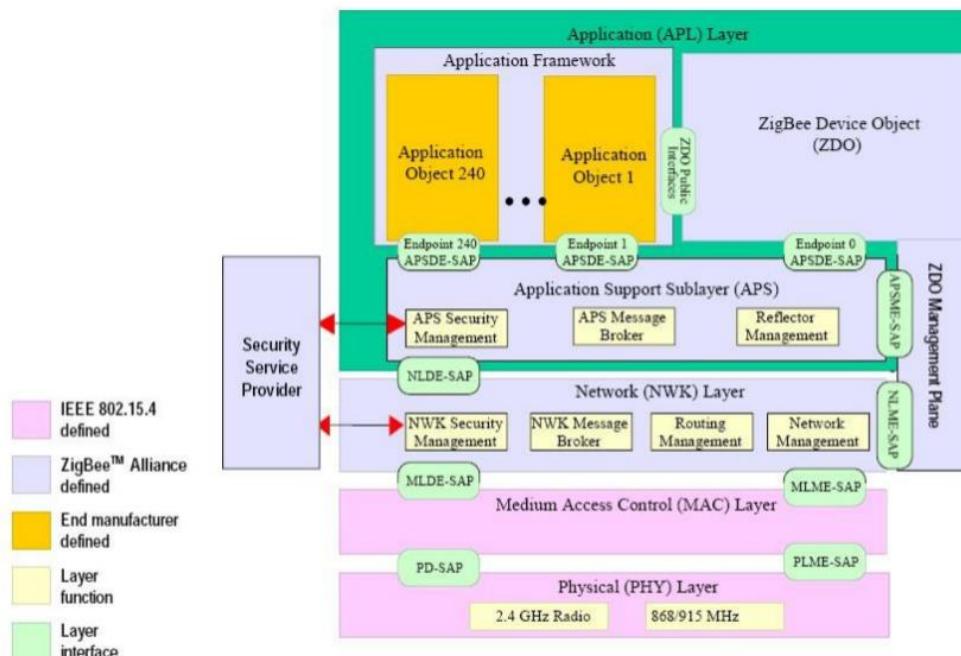


Figura 9 - Arquitectura de capas de ZigBee

- **Capa física:** La capa física de ZigBee está definida por el estándar IEEE 802.15.4. La capa física es la encargada de la activación y la desactivación de la transmisión y la recepción de datos. También es donde se realiza la selección de los canales. Los canales de ZigBee se dividen en tres bandas: 2.4 GHz donde hay 16 canales disponibles, 915 MHz donde hay 10 canales y 868 MHz donde hay un canal. Dependiendo de la región donde se use el producto ZigBee se usará una banda u otra (2.4 GHz en todo el mundo, 915 MHz en Estados Unidos y 868 MHz en Europa). En esta capa se realiza la detección de energía del canal que se está usando para poder rechazar los canales con interferencias. La modulación de ZigBee es O-QPSK para la banda de 2.4 GHz y BPSK para las bandas de 868-928 MHz.
- **Capa MAC:** La capa MAC está definida por el estándar IEEE 802.15.4. Las principales funciones de esta capa son el manejo y la gestión de los datos. En esta capa se gestionan las transmisiones de radiofrecuencia entre dispositivos (punto a punto) incluyendo servicios como la retransmisión si no llega algún ACK y técnicas de evitación de colisiones (CSMA-CA). Esta capa es también responsable de la generación y gestión de tramas *beacon*. La capa MAC implementa GTS (Guaranteed Time Slot) y se encarga de administrar la reserva de los slots para su uso. Como se ha mencionado anteriormente, en la capa MAC se validan las tramas y se realiza el *acknowledgement* de la información.
- **Capa de red:** La capa de red de ZigBee es la encargada de enrutar los paquetes y establecer el tipo de topología de red (estrella, *mesh* o árbol). Cuando un coordinador intenta establecer una red ZigBee, se hace un escaneo energético para encontrar el mejor canal RF para la red. Cuando se escoge el canal, el coordinador asigna un PAN-ID que se aplicará a todo nodo que se conecte o forme parte de la red. El PAN-ID es un número de 16 bits que sirve de identificador de red. Un nodo se puede comunicar dentro de una red después de haber pasado por un proceso de asociación. Este proceso sirve para unir un nodo a un “padre” (*parent*). Los nodos que no tienen padre se consideran huérfanos. Normalmente los nodos se vuelven huérfanos cuando son dispositivos móviles y pierden conexión debido al rango. En estos casos se hace un escaneado haciendo un broadcast de una trama de notificación de huérfano (*orphan notification command frame*) que ayuda a los nodos huérfanos a encontrar a su padre. Cuando un parent recibe un comando de este tipo, responde el nodo que lo ha enviado de modo que se pueda unir de nuevo. La capa de red se asegura que la comunicación sea autenticada y confidencial. También se encarga de la seguridad en la comunicación.
- **Capa de aplicación:** La capa de aplicación de ZigBee está dividida en tres partes: Application Framework, ZigBee Device Object (ZDO) y Application Support Sublayer (APS).
  - **Application Framework:** El Application Framework depende del vendedor del producto ZigBee. El fabricante define cómo se implementan los *end points* y cómo se hacen las peticiones y las confirmaciones de los datos.
  - **ZigBee Device Object (ZDO):** Es el protocolo encargado de la gestión de los dispositivos, las claves de seguridad y las políticas. Es el responsable de definir el rol de los dispositivos como coordinador o dispositivo final, de hacer el descubrimiento de dispositivos *one-hop* en la red y la identificación de los servicios que estos ofrecen. Después de descubrirlos crea vínculos seguros con ellos y responde a sus peticiones como se requiera.

- **Application Support Sublayer (APS):** Esta capa es responsable de filtrar paquetes para los dispositivos finales que no están registrados o de perfiles que no cuadran con la tabla de vinculación. Se filtran para evitar duplicidades dado que en una red que solicita reenvíos automáticamente es algo muy común. Esta capa está involucrada en mantener la tabla de vinculación (binding table). Esta tabla asocia MACs de 64 bits con direcciones de red ZigBee de 16. Por tanto, esta capa se encarga de mantener las *binding tables*, definir las direcciones, mapearlas y manejarlas.

#### 2.1.2.3. Bluetooth y Bluetooth Low Energy (BLE)

Bluetooth [10] es un tipo de red WPAN que funciona en la banda ISM de 2.4 GHz destinada a facilitar la comunicación entre dispositivos móviles, eliminar cables y conectores entre dispositivos y posibilitar la creación de redes inalámbricas para facilitar la sincronización de datos entre equipos personales. Esta tecnología fue introducida por primera vez en 1989 y ha tenido 5 versiones distintas hasta el momento. En cada versión se ha mejorado, entre otras cosas, la velocidad de transmisión y el rango de la comunicación. En la versión 5.0 de Bluetooth llega a una velocidad de 50 Mbps y un rango de 100 metros que, comparado con los 1 Mbps y 1 m de la primera versión, supone un gran avance. Bluetooth fragmenta los datos en paquetes ya estos son enviados a través de uno de los 79 canales disponibles de 1 MHz de ancho de banda.

En el año 2011 se creó un subtipo de Bluetooth, la versión 4.0, llamado Bluetooth Low Energy (BLE), el cual optimiza el consumo energético y hace mucho más viable usar Bluetooth en aplicaciones M2M o en soluciones IoT. Las baterías de los dispositivos que implementan BLE pueden durar 4 o 5 años gracias a la mejoría en el consumo de energía. BLE opera en la misma banda que Bluetooth. Otra mejoría notable es la velocidad con la que BLE establece conexión. BLE tarda unos cuantos milisegundos establecer conexión frente a los aproximadamente 100 milisegundos que tarda Bluetooth. Otro de los cambios importantes es la posibilidad de crear una malla (*mesh*) entre dispositivos, al contrario que Bluetooth que es una tecnología *one-hop*. Estas mejoras han hecho que BLE sea una opción viable para su aplicación en la automatización de viviendas e industrias.

La siguiente imagen describe BLE a nivel de su arquitectura de capas:

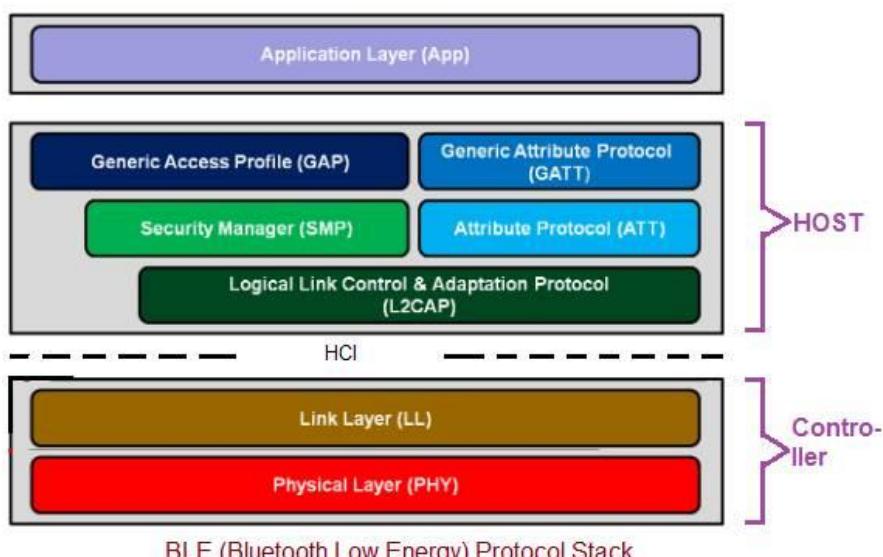


Figura 10 - Arquitectura de capas de BLE

- **Capa Física:** BLE usa modulación GFSK y opera en la banda de 2.4 GHz. Usa un transceptor con salto de frecuencia para cambiar de canal. Existen dos esquemas de modulación que permiten el envío de 1 Msym/s (1 Mbps) o 2 Msym/s (2 Mbps). Existen dos tipos de capa física para BLE, la *coded* y la *uncoded*. La capa física *coded* se introdujo en la versión 5.0 de Bluetooth y permite rangos de comunicación más extensos manteniendo la BER máxima gracias a que tiene un mejor sistema de detección de errores de bit y de corrección de estos. En ambos modos de capa física se emplea una topología Time Division Duplex (TDD).
  - **Capa de enlace de datos:** La capa de enlace de datos de BLE es responsable de avisar, crear, escanear y mantener las conexiones. Los dispositivos BLE tienen distintos roles, los más comunes son: Advertiser/Scanner, Slave/Master y Broadcaster/Observer. Las operaciones de la capa de enlace de BLE se pueden describir con el diagrama de estados siguiente:

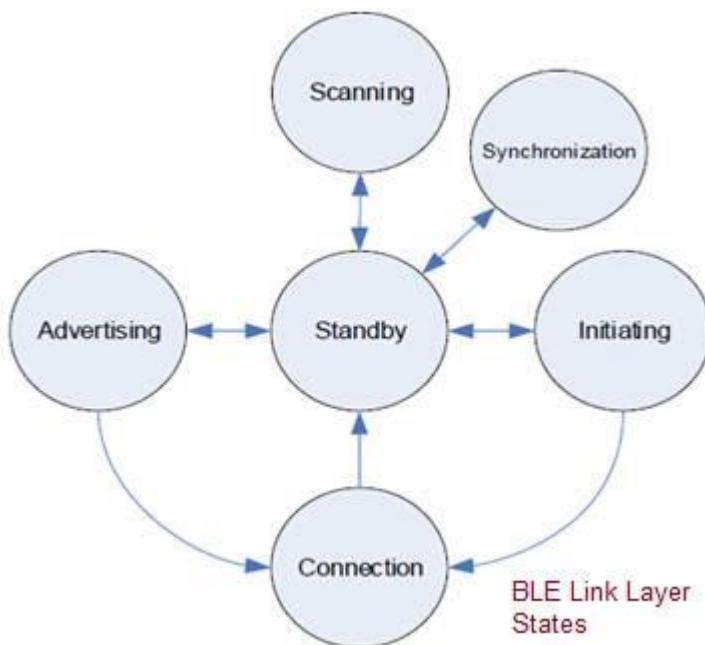


Figura 11 - Diagrama de estados de la capa DLL de BLE

- **Standby** en este estado no se recibe ni se envía ningún paquete. Se puede entrar en este estado desde cualquiera de los otros estados.
  - **Advertising:** El nodo que se encuentra en este estado es llamado Advertiser. Se puede alcanzar este estado desde el estado de Standby. DLL transmitirá Advertising Packets y escuchará y responderá a las respuestas ocasionadas del envío inicial.
  - **Scanning:** En este estado se escuchan los Advertising Packets enviados desde los nodos Advertisers. Los nodos en este estado son llamados Scanners. Se puede acceder a este estado desde el estado Standby.
  - **Initiating:** Los nodos en este estado se llaman Iniciadores (initiators). Se puede acceder a este estado desde Standby. Se escuchan los Advertising Packets de determinados dispositivos BLE. También se responde a estos paquetes para iniciar una conexión desde otro dispositivo.

- **Connection:** Se puede acceder a este estado tanto desde Initiating State como desde Standby State. En este estado hay dos roles para los nodos, pueden ser *master/slave*. Cuando se entra a este estado desde Initiating State, será *master*. Si se entra desde Advertising State, será *slave*. Cuando se es máster, la capa de enlace de datos se comunicará con el dispositivo en el rol de *slave* y definirá los tiempos de transmisión.
- **Synchronization:** Se puede llegar a este estado desde el estado Standby. En este estado, se escuchan los paquetes de canal periódicos que llegan de un dispositivo específico que transmite Advertising Packets periódicamente.
- **HCI:** Da comunicación entre el controlador y el host mediante tipos de interfaz estándares. Se puede implementar usando API o mediante interfaces como UART/SPI/USB. Los comandos y eventos de HCI están definidos en las especificaciones de Bluetooth.
- **L2CAP:** Esta capa ofrece los servicios de encapsulamiento de paquetes para las capas superiores. Esto permite la comunicación lógica entre dispositivos finales.
- **SMP:** La capa Security Manager provee métodos de emparejamiento y distribución de claves. Ofrece servicios a otras capas del *stack* para así proporcionar una conexión segura y un intercambio de datos privado y seguro entre dispositivos BLE.
- **ATT:** Esta capa permite que el dispositivo BLE exponga ciertos atributos.
- **GAP:** Esta capa hace de interfaz directamente con la capa de aplicación y con los perfiles que haya en ella. Se encarga del descubrimiento de dispositivos y de cuestiones relacionadas con servicios de conexión para el dispositivo BLE. También se encarga de la iniciación de las características de seguridad.
- **GATT:** Esta capa especifica los procedimientos para usar ATT. Las comunicaciones de datos entre dispositivos BLE se manejan a través de estos procedimientos. Las aplicaciones o los perfiles utilizarán directamente GATT.
- **Capa de aplicación:** Las capas del *stack* interactúan con aplicaciones o perfiles de la capa de aplicación. La interoperabilidad de las aplicaciones de Bluetooth se consigue mediante perfiles Bluetooth. Estos perfiles definen la interacción vertical entre las capas, así como las interacciones *peer-to-peer* entre capas específicas de dispositivos diferentes. Un perfil está compuesto de uno o más servicios para abordar un caso de uso particular. Un servicio consta de características o referencias a otros servicios. Todos los perfiles o las aplicaciones se ejecutan sobre las capas GAP/GATT de la pila de protocolo de BLE. Esta capa maneja el descubrimiento de dispositivos y servicios relacionados con la conexión.

## 2.2. Comunicaciones celulares

Una vez los paquetes de información de los sensores han llegado al *gateway*, es necesaria una tecnología celular que dé conexión a internet a la pasarela para que ésta pueda direccionar los paquetes hacia el *backend* a través de internet usando un protocolo de transporte. A continuación, se introducirán las redes celulares y posteriormente se presentarán dos de las tecnologías celulares más usadas en el mundo IoT.

Las tecnologías celulares, como indica su nombre, están basadas en células normalmente hexagonales en donde se hallan emisores-receptores centrales que permiten una conexión permanente a pesar de que el usuario se mueva. Esto es posible, ya que antes de salir de una

celda el dispositivo móvil, se pone en contacto con la estación base siguiente, y una vez tiene el canal y la frecuencia correctas, se desconecta de la anterior, esto es llamado *handover*.

El tamaño de las células difiere entre áreas rurales o urbanas y cada una de las células tiene una frecuencia distinta para evitar interferencias. La imagen siguiente muestra una posible distribución de frecuencias y la disposición de las celdas y las BTS (Base Transceiver Station) en una red celular.

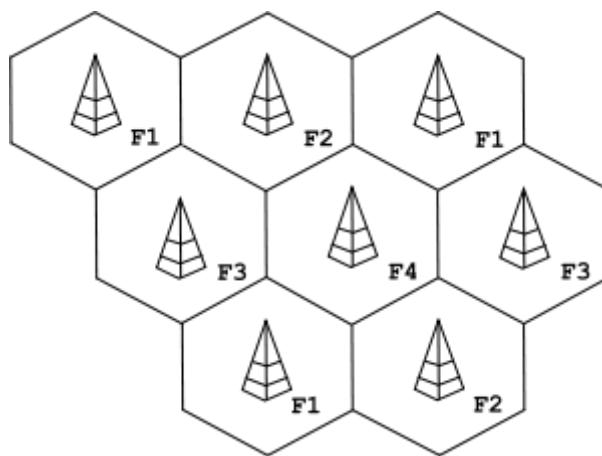


Figura 12 - Red de celdas

Para que las células no interfieran entre si hay diferentes protocolos de acceso al medio. Estos protocolos pueden combinarse para obtener un número mayor de canales y así permitir un número de usuarios mayor por celda. Los protocolos de acceso al medio más usados en tecnología celular son los siguientes:

- **Frecuency Division Multiple Access (FDMA):** En la figura anterior cada celda tiene una frecuencia distinta a las adyacentes. Sólo se puede rehusar la frecuencia de una celda cuando hay una en medio con frecuencia distinta. Aun así, puede haber interferencias, es por esto por lo que se suele combinar FDMA con otros protocolos de acceso al medio como CDMA o TDMA.
- **Code Division Multiple Access (CDMA):** En CDMA se asigna un código único para cada usuario conectado a la red. Estos códigos son ortogonales entre sí. Aunque el receptor capte combinaciones lineales de códigos de dos usuarios o más, si conoce el código de transmisión del usuario de interés y realiza el producto escalar de la señal con el código de este usuario, podrá aislar sus datos de los del resto de usuarios. Esto se da porque el producto escalar entre vectores ortogonales es cero, luego las señales de los usuarios que no interesan serían anuladas.
- **Time Division Multiple Access (TDMA):** TDMA consiste en la división del continuo temporal en ranuras o slots (time-slots) de tiempo. En cada segundo hay una cantidad de ranuras de tiempo que en la práctica se convierten en canales. Cada usuario de una red con TDMA tiene signado uno de estos slots temporales para realizar su transmisión. EN este caso no hay una división física de la señal, en el caso de que dos usuarios transmitan a la vez puede haber interferencias. TDMA se suele usar en conjunto con FDMA o CDMA en sistemas de comunicaciones móviles.
- **Polarization Division Multiple Access (PDMA):** En PDMA se usan antenas con polarización diferente de modo que los vínculos de comunicación se establezcan entre usuarios con la misma polarización y no haya interferencias con las comunicaciones que

se dan con una polarización diferente. Este tipo de método de acceso al medio permite la reutilización de frecuencias.

La familia de tecnologías celulares de 3GPP (3rd Generation Partnership Project) ha ido evolucionando. Estas generaciones se conocen como 1G, 2G, 2.5G, 3G, 4G y 5G. Las características principales de cada una de estas evoluciones son las siguientes:

- **Sistemas 1G:** Transmisión analógica mediante modulación FM y sólo útil para aplicaciones de voz.
- **Sistemas 2G (GSM):** Velocidad de transmisión de 9.6 kbps. Transmisión digital de datos SMS y voz, usado para aplicaciones de telefonía y PCs. Transmite a partir de una conexión permanente (flujo de datos permanente).
- **Sistemas 2.5G (GPRS):** Velocidad de transmisión de 56 kbps a 114 kbps. Transmisión digital de voz y datos, pero con más velocidad, y empaquetado de los datos con protocolo IP. No necesita mantener una conexión permanente entre transmisor-receptor, envía la información en paquetes. Se puede acceder a internet a través de esta red gracias a que usa el protocolo IP. Transmisión bidireccional de los datos (cliente-servidor, servidor-cliente).
- **Sistemas 3G (UMTS):** Velocidad de transmisión de 384 kbps a 2 Mbps. Transmisión digital de voz y datos a más velocidad que 2.5G y empaquetado de datos con protocolo IP que permite la conexión a internet. Proporciona lo mismo que 2.5G, pero a más velocidad. Permite el *roaming* global.
- **Sistemas 4G (LTE):** Velocidad de transmisión de 50 Mbps a 100 Mbps. Proporciona lo mismo que 3G, pero a mucha más velocidad y mejora la compatibilidad y movilidad entre los sistemas de modo que se pueda utilizar el servicio de banda ancha en cualquier momento y lugar.
- **Sistemas 5G:** Velocidad de transmisión de entre 50 Mbps y 4 Gbps. Transmisión digital de voz y datos a gran velocidad. Tecnología MIMO mejorada, conectividad sin fisuras y de baja latencia muy útil para aplicaciones IoT como los coches autónomos, manipulación de máquinas a distancia, *Smart Cities*, etc.

Las tecnologías celulares que se explican a continuación están adaptadas para su uso en casos dónde los dispositivos no transmiten mucha cantidad de datos y no dependen de la latencia de la red ni de la velocidad de transmisión para operar correctamente. Otro punto importante de estas redes es el bajo consumo energético que tienen que hace que los nodos puedan funcionar durante periodos de tiempo muy largos sin necesitar un cambio de batería.

### 2.2.1. NB-IoT

Narrow Band IoT o NB-IoT [11] es un estándar abierto de tecnología celular desarrollada con estándares de LPWAN por 3GPP (3rd Generation Partnership Project) destinada a conectar a internet objetos cotidianos que no transmitan grandes volúmenes de datos. Estos dispositivos suelen transmitir con poca frecuencia y además la transmisión se hace con poca potencia esto hace que las baterías de los nodos puedan durar hasta 10 años. NB-IoT se implementa en la capa de acceso de red y aprovecha las instalaciones de las redes móviles existentes. Esto hace que usar esta tecnología solo suponga un contrato con un proveedor de servicios y un transceptor con tecnología celular, lo que aminora los costes del despliegue. NB-IoT aprovecha el espectro de GSM y LTE usando frecuencias de portadora en la banda de GSM o bien usando un bloque de recursos no utilizados dentro de la banda de guarda de LTE. El ancho de banda tanto de *downlink* como de *uplink* es de unos 180 khz Las características que más destacan de esta tecnología son

su cobertura, (que es 20 dB mejor que la cobertura de GSM), su penetración en interiores y bajo tierra y la cantidad de conexiones que permite realizar por celda. NB-IoT permite la conexión de alrededor de unos 100.000 dispositivos por celda simultáneamente. Los enlaces tienen rangos de hasta 20 km en áreas rurales y 5 km en áreas urbanas. La velocidad de los enlaces de subida y de bajada oscila entre los 160 kbps y los 250 kbps. Esto hace que la latencia de la red sea de unos 1.6 a 10 segundos. Como la aplicación de esta tecnología suele darse en casos donde los nodos transmiten información cada cierto tiempo, y la cantidad de información transmitida no es grande, el hecho de que la latencia sea alta no es un problema ya que se prioriza la cobertura y la llegada de los datos de forma segura al destino.

La figura siguiente muestra un análisis e la arquitectura de capas de NB-IoT. A continuación, se procederá a comentar la figura y las capas que conforman NB-IoT.

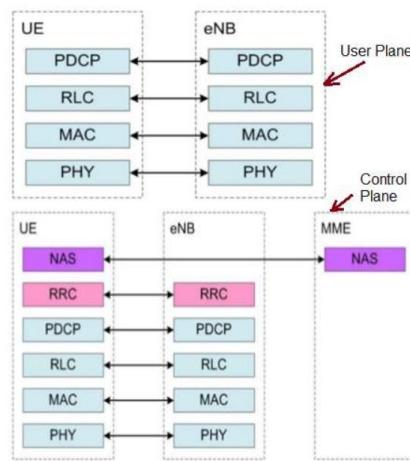


Figura 13 - Arquitectura de capas de NB-IoT

Las capas de NB-IoT y su función son las siguientes:

- **Capa Física (PHY):** La capa física permite la comunicación entre eNB y UE. También hace posible el transporte de datos de y hacia otras capas superiores. Se encarga de la detección de errores, FEC, procesado de antena, sincronización, etc. La capa física consiste en señales y canales físicos. Las señales físicas se usan para la sincronización, la identificación de celdas y estimación de canales. Los canales físicos se usan para controlar el transporte, programación y procesado del *payload* de usuario de capas más altas. La modulación de NB-IoT es OFDMA o 16 QAM para *downlink* y SC-FDMA para *uplink*.
- **Capa MAC:** La capa MAC se encarga de los mensajes relacionados con el acceso celular entre el UE y la red. Además, la capa MAC realiza las siguientes funciones: mapeo de canales lógicos a canales de transporte, multiplexación de MAC SDU desde uno o diferentes canales lógicos en bloques de transporte para ser entregados a la capa física en el lado UE, corrección de errores mediante retransmisión HARQ, manejo de prioridades entre UEs mediante programación dinámica, priorización de canales lógicos, selección de formato de transporte y selección de tamaño de TB (unidad básica de intercambio entre la capa MAC y la capa física).
- **Capa RLC:** La capa RLC se encarga de la transferencia de PDUs de capas superiores, de la corrección de errores mediante ARQ (solo para transferencia de datos AM), de la concatenación, segmentación y reensamblaje de SDUs de RLC (UM y AM), realiza la re-

segmentación de PDU de datos RLC (AM), también reordena las PDU de datos RLC (UM y AM), detecta los duplicados (UM y AM) y realiza el descarte de SDUs de RLC (UM y AM). La capa RLC también se encarga de la detección y recuperación de errores de protocolo. Hay tres modos de transmisión admitidos por RLC: modo *transparent* (adecuado para transmitir voz), modo *unacknowledged* (adecuado para transmitir tráfico continuo) y modo *acknowledged* (adecuado para transportar tráfico TCP).

- **Capa PDCP:** La capa PDCP se encarga de lo siguiente: transferencia de datos (plano C, plano U) entre RLC y la interfaz del plano U superior, mantenimiento del número de secuencia de PDCP, transferencia del estado del número de secuencia para su uso en el *handover* u otros procesos, implementa ROHC (Robust Header Compression), se encarga de la entrega en secuencia de las PDUs de la capa superior en el restablecimiento de la capa inferior, de la eliminación de SDUs de capa inferior duplicadas en el restablecimiento de la capa inferior para RLC AM, cifrado y descifrado de datos del plano C y del plano U, protección de la integridad y verificación de la integridad de los datos del plano C, descarte basado en temporizador, descarte de duplicados, enrutamiento y reordenamiento (solo para portadoras divididas LWA). Los cambios en la capa PDCP de NB-IoT en comparación con PDCP en LTE son los siguientes:
  - El tamaño máximo de la SDU y la PDU de control PDCP es de 1600 bytes
  - La operación de recepción del informe de estado de PDCP no es una aplicación en NB-IoT.
  - En LTE, las PDU que transportan datos de DRB se mapean en RLC UM, pero en el caso de NB-IoT, las DRB se mapean en RLC AM.
  - NB-IoT usa un número de secuencia en PDCP de 7 bits para DRB.
- **Capa RRC:** El UE debe realizar la transición al "modo conectado RRC" antes de transferir cualquier dato de la capa de aplicación o completar cualquier procedimiento de señalización. El establecimiento de la conexión RRC es un proceso 3-way handshake entre UE y eNB. Se utiliza para realizar la transición del UE de "RRC IDLE" a "RRC connected mode". Los mensajes intercambiados entre UE y eNB necesarios para completar el procedimiento de establecimiento de conexión RRC son los siguientes: solicitud de conexión RRC (UE → eNB), configuración de conexión RRC (eNB → UE) y configuración de RRC completada (UE → eNB). El procedimiento de establecimiento de la conexión RRC siempre lo inicia el UE, pero puede ser iniciado por el UE o la red. La liberación de la conexión RRC siempre es iniciada por eNB. El mensaje NAS inicial se transfiere como parte del procedimiento de establecimiento de la conexión RRC para reducir la demora de establecimiento.

### 2.2.2. LTE-M/CAT-M1

LTE-M (Long Term Evolution for Machines) [12] es un estándar desarrollado por 3GPP que mejora la velocidad de datos respecto a NB-IoT. Es una tecnología pensada para el uso en aplicaciones M2M (Machine to Machine) e IoT. Usa la infraestructura de antenas de LTE y está optimizada para un ancho de banda mayor que NB-IoT y para conexiones móvil que incluyan voz. El ancho de banda de LTE-M es de 1.4 MHz y la velocidad máxima de los datos es de 384 kbps con una velocidad de bajada y de subida de hasta 1 Mbps. El incremento en la velocidad mejora en gran medida la latencia respecto a NB-IoT, en el caso de LTE-M la latencia es de aproximadamente 50 a 10 milisegundos. Si la red se mantiene funcionando a la mitad de la velocidad, las baterías de los dispositivos pueden llegar a durar 10 años. Esto es porque el consumo al total de la velocidad de la red es mayor al que podría tener NB-IoT. En cuanto al

coste por módulo, está alrededor de los 10 a los 15 dólares, que comparado con los 5 o 7 dólares de los módulos NB-IoT lo hace una tecnología más cara. LTE-M implementa VoLTE (Voice over LTE) que permite enviar voz, esta prestación no la tiene NB-IoT. El rango de los enlaces de LTE-M es de unos 5 Km, algo menor que los de NB-IoT. LTE-M es una mejor solución que NB-IoT cuando lo que se busca es poca latencia y una mayor velocidad en la transferencia de los datos.

La figura siguiente muestra un análisis de la arquitectura de capas de LTE-M. A continuación, se procederá a comentar la figura y las capas que conforman LTE-M.

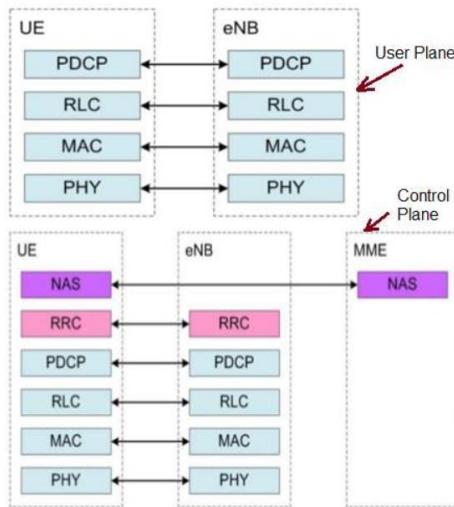


Figura 14 - Arquitectura de capas de LTE-M

Las capas de LTE-M y su función son las siguientes:

- **Capa Física (PHY):** La capa física se encarga de insertar el código de detección de errores o CRC. La modulación usada por LTE-M es OFDMA para *downlink* y SC-FDMA o 16 QAM para *uplink*. Implementa MIMO (Multiple Input Multiple Output) y HARQ (Hybrid Automatic Repeat Request). Esta capa también realiza medidas.
- **Capa MAC:** En *uplink* hace el mapeo de canales, multiplexa, maneja los elementos de control, realiza el *random access procedure*, prioriza los canales según PBR (Priorized Bit Rate), implementa HARQ y realiza el envío de BSR (Buffer Status Report). Para el enlace *downlink* realiza el mapeo de canales, de-multiplexa, implementa DRX (Discontinuous Reception), maneja los elementos de control, implementa HARQ.
- **Capa RLC:** En *uplink* provee el BSR, segmenta y concatena e implementa ARQ (Automatic Repeat reQuest) para el modo AM.
- **Capa PDCP:** En *uplink* añade el número de secuencia, realiza el *handover* y el manejo de los datos, protege la integridad de los datos, realiza el cifrado y la compresión del encabezado. En *downlink*, entrega de numero de secuencia, detección de paquetes duplicados, validación de la integridad, descifrado y descompresión del encabezado.
- **Capa RRC:** Esta capa se encarga de gestionar la conexión, la configuración, la seguridad, controla el paginado, realiza los broadcasts, gestiona la configuración de las medidas que se toman en la capa física, reporta las medidas, selecciona las celdas y gestiona la movilidad.

## 2.3. Protocolos de transporte

Una vez los datos de los sensores han llegado al *gateway* y éste tiene conectividad a internet mediante una comunicación celular, se necesita un protocolo de transporte/mensajería que envíe los datos del *gateway* al *backend* a través de internet. Estos datos deben ser consumibles en cualquier momento de modo que deberían estar guardados en algún lugar. A continuación, se plantean dos protocolos de mensajería que se podrían usar en el paradigma de este proyecto.

### 2.3.1. CoAP (Constrain Application Protocol)

CoAP [13] es un protocolo de software a nivel de aplicación destinado al uso en dispositivos con recursos limitados (constrained resources), se diseñó para usar el modelo cliente/servidor de HTTP en sensores de baja potencia. Permite *multicast*, gasta pocos recursos y tiene una gran simplicidad. Esto lo hace un protocolo ideal para soluciones IoT y M2M porque, como se ha dicho antes, normalmente los dispositivos usados en este tipo de paradigmas no tienen muchos recursos ni potencia disponibles. CoAP funciona en la mayoría de los dispositivos que soportan UDP o un análogo de UDP. CoAP implementa el modelo REST (Representational Estate Transfer) de HTTP (con las primitivas GET, POST, PUT y DELETE), usa cabeceras reducidas, limita el intercambio de mensajes y añade mecanismos de seguridad específicos.

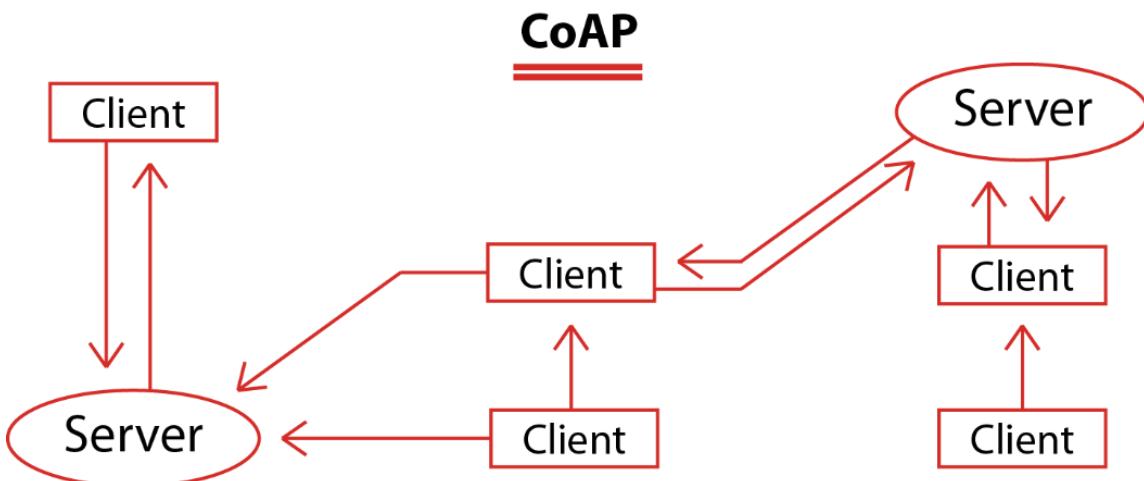


Figura 15 - Protocolo CoAP

En CoAP los clientes hacen *requests* (usando Method code) al servidor y este responde con un código de respuesta (Response Code). La diferencia principal con HTTP es que CoAP hace este proceso de manera asíncrona. CoAP intercambia los mensajes entre puntos finales usando UDP. Hay cuatro tipos de mensaje: Confirmables (si son recibidos ha de llegar un ACK), No-Confirmables (no hay ACK al recibirse), ACK y Reset. Si se busca más robustez se optará por los Confirmables, si en cambio se busca velocidad y no importa tanto la pérdida de información se usan los No-Confirmables.

En CoAP es necesario para el cliente conocer la URI del servidor al que van a realizar la petición. Esto hace que si hay nuevos nodos se deba recurrir al *caching* para reconocerlos antes de poder servir sus *requests*.

### 2.3.2. MQTT (Message Queuing Telemetry Transport)

MQTT [14] se trata de un protocolo de comunicación M2M (machine to machine) de tipo message queuing muy usado para la comunicación entre dispositivos IoT. Está basado en la capa TCP/IP como base para la comunicación y a diferencia de otros protocolos como HTTP cada conexión de MQTT se mantiene abierta y se reutiliza en cada comunicación.

MQTT es un servicio de mensajería *push* con patrón publicador/subscriptor. Los clientes publican en *topics* organizados jerárquicamente y es necesario de un bróker que filtre esos mensajes y e os haga llegar al cliente (o clientes) suscrito a ese *topic*.

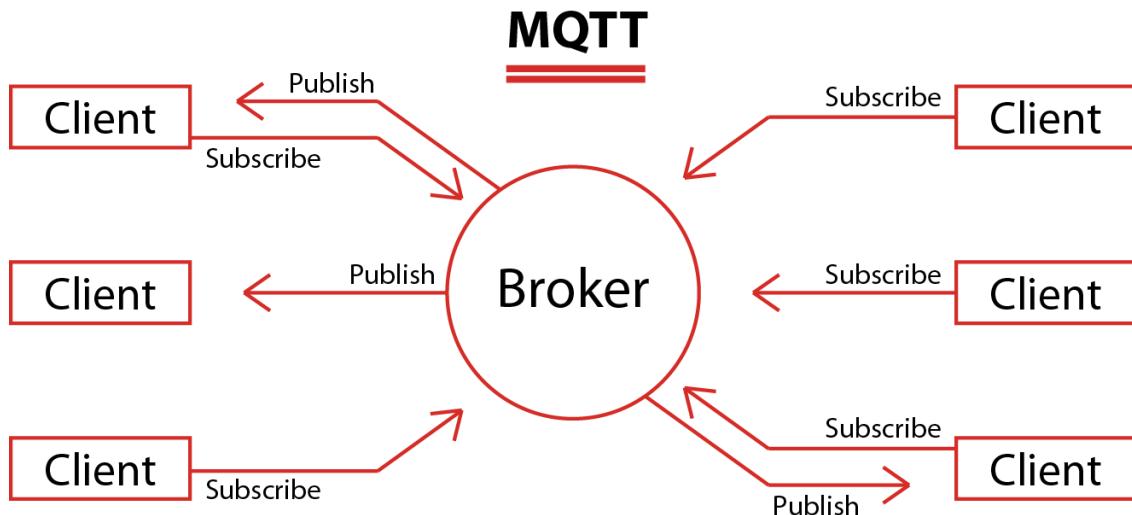


Figura 16 - Protocolo MQTT

Los clientes mantienen una conexión TCP/IP con el *broker* y este mantiene un registro de los clientes conectados. Para realizar esta conexión el cliente envía un mensaje de CONNECT con la información necesaria para la conexión (client-id, nombre, usuario, contraseña...), cuando el *broker* lo recibe, envía un mensaje de CONNACK al cliente con el resultado de la petición (aceptada, rechazada...). Para enviar mensajes a un *topic*, los clientes usan mensajes PUBLISH, que contienen el *topic* y el *payload* a entregar. Para suscribirse o desuscribirse se emplean mensajes de SUBSCRIBE y UNSUBSCRIBE que al ser recibidos por el *broker* son respondidos con SUBACK y UNSUBACK respectivamente. Para asegurar que la conexión está activa en todo momento los clientes mandan periódicamente un mensaje de PINGREQ que es respondido por el *broker* con un PINGRESP. Finalmente, el cliente se desconecta enviando un mensaje de DISCONNECT.

Las ventajas de usar MQTT respecto a otros protocolos son la escalabilidad, el asincronismo y el desacoplamiento entre clientes que da el patrón pub-sub. Al ser un protocolo sencillo y ligero muy adecuado para aplicaciones IoT donde se emplean dispositivos de baja potencia. Como MQTT necesita pocos recursos el consumo de energía es bajo. MQTT dispone de más de un mecanismo de calidad de servicio (QoS) que permite gestionar el nivel de robustez en el envío de mensajes con tres configuraciones posibles, lo que lo hace una muy buena solución para prevenir la pérdida de datos en caso de desconexión.

A pesar de sus características, MQTT puede suponer un problema para algunos dispositivos muy restrictivos, por el hecho de ir sobre TCP y de manejar nombres de *topics* largos. Esto se soluciona con la variante MQTT-SN que utiliza UDP y soporta indexación de nombres de *topics*.

## 2.4. Conclusiones

A continuación, se presenta el sistema final que se va a implementar en el proyecto y las razones por las que se ha escogido cada tecnología.

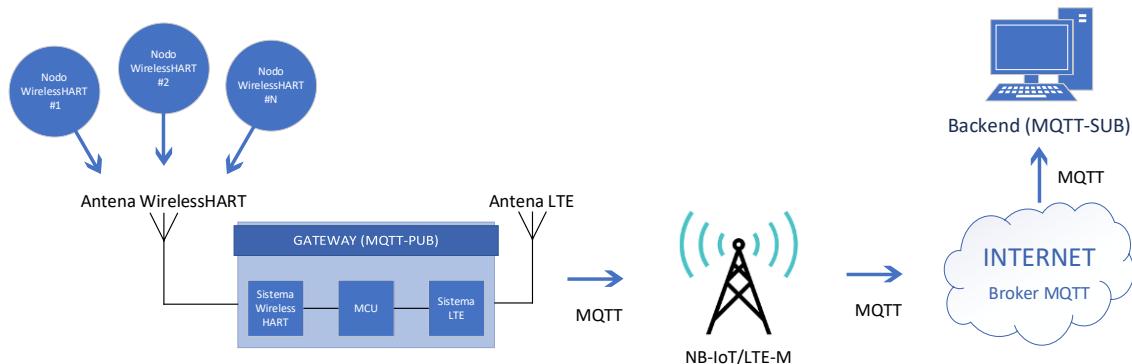


Figura 17 - Diagrama final del sistema

La tecnología WirelessHART ha sido elegida para la comunicación con el *gateway* dado a que los sensores desplegados en la marina eran de esta tecnología. La tecnología WirelessHART es una buena opción para este proyecto ya que está destinada a la captación de datos de sensores de redes de sensores industriales. Si extrapolamos la función que tiene WirelessHART dentro de la industria a otros entornos nos damos cuenta de que es una tecnología extremadamente útil para transportar datos de sensores hacia un *gateway* o un *backend* WirelessHART. Al ser una tecnología que se recupera y optimiza de forma autónoma, la implementación de ésta prevendrá posibles intervenciones humanas posteriores al despliegue. También se debe tener en cuenta que es una tecnología segura, algo importante si se tiene en cuenta que se va a desplegar en un entorno urbano accesible para muchos dispositivos.

Como protocolo de transporte, se ha escogido MQTT. Aún que el consumo de recursos y el consumo energético de MQTT sean algo superiores a los de CoAP, una de las principales ventajas de MQTT respecto a CoAP es su patrón pub/sub. Con MQTT la comunicación entre los nodos y el *backend* se hace mediante un *topic*, de modo que quien envía los datos no tiene por qué conocer absolutamente nada del destinatario de los datos. El destino de los datos es el *topic* que se haya configurado y, por ende, todo aquel que esté suscrito a él tiene acceso a ellos. En CoAP es necesario que el cliente conozca la URI del servidor de su petición para poder realizar peticiones. En CoAP se podría configurar un servidor proxy a modo de *broker*. Este sería el intermediario de todas las peticiones de los clientes. El problema de esto es la complejidad de la gestión del sistema y si tenemos en cuenta que MQTT ya implementa esta solución es innecesario. Uno de los posibles problemas de MQTT puede ser la caída del *broker*. Si cae el *broker* la comunicación se sesga por completo. Una solución posible a este problema es tener un "mqtt cluster" o un "MQTT Bridge" en caso de *mosquitto*. Se trata de varios servidores que funcionan como uno, de modo que si alguno de ellos cae (desconexión, fallo en el hardware, etc.) los otros lo reemplazan. Este protocolo ya estaba siendo usado en el sistema que está instalado actualmente en las marinas.

Para las comunicaciones celulares, que permiten que los datos de la red de sensores lleguen a Internet, se ha escogido Narrowband IoT. Esto se debe a que es una red celular con cobertura a nivel mundial. La red LTE-M no tiene la misma cobertura global y si se desea que el producto final se pueda usar a nivel internacional, la tecnología celular implementada debe permitirlo. A nivel de velocidad y ancho de banda, es suficiente para las cantidades de datos que maneja nuestro sistema. Al ser una red que aprovecha las antenas de telefonía ya instaladas de modo que no hace falta desplegar ningún tipo de infraestructura. Además, los contratos para usar esta red son muy baratos tan solo hace falta obtener una tarjeta SIM o eSIM con contrato de datos, esto suele suponer un gasto anual de 1 € a 1.60 € por cada nodo de la red.

Cuando se está eligiendo una tecnología para aplicarla como solución a un problema en un proyecto, se deben tener en cuenta las restricciones por normativa. La regulación marca que sólo se puede estar transmitiendo un 1% del tiempo en algunas bandas y con un máximo de potencia de 10 dBm (100 mW). Las frecuencias donde operan estas tecnologías pueden variar dependiendo del país donde se usen. De modo que hay que tener en cuenta, no sólo la aplicación de la tecnología y sus ventajas, sino también las restricciones que pueden hacer inviable una solución. En el caso de este proyecto no se transmite durante más del 1% del tiempo ni se superan los 10 dBm y las bandas usadas son la de 2.4 GHz (WirelessHART) y las bandas de NB-IoT, las cuales no requieren de licencia.

### 3. Diseño y desarrollo del hardware

En esta sección se explica el diseño de las diferentes partes que componen el proyecto. Estas se han visto en el diagrama de la Figura 2.

El programa que se va a usar durante el desarrollo de este apartado será Circuit Maker, una versión enfocada a la comunidad *maker* basada en el software Altium.

#### 3.1. Selección de componentes

Los componentes principales que se han usado para el diseño y desarrollo del proyecto han sido seleccionados de entre las posibles soluciones teniendo en cuenta los requerimientos funcionales y la simplicidad en la integración.

Para la elección del módem NB-IoT/CAT-M1 se ha seguido el razonamiento siguiente. Por una parte, la integración de una MCU con una radio externa permite el poder escoger que MCU se quiere usar y que radio se quiere usar, pero su integración es un proceso mucho más extenso que usar un SOM (System On Module). Por otra parte, si se usa un SOM, la radio, la MCU y los sistemas que integre el componente no son elegibles y el componente en sí es más caro, pero su integración es mucho más rápida y sencilla. Aunque los SOMs sean más caros que optar por la otra solución, se usará este tipo de componentes ya que para este proyecto es prioritario la sencillez en la integración antes que su precio.

Hay diversas soluciones en el mercado como nRF9160 de Nordic Semiconductor o SARA-N3 [15] y SARA-R5 [16] de Ublox pero estas dos últimas no llevan MCU integrada. Integrar una MCU a parte resulta en un proceso de integración más complejo y largo, y como se ha mencionado antes, la prioridad es la facilidad en la integración. Por ello se ha elegido nRF9160 como módem LTE. En la figura siguiente se pueden ver las características del componente.

Features:	
<b>Microcontroller:</b> <ul style="list-style-type: none"> <li>ARM® Cortex® -M33</li> <li>243 EEMBC CoreMark score running from flash memory</li> <li>Data watchpoint and trace (DWT), embedded trace macrocell (ETM), and instrumentation trace macrocell (ITM)</li> <li>Serial wire debug (SWD)</li> <li>Trace port</li> <li>1 MB flash</li> <li>256 kB low leakage RAM</li> <li>ARM® Trustzone®</li> <li>ARM® Cryptocell 310</li> <li>Up to 4x SPI master/slave with EasyDMA</li> <li>Up to 4x I2C compatible two-wire master/slave with EasyDMA</li> <li>Up to 4x UART (CTS/RTS) with EasyDMA</li> <li>I2S with EasyDMA</li> <li>Digital microphone interface (PDM) with EasyDMA</li> <li>4x pulse width modulator (PWM) unit with EasyDMA</li> <li>12-bit, 200 ksps ADC with EasyDMA - eighth configurable channels with programmable gain</li> <li>3x 32-bit timer with counter mode</li> <li>2x real-time counter (RTC)</li> <li>Programmable peripheral interconnect (PPI)</li> <li>32 general purpose I/O pins</li> <li>Single supply voltage: 3.0 – 5.5 V</li> <li>All necessary clock sources integrated</li> <li>Package: 10 x 16 x 1.04 mm LGA</li> </ul>	<b>LTE modem:</b> <ul style="list-style-type: none"> <li>Transceiver and baseband</li> <li>3GPP LTE release 13 Cat-M1 and Cat-NB1 compliant <ul style="list-style-type: none"> <li>3GPP release 13 coverage enhancement</li> </ul> </li> <li>3GPP LTE release 14 Cat-NB2 compliant</li> <li>GPS receiver <ul style="list-style-type: none"> <li>GPS L1 C/A supported</li> </ul> </li> <li>RF transceiver for global coverage <ul style="list-style-type: none"> <li>Up to 23 dBm output power</li> <li>-108 dBm sensitivity (LTE-M) for low band, -107 dBm for mid band</li> <li>Single 50 Ω antenna interface</li> </ul> </li> <li>LTE band support in hardware: <ul style="list-style-type: none"> <li>Cat-M1: B1, B2, B3, B4, B5, B8, B12, B13, B14, B17, B18, B19, B20, B25, B26, B28, B66</li> <li>Cat-NB1/NB2: B1, B2, B3, B4, B5, B8, B12, B13, B17, B19, B20, B25, B26, B28, B66</li> </ul> </li> <li>Supports SIM and eSIM with an ETSI TS 102 221 compatible UICC interface</li> <li>Power saving features: DRX, eDRX, PSM</li> <li>IP v4/v6 stack</li> <li>Secure socket (TLS/DTLS) API</li> </ul>
	<b>Current consumption @ 3.7 V:</b> <ul style="list-style-type: none"> <li>Power saving mode (PSM) floor current: 2.7 μA</li> <li>eDRX @ 82.91s: 18 μA in Cat-M1, 37 μA in Cat-NB1 (UICC Included)</li> </ul>

Figura 18 - Características de nRF9160

En el proyecto la alimentación tiene que ser de 3.3V y nRF9160 se puede alimentar entre 3 y 5 Volts. También incluye un receptor GPS. Esto permitirá el uso de la señal de GPS para poner *timestamp* a los paquetes que se transmitan con el dispositivo, el error en el tiempo global será del orden de milisegundos.

Para la recepción de los datos de los sensores WirelessHART se ha optado por usar el SOM ITM-DYUF-B-02 de Midatronics. Este SOM integra el chip LTC5800 de Analog Devices. Durante la memoria se usa Dusty para referirse a este SOM. Dusty se ha seleccionado por dos motivos. Por una parte, simplifica la integración al tratarse de un SOM que incorpora el microcontrolador (ARM Cortex M3), la radio IEEE802.15.4, los cristales y los conectores de antena. Por otra parte, se trata de una solución que integra la coordinación de una red IEEE802.15.4E con un consumo muy reducido. Otro factor importante en la elección ha sido que ahora mismo en el mercado hay pocas alternativas que implementen un *gateway* para IEEE802.15.4E y este ya se está usando actualmente en la solución.

Por último, hay que destacar que los costes de usar SOMs en vez de diseñar el circuito que implemente la funcionalidad deseada, son más altos por unidad. Debido a la poca producción en la fase de prototipado el impacto económico respecto al coste del proyecto es bajo. En una situación de producción en masa este coste sería significativo y resultaría más rentable realizar el diseño e integración de una solución propia basada en el chip Analog Devices LTC5800.

### 3.2. Diseño del subsistema nRF9160

Para el diseño del *hardware* se han tomado como referencia las “*hardware files*” de la placa de desarrollo nRF9160DK, que se pueden encontrar en la página web de Nordic [17]<sup>1</sup>. Se parte del siguiente circuito, sacado de la página 389 del documento *Product specifications* de Nordic Semiconductor [18], que indica los componentes mínimos para el funcionamiento del chip nRF9160.

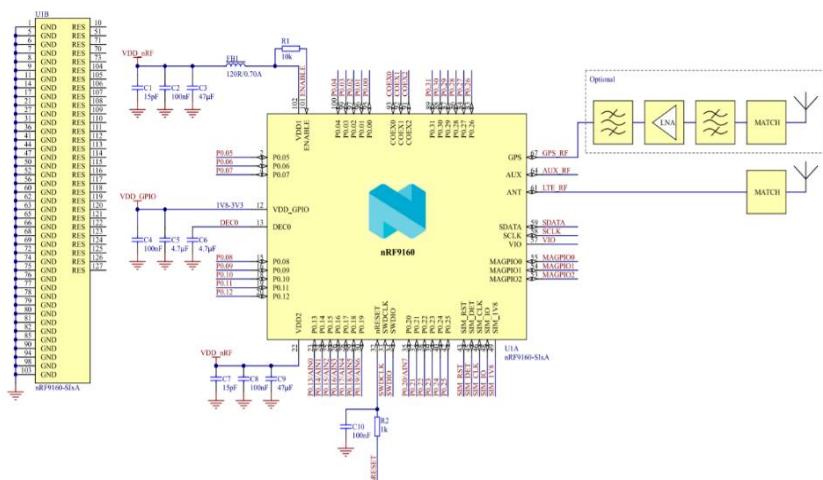


Figura 19 - Circuito básico para nRF9160

<sup>1</sup> Dentro del archivo .zip en la ruta *nRF9160-DK - Hardware files 0\_15\_0\PCA10090-nRF9160 Development Board 0\_15\_0\Schematic\_Layout pdf files* se encuentra el archivo *PCA10090\_Schematic\_And\_PCB.pdf* donde se recogen los esquemáticos de todas las regiones de la placa de desarrollo. Se usará este PDF, el BOM y los *datasheet* (y guías de implementación) de los componentes usados para replicar los circuitos con las funcionalidades que se van a incluir en el producto final del proyecto. Los *datasheet* de los componentes se pueden consultar en la bibliografía.

Se trata del circuito que provee de alimentación, señal de *reset* y conectividad GPS/LTE al chip nRF9160. Se pueden ver las redes de condensadores para el filtrado de alimentación en los pines VDD1, VDD2, VDD\_GPIO, DEC0, así como el filtrado de corriente que efectúa la bobina FB1.

Los componentes recomendados por el fabricante se hallan en una tabla que se puede consultar en la página 390 de el mismo documento. Cada componente tiene un designador, el cual se corresponde con un valor, una descripción del componente y un tamaño o *footprint*. La tabla en cuestión es la siguiente:

Designator	Value	Description	Footprint
C1, C7	15 pF	Capacitor, NPO, $\pm 0.1\mu\text{F}$ , 25 V	0201
C2, C4, C8, C10	100 nF	Capacitor, X5R, $\pm 10\%$	0201
C3, C9	47 $\mu\text{F}$	Capacitor, X5R, $\pm 20\%$ , 10 V	0805
C5	4.7 $\mu\text{F}$	Capacitor, X5R, $\pm 20\%$	0402
C6	4.7 $\mu\text{F}$	Capacitor, X5R, $\pm 10\%$ , 16 V	0805
R1	10 k $\Omega$	Resistor, $\pm 1\%$ , 0.05 W	0201
R2	1 k	Resistor, $\pm 1\%$ , 0.05 W	0201
FB1	120 R / 0.70 A	Ferrite Bead, 120 $\Omega$ @ 100 MHz, 700 mA, 130 m $\Omega$ Max	0402
U1	nRF9160-SIxA	Low power System-in-Package (SiP) with integrated LTE-M/NB-IoT modem and GPS	LGA

Figura 20 - BOM de Nordic para el funcionamiento mínimo de nRF9160

Tras buscar los componentes en la librería online de Circuit Maker e implementarlos, se consigue el esquemático siguiente:

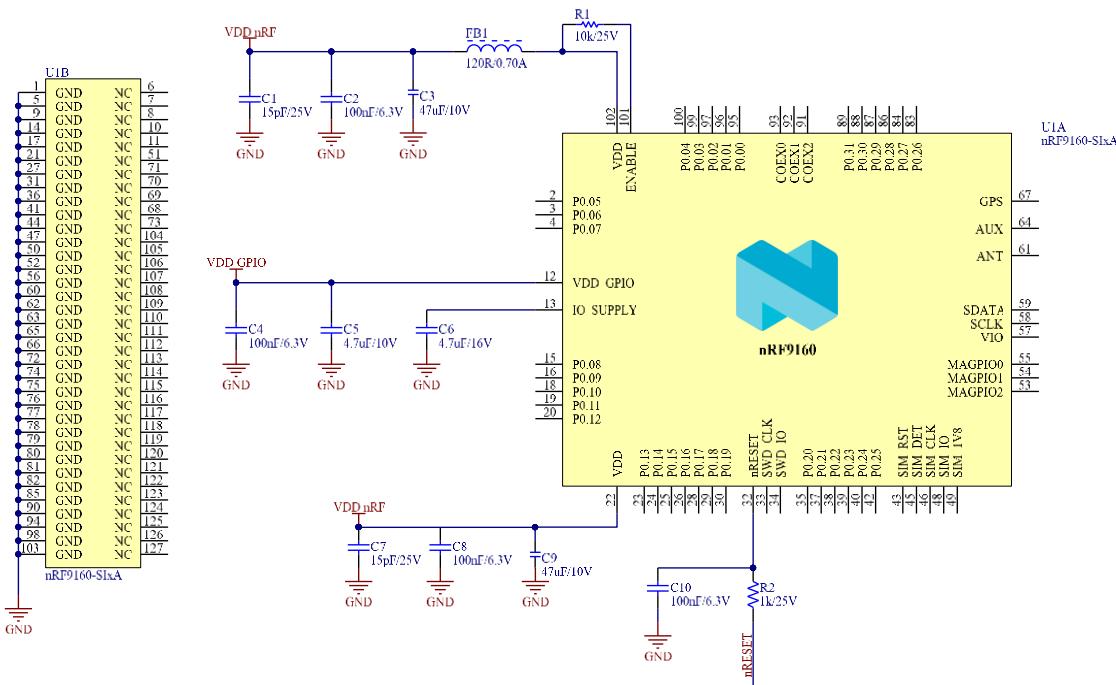


Figura 21 – Circuito básico para nRF9160

Posteriormente, se han añadido las conexiones necesarias para, comunicarse con la SIM, lograr la comunicación entre el chip nRF9160 y el SOM que implementa el protocolo WirelessHART, el control de los LEDs y los botones, la conexión de las antenas de GPS y LTE, las señales de control de la alimentación, las señales de *reset* y el control del encendido de Dusty. La funcionalidad de cada una de estas señales será explicada posteriormente en una tabla.

En cuanto a las UARTs, la UART1 se destinará a *debugging* del chip nRF9160 mediante un cable FTDI y la UART2 se utilizará para comunicar el módulo Dusty con el módulo nRF1960. Por último, el componente J7 se usará para programar el chip nRF9160 mediante una conexión SWD (Serial Wire Debug).

La figura siguiente es el circuito de la Figura 21 una vez se le agregan las señales de los subsistemas que componen la solución.

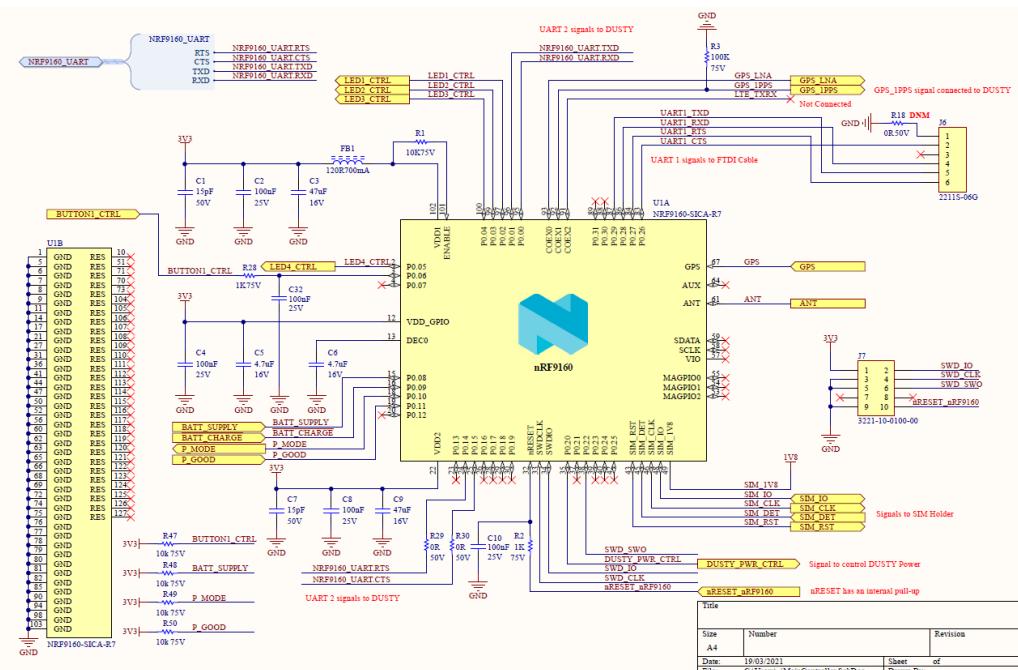


Figura 22 - Circuito para nRF9160 con todas las señales

A continuación, se muestra una tabla con todas las señales del circuito anterior con las funciones que tienen, tipo de pin, numero de pin de nRF9160, etc.

nRF9160 Pin Number	nRF9160 Pin Name	nRF9160 Pin Type	Schematic signal name	Function
2	P0.05	Output	LED4_CTRL	Control del LED rojo
3	P0.06	Input w/ internal pull-up	BUTTON1_CTRL	Botón programable
15	P0.08	Input	BATT_SUPPLY	Indica si la fuente de alimentación principal está conectada. HIGH si está desconectada y LOW siempre que esté conectada.
16	P0.09	Input	BATT_CHARGE	Indica si la batería está cargando. HIGH para batería cargada y LOW para batería cargando.
18	P0.10	Output	P_MODE	Permite escoger la eficiencia del regulador en función del consumo esperado. Existen dos modos, modo PWM y modo PFM. HIGH para PWM y LOW para PFM

19	P0.11	Input	P_GOOD	Indica si los valores del voltaje son correctos. HIGH para valores por encima del 95% del voltaje nominal y LOW para valores inferiores al 90%.
24	P0.14	Output	NRF9160_UART.RTS	Función RTS
25	P0.15	Input	NRF9160_UART.CTS	Función CTS
32	nRESET	Input w/ internal pull-up	nRESET_nRF9160	Botón de reset de la placa
33	SWD_CLK	Input	SWD_CLK	Clock
34	SWD_IO	Input/Output	SWD_IO	Debug IO
35	P0.20	Output	DUSTY_PWR_CTRL	Control de la alimentación de Dusty
38	P0.22	Output	SWD_SWO	Serial Data
44	SIM_RST	Output	SIM_RST	SIM Reset
45	SIM_DET	Input	SIM_DET	SIM Detection
46	SIM_CLK	Output	SIM_CLK	SIM Data
48	SIM_IO	Input/Output	SIM_IO	SIM Clock
49	SIM_1V8	Output	SIM_1V8	SIM Voltage
61	ANT	Input/Output	ANT	LTE Antena Signal
67	GPS	Input	GPS	GPS Antena Signal
83	P0.26	Input	UART1_CTS	UART1
84	P0.27	Output	UART1_RTS	UART1
86	P0.28	Input	UART1_RXD	UART1 RX
87	P0.29	Output	UART1_TXD	UART1 TX
91	COEX2	Output	LTE_TXRX	LTE
92	COEX1	Output	GPS_1PPS	Pulse per second GPS
93	COEX0	Output	GPS_LNA	GPS LNA
95	P0.00	Input	NRF9160_UART.RXD	UART2 a Dusty
96	P0.01	Output	NRF9160_UART.TXD	UART2 a Dusty
97	P0.02	Output	LED1_CTRL	Control del LED verde
99	P0.03	Output	LED2_CTRL	Control del LED amarillo
100	P0.04	Output	LED3_CTRL	Control del LED naranja

Tabla 1 - Tabla de señales usadas por nRF9160

### 3.3. Diseño del subsistema NB-IoT/CAT-M1 y GPS

A continuación, se explica el diseño de las antenas NB-IoT/CAT-M1 y GPS, así como la interconexión con la SIM del subsistema NB-IoT/CAT-M1.

#### 3.3.1. Antenas NB-IoT/CAT-M1 y GPS

A partir de diseño de referencia del hardware files de Nordic [17], en la página 2 del PDF con los esquemáticos se encuentra el esquemático que implementa la funcionalidad de las antenas de NB-IoT/CAT-M1 y GPS. La parte de circuito que se usará en esta sección es la siguiente.

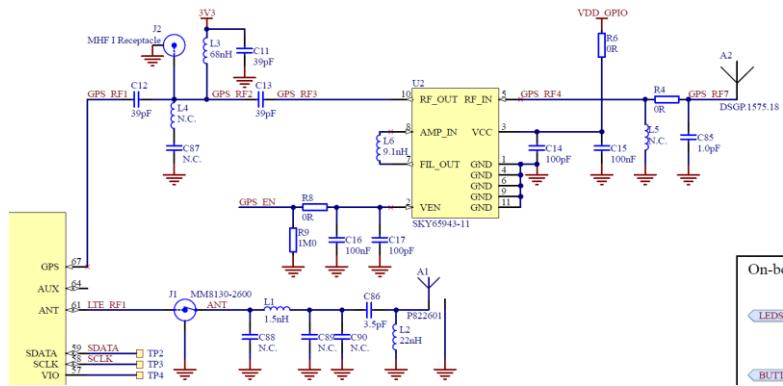


Figura 23 - Antenas NB-IoT/CAT-M1 y GPS implementadas en los archivos de hardware.

El chip nRF9160 va conectado a la antena GPS a través de dos redes de adaptación de impedancias junto con un *Low Noise Amplifier* (U2, SKY65943-11) que se encarga de preamplificar la señal recibida. Este LNA irá conectado al chip nRF9160 en el pin COEX0, esto se puede ver en la Figura 22. En el diseño se ha optado por dejar sin montar (D.N.M.) los condensadores de las redes de adaptación de la antena GPS, dejando solo su *footprint* en la PCB para su futura instalación. Tampoco se ha incluido J2 que es un conector coaxial usado para leer la salida GPS del chip nRF9160.

Respecto a la antena del subsistema NB-IoT/CAT-M1, se ha optado también por dejar solo los *footprints* de los condensadores de la red de adaptación. Tampoco se ha implementado J1, un conector coaxial con *switch* integrado usado para leer la salida NB-IoT/CAT-M1 del chip nRF9160 durante el proceso de *debug* de la placa.

Es importante mencionar que las antenas físicas no aparecen en el esquemático final, dado que no van integradas en la PCB. Las antenas van enroscadas en los terminales coaxiales J1 y J2 utilizando un conector SMA (SubMiniature versión A) hembra. Esto es debido a que la placa irá dentro de una caja estanca IP-67 y las antenas se pondrán en el exterior (a través de un pasamuros) para asegurar una buena cobertura.

El resultado después de copiar el circuito proporcionado por Nordic y de efectuar los cambios mencionados, es el siguiente.

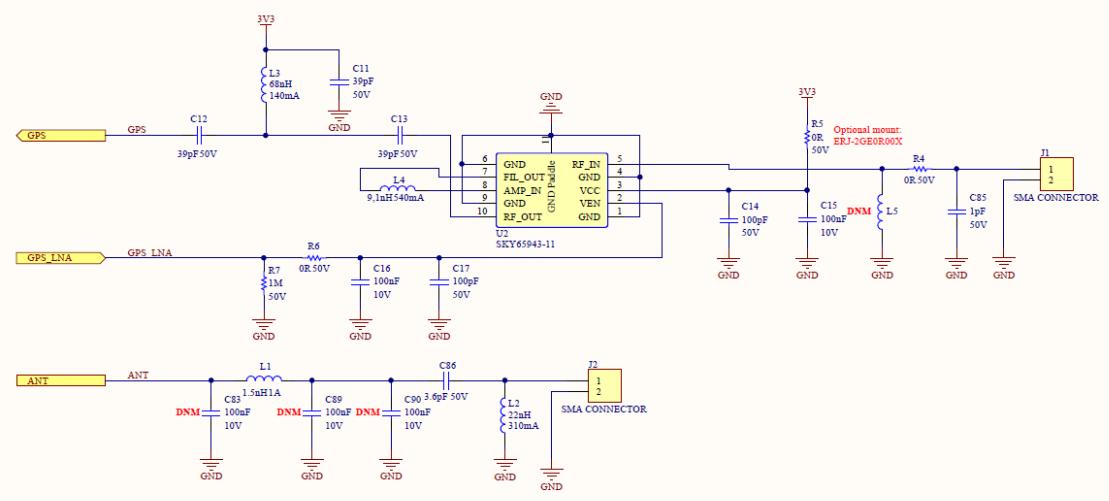


Figura 24 - Circuito que implementa las antenas de NB-IoT/CAT-M1 y GPS

### 3.3.2. Diseño del circuito para la SIM

Para tener conexión a NB-IoT/CAT-M1 se necesita una SIM proporcionada por el operador que prestará el servicio de datos. Tal y como se indica en los requerimientos del proyecto, se ha optado por instalar un *SIM holder*, ya que uno de los requisitos es el poder cambiar de tarjeta en función del país donde vaya desplegado el sistema.

El circuito para las SIMs que tiene la placa de desarrollo nRF9160DK se muestra en la Figura 25. Se trata de un circuito con un *switch* (U21) que permite al usuario escoger entre las dos posibles SIMs (SIM o eSIM) a través de un selector externo (SW7). Por defecto el sistema está conectado a la SIM dado que el multiplexor utiliza los pines NC. Una vez el *switch* SW7 pasa a la posición donde se encuentra el GND, el multiplexor usa los pines NO y por tanto el sistema está conectado a la eSIM. Las señales que usan ambas SIMs son 1V8 para la alimentación, IO para los datos, RST para el reset y CLK para el reloj.

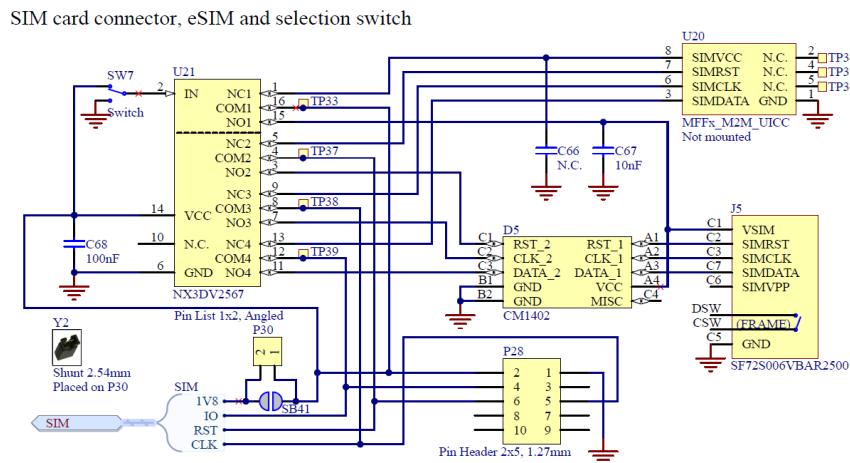


Figura 25 - Circuito para implementar las SIMs de la placa de desarrollo nRF9160

Como en nuestro caso no usaremos la eSIM (la cual se suelda directamente a la placa), podemos obviar tanto el *switch* (U21) como la eSIM (U20). Tampoco se instalarán SW7, P28 ni P30.

El resultado es el siguiente:

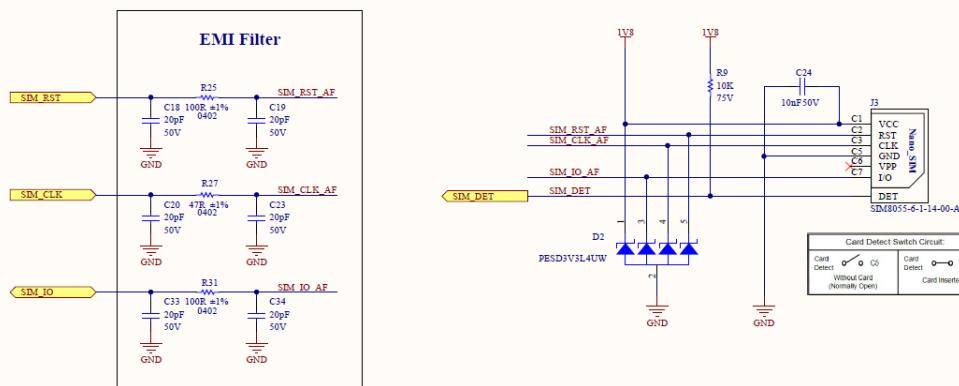


Figura 26 - Circuito diseñado para implementar la SIM

Finalmente, J3 es el componente donde se inserta la SIM (SIM Holder) el cual tiene detector de SIM, que servirá para informar al chip nRF9160 de si se ha instalado o no una SIM en el

componente. También, se ha implementado un filtro de EMI (Electromagnetic Interferences) que protege la SIM de posibles subidas de tensión.

### 3.4. Diseño del subsistema IEEE802.15.4E

LTC5800 de Analog Devices implementa el protocolo WirelessHART el cual toma la capa física y de enlace de datos de IEEE802.15.4E que incorpora la extensión TSCH (Time Synchronized Channel Hopping). Este chip permite implementar la funcionalidad tanto del nodo como del *gateway*.

Para facilitar la integración del chip LTC5800 en la placase ha optado por utilizar un SOM (System On Module), concretamente Dusty (ITM-DYUF-B-02). Este SOM incorpora el chip LTC5800, dos relojes (32 kHz y 20 MHz) y una antena de 2.4 GHz para IEEE802.15.4E.

El esquemático siguiente (pág. 5 y 6 de “Eterna LTP5901 / LTP5902 Integration Guide” [19]) se va a usar para instalar los puertos de *debugging* (JTAG) y de programación de la memoria Flash del SOC que se usa en el proyecto.

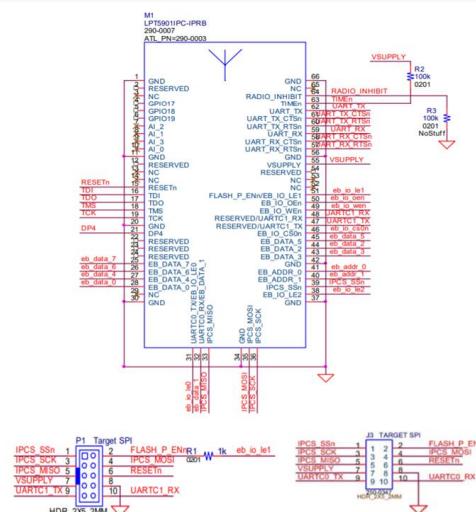


Figura 27 - Esquemático instalación de JTAG y debugging

Se han instalado las J4 y J5 (TC 2050-IDC). J4 servirá para poder programar la memoria Flash del MCU (U5, Dusty) y J5 para permitir el debug del mismo. Una vez instalado esto el resultado es el siguiente:

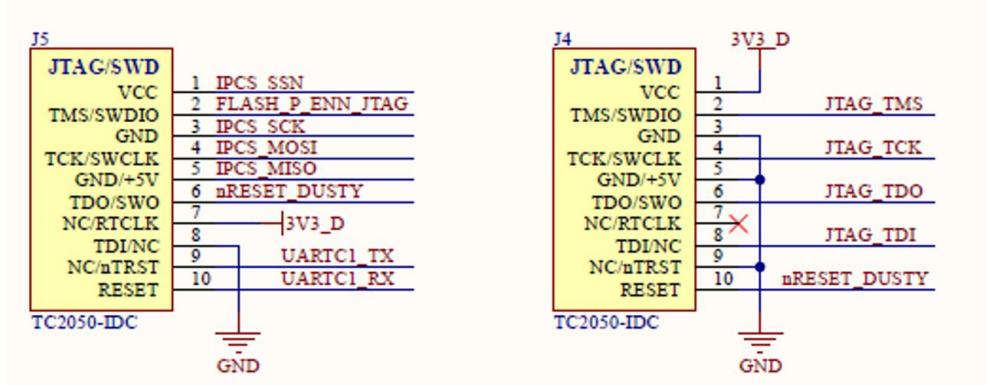


Figura 28 - Conectores de JTAG y SWD para Dusty

Como se ha comentado en la introducción, uno de los requerimientos del sistema es que debe ser capaz de manejar una red WirelessHART de hasta 100 nodos. Puesto que el componente al que se conectan los sensores WirelessHART (U5, Dusty, ITM-DYUF-B-02) sólo aguanta un número limitado de 32 dispositivos, se debe incrementar la RAM. Para ello se han seguido las páginas 19, 20, 21 del documento “Eterna Integration Guide” [20] donde se explica cómo integrar la RAM que se necesita para soportar esta funcionalidad. El esquemático de la Figura 29 es el que se seguirá para la ampliación de memoria. Éste se halla en la página 20 del documento. En la misma página y en la siguiente se especifican en un BOM los componentes usados en el esquemático.

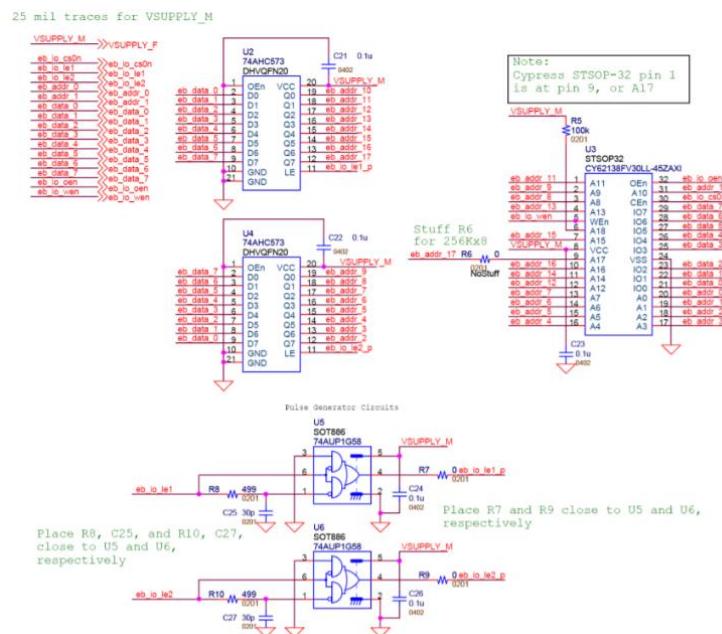


Figura 29 - Esquemático de Eterna para la instalación de la RAM

El resultado de la implementación es el siguiente:

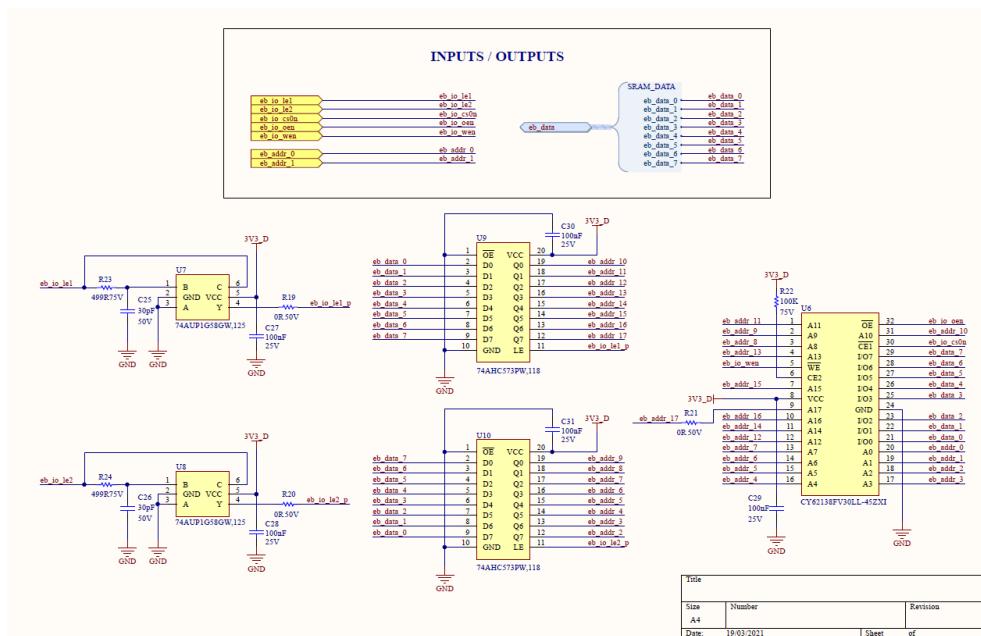


Figura 30 - Implementación de la RAM

Se necesita una memoria de 256 K para soportar la funcionalidad con 100 nodos WirelessHART, por tanto, se requiere de 18 bits para direccionar 256 K espacios de memoria de 8 bits cada uno. Dado que el módulo Dusty no tiene los suficientes pines libres como para direccionar esas 256 K posiciones y además escribir y leer 8 bits de datos en cada una de estas ( $18 + 8 = 26$  bits o pines necesarios, si se suman lectura y escritura de la ram son 28), se utiliza un sistema multiplexado.

Así pues, el funcionamiento del sistema para hacer uso de la memoria RAM externa es el siguiente. En el primer y segundo ciclo de reloj, los pines eb\_data\_x se configuran como salida y se carga la dirección alta (eb\_addr\_[17...10]) y baja (eb\_addr\_[9...2]) de memoria a través de los latches U9 y U10 respectivamente. Para activar los *latches* de manera secuencial se utilizan las señales eb\_io\_le1\_p y eb\_io\_le2\_p que salen de los circuitos de generación de pulsos implementados por U7 y U8. Finalmente, en el tercer ciclo, se leen o escriben los datos en la posición de memoria de la RAM definida en los ciclos anteriores, dependiendo de las señales eb\_io\_oen (leer) y eb\_io\_wen (escribir).

A continuación se muestra la figura de el SOM Dusty con las señales pertinentes para comunicarlo con nRF9160 mediante UART y para hacer un reset a Dusty que se han obtenido usando la guia de usuario de Dusty [21].

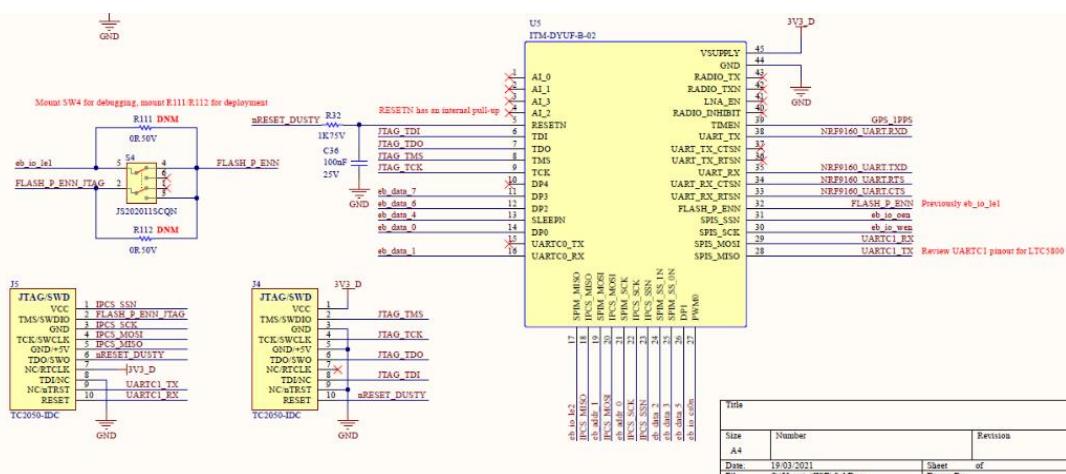


Figura 31 - Implementación de Dusty y señales de las UARTs

Un problema que se ha tenido que solucionar es que la señal FLASH\_P\_ENN, presente en el pin número 32 de Dusty (cuya función es la de permitir la depuración de este), entra en conflicto directo con la señal eb\_io\_le1 (cuya función es activar o desactivar el *latch* U9). Para solucionar el problema se ha optado por instalar un *switch* (SW4) y las resistencias R111 y R112 de modo que cuando se use la función de programación del MCU, este *switch* se debe mover a la posición 2. Mientras este esté en la posición 1 (la que se ve en el esquemático anterior) la señal habilitará o deshabilitará el *latch* U9. Una vez se proceda a la producción del dispositivo, se procederá a instalar únicamente una de las resistencias, o bien solo se usará la función de JTAG (instalando la resistencia R112) o la de habilitar el *latch* (instalando la resistencia R111). Es obvio que para el correcto funcionamiento del producto final se acabará optando por la segunda opción.

### 3.5. Diseño del subsistema de alimentación

Este sistema alimenta a todos los subsistemas de la placa a 3.3 VDC. Para que esto sea posible y para permitir la alimentación mediante una batería de emergencia se requiere de varias etapas

para regular el voltaje y la implementación de tres componentes. Se ha dividido el diseño del subsistema de alimentación de la placa en cuatro bloques. Tres de estos bloques, mostrados en la Figura 32, son para la alimentación común de toda la placa y uno de ellos solo distribuye la tensión al SOM Dusty. A continuación, se explica el funcionamiento de los tres bloques principales y posteriormente se procederá a explicar el funcionamiento del cuarto bloque dedicado a alimentar el SOM Dusty.

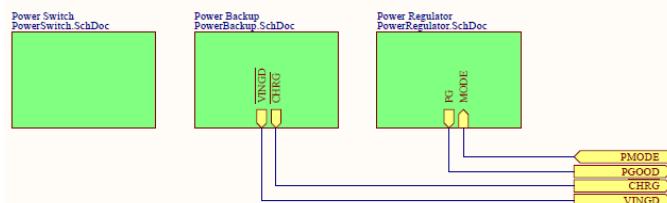


Figura 32 - Diagrama de bloques de la alimentación de la placa

En el primer bloque (Power Switch) se halla el *switch* que conecta o desconecta la batería principal del sistema y la entrada de alimentación principal. En el segundo bloque (Power Backup) se encuentra la implementación del chip que gestiona la batería de *backup* y el conector para esta batería. Finalmente, en el tercer bloque (Power Regulator) se encuentra el regulador de tensión que dará la alimentación de 3.3 VDC al sistema.

### 3.5.1. Power Switch

En el circuito Power Switch se conecta la batería principal a J9 que puede recibir voltajes de entre 8 y 40 VDC y tiene que soportar hasta 2A. SW1 se encarga de dejar pasar o no la corriente, por tanto, de conectar o desconectar la alimentación principal del sistema.

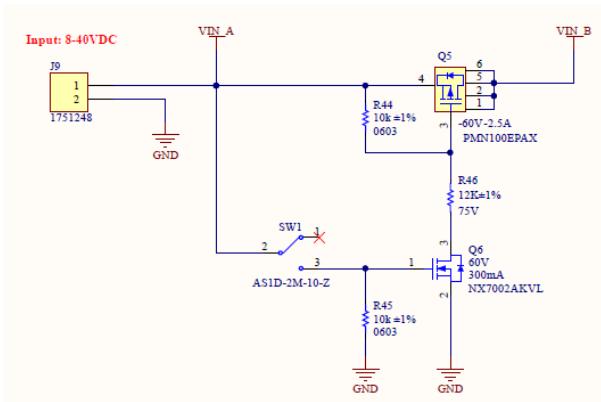


Figura 33 - Power Switch

Cuando SW1 está en la posición 2-1, Q6 está desactivado (tiene 0V en la base) y por tanto Q5 es un circuito abierto y no deja pasar corriente entre VIN\_A y VIN\_B. Si SW1 está en la posición 2-3, Q6 se activa y por tanto hay corriente en R46 y R44, esto crea un divisor de tensión en estas dos resistencias. El divisor de tensión hace que haya más tensión en el emisor de Q5 que en su base y al ser un transistor PNP se activa dado que Vbe es negativa. El hecho de que Q5 se active hace que VIN\_B sea igual a VIN\_A (el voltaje de la fuente alimenta el circuito). R45 hace la función de *pull-down*, para descargar el circuito y asegurar que no hay tensión en Q6 una vez se apague el circuito.

### 3.5.2. Power Backup

El circuito de Power Backup regula la alimentación que proviene de la fuente de alimentación principal y de la batería de emergencia. El voltaje proveniente de Power Switch (VIN\_B), que está entre los 8 y los 40 Volts, es convertido por el dispositivo LTC4091 [22] a una tensión de entre 3.5 VDC y 4.45 VDC llamada VSYS. Como se ha dicho anteriormente U3 también se encarga de gestionar la carga de la batería de emergencia. Esta batería está conectada al regulador en VBAT.

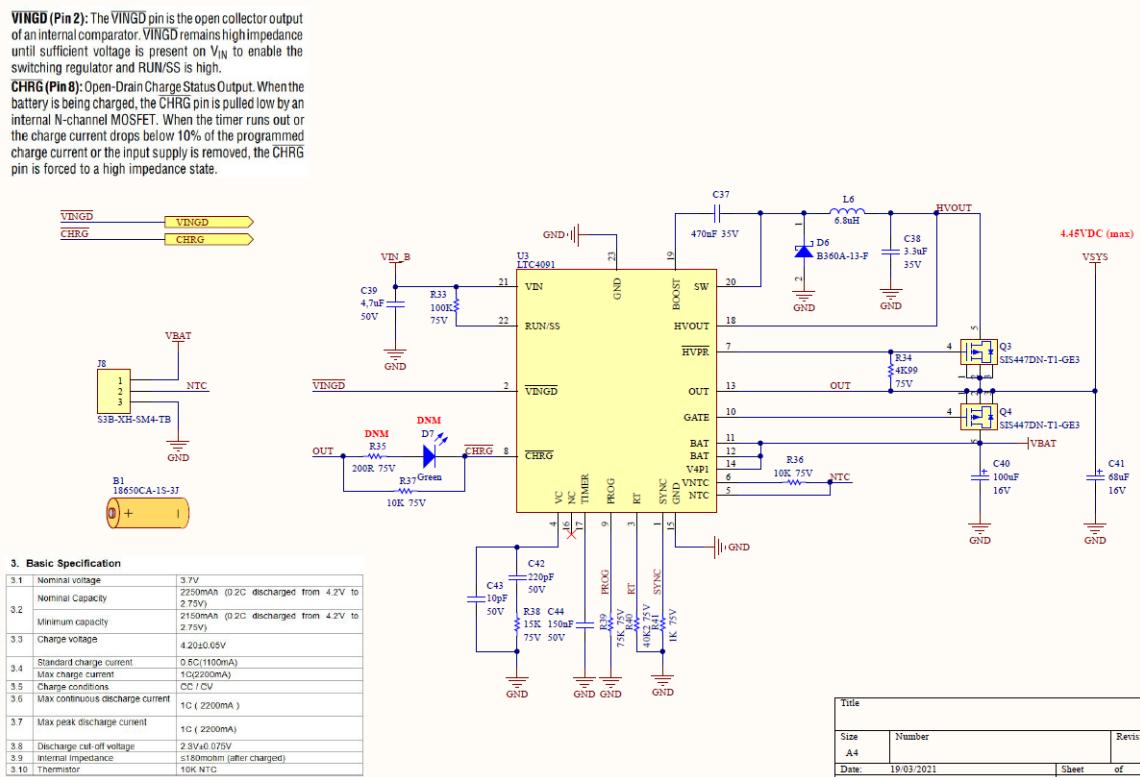


Figura 34 - Power Backup

Del circuito salen dos señales, VINGD y CHRG. Por un lado, VINGD (posteriormente llamada BATT\_SUPPLY) es 0 siempre que la fuente principal esté conectada. Si en algún momento se desconecta la fuente, el pin pasa a estar en alta impedancia. Con la resistencia de *pull-up* R48 (ver Figura 22) la señal de alta impedancia pasaría a ser un 1, esto quiere decir que, en caso de desconexión de la fuente principal, a nRF9160 le llegaría un 1 por el pin P0.08.

Por otro lado, la señal CHRG (posteriormente nombrada BAT\_CHARGE) está a 0 mientras la batería de emergencia está cargándose. Una vez la batería se haya cargado por completo el pin pasa a estar en alta impedancia. Con la resistencia R37 haciendo la función de *pull-up*, nRF9160 detectará un 1 cuando la batería esté cargada y un 0 cuando se esté cargando. En el caso en el que no se detecte batería de emergencia se leerá un 1. La batería de emergencia se conecta en J8. La señal NTC sirve para monitorizar la temperatura de la batería en todo momento. Si en algún momento la batería al cargarse subiera la temperatura por encima de cierto umbral, se desconectaría automáticamente para evitar daños mayores. En el capítulo 3.5.4. se habla de la batería escogida y se demostrará, mediante una estimación de consumo, que es capaz de alimentar el sistema ininterrumpidamente durante un mínimo de 48h.

### 3.5.3. Power Regulator

La función principal del circuito Power Regulator es la de convertir el voltaje VSYS, que proviene de Power Backup y cuyo valor está entre 2.5 y 4.45 VDC, a 3.3 VDC para alimentar todo el sistema.

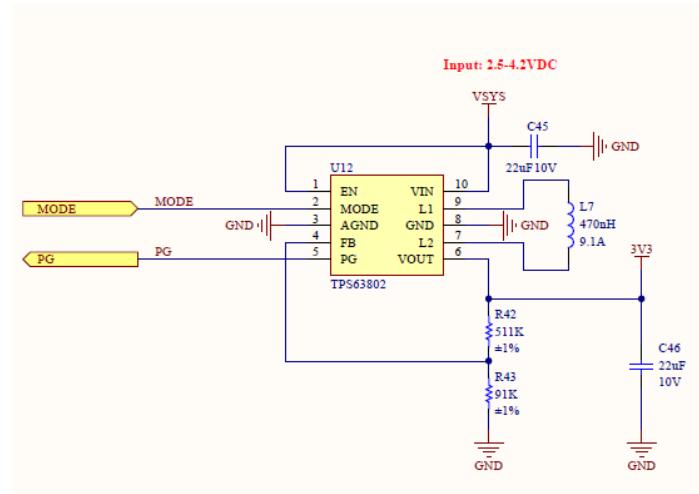


Figura 35 - Power regulator

El componente TPS63802 [23] (U12) es un regulador que se encarga de gestionar el voltaje que proviene de LTC4091 para convertirlo a 3.3 VDC. Si el voltaje está por encima de los 3.3 VDC, el regulador regula (función *buck*) la tensión para mantenerla estable en los 3.3 VDC. Si el voltaje es inferior a los 3.3 VDC, normalmente a causa de la descarga de la batería de emergencia cuando no hay fuente de alimentación principal, el regulador regula la tensión (función *boost*) para lograr mantenerla en los 3.3 VDC. Este tipo de reguladores se llaman Buck-Boost debido a que no solo bajan la tensión, sino que también pueden subirla a un voltaje determinado.

El pin PG (Power Good) indica si el voltaje está en valores correctos. Si el voltaje de salida está por encima del 95% del voltaje nominal el pin PG estará en alta impedancia, y si el voltaje de la salida baja del 90% del voltaje nominal el pin PG estará en *low* (0). Con la resistencia R50 haciendo de *pull-up*, cuando el pin PG esté en alta impedancia, nRF9160 detectará un 1. De este modo se puede saber si hay algún problema con la alimentación. A nivel de software, se configurará una rutina de apagado y aviso si se detecta que PG está a 0 ya que esto implicaría un error o mal función en el sistema.

El regulador tiene la posibilidad de configurarse con dos modos de eficiencia, PWM o PFM. El modo PFM (pulse-frequency-modulation) es más eficiente en casos de uso con corrientes de salida bajas. Estos modos se configuran usando el pin MODE del regulador. Este pin no se puede dejar flotando y será controlado por nRF9160 mediante software con la señal P\_MODE del pin P.10.

### 3.5.4. Estimación de consumo energético y elección de la batería de emergencia

La batería que se ha escogido es TLI-1550S de Tadiran Batteries [24]. A continuación, se van a realizar los cálculos de consumo necesarios para demostrar que cumple con las especificaciones necesarias para poder alimentar el sistema de manera ininterrumpida durante un fin de semana, es decir, un mínimo de 48h.

Para el cálculo del consumo del chip nRF9160, se va a usar un *power profiler* que se puede consultar en la web de Nordic Semiconductor [25]. Se ha configurado el *power profiler* con los

valores mostrados en la Figura 35. Estos valores se han escogido teniendo en cuenta el caso de uso donde el sistema esté consumiendo más recursos, es decir el peor caso en cuanto a consumo energético (100 nodos transmitiendo un paquete de 184 bytes cada 60 segundos).

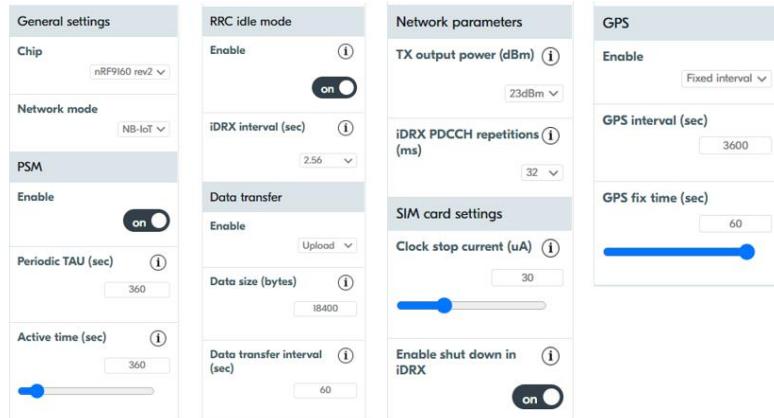


Figura 36 - Valores de los parámetros en la configuración del *power profiler*.

Para el cálculo del tamaño de los datos se ha sumado el tamaño máximo de un paquete WirelessHART con el tamaño máximo de las cabeceras de MQTT, TCP e IP. Esto es:

$$\text{WirelessHART}_{\text{Packet}} + \text{MQTT}_{\text{Header}} + \text{TCP}_{\text{Header}} + \text{IP}_{\text{Header}}$$

$$59\text{bytes} + 5\text{bytes} + 60\text{bytes} + 60\text{bytes} = 184\text{bytes}$$

Teniendo en cuenta que cada *gateway* puede soportar hasta 100 nodos tenemos que el tamaño máximo de los datos es de:

$$184\text{bytes} \times 100\text{Nodos} = 18400\text{bytes}$$

Una vez configurado el *power profiler*, este nos da el resultado del consumo. A continuación, se muestra el sumario del consumo.

General		Current consumption	
Chip	nRF9160 rev2		
Voltage	3.7		
<b>PSM</b>			
Periodic TAU	6.0 min		
Periodic TAU timer element	'10001I00'		
Active time timer element	'0010010'		
<b>RRC connected mode</b>			
Data upload interval	60.0 s		
Data upload charge	530.22 mC		
Connection management charge	79.09 mC		
Total charge	609.31 mC		
<b>RRC idle mode</b>			
Time in iDRX	45.44 s		
iDRX average current	899.74 µA		
Total charge	39.16 mC		
<b>GPS</b>			
GPS average current	748.33 µA		
GPS fix total charge	2.69 C		
		LTE event total charge	648.47 mC
		PSM floor current	2.7 µA
		Total average current	11.95 mA

Figura 37 - Resultados del *power profiler* de nRF9160

Entonces tenemos que el consumo de nRF9160 en el peor de los casos es de unos 11.95mA. Ahora se va a calcular el consumo en el peor de los casos del mánager Dusty.

Para saber el consumo del mánager Dusty en el peor de los casos se ha utilizado un documento Excel que proporciona Analog Devices en su página web [26].

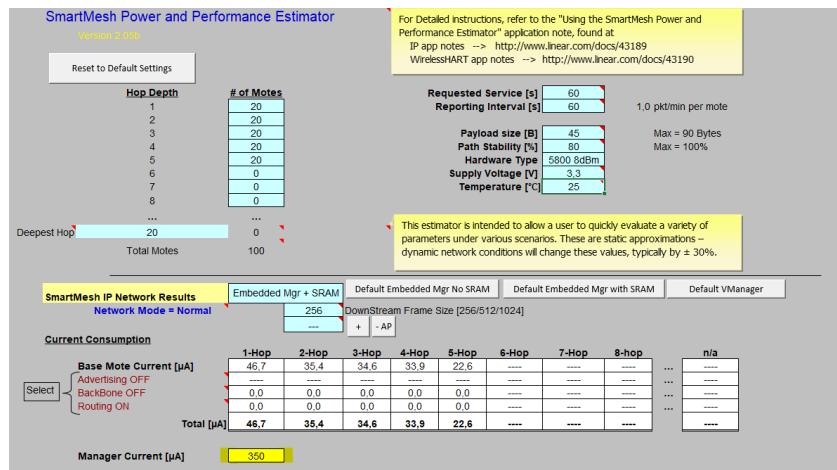


Figura 38 - Estimación del consumo de corriente del mánager

Para el cálculo se ha usado el número máximo de motas que puede soportar un *gateway* (100), un *requested service* de 60 segundos, un *reporting interval* de 60 segundos, un voltaje de 3.3 VDC y los demás parámetros se han dejado por defecto. Esto sería uno de los peores casos de consumo a los que nos podríamos enfrentar en cuanto a Dusty. El consumo medio estimado para el mánager que estima el algoritmo es de unos 350  $\mu$ A.

Ahora que tenemos las medidas aproximadas de consumo de ambos módulos, se plantean dos casos con un consumo distinto:

- **Caso A:** La alimentación del sistema proviene de la fuente principal y por tanto se usan los dos reguladores del circuito de alimentación. En este caso tenemos que el consumo del sistema es el consumo de nRF9160 y el del mánager Dusty, sumado al consumo pasivo del circuito el cual se despreciará para realizar los cálculos ya que se espera que sea muy bajo en comparación con los otros dos consumos.

$$I_{System} = I_{DustyManager} + I_{nRF9160} + I_{Passive} \approx I_{DustyManager} + I_{nRF9160}$$

$$I_{System} = 350\mu A + 11'95mA = 12'3mA$$

Ahora se debe tener en cuenta el rendimiento del regulador TPS63802 ya que es el elemento que alimenta directamente a estos dos módulos. El rendimiento de este dispositivo se puede hallar en el *datasheet* del mismo. La figura siguiente muestra este rendimiento:

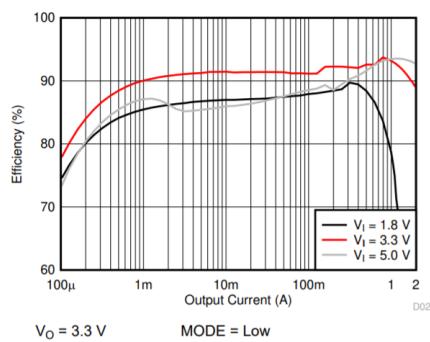


Figura 39 - Eficiencia respecto la corriente de salida del chip TPS63802

En nuestro caso el voltaje de entrada estará entre la línea roja y la línea gris del gráfico anterior dado que LTC4091 da la tensión a TPS63802 con un valor entre 3.5V y 4.45V. Teniendo en cuenta que en el peor caso se necesitan 12.3mA, podemos aproximar la eficiencia del dispositivo a 87%. Teniendo en cuenta la eficiencia del regulador tenemos que:

$$I_{TPS\ 63802} = \frac{I_{System}}{\eta_{TPS\ 63802}} = \frac{12.3mA}{0.87} = 14.14mA$$

Una vez tenemos el consumo en el regulador de voltaje U12, necesitamos el rendimiento de U3 (LTC4091) cuando este regula el voltaje de la fuente principal. Este rendimiento se puede hallar del mismo modo que el del anterior dispositivo. En el *datasheet* de U3 tememos el siguiente gráfico:

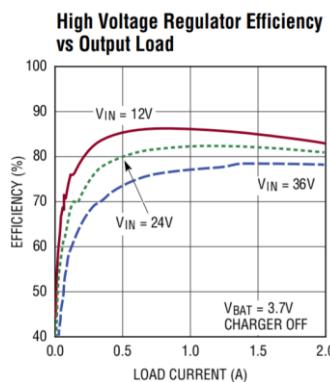


Figura 40 - Eficiencia respecto la corriente de salida del chip LTC4091

Así pues, tenemos que para un voltaje de entrada de 12 Volts (fuente de alimentación usada en el proyecto) y una corriente de salida de pocos miliamperios, nos situamos en un rendimiento de alrededor del 55%. Entonces:

$$I_{Total} = \frac{I_{TPS\ 63802}}{\eta_{LTC\ 4091_{Fuente\ Principal}}} = \frac{14.14mA}{0.55} = 25.7mA$$

Finalmente, el resultado del consumo total del sistema es de aproximadamente 25.7 mA cuando se usa como alimentación una fuente de 12 Volts. Dado que en este caso la alimentación no proviene de la batería de emergencia, no se van a realizar los cálculos de duración de esta. En este caso lo más importante es asegurar que ambos dispositivos soportan la corriente de pico del sistema que puede estar alrededor de 1 A.

- **Caso B:** La alimentación proviene de la batería de emergencia y, por tanto, U3 (LTC4091) no hace su función de regulador, sino que deja pasar directamente la alimentación proveniente de la batería de emergencia. Esto implica que su rendimiento estará alrededor del 100%. Teniendo en cuenta que puede haber fugas de corriente se tomará el rendimiento como 99%. En este caso tenemos que:

$$I_{Total} = \frac{I_{TPS\ 63802}}{\eta_{LTC\ 4091_{Bateria\ Emergencia}}} = \frac{14.14mA}{0.99} = 14.28mA$$

En el *datasheet* de la batería TLI-1550S se indica la capacidad de la batería cuando esta se descarga a 50mA como máximo. Esta capacidad es de 1000mAh. Por tanto, tenemos que:

$$T = \frac{C_{Bateria}}{I_{Total}} = \frac{1000mAh}{14.28mA} = 70.03h \approx 3\text{días}$$

Con estos cálculos se demuestra que, incluso en el peor de los escenarios, se tienen aproximadamente 3 días de funcionamiento del sistema con la alimentación de la batería de emergencia, uno más que lo que se exponía en el apartado de requerimientos del proyecto.

### 3.5.5. Control de alimentación para el subsistema IEEE802.15.4E

Si bien no es un requerimiento funcional del proyecto, se ha decidido controlar el encendido del módulo Dusty. Para lograr esto se ha incorporado el chip TPS22917, el cual incorpora un transistor con una corriente de fuga baja, con un consumo muy bajo. Este componente permite controlar el paso de la alimentación de VIN a VOUT (ver en la figura siguiente) en función del estado del pin ON. En caso de que ON esté a nivel alto (HIGH), entonces, VIN pasa a VOUT y, por tanto, el módulo Dusty estará alimentado (encendido). En caso contrario, si ON es a nivel bajo (LOW), VOUT queda flotando (desconectado de VIN) y la tensión restante se descarga a través de RQOD a tierra, para asegurar que el circuito esté parado.

El *datasheet* del producto [27] recomienda implementarlo del siguiente modo:

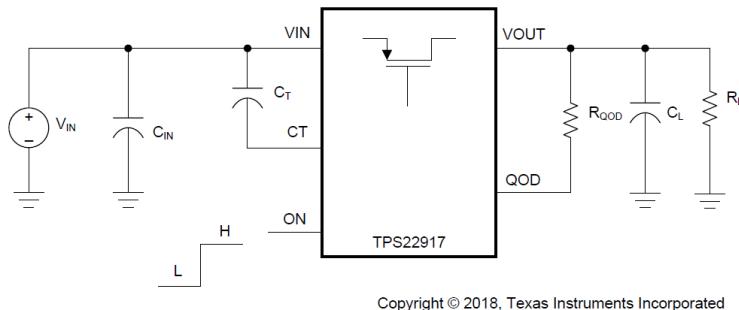


Figura 41 - Circuito para implementar TPS22917

En la página 6 se encuentran los parámetros de diseño de un ejemplo, estos son:

#### 9.2.1 Design Requirements

For this design example, use the values listed in Table 3 as the design parameters:

Table 3. Design Parameters

DESIGN PARAMETER	EXAMPLE VALUE
Input Voltage (V <sub>IN</sub> )	3.6 V
Load Current / Resistance (R <sub>L</sub> )	1 kΩ
Load Capacitance (C <sub>L</sub> )	47 μF
Minimum Fall Time (t <sub>f</sub> )	40 ms
Maximum Inrush Current (I <sub>RUSH</sub> )	150 mA

Figura 42 - Parámetros de diseño

En la página 17 se explican las fórmulas para hallar los valores de los componentes. Dado que en la aplicación de este proyecto no es importante la velocidad del cambio de estado del transistor, se han escogido los valores mínimos expresados en el *datasheet* para los parámetros de diseño de ejemplo.

### 9.2.2.1 Limiting Inrush Current

Use [Equation 5](#) to find the maximum slew rate value to limit inrush current for a given capacitance:

$$(\text{Slew Rate}) = I_{\text{RUSH}} + C_L$$

where

- $I_{\text{INRUSH}}$  = maximum acceptable inrush current (mA)
  - $C_L$  = capacitance on VOUT ( $\mu\text{F}$ )
  - Slew Rate = Output Slew Rate during turn on ( $\text{mV}/\mu\text{s}$ )
- (5)

Once the required slew rate shown in [Equation 1](#) can be used to find the minimum CT capacitance

$$CT = SR_{\text{ON}} + (\text{Slew Rate}) \quad (6)$$

$$CT = 1900 + 3.2 = 594 \text{ pF} \quad (7)$$

To ensure an inrush current of less than 150 mA, choose a CT value greater than 594 pF. An appropriate value should be placed on such that the  $I_{\text{MAX}}$  and  $I_{\text{PLS}}$  specifications of the device are not violated.

Figura 43 - Valor de CT

### 9.2.2.3 Setting Fall Time for Shutdown Power Sequencing

Microcontrollers and processors often have a specific shutdown sequence in which power must be removed. Using the adjustable Quick Output Discharge function of the TPS22917, adding a load switch to each power rail can be used to manage the power down sequencing. To determine the QOD values for each load switch, first confirm the power down order of the device you wish to power sequence. Be sure to check if there are voltage or timing margins that must be maintained during power down.

Once the required fall time is determined, the maximum external discharge resistance ( $R_{\text{DIS}}$ ) value can be found using [Equation 4](#):

$$t_{\text{FALL}} = 2.2 \times (R_{\text{DIS}} \parallel R_L) \times C_L \quad (8)$$

$$R_{\text{DIS}} = 630 \Omega \quad (9)$$

[Equation 3](#) can then be used to calculate the  $R_{\text{QOD}}$  resistance needed to achieve a particular discharge value:

$$R_{\text{DIS}} = \text{QOD} + R_{\text{QOD}} \quad (10)$$

$$R_{\text{QOD}} = 480 \Omega \quad (11)$$

To ensure a fall time greater than, choose an  $R_{\text{QOD}}$  value greater than 480  $\Omega$ .

Figura 44 - Valor de Rqod

El resultado es el siguiente:

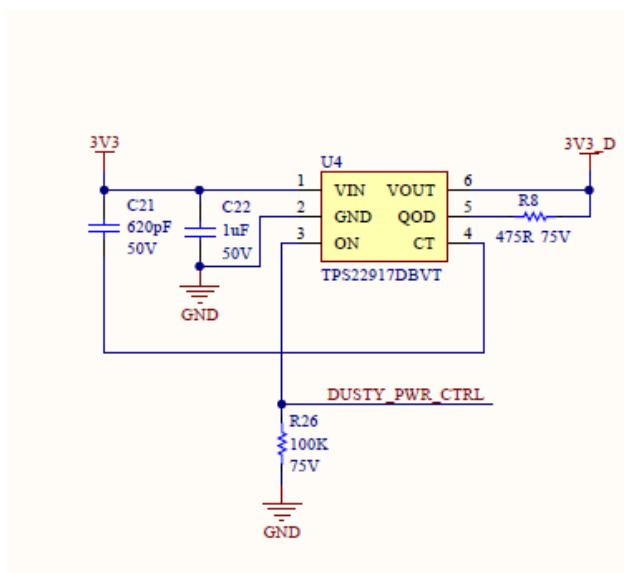


Figura 45 - Circuito para TPS22917

La señal 3V3, que proviene de Power Regulator, puede seguir su curso hacia el pin 6 de U4 (3V3\_D) en el caso de que la señal DUSTY\_PWR\_CTRL esté a nivel alto (un 1). Esta señal la controla el chip nRF9160 a través del pin 35 (P0.02). Si la señal DUSTY\_PWR\_CTRL se encuentra a 0, Dusty no recibirá alimentación y por tanto se mantendrá apagado.

Nótese que se ha incluido una resistencia de *pull-down* (R26) en paralelo a la señal DUSTY\_PWR\_CTRL para asegurar que al encenderse la placa la señal ON del chip (TPS229) está conectada a tierra, luego VIN y VOUT están desconectados y por tanto el módulo DUSTY está apagado.

### 3.6. Diseño del subsistema de interfaz con el usuario

Para obtener señales visuales del estado de la placa se incorporarán 4 LEDs. El LED verde (D1) está destinado a encenderse cuando la placa esté ON. La funcionalidad de los demás LEDs se podrá determinar según las necesidades del producto final a través del software.

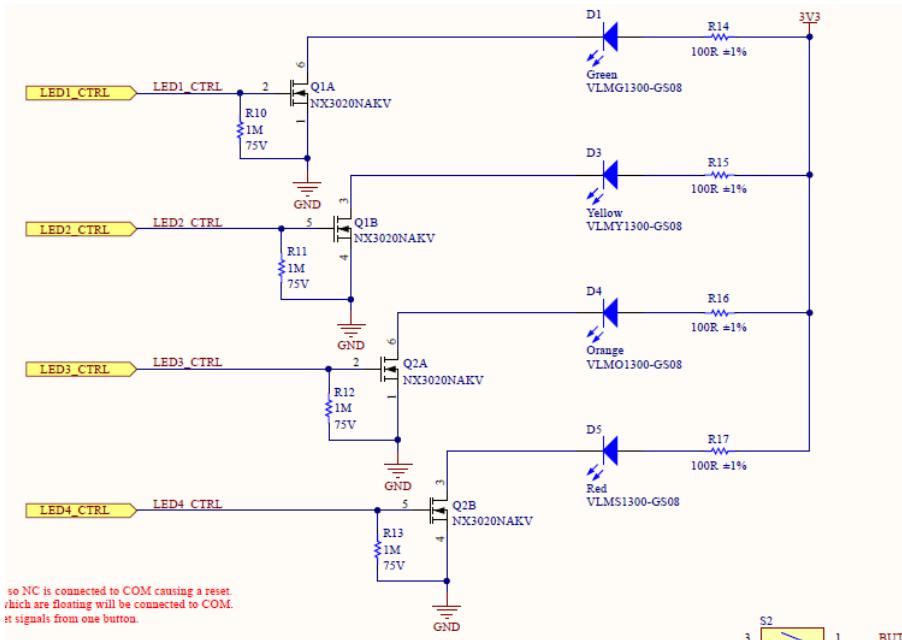


Figura 46 - Circuito para los LEDs

Las señales de control de los LEDs van conectadas a los pines P0.02, P0.03, P0.04, P0.05 del MCU nRF9160. Las resistencias de *pull-down* (R14, R15, R16, R17) aseguran que cuando la placa se enciende, los transistores tienen 0V en la base, de modo que los LEDs se encuentran apagados a pesar de que el pin GPIO esté flotando y pueda existir una tensión transitoria. Cuando un transistor recibe un nivel alto de tensión (un 1, VCC) el transistor (tipo N) se activa y deja pasar la corriente desde 3V3 hasta GND de modo que el LED correspondiente se enciende.

Como el circuito está alimentado a 3V3, y los LEDs tienen una caída de tensión de entre 2 y 2.4V y soportan una corriente máxima de 20mA

$$R = \frac{V_{out} - V_{LED}}{I_{LED\_MAX}} = \frac{3.3 - 2}{20 \cdot 10^{-3}} = 65\Omega$$

Finalmente se ha escogido una resistencia de  $100\Omega$ , de modo que la corriente será de 13mA y los LEDs brillarán con menor intensidad. Este valor se ha escogido para simplificar el BOM.

También se han implementado dos botones (ver circuito de la figura siguiente). Hay que destacar, que en ambos casos se debe utilizar la resistencia de *pull-up* que incorpora el microcontrolador, de modo que el pin de entrada tenga una tensión determinada (1 cuando no se pulsa, 0 cuando se pulsa). El botón de reset (SW2) servirá para reiniciar la placa de manera

asíncrona a través de la pulsación del usuario. La funcionalidad de SW3 se determinará según las necesidades del producto final mediante software.

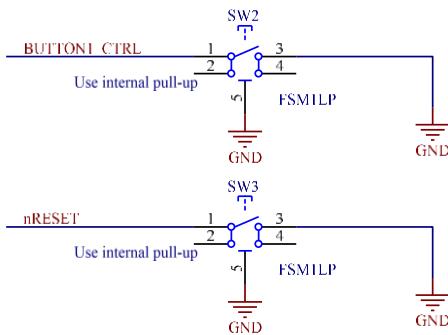


Figura 47 - Circuito de botones básicos

Posteriormente se ha modificado el circuito del botón de *reset* para dividir esta señal en dos, de modo que tanto nRF9160 como DUSTY tengan su propia señal de *reset* y de esta forma no se reinicen ambas cuando se está haciendo *debugging* o se está programando cada uno de ellos.

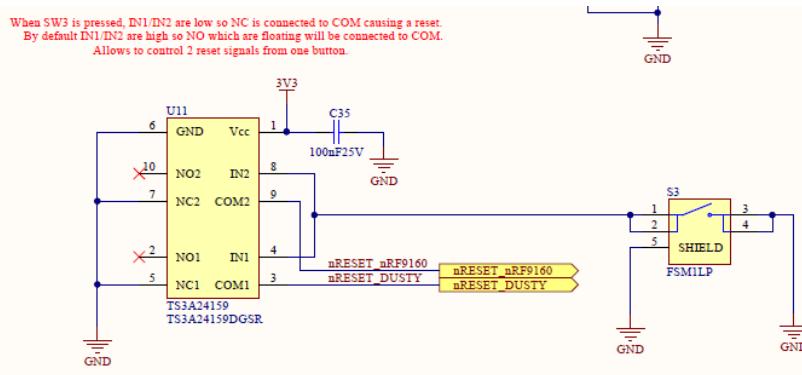


Figura 48 - Circuito para el multiplexado de las señales de reset

Para ello, se ha instalado un multiplexor (U11) que commuta la señal de *reset* dependiendo del estado del botón S3. Cuando SW3 está sin presionar, IN1 e IN2 detectan VCC y por tanto los puertos NO1 y NO2 están conectados a COM1 y COM2 respectivamente. En este caso NO1 y NO2 están flotando y, por tanto, las señales nRESET\_nRF9160 y nRESET\_DUSTY tendrán el valor definido por el *pull-up* interno o por la señal de *reset* del JTAG. Cuando S3 es presionado, IN1 e IN2 están conectados a GND y por tanto los puertos NC1 y NC2 están conectados a COM1 y COM2 respectivamente. Este es el caso en el que las señales de *reset* toman el valor de GND y causan un *reset* en ambos controladores.

### 3.7. Estimación de costes

El BOM (Bill of Materials) es la tabla donde se registran los componentes que se han de adquirir para la producción del producto final, sus características y su precio. A cada componente de los diseños se le asigna un identificador único en función de su tipología (por ejemplo, R para las resistencias y C para los condensadores). En cada fila del documento se describe el componente, en algunos casos se especifica si se ha de montar o no, se especifica el *footprint*, el fabricante y el *part number* que le da el fabricante al componente. Cuando no se especifica el *part number* del componente ni el fabricante, es porque es indiferente y lo que realmente importa es el

valor/características de este. El BOM de este proyecto se ha adjuntado en un anexo al final de esta memoria.

A continuación, se procederá a incluir el precio de cada componente de manera que se pueda hacer un presupuesto del producto para diferente cantidad de unidades. Además, también se deberá presupuestar la fabricación de la PCB y el montaje de los componentes. Las cantidades para las cuales se va a hacer la aproximación del presupuesto son 1, 100 y 1000.

Manufacturer_Part_Number	1	100	1000	Precio para 1	Precio para 100	Precio para 1000
TLI-1550S	25,78	21,23	20,51	25,78	21,23	20,51
885012006052	0,10	0,024	0,02642	0,2	0,048	0,05284
885012206071	0,01	0,0406	0,04288	0,12	0,4872	0,51456
GRM32EC81C476KE15L	0,62	0,2865	0,18037	1,24	0,573	0,36074
EMK212B7475KG-T	0,27	0,1056	0,0652	0,54	0,2112	0,1304
0402N390J500CT	0,1	0,0124	0,00696	0,3	0,0372	0,02088
CC0402JRNPO9BN101	0,1	0,0088	0,00497	0,2	0,0176	0,00994
885012205018	0,1	0,0154	0,02066	0,2	0,0308	0,04132
C0603C621J5GACTU	0,36	0,1511	0,0957	0,36	0,1511	0,0957
GCM21BR71H105KA03L	0,34	0,1339	0,0827	0,34	0,1339	0,0827
GRM1885C1H103JA01D	0,2	0,0682	0,04126	0,2	0,0682	0,04126
CC0603JRNPO9BN300	0,1	0,0316	0,01774	0,2	0,0632	0,03548
CGA3E1X7R1V474K080AC	0,24	0,0935	0,05775	0,24	0,0935	0,05775
CGA4J1X7R1V335K125AC	0,45	0,2079	0,1309	0,45	0,2079	0,1309
GRM31CR71H475KA12L	0,46	0,19	0,12033	0,46	0,19	0,12033
EEEFK1C101P	0,53	0,2225	0,14093	0,53	0,2225	0,14093
C3216X5R1A686M160AC	1,44	0,8539	0,56925	1,44	0,8539	0,56925
MC0603B221K500CT	0,04	0,03	0,017	0,04	0,03	0,017
CC0603JRNPO9BN100	0,1	0,0181	0,01018	0,1	0,0181	0,01018
C0603C154K5RACTU	0,18	0,0595	0,03455	0,18	0,0595	0,03455
C1206C226K8RACTU	0,74	0,3762	0,25079	1,48	0,7524	0,50158
04025A1R0BAT2A	0,3	0,1161	0,07173	0,3	0,1161	0,07173
GJM1555C1H3R6CB01D	0,12	0,0377	0,02118	0,12	0,0377	0,02118
VLMG1300-GS08	0,39	0,1633	0,08289	0,78	0,3266	0,16578
VLMY1300-GS08	0,39	0,1633	0,08289	0,39	0,1633	0,08289
VLMO1300-GS08	0,39	0,1633	0,08289	0,39	0,1633	0,08289
VLMS1300-GS08	0,36	0,1537	0,07802	0,36	0,1537	0,07802
B360A-13-F	0,4	0,2218	0,12474	0,4	0,2218	0,12474
BLM15AX121SN1D	0,1	0,0399	0,02299	0,1	0,0399	0,02299
1028C	0,83	0,7541	0,46758	0,83	0,7541	0,46758
SMA CONNECTOR	1,58	1,11	1	3,16	2,22	2
SIM8055-6-1-14-00-A	1,48	1,125	0,92744	1,48	1,125	0,92744
2211S-06G	0,043	0,031	0,027	0,043	0,031	0,027
3221-10-0100-00	0,65	0,483	0,42	0,65	0,483	0,42
1028	2,03	1,4132	1,13989	2,03	1,4132	1,13989
1751248	1,34	1,1984	0,952	1,34	1,1984	0,952
MLG1005S1N5BT000	0,14	0,085	0,05954	0,14	0,085	0,05954

LQW15AN22NH00D	0,17	0,104	0,07276	0,17	0,104	0,07276
LQW15AN68NG00D	0,18	0,1089	0,07623	0,18	0,1089	0,07623
LQW15AN9N1G00D	0,18	0,1089	0,07623	0,18	0,1089	0,07623
XAL4030-682MEC	3,4	1,92	1,16	3,4	1,92	1,16
XFL4015-471MEC	2,5	2,05	1	2,5	2,05	1
NX3020NAKVYL	0,474	0,194	0,082	0,948	0,388	0,164
SIS447DN-T1-GE3	0,97	0,6568	0,41536	1,94	1,3136	0,83072
PMN100EPAX	0,48	0,264	0,1485	0,48	0,264	0,1485
NX7002AKVL	0,181	0,08	0,036	0,362	0,16	0,072
NX3008PBKW	0,403	0,118	0,056	0,403	0,118	0,056
CRCW060310K0FKEA	0,1	0,0156	0,00701	1,3	0,2028	0,09113
CRCW06031K00FKEA	0,1	0,0145	0,00652	0,4	0,058	0,02608
CRCW0603100KFKEA	0,1	0,0156	0,00701	0,4	0,0624	0,02804
MCS04020C1004FE000	0,13	0,0465	0,02046	0,13	0,0465	0,02046
CRCW0603475RFKEA	0,1	0,0145	0,00652	0,1	0,0145	0,00652
CRCW06031M00FKEA	0,1	0,0156	0,00701	0,4	0,0624	0,02804
CR0603-FX-1000ELF	0,1	0,0068	0,00305	0,4	0,0272	0,0122
ERJ3EKF4990V	0,1	0,0149	0,00582	0,2	0,0298	0,01164
NTCLE300E3103SB	2,76	1,4903	1,15497	2,76	1,4903	1,15497
CRCW06034K99FKEA	0,1	0,0156	0,00701	0,1	0,0156	0,00701
CRCW0603200RFKEA	0,1	0,0156	0,00701	0,1	0,0156	0,00701
CRCW060315K0FKEA	0,111	0,017	0,008	0,111	0,017	0,008
MCT06030C5603FP500	0,13	0,0465	0,02046	0,13	0,0465	0,02046
CRCW060340K2FKEA	0,1	0,0156	0,00701	0,1	0,0156	0,00701
ERJ3EKF5113V	0,1	0,0149	0,0067	0,1	0,0149	0,0067
CR0603-FX-9102ELF	0,101	0,012	0,007	0,101	0,012	0,007
CRCW060312K0FKEA	0,1	0,0156	0,00701	0,2	0,0312	0,01402
CRCW060327K4FKEA	0,1	0,0156	0,00701	0,1	0,0156	0,00701
FSM1LP	0,35	0,2798	0,21254	0,7	0,5596	0,42508
JS202011SCQN	0,55	0,4418	0,338	0,55	0,4418	0,338
AS1D-2M-10-Z	2,27	1,7451	1,36336	2,27	1,7451	1,36336
NRF9160-SICA-R7	23,73	18,7139	16,99022	23,73	18,7139	16,99022
SKY65943-11	3,27	2,186	1,5694	3,27	2,186	1,5694
LTC4091EDJC#PBF	7,8	5,8302	5,0985	7,8	5,8302	5,0985
TPS22917DBVT	0,65	0,4371	0,28525	0,65	0,4371	0,28525
ITM-DYUF-B-02	32,25	32,25	32,25	32,25	32,25	32,25
CY62138FV30LL-45ZXI	4,42	3,49875	3,35859	4,42	3,49875	3,35859
74AUP1G58GW,125	0,48	0,2244	0,11813	0,96	0,4488	0,23626
74AHC573PW,118	0,45	0,2882	0,17497	0,9	0,5764	0,34994
TS3A24159DGSR	1,24	0,8631	0,6195	1,24	0,8631	0,6195
TPS63802DLAT	2,08	1,5014	1,16375	2,08	1,5014	1,16375
EMIF03-SIM02M8	0,5	0,4181	0,25733	0,5	0,4181	0,25733
Precio para 1 unidad (En dólares)				146,1	112,2	99,547

Tabla 2 - Precio de los componentes en dólares

Ahora se debe sumar el coste de la producción a cada caso y se obtiene el coste total de una placa para cada caso.

- **Para 1 placa:** El coste estimado para una placa sigue la siguiente fórmula

$$\$_{Unidad} = \$_{Materiales} + \$_{PCB} + \$_{Producción}$$

$$\$_{Unidad} = 146'10\$ + 28'10\$ + 69'30\$ = 174'20\$$$

- **Para 100 placas:** El coste unitario estimado si se producen 100 placas es el siguiente

$$\$_{Unidad} = 112'20\$ + 20'80\$ + 17'70\$ = 150'70\$$$

- **Para 1000 placas:** El coste unitario estimado si se producen 1000 placas es el siguiente

$$\$_{Unidad} = 99'547\$ + 16'60\$ + 11'54\$ = 127'687\$$$

El coste de los componentes y, por ende, el coste total, se reduce cuantas más unidades se produzcan. A este coste se le debe sumar los costes de ingeniería (diseño, desarrollo y el precio/hora) y los costes de la certificación del producto (para Europa el certificado de la UE y para América el certificado FCC). Todo esto forma parte de los costes CAPEX, una vez se despliegue el producto, se deben tener en cuenta los costes de la operación o OPEX (contrato de las tarjetas SIM de cada *gateway*, mantenimiento, etc.).

## 4. Validación del hardware

En este capítulo se explica que es la validación del hardware. También se expone el plan de validación de hardware que se va a seguir y los dos tipos de validaciones que se van a llevar a cabo, la eléctrica y la lógica.

### 4.1. Plan de validación

El plan de validación sirve para testear el correcto funcionamiento del sistema diseñado una vez se produzca. Este plan de validación consta de dos partes principales: la validación eléctrica y la validación lógica.

Con la validación eléctrica se busca comprobar que no hay ningún error o defecto en la electrónica. No sólo interesa ver que al conectar la placa no hay componentes funcionando de manera incorrecta, también se busca medir las tensiones e intensidades de los puntos críticos del sistema para asegurar la integridad de este. Esto se logra de forma visual y de forma métrica. Con el primer contacto con la fuente de alimentación se puede comprobar que no hay cortocircuitos a simple vista, posteriormente las mediciones dictaminarán si el sistema está funcionando de forma correcta o si, en cambio, hay algún problema, en cuyo caso se buscará la causa del mal funcionamiento y se corregirá para aplicarse en la siguiente versión. Estos problemas pueden deberse al diseño o a la fabricación.

Con la validación lógica se busca comprobar que el sistema funciona correctamente a nivel lógico, es decir que todos los componentes responden correctamente y de forma lógica al programar el sistema. Esto se logra cargando programas que validan cada módulo del sistema y la interacción entre ellos (SIM y LTE, LEDs y Botones, comunicación entre nRF9160 y Dusty mediante las UARTs, etc.).

### 4.2. Validación eléctrica

Después de realizar una inspección visual de la placa, antes de realizar las medidas de las tensiones e intensidades, es una buena práctica el comprobar el valor de las resistencias más críticas del circuito con tal de prevenir fallos al alimentar el sistema y para asegurar que las resistencias que se han colocado son las correctas. Posteriormente comprueba que, al conectar la fuente de alimentación principal, no hay ningún cortocircuito. Para ello se utilizará una fuente de alimentación regulable con la corriente limitada para evitar que, en el caso de que exista un cortocircuito, no se quemen componentes. Esto se hará tanto para la entrada de alimentación principal, como para la de la batería de emergencia. Finalmente, se realizarán mediciones en el circuito de alimentación para comprobar que la alimentación y la funcionalidad es correcta.

Estas se van a realizar como sigue.

Primero se realizarán las mediciones en el circuito de Power Switch.

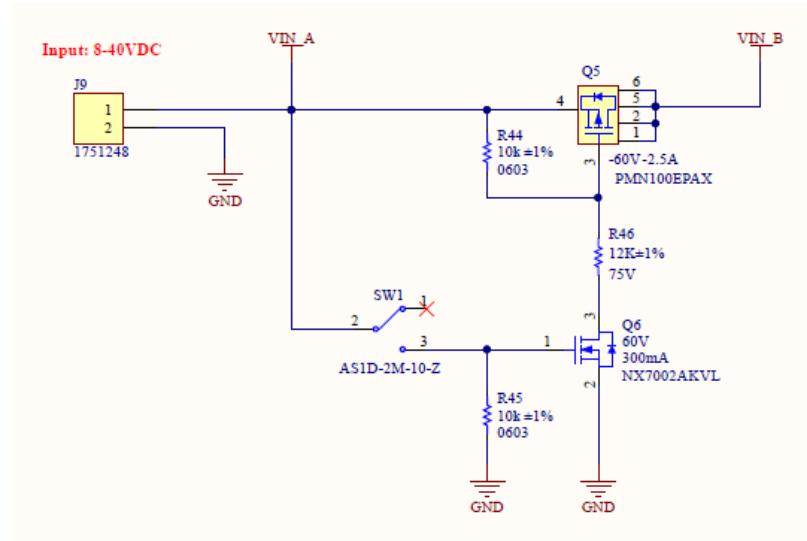


Figura 49 - Power Switch

En el circuito Power Switch se medirán con un multímetro los puntos VIN\_A, VIN\_B, Q6-1 y Q6-3. Los puntos VIN\_A y VIN\_B deben tener la misma tensión de 12 Volts, provenientes de la fuente de alimentación principal, una vez el switch SW1 esté en la posición 2-3. La tensión en Q6-1 debería ser de 0 Volts cuando SW1 está en la posición 2-1, y debería ser igual que VIN\_A cuando SW1 se encuentra en la posición 2-3. La tensión en el punto Q6-3 debería ser igual a VIN\_A cuando SW1 se encuentre en la posición 2-1, y 0V cuando SW1 se encuentre en la posición 2-3. También se debe medir el valor de la tensión entre R44 y R46 para asegurar que las resistencias del divisor de tensión son correctas.

Una vez comprobado el circuito de Power Switch se realizarán las mediciones pertinentes en el circuito de Power Backup.

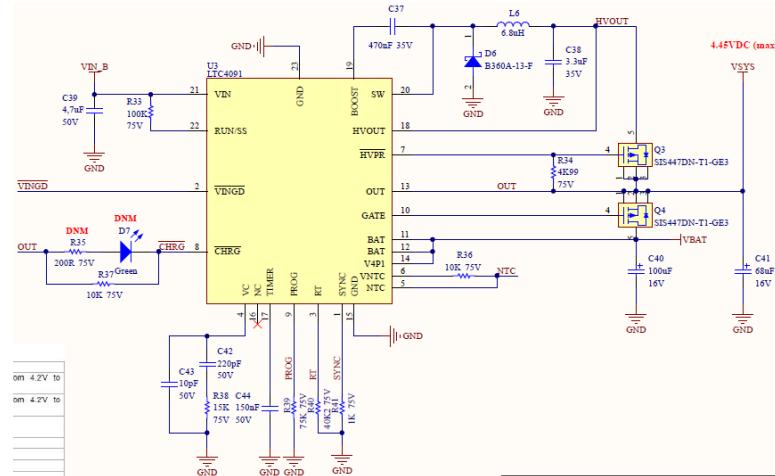


Figura 50 - Power Backup

Para testear el correcto funcionamiento de este circuito se medirá la salida de LTC4091 con y sin conexión a la fuente de alimentación principal. El primer paso es medir la salida con a la fuente de alimentación principal conectada. El valor de VSYS (medido en el pin OUT de U3) debería de estar entre los 3.5 Volts y los 4.45 Volts como indica el fabricante en el *datasheet* del componente, aunque el valor exacto dependerá de cuán cargada esté la batería. Posteriormente se quitará la alimentación principal y se realizará nuevamente la medición de VSYS. En este caso

el voltaje medido es el de la batería de emergencia y deberá estar entre los 2.5V y los 4.2V como indica su *datasheet*. Además de las mediciones anteriores se deben hacer las comprobaciones siguientes: que el valor del pin CHRG es 1 cuando la batería se está cargando y 0 cuando la batería no se está cargando, que el valor de NTC sea el esperado según el sensor NTC montado y que los valores de PROG, RT y SYNC se correspondan con los que recomienda el *datasheet*. También se debe hacer un test del ciclo de carga de una batería para asegurar que el componente es capaz de cargar correctamente una batería externa. Para este test se medirá el amperaje de una batería descargada y se comprobará que cuando este amperaje baja, el voltaje sube para que la corriente suba de nuevo. Si las subidas son graduales y la batería se carga de forma adecuada se habrá pasado el test satisfactoriamente.

En el siguiente test se realizarán las medidas al circuito de Power Regulator.

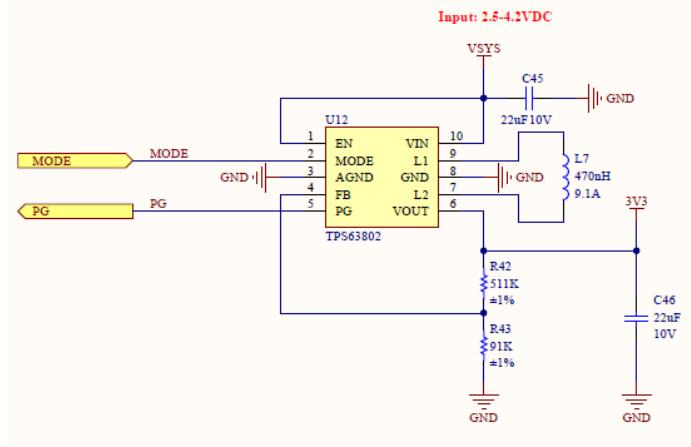


Figura 51 – Power Regulator

En el circuito de Power Regulator se realizará una medición en 3V3 para comprobar que realiza la conversión del voltaje. Para esto se deben crear artificialmente las situaciones en las que el componente debe hacer la función de *boost* y las situaciones en la que el componente debe realizar la función de *buck*. Para ello se debe desconectar la fuente de alimentación principal y se debe usar una fuente de alimentación a la que se le pueda regular el voltaje para simular una batería cargada y otra descargada. Tanto cuando se simule la batería cargada (TPS63802 realizará la función *buck*) como cuando se simule la batería descargada (TPS63802 realizará la función *boost*), el voltaje en 3V3 debe de ser aproximadamente 3.3 Voltios, con un error máximo de  $\pm 0.3\%$ , de forma continuada. Esto también ha de pasar cuando se le conecta la fuente de alimentación principal al sistema. Además de las tensiones se debe comprobar que el valor de la señal PG es el correcto en cada caso, el divisor de tensión de R42 y R43 es correcto midiendo el voltaje en el pin FB y que la señal MODE tiene continuidad con el procesador.

El siguiente test sirve para comprobar el correcto funcionamiento del dispositivo que alimenta el SOM Dusty.

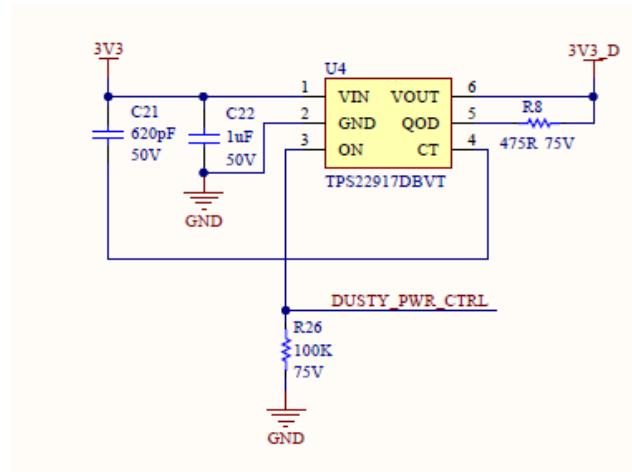


Figura 52 - Circuito para el control de alimentación del subsistema IEEE802.15.4E

Una vez comprobada la salida de Power Regulator, se debe comprobar que cuando la señal DUSTY\_PWR\_CTRL es un 1 lógico, el voltaje de 3V3 es el mismo que el de 3V3\_D y que cuando esta señal es un 0 lógico, 3V3\_D es 0 Volts y por tanto Dusty no está alimentado. También se debe comprobar la continuidad de la señal con el procesador.

Finalmente, si los voltajes son correctos, se mide la corriente pico del sistema y si esta no supera los 1.5 Amperios que se esperan como máximo cuando el módem NB-IoT está transmitiendo, se procede a la validación lógica.

En esta sección se han hecho comprobaciones sobre los modos de fallo que puede haber al testear la placa, pero al producirla hay que incluir otros tipos de modos de fallo en los tests y compilarlos en los manuales de usuario y el *datasheet* del producto. Estos pueden ser, por ejemplo: Modos de fallo causados por el usuario (¿Qué pasa si el usuario instala la batería al revés?, ¿Y si usa una fuente de características distintas a las que se especifican en el *datasheet* del producto final?, etc.), modos de fallo causados por el paso del tiempo (¿Qué pasa si la batería se gasta y empieza a verter líquido y este crea un cortocircuito?, ¿Y si algún componente falla debido al desgaste con el tiempo?).

### 4.3. Validación lógica

Una vez se ha realizado la validación eléctrica, se procede a realizar la validación lógica que consiste en una serie de programas que comprueban el correcto funcionamiento de la placa a nivel lógico. Estos programas son:

- **Programa para LEDs y Botones:** Este programa comprueba que tanto los leds como los botones que se han instalado, funcionan correctamente. Para esto lo que se ha programado es una rutina que enciende y apaga los LEDs cuando se pulsan los botones, en este caso cuando se pulsa el botón programable ya que el otro está dedicado a resetear la placa. De forma visual se puede comprobar que, si los LEDs se encienden cada vez que se pulsa un botón, estos funcionan bien a nivel electrónico y responden a comandos de nRF9160.
- **Programa para GPIOs:** El programa que comprueba los pines GPIO consta de varios tests. Se testean todas las señales involucradas en el funcionamiento de la placa y que de algún modo son programables desde nRF9160. Estas señales son:
  - **BATT\_SUPPLY:** Para el testeo de la señal BAT\_SUPPLY, se le pide al usuario desconectar la fuente de alimentación principal, de este modo la lectura en el

pin P0.08 debería de ser un 1. Posteriormente, si se cumple la condición anterior, se le pide al usuario que conecte la fuente de alimentación principal de modo que la lectura sea un 0. Si esta condición se cumple, se pasa al testeo de la siguiente señal.

- **BATT\_CHARGE:** Para el testeo de la señal BATT\_CHARGE, se le pide al usuario que quite la batería de emergencia. El programa realiza una lectura del pin P0.09 y este debe estar a 1 mientras no haya batería. Si esto se cumple, se le pide al usuario que inserte una batería descargada de modo que la lectura sea un 0. Finalmente, y si la condición anterior se cumple, se le pide al usuario insertar una batería completamente cargada de modo que la lectura sea 1. Con este método se comprueba el correcto funcionamiento de todos los posibles valores lógicos de la señal.
  - **P\_GOOD:** Para el testeo de esta señal, se le pide al usuario que desconecte la fuente de alimentación principal y que instale una batería de emergencia descargada. De este modo el voltaje a la salida de TPS63802 será menor al 90% del voltaje nominal y la lectura del pin P0.12 deberá ser un 0. Posteriormente si la condición anterior se cumple, se le pide al usuario que vuelva a conectar la fuente de alimentación principal y cambie la batería por una cargada de modo que el voltaje a la salida de TPS63802 sea mayor al 95% del voltaje nominal y por tanto se lea un 1 en el pin P0.12. Si se cumplen ambas condiciones, se pasa al testeo de la siguiente señal.
  - **P\_MODE:** Se configura nRF9160 para tener un 1 en el pin P0.10 y se le pide al usuario que mida la salida. Posteriormente, si el usuario introduce un comando afirmando que la medición era correcta, se configura nRF9160 para tener un 0 en el mismo pin. Como antes, se le pide al usuario que mida la salida y confirma mediante un comando que es un 0. Si se cumple correctamente se pasa al testeo de la siguiente señal.
  - **DUSTY\_PWR\_CTRL:** El testeo de esta señal se realiza del mismo modo que el anterior, se le pide al usuario que haga las mediciones y se configura el pin en ambos estados (0 y 1). Posteriormente si se cumplen ambos estados de forma lógica y correctamente, se pasa al testeo de la siguiente señal.
  - **GPS\_1PPS:** Se configura el pin COEX1 para que tenga una señal de un pulso por segundo y luego se pide al usuario que realice la medición y confirme mediante un comando si esto es correcto. Si esto se cumple, se pasa al testeo de la siguiente señal.
  - **GPS\_LNA:** El testeo de esta señal se realiza del mismo modo que el de las señales P\_MODE y DUSTY\_PWR\_CTRL, se le pide al usuario que haga las mediciones y se configura el pin en ambos estados (0 y 1).
- **Programa UARTs:** Con este programa se busca comprobar que la comunicación entre ambos módulos (nRF9150 y Dusty) existe y se da de forma correcta. Para ello se ha programado que nRF9160 pregunte por el modelo de hardware, la revisión del hardware y la MAC al módulo Dusty. Dusty tiene que devolver por puerto serie esta información y si lo hace significa que hay una conexión establecida entre ambos módulos y que esta se da de forma correcta ya que se entregan los datos sin errores.
  - **Programa LTE y SIM:** Este programa comprueba en primera instancia si la señal SIM\_DET del portaSIM funciona correctamente. Para ello se pide que el usuario saque o introduzca la tarjeta a la vez que se hace una lectura del pin donde se halla la señal SIM\_DET. Si la acción del usuario se corresponde con la lectura, se informa al usuario

que el sistema funciona correctamente. Posteriormente, la rutina inicia el modem LTE para comprobar que cuando la SIM está insertada, la conexión LTE es posible y no hay ningún error. Esta segunda parte del código avisará al usuario de cuando se establece la conexión y cuando no hay conexión LTE. De este modo, si el usuario está en un área con cobertura LTE y con la sim insertada en el sistema, pero la placa no se conecta, sabrá que se debe a un error.

Si no ha habido ningún error el programa debe recorrer todos los tests, si hay cualquier error, el programa no avanza y se queda en la fase donde se esté produciendo dicho error.

## 5. Diseño y desarrollo del firmware

En esta sección se va a explicar el firmware que se cargará en el sistema. Este firmware debe recibir los paquetes provenientes de la red de sensores de WirelessHART a través de la conexión UART con el módulo Dusty, preparar estos paquetes para su transmisión por MQTT y finalmente prepararlos para transmitirse por NB-IoT/CAT-M1 y finalmente, transmitirlos. El firmware debe permitir monitorizar de forma remota la red WirelessHART.

Para esto son necesarios varios elementos. Estos elementos son: la API de Dust Networks que se debe integrar dentro de los *drivers* UART de Nordic Semiconductor, dos colas FIFO (una *upstream* y otra *downstream*), dos *threads* (uno para controlar la conexión MQTT y otro para la comunicación Dusty - nRF9160), un broker MQTT y la configuración de un *backend* (una aplicación final que gestione los datos que llegan por MQTT). El sistema operativo de nRF9160 es Zephyr, que es un RTOS (Real Time Operating System). Es necesario un RTOS para la ejecución de manera concurrente de los procesos (*threads*) necesarios para hacer posible el funcionamiento del sistema.

En el siguiente apartado se explica la arquitectura que tiene este firmware, posteriormente se explicará su implementación y se comentará con más profundidad las funciones usadas y la funcionalidad del programa en sí.

### 5.1. Arquitectura de la solución

La Figura 53 es un diagrama de secuencia donde se puede ver la arquitectura de la solución y cómo funciona el firmware.

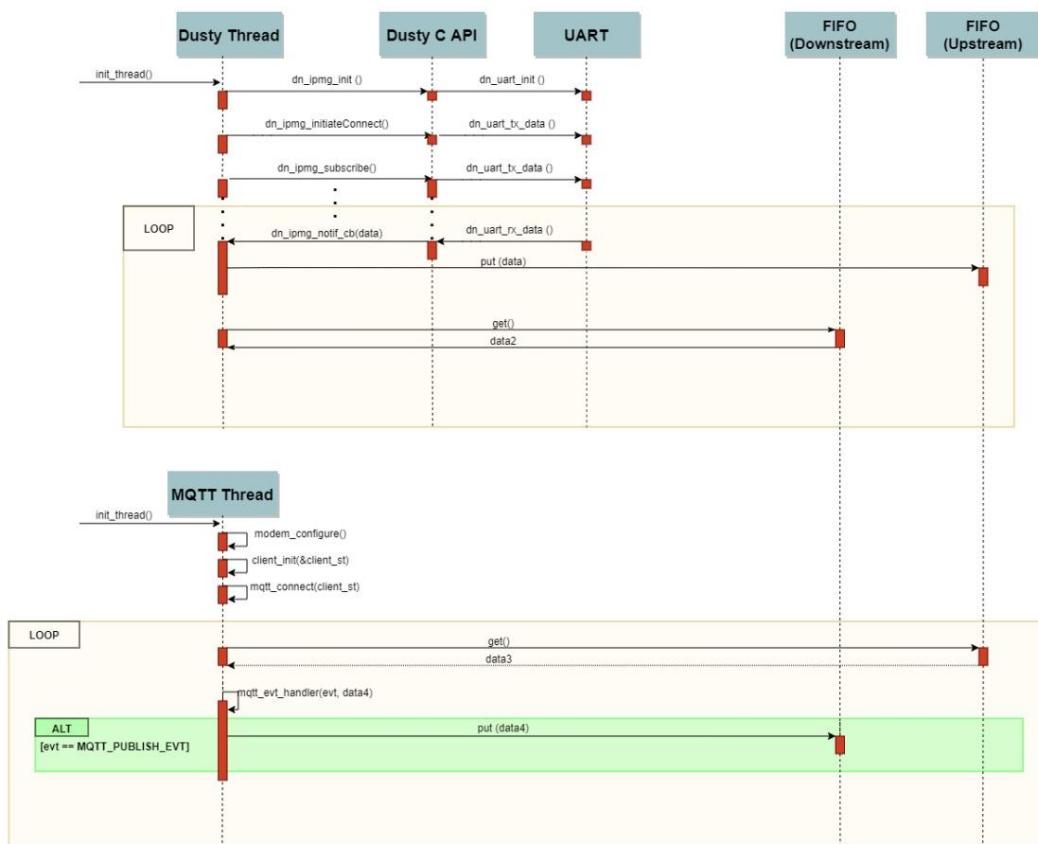


Figura 53 - Diagrama de secuencia del firmware

Como se puede observar, la solución consta de dos *threads* que se ejecutan en paralelo una vez la placa se enciende. Estos *threads* son Dusty Thread y MQTT Thread. También se puede ver en el diagrama que estos *threads* se pasan información entre ellos usando dos colas FIFO, una *upstream* y la otra *downstream*.

Dusty Thread es la rutina encargada de comunicar el módulo nRF9160 con el *manager* Dusty. Esto lo hace mediante el envío de comandos a la API de Dusty (ver en la figura anterior). Dado que el módulo Dusty no es programable (si configurable), toda interacción con él debe realizarse usando la UART y la API de Dusty. Solo se le pueden consultar datos mediante comandos enviados por UART y así es como se recogen los datos de los sensores Wireless HART.

Por otro lado, MQTT Thread es la rutina encargada de recibir los datos provenientes del *manager* Dusty a través del *thread* de Dusty, mediante la cola FIFO *upstream* y enviarlos hacia el *backend* usando el protocolo MQTT y la conexión LTE. La parte del envío por LTE no está representada en el diagrama dado que se realiza internamente en una librería. El envío de los datos hacia el *backend* se realiza en el bucle de la función `MQTT_thread`. Aún se debe de implementar (pero se puede ver en el diagrama) un camino inverso. Este camino servirá para que, si se recibe algo desde el *backend* mediante MQTT, el sistema pueda recibir este paquete y depositarlo en la cola *downstream* y que este se envíe a la API de Dusty lo cual se usaría para cambiar la configuración del *manager* de manera remota.

## 5.2. Implementación

En esta sección se explicarán con más detalle las funciones de más importancia usadas en el firmware para cada uno de los dos *threads* y para la UART.

- **`init_thread()`**: Crea e inicializa un *thread*.
- **Dusty Thread**
  - **`dn_ipmg_init()`**: Inicializa a conexión serie (UART), resetea y asigna a variables la dirección de memoria de las *callbacks* necesarias para la conexión entre nRF9160 y el manager Dusty.
  - **`dn_ipmg_initiateConnect()`**: Inicia la conexión entre nRF9160 y el módulo Dusty mediante UART.
  - **`dn_ipmg_subscribe()`**: nRF9160 se suscribe a ciertas notificaciones periódicas del manager Dusty. Estas notificaciones son seleccionables en la misma función mediante una máscara de bits.
  - **`dn_ipmg_notif_cb(data)`**: *Callback* con los datos solicitados por el cliente nRF9160 en la configuración de las notificaciones de la función anterior. Estos datos se entregan periódicamente.
- **MQTT Thread**
  - **`modem_configure()`**: Configuración del módem LTE para ofrecer conexión.
  - **`client_init(&client_st)`**: Inicializa la estructura (*struct*) del cliente MQTT. Se proveen datos como el *broker* que se va a usar, el id de cliente MQTT de nRF9160, usuario y contraseña en caso de tratarse de una conexión privada, el QoS de la conexión, etc.
  - **`mqtt_connect(client_st)`**: Establece la conexión entre el cliente MQTT (nRF9160) y el broker MQTT.
  - **`mqtt_evt_handler(evt, data4)`**: Dependiendo del evento MQTT que suceda, gestiona flujo de trabajo de MQTT. En el caso en el que el *broker* detecte una publicación en el tópico suscrito por el nRF9160, el *event handler* provee

los datos que se deben poner en la cola *downstream* para hacer un posible cambio de configuración de Dusty de forma remota. En caso de que se haya una desconexión de LTE o caiga el *broker* y por tanto se pierda la sesión MQTT, el *event handler* informa de que hay una desconexión y hace que nRF9160 se intente reconnectar. Esta función se puede programar de un modo más extenso para optimizar el funcionamiento del sistema y cubrir otros posibles eventos.

- **UART**

- **dn\_uart\_init()**: Inicializa a conexión serie (UART), asigna a una variable la *callback* de recepción de bytes (**dn\_uart\_rx\_data()**).
- **dn\_uart\_tx\_data()**: Coloca en el buffer de transmisión los datos a ser transmitidos. Estos datos se envían de byte en byte.
- **dn\_uart\_rx\_data()**: *Callback* de recepción de datos. Recibe los datos que vienen de la UART y los almacena en un buffer de entrada.

Tanto las funciones de Dusty Thread, como las funciones de UART, son de la API de Dust Networks. Antes de ser usada, la API debe ser integrada en el proyecto del firmware. Para ello se debe programar el módulo UART, el cual viene en blanco dentro de los ficheros de la API dado que la funcionalidad de la UART depende completamente del hardware donde se vaya a ejecutar el firmware.

## 6. Validación de la solución

La última parte del proyecto consiste en la validación de la solución. Una vez se produce la placa prototipo es necesario probar el sistema entero. Para esto se debe cargar el firmware en la placa y dejar el sistema funcionando durante unos días. Si no llegan los paquetes a su destino, o estos llegan en un formato incorrecto, o hay alguna otra clase de error, el problema es probablemente del firmware, ya que se han realizado los tests eléctricos y lógicos previamente. La única parte del sistema que puede dar problemas en esta etapa es el firmware o la configuración del *backend* y el servidor MQTT de modo que detectar y, por tanto, solucionar los posibles errores que se den es fácil. Esta validación se puede realizar con la placa de desarrollo nRF9160-DK o con la placa diseñada a lo largo de esta memoria.

Para realizar la validación previa a la producción de la placa que se ha diseñado, se necesita una placa de desarrollo nRF9160-DK, dos o más chips LTP5901 (uno actuando como *manager* y los otros actuando como dispositivos sensores) y seguir los siguientes pasos:

1. Instalar y configurar el entorno de desarrollo de Nordic Semiconductor el cual está compuesto por nRF Connect, el Toolchain Manager y el *programmer* Segger.
2. Poner el *switch* SW5 de la placa de desarrollo en la posición para cargar programas en el chip nRF52 y cargar la configuración de los pines GPIO y de la UART necesaria para el correcto funcionamiento del sistema.
3. Los pines UART (RTS, CTS, TXD y RXD) del dispositivo que hace la función de *manager* deben conectarse a nRF9160-DK según se hayan configurado en el firmware cargado a nRF52.
4. Poner el *switch* SW5 de la placa de desarrollo en la posición para cargar programas en nRF9160 (posición nRF91) y cargar el firmware descrito en el capítulo anterior. Nota: en el archivo de configuración del firmware hay que poner correctamente la configuración de la tarjeta SIM que se esté usando y del *broker* MQTT.
5. Configurar el *backend* con Grafana [28] y un cliente MQTT u observar el flujo de mensajes usando un cliente MQTT suscrito al tópico donde se publican los datos provenientes de nRF9160.
6. Finalmente alimentar el sistema y comprobar que los datos que llegan se corresponden con el formato JSON que se muestra en la Figura 54:

```

Packet format

The format of the sent packet from the Dusty manager to the MQTT endpoint is as follows:

{
  "name": "notifData",
  "fields": {
    "macAddress": "00-17-0d-00-00-XX-XX-XX",
    "data": [byte_1, byte_2, ... ,byte_n]
  },
  "gateway": "uoc-1",
  "timestamp": YYYYYYYY
}

Where:


- byte_1, bytes_2, ... ,byte_n are data generated by a Dusty mote.
- XX-XX-XX are the last 3 bytes from the MAC address of the dusty mote which generated the data.
- YYYYYYYY is the Unix epoch time from when the packet was generated.

```

Figura 54 - Formato de los datos enviados por MQTT

Dado que la placa diseñada aún no se ha producido, se ha realizado la verificación de la solución con la placa de desarrollo. Se ha usado HiveMQ [29] como *broker* MQTT. En los archivos de configuración del *firmware* para nRF9160, se han establecido los parámetros del *broker* (Host broker.hivemq.com, puerto 1883 y tópicos de suscripción y de publicación). Una vez hecho esto se ha cargado el firmware en la placa y se han alimentado los nodos sensores. Estos nodos sensores están configurados para transmitir cada 2 minutos. Para comprobar que la placa se conecta a NB-IoT/CAT-M1, se suscribe a las notificaciones del *manager* Dusty y se conecta correctamente al *broker*, se ha de mirar en la consola. En la Figura 55 se muestra el diálogo de la consola.

```
*** Booting Zephyr OS build v2.4.99-ncs1 ***
[00:00:01.212,280] <inf> app: [DUSTY_Thread] Initiating connection...
[00:00:01.215,667] <inf> app: [DUSTY_Thread] CONNECTED successfully!
[00:00:01.215,759] <inf> app: [DUSTY_Thread] Subscribing to data notifications...
[00:00:01.219,085] <inf> app: [DUSTY_Thread] SUBSCRIBED to data notifications successfully!

[00:00:05.212,219] <inf> app: [MQTT_Thread] The MQTT simple sample started
[00:00:05.212,219] <inf> app: Disabling PSM and eDRX
[00:00:05.228,393] <inf> app: LTE Link Connecting...
+CEREG: 2,"08A6","00C9A015",7,0,0,"11100000","11100000"
+CSCON: 1
+CEREG: 1,"08A6","00C9A015",7...,"11100000","11100000"
[00:00:08.519,317] <inf> app: LTE Link Connected!
[00:00:08.729,400] <inf> app: IPv4 Address found 3.122.77.42
[00:00:08.729,431] <inf> app: [MQTT_Thread] Attempt #0 to connect MQTT
[00:00:09.055,084] <inf> app: [MQTT_Thread] MQTT client connected!
[00:00:09.055,114] <inf> app: [MQTT_Thread] Subscribing to: imanol/nRF9160/sub len 18

[00:00:09.201,110] <inf> app: [MQTT_Thread] SUBACK packet id: 1
```

Figura 55 - Dialogo de la consola que muestra la conexión del manager, LTE y MQTT

Lo primero que confirma la placa es la conexión con el *manager* Dusty, posteriormente se confirma la conectividad NB-IoT/CAT-M1 y finalmente el enlace con el *broker* y la suscripción al *topic* que servirá para realizar cambios de configuración en el manager de forma remota.

Para confirmar que los paquetes que recibe el *manager* son enviados, el *firmware* está programado para avisar cuando llega un paquete desde Dusty y cuando lo envía por MQTT muestra el formato de éste. Esto se puede ver en la Figura 56.

```
Date time: 1630517 333715
[MQTT_Thread] Receive from Dusty Thread: {
  "name": "notifData",
  "fields": {
    "macAddress": "00-17-0d-00-00-33-f8-c0",
    "data": [40,4,80,128,207,3,0,20,49,192,14,252,0,0,64,1,80,130,207,3,0,148,64,192,14,252,0,1,64,1,80,142],
  },
  "gateway": "uoc-1",
  "timestamp": "1025667564"
}

[00:01:08.788,482] <inf> app: [DUSTY_Thread] data notification received from the manager!
[00:01:08.788,574] <inf> app: [DUSTY_Thread] data successfully written in the message queue!

[00:01:08.859,252] <inf> app: [MQTT_Thread] data read from FIFO successfully!
[00:01:08.938,018] <inf> app: [MQTT_Thread] to topic: imanol/nRF9160/pub len: 18

[00:01:09.410,247] <inf> app: [MQTT_Thread] PUBACK packet id: 52697
+CSCON: 0
```

Figura 56 - Confirmación de recibo de información desde Dusty y publicación del mensaje por MQTT

En la figura anterior no se muestra el mensaje al completo ya que el formato con el que se muestra en la consola impide capturar todos los datos que contiene el JSON. Notar, que cuando se publica el mensaje y se recibe el PUBACK conforme este se ha recibido en destino, la consola muestra el PUBACK con el id del paquete cuya llegada se ha confirmado.

El último paso es confirmar que se puede leer ese mensaje desde un cliente MQTT y que su formato corresponde con el visto en la Figura 54. Para esto se va a usar un cliente web. El cliente es HiveMQ Websocket [30]. Después de configurar el cliente con el mismo host y con el puerto 8000 (se trata de un cliente web de modo que no se le puede poner el puerto 1883 o no veríamos nada), se procede a la suscripción en el *topic* donde nRF9160 está enviando los mensajes (imanol/nRF9160/pub). La figura 57 muestra el resultado.

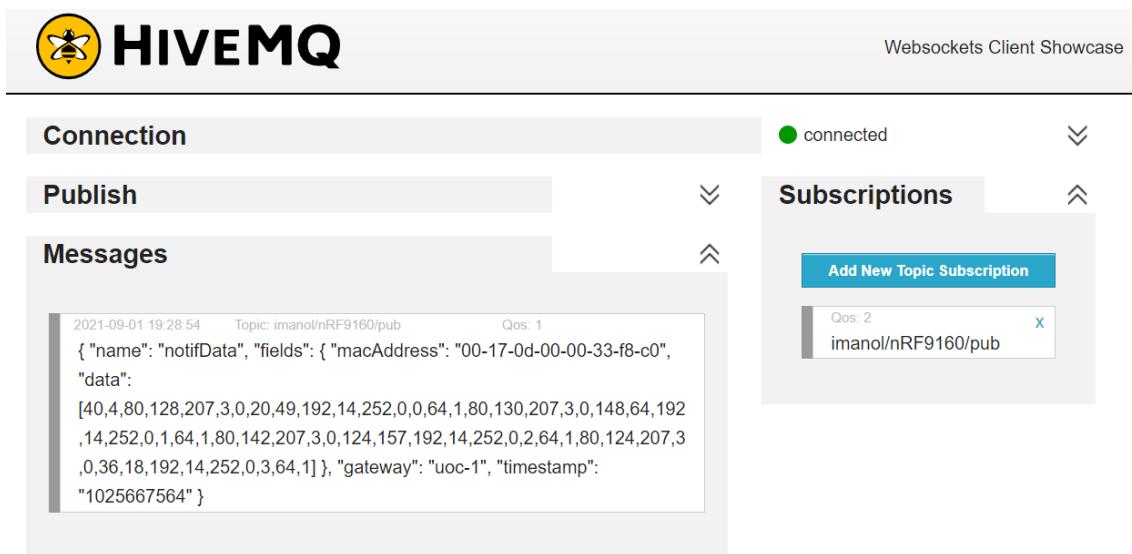


Figura 57 - Resultado de la suscripción al *topic* Imanol/nRF9160/pub en el *websocket* de HiveMQ

Confirmamos que tanto la MAC del nodo que ha enviado el mensaje como el *timestamp* coinciden con el mensaje que se veía en la consola en la Figura 56. Además, vemos que el formato del JSON se corresponde con el mencionado anteriormente.

## 7. Conclusiones y trabajo futuro

Como se ha descrito al inicio del documento, el objetivo de este proyecto es diseñar un *gateway* que permita reenviar los datos provenientes de una red de sensores que utiliza el protocolo WirelessHART a través de una red de celular NB-IoT/CAT-M1 y mediante el protocolo de mensajería MQTT. Tras pasar por las distintas etapas del proceso de diseño y creación de un producto/solución, se ha logrado cumplir con los requerimientos del proyecto.

Aún no ha sido posible producir el prototipo de la placa que se ha diseñado, como se indica a lo largo de esta memoria. Lo último que se ha hecho es el diseño del *layout* de la placa y está todo listo para producirse tan pronto se compren los materiales, estos se reciban y se configuren las máquinas. La imagen siguiente es el *layout* final del sistema que se ha diseñado:

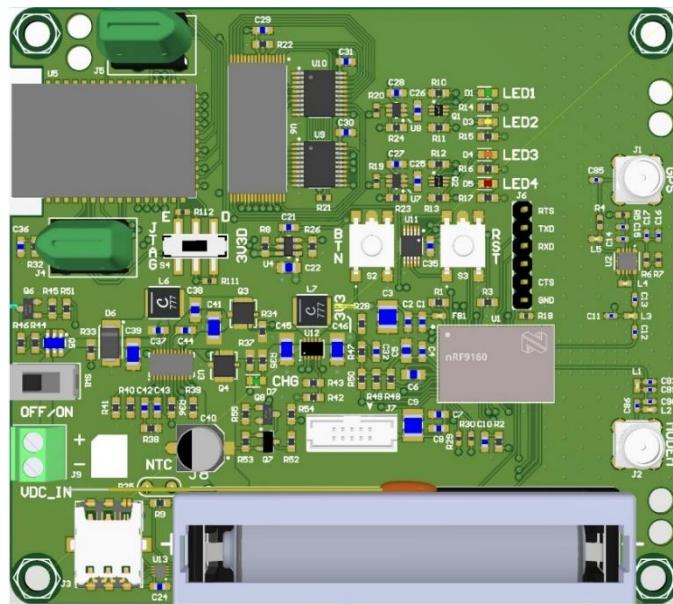


Figura 58 - *Layout* de la solución

Una vez se produzcan los prototipos del sistema (se prevé que sean 5), se validarán utilizando la metodología descrita en esta memoria. Posteriormente se optimizará el firmware y se le añadirán funciones que mejoren el rendimiento del sistema, como por ejemplo diferentes maneras de funcionamiento dependiendo del modo de alimentación (batería de emergencia o fuente de alimentación principal), para que, en caso de estar usando la batería, el sistema consuma menos. Cuando se cargue el firmware a la placa, se realizará la validación de la solución que se ha descrito en el apartado anterior de la memoria<sup>2</sup>. Si no hay ningún error, se procederá a validar el sistema a nivel de consumo energético y a hacer los tests de los distintos modos de fallo de los que se ha hablado anteriormente. Finalmente, se ha de certificar el producto a nivel de radiofrecuencia, certificación UE para Europa y FCC para América. Una vez se haya realizado todo este proceso, se podrá iniciar la producción en masa. Para ello, se ha de prestar atención a los esquemáticos, ya que hay algunos elementos que no se incluyen en la producción final, por ejemplo, el *switch* SW4.

---

<sup>2</sup> La validación de la placa diseñada no necesitará de la preconfiguración de GPIOs y UART que se hace en nRF9160-DK mediante el chip nRF52.

Una vez se haya producido en masa el sistema, FALCO se encargará de realizar el despliegue y colocará un *gateway* por cada 100 nodos WirelessHART. De esta manera, se habrá mejorado el sistema instalado anteriormente y se habrá logrado cumplir con los objetivos del proyecto.

## 8. Bibliografía

### 8.1. Referencias

- [1] Institut National de Recherche en Informatique et en Automatique (INRIA). Web del proyecto SmartMarina. Última consulta: 07/08/2021. Disponible en línea: <https://project.inria.fr/smartmarina/>
- [2] FALCO. Web de la compañía FALCO. Última consulta: 07/08/2021. Disponible en línea: <https://wefalco.com/>
- [3] Internet Engineering Task Force (IETF). RFC8376: Low-Power Wide Area Network (LPWAN) Overview. Última consulta: 07/08/2021. Disponible en línea: <https://datatracker.ietf.org/doc/html/rfc8376>
- [4] Sigfox. Sigfox Technology. Última consulta: 09/08/2021. Disponible en línea: <https://www.sigfox.com/en/what-sigfox/technology#:~:text=The%20Sigfox%20protocol%20stack%20is%20the%20software%20used%20by%20connected,system%20on%20chips%20or%20modules>
- [5] LoRa. About LoRaWAN. Última consulta: 09/08/2021. Disponible en línea: <https://lora-alliance.org/about-lorawan/>
- [6] Institute of Electrical and Electronics Engineers (IEEE). IEEE 802.15.4-2006: IEEE Standard for Information technology-- Local and metropolitan area networks-- Specific requirements-- Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs). Última consulta: 10/08/2021. Disponible en línea: <https://ieeexplore.ieee.org/document/1700009>
- [7] IEC. IEC/PAS 62591. Última consulta: 10/08/2021. Disponible en línea: [https://webstore.iec.ch/p-preview/info\\_iecpas62591%7Bed1.0%7Den.pdf](https://webstore.iec.ch/p-preview/info_iecpas62591%7Bed1.0%7Den.pdf)
- [8] IEC/PAS 62591:2009, Industrial communication networks - Fieldbus specifications - WirelessHART TM communication network and communication profile
- [9] ZigBee Alliance. ZigBee Specification. Última consulta: 07/08/2021. Disponible en línea: <https://zigbeealliance.org/wp-content/uploads/2019/11/docs-05-3474-21-0csg-zigbee-specification.pdf>
- [10] Institute of Electrical and Electronics Engineers (IEEE). IEEE 802.15.1 offsite links page. Última consulta: 09/08/2021. Disponible en línea: <https://ieee802.org/15/Bluetooth/index.html>
- [11] GSMA. NB-IoT deployment guide to basic feature set requirements. Última consulta: 01/09/2021. Disponible en línea: <https://www.gsma.com/iot/wp-content/uploads/2019/07/201906-GSMA-NB-IoT-Deployment-Guide-v3.pdf>
- [12] GSMA. LTE-M deployment guide to basic feature set requirements. Última consulta: 01/09/2021. Disponible en línea: <https://www.gsma.com/iot/wp-content/uploads/2019/08/201906-GSMA-LTE-M-Deployment-Guide-v3.pdf>
- [13] Internet Engineering Task Force (IETF). The Constrained Application Protocol (CoAP). Última consulta: 07/08/2021. Disponible en línea: <https://tools.ietf.org/html/rfc7252>

- [14] OASIS. MQTT Version 3.1.1 Plus Errata 01. Última consulta: 01/09/2021. Disponible en línea: <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html>
- [15] u-blox. SARA-N3 series Multi-band NB-IoT (LTE Cat NB2) modules. Última consulta: 15/03/2021. Disponible en línea: <https://www.u-blox.com/en/product/sara-n3-series>
- [16] u-blox. SARA-R5 series LTE-M / NB-IoT modules with secure cloud. Última consulta: 15/03/2021. Disponible en línea: <https://www.u-blox.com/en/product/sara-r5-series>
- [17] Nordic Semiconductor. Hardware Files “Layout and BOM resources v0.15.0”. Última consulta: 23/02/2021. Disponible en línea: [https://www.nordicsemi.com/-/media/Software-and-other-downloads/Dev-Kits/nRF9160-DK/nRF9160-DK---Hardware-files-0\\_15\\_0.zip](https://www.nordicsemi.com/-/media/Software-and-other-downloads/Dev-Kits/nRF9160-DK/nRF9160-DK---Hardware-files-0_15_0.zip)
- [18] Nordic Semiconductor. Product specification nrf9160 DK. Última consulta: 23/02/2021. Disponible en línea: [https://infocenter.nordicsemi.com/pdf/nRF9160\\_PS\\_v2.0.pdf](https://infocenter.nordicsemi.com/pdf/nRF9160_PS_v2.0.pdf)
- [19] Dust Networks. Eterna LTP5901 / LTP5902 Integration Guide. Última consulta: 23/02/2021. Disponible en línea: <http://www.farnell.com/datasheets/1975853.pdf>
- [20] Dust Networks. Etherena integration guide. Última consulta: 23/02/2021. Disponible en línea: [https://www.analog.com/media/en/technical-documentation/user-guides/Eterna\\_Integration\\_Guide.pdf](https://www.analog.com/media/en/technical-documentation/user-guides/Eterna_Integration_Guide.pdf)
- [21] Midatronics. Dusty User's guide. Última consulta: 23/02/2021. Disponible en línea: <https://midatronics.com/wp-content/uploads/2019/10/MIDATRONICS-ITM-DYPA-or-DYUF-B-02-User-Guide-Dusty-Rev-1.8.pdf>
- [22] Linear Technology. Datasheet LTC4091. Última consulta: 05/08/2021. Disponible en línea: <https://www.analog.com/media/en/technical-documentation/data-sheets/4091fa.pdf>
- [23] Texas Instruments. Datasheet TPS63802. Última consulta: 05/08/2021. Disponible en línea: [https://www.ti.com/lit/ds/symlink/tps63802.pdf?ts=1619171791284&ref\\_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FTPS63802](https://www.ti.com/lit/ds/symlink/tps63802.pdf?ts=1619171791284&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FTPS63802)
- [24] Tadiran Batteries LTD. Datasheet TLI-1550S. Última consulta: 05/08/2021. Disponible en línea: <https://www.tadiranbat.com/assets/tli-1550s-rev.-c-dec-2019.pdf>
- [25] Nordic Semiconductor. Online Power Profiler for LTE. Última consulta 05/08/2021. Disponible en línea <https://devzone.nordicsemi.com/nordic/power/w/opp/3/online-power-profiler-for-lte>
- [26] Analog Devices. SmartMesh Power and Performance Estimator Excel spreadsheet. Última consulta: 09/08/2021. Disponible en línea: [https://www.analog.com/media/en/simulation-models/software-and-simulation/smartmesh\\_power\\_and\\_performance\\_estimator.xls](https://www.analog.com/media/en/simulation-models/software-and-simulation/smartmesh_power_and_performance_estimator.xls)
- [27] Texas Instruments. Datasheet TPS22917. Última consulta: 05/08/2021. Disponible en línea: [https://www.ti.com/lit/ds/symlink/tps22917.pdf?ts=1628171675670&ref\\_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FTPS22917](https://www.ti.com/lit/ds/symlink/tps22917.pdf?ts=1628171675670&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FTPS22917)
- [28] Grafana Labs. Download Grafana. Última consulta: 31/08/2021. Disponible en línea: <https://grafana.com/grafana/download?pg=get&plcmt=selfmanaged-box1-cta1>
- [29] HiveMQ. Public MQTT *broker*. Última consulta: 07/09/2021. Disponible en línea: <https://www.hivemq.com/public-mqtt-broker/>

## 8.2. Figuras

Figura 4 <https://www.researchgate.net/profile/Pietro-Tedeschi/publication/327605635/figure/fig1/AS:670107357437959@1536777452745/Sigfox-Protocol-Stack-compare-with-OSI-model.png>

Figura 5 <https://www.rfwireless-world.com/images/LoRa-protocol-stack.jpg>

Figura 6 <https://www.controlglobal.com/assets/12WPpdf/120904-emerson-wirelesshart-isa.pdf>

Figura 7 <https://iopscience.iop.org/article/10.1088/1757-899X/466/1/012077/pdf>

Figura 8 [https://www.incibe-cert.es/sites/default/files/blog/seguridad-comunicaciones-zigbee/red\\_zigbee.jpg](https://www.incibe-cert.es/sites/default/files/blog/seguridad-comunicaciones-zigbee/red_zigbee.jpg)

Figura 9 [https://rua.ua.es/dspace/bitstream/10045/1109/7/Informe\\_ZigBee.pdf](https://rua.ua.es/dspace/bitstream/10045/1109/7/Informe_ZigBee.pdf)

Figura 10 <https://www.rfwireless-world.com/images/BLE-Protocol-Stack.jpg>

Figura 11 <https://www.rfwireless-world.com/images/BLE-States.jpg>

Figura 12 [https://commons.wikimedia.org/wiki/File:Reutilizacion\\_frecuencia.svg](https://commons.wikimedia.org/wiki/File:Reutilizacion_frecuencia.svg)

Figura 13 <https://www.rfwireless-world.com/images/NB-IoT-Protocol-Stack.jpg>

Figura 14 <https://www.rfwireless-world.com/images/LTE-M-Protocol-Stack.jpg>

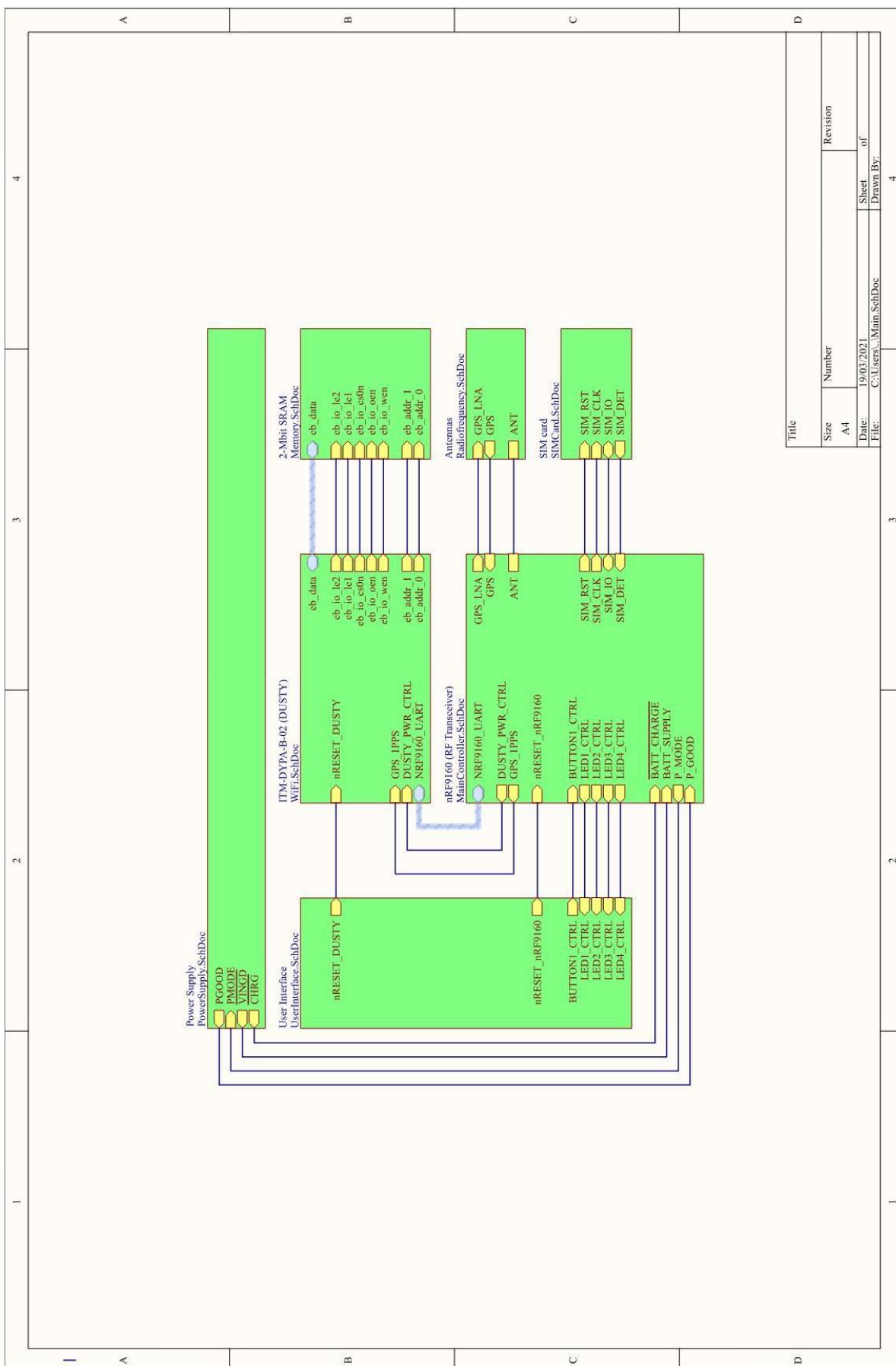
Figura 15 [https://www.pickdata.net/sites/default/files/coap\\_diagrama\\_example.png](https://www.pickdata.net/sites/default/files/coap_diagrama_example.png)

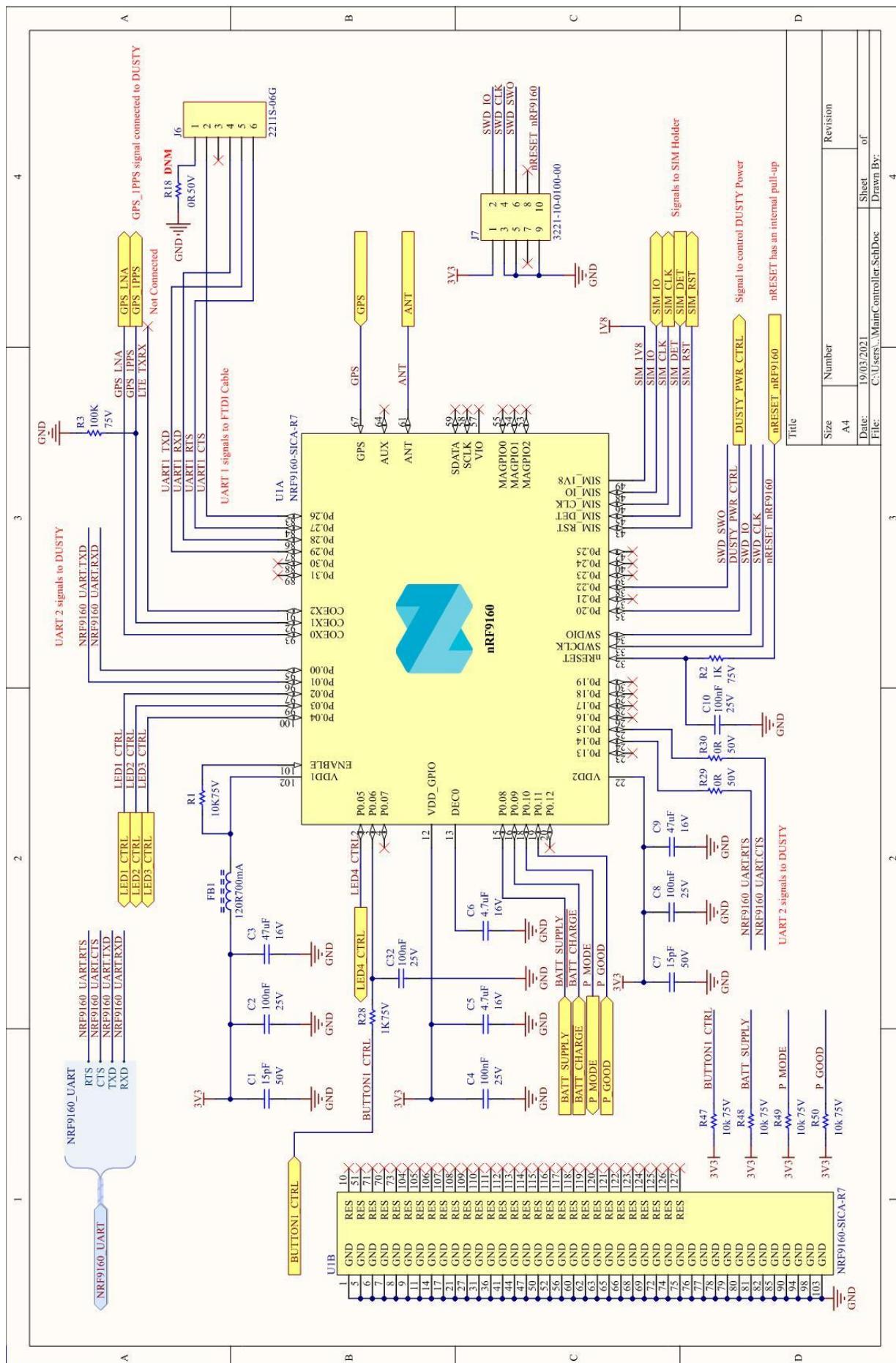
Figura 16 [https://www.pickdata.net/sites/default/files/mqtt\\_diagram\\_example.png](https://www.pickdata.net/sites/default/files/mqtt_diagram_example.png)

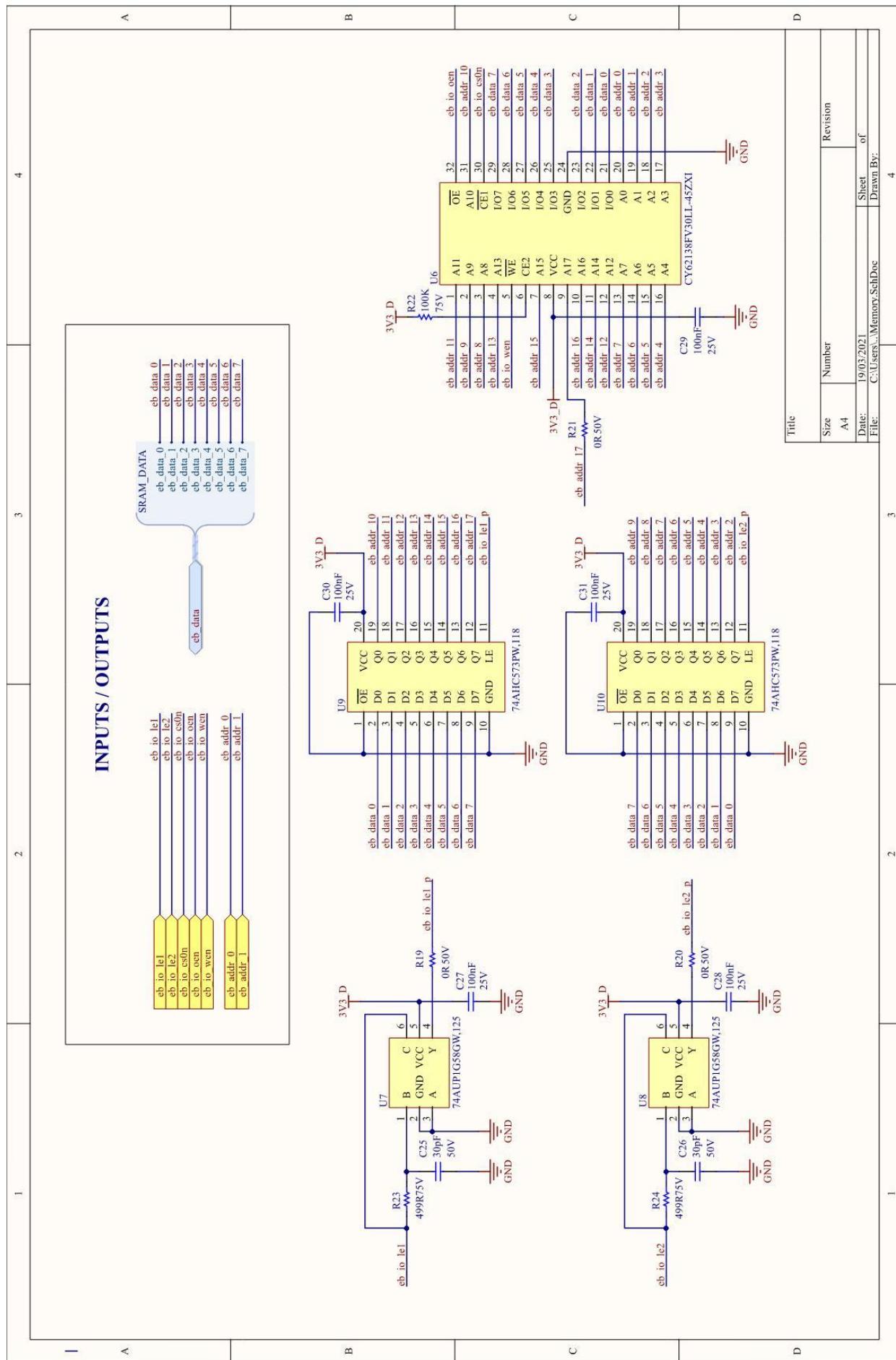
Figura 18

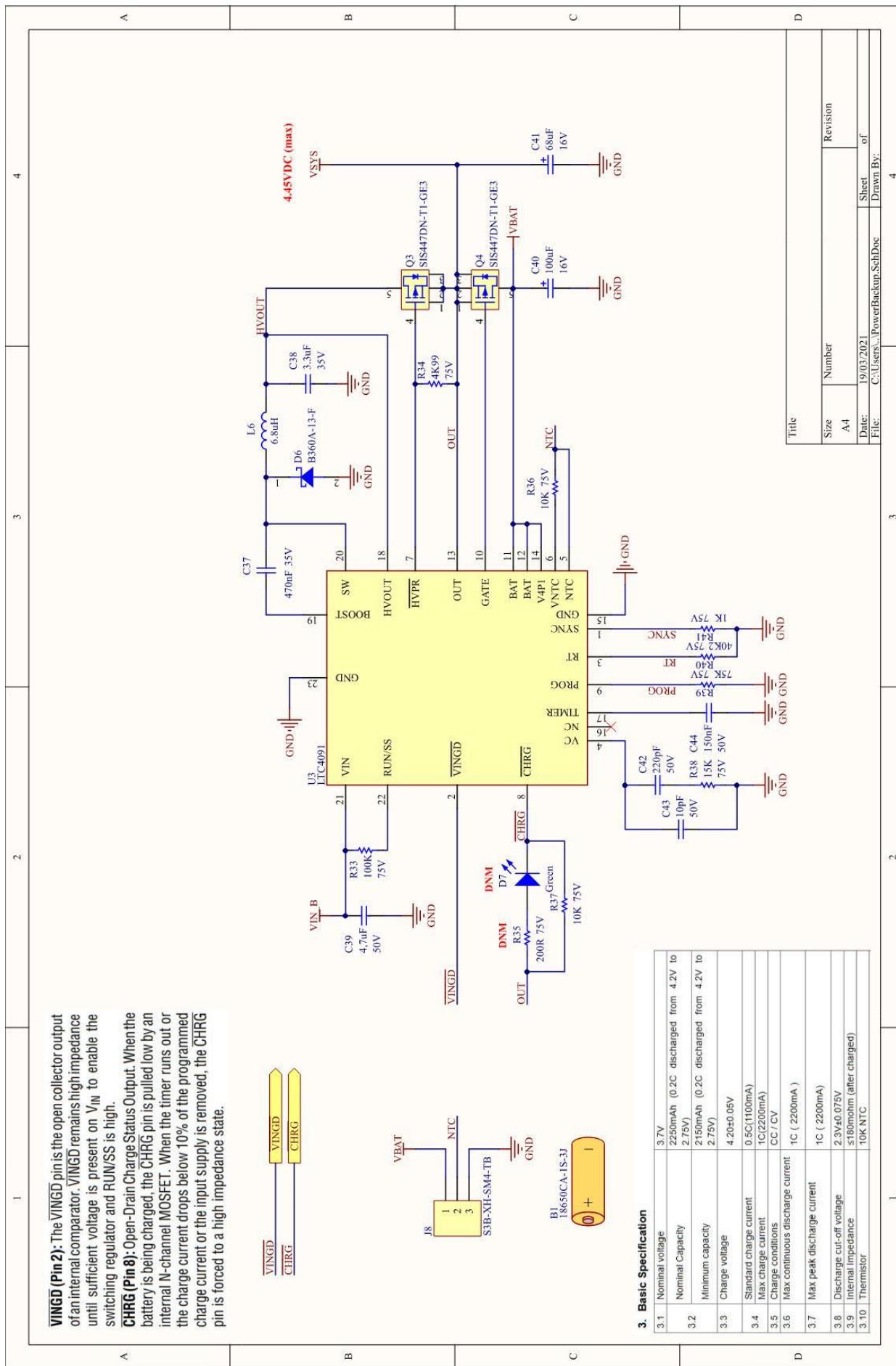
[https://infocenter.nordicsemi.com/topic/ps\\_nrf9160/nRF9160\\_html5\\_keyfeatures.html](https://infocenter.nordicsemi.com/topic/ps_nrf9160/nRF9160_html5_keyfeatures.html)

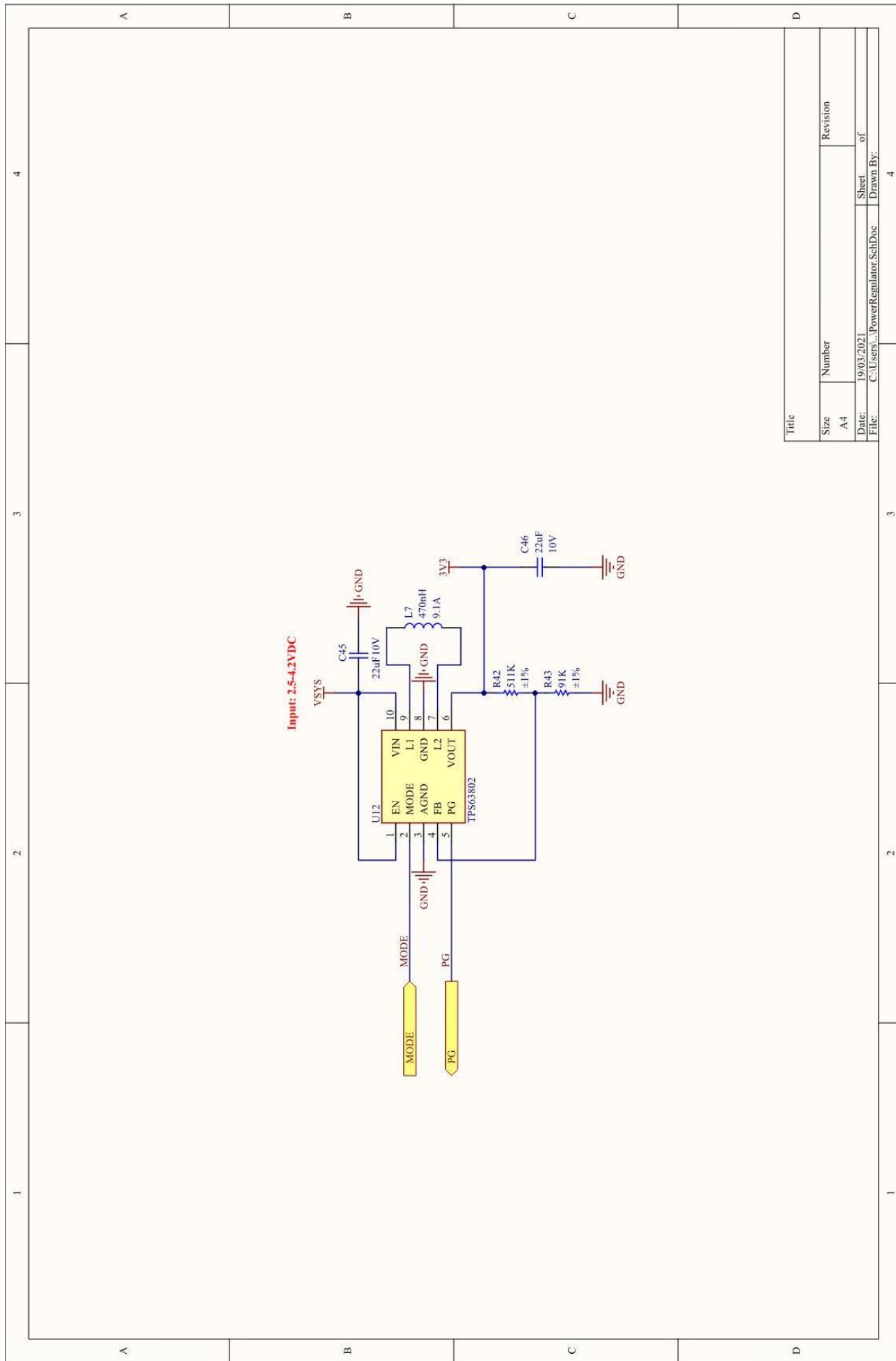
## 9. Anexo I: Esquemáticos

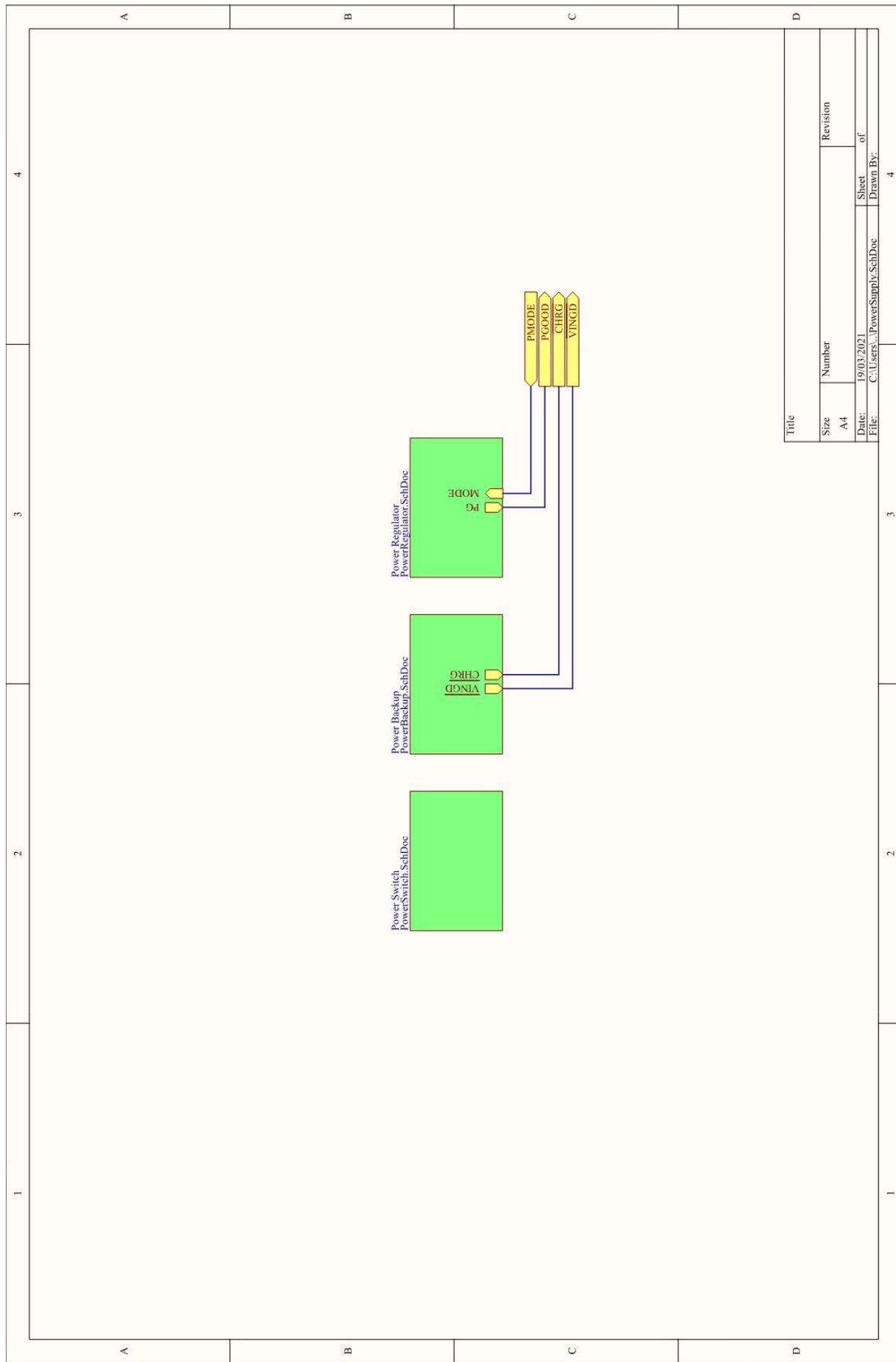


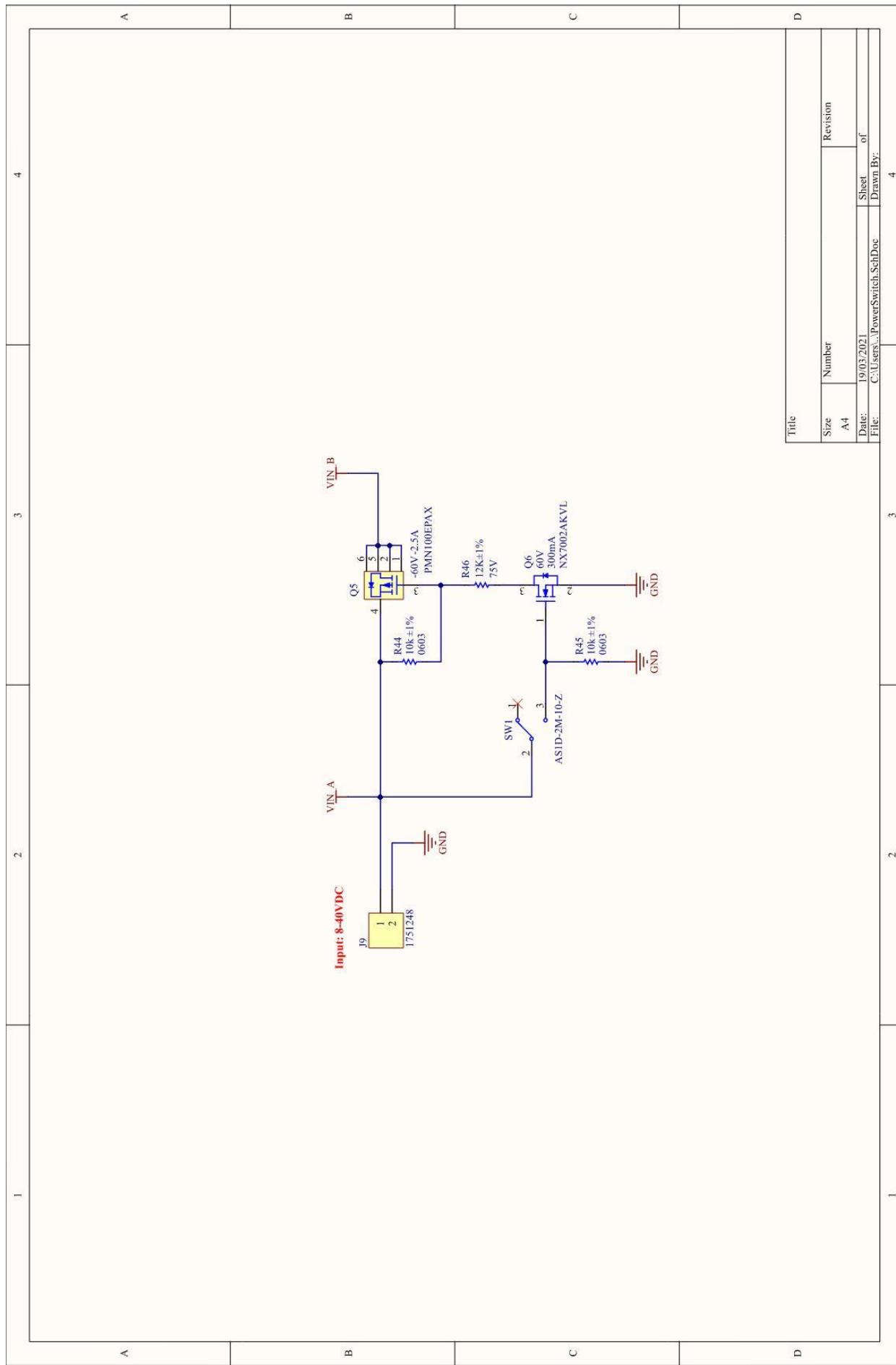


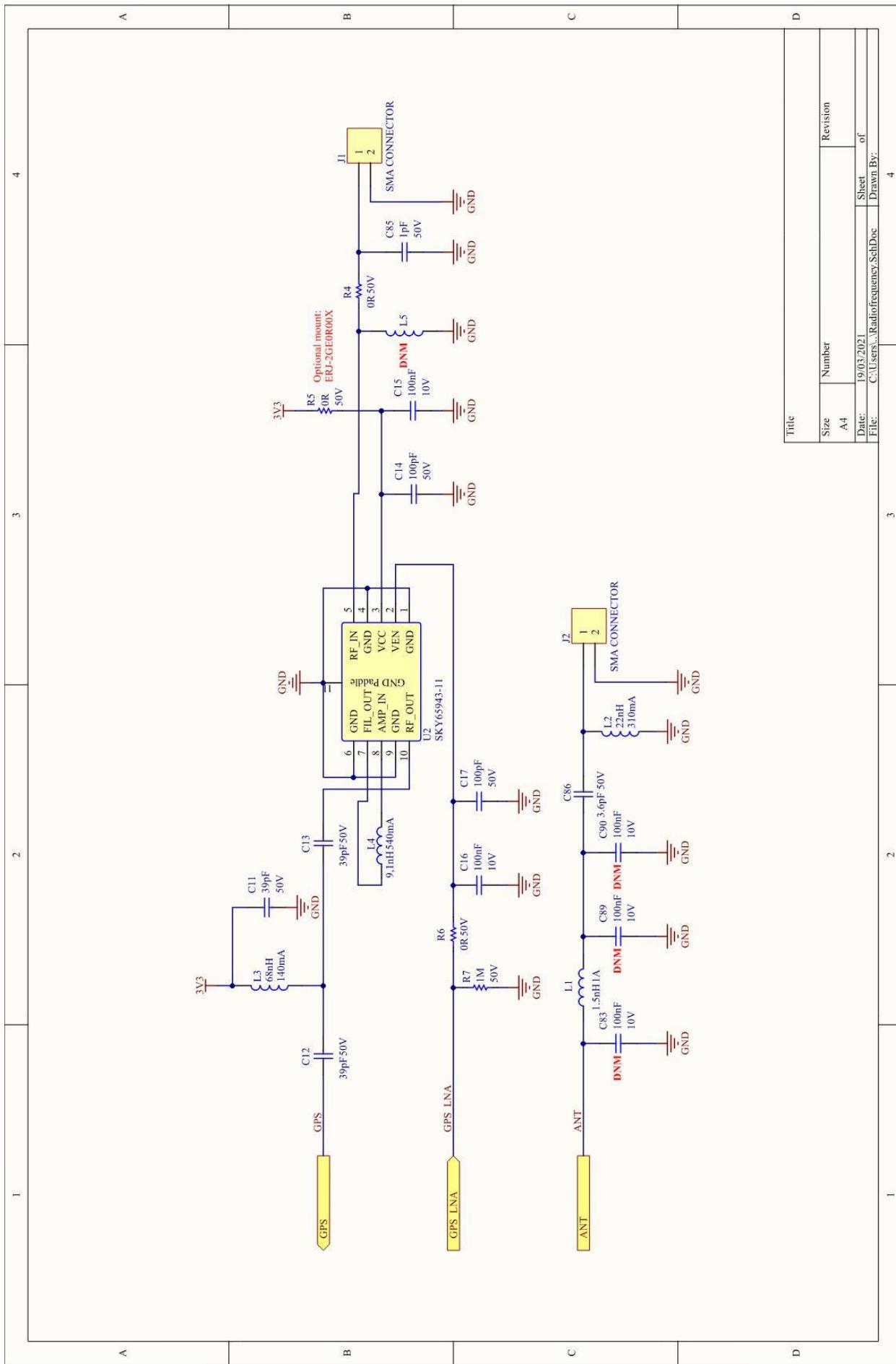


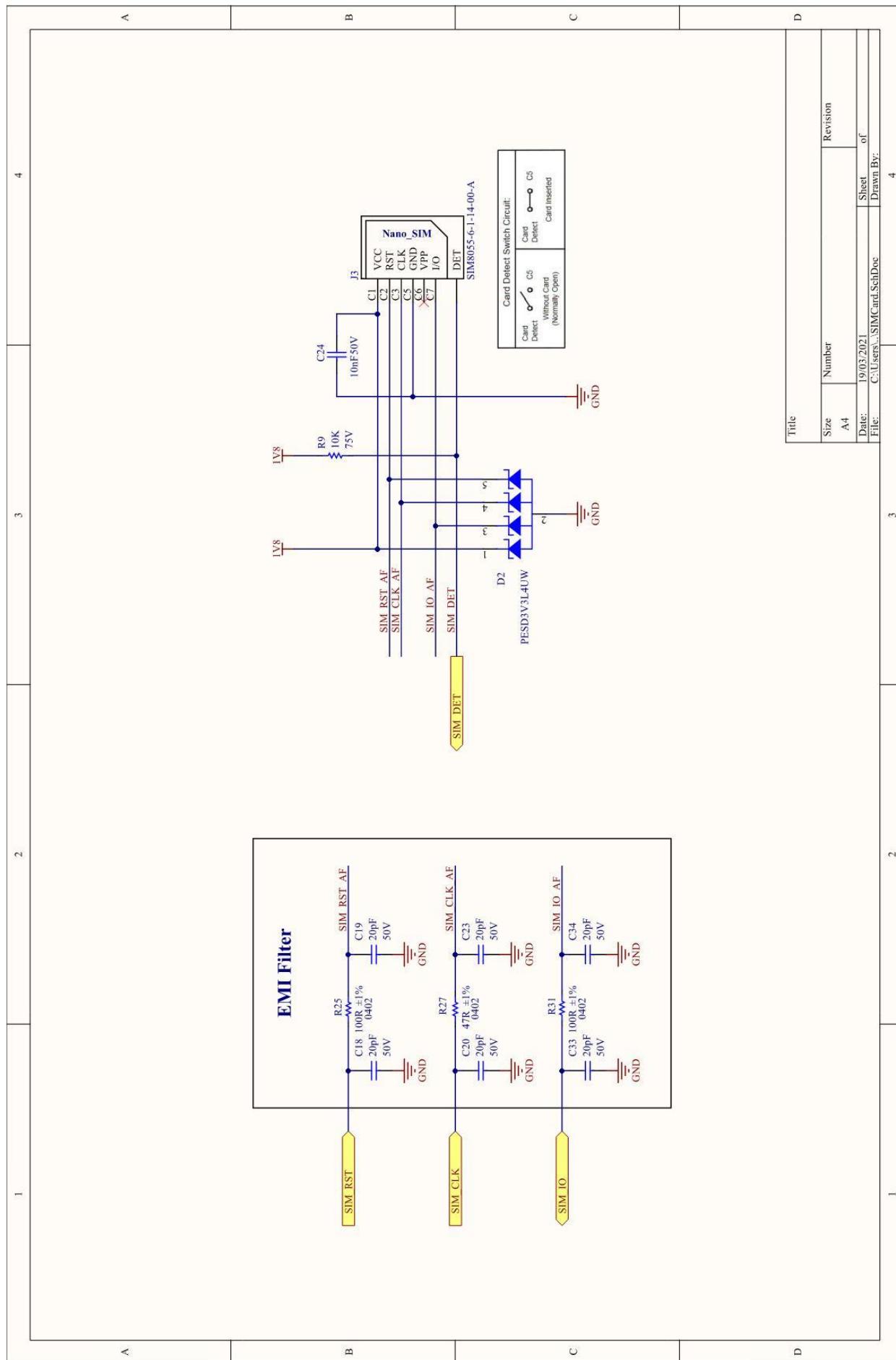


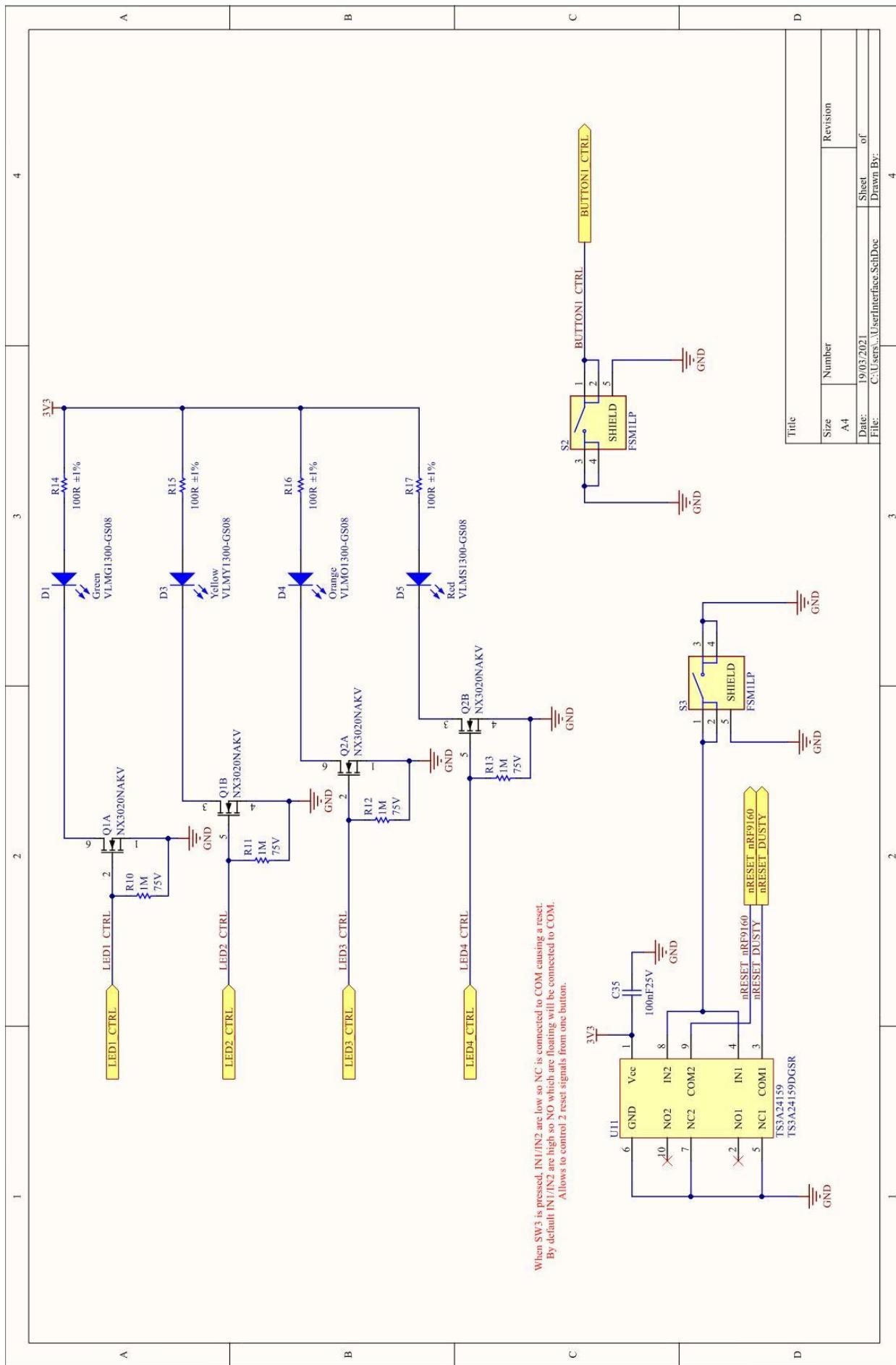


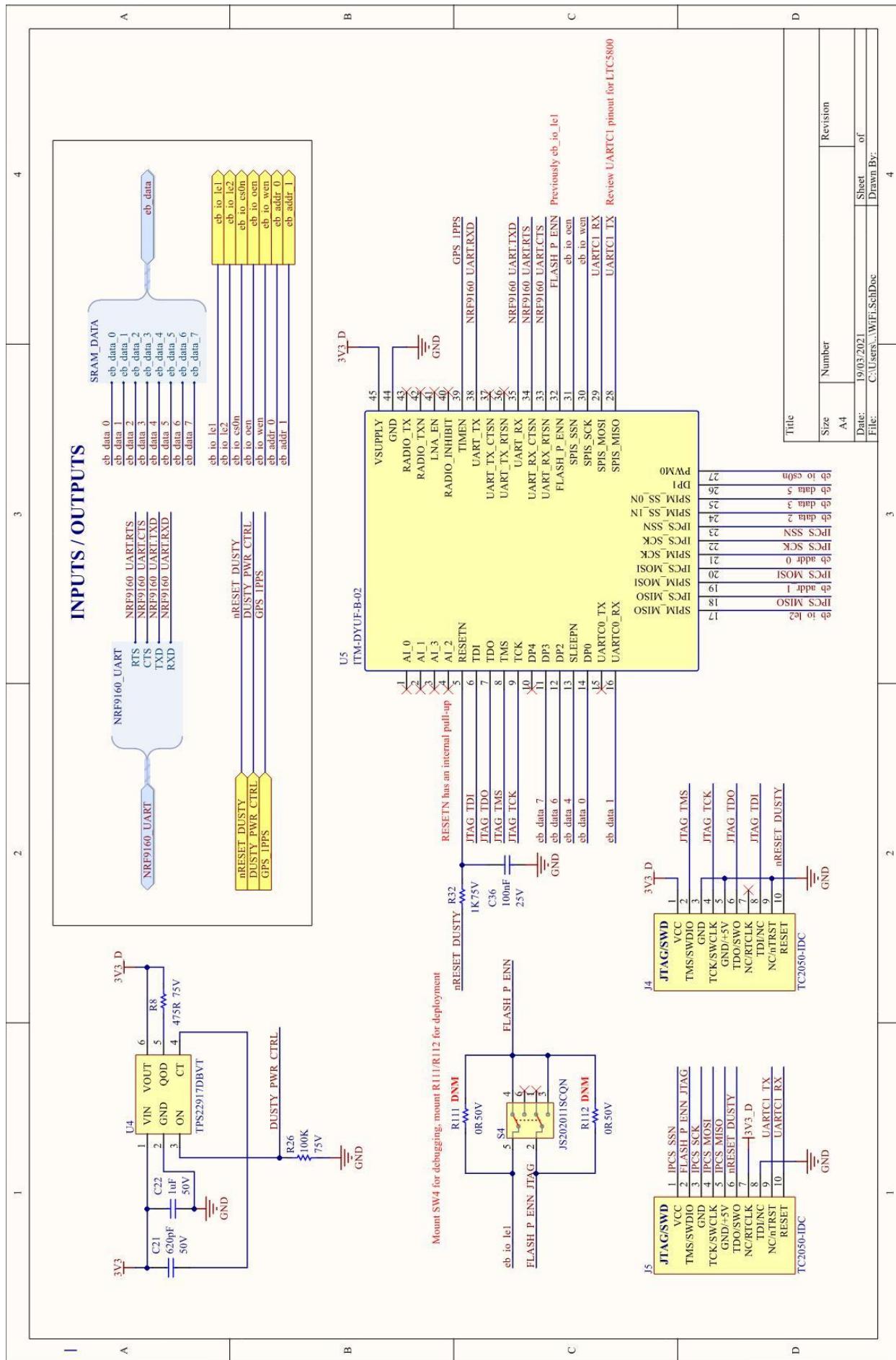












## 10. Anexo II: Bill of Materials (BOM)

Quantity	Designator	Manufacturer_Part_Number	Description	Supplier Part Number 1	Supplier Part Number 2	Supplier Part Number 3	Supplier Part Number 4
1	B1	TLI-1550S	AA Size Lithium Ion Rechargeable Battery				
2	C1, C7	885012006052	15pF, 50V, ±5%, Cer. COG, 0603	2495 134	710-885012006052		732-7794-1-ND
12	C2, C4, C8, C10, C27, C28, C29, C30, C31, C32, C35, C36	885012206071	100nF, 25V, ±10%, Cer. X7R, 0603	2495 173	710-885012206071		732-7989-1-ND
2	C3, C9	GRM32EC81C476KE15L	47uF, 16V, 1210, ±10%, Cer. X6S	1907 527	81-GRM32EC81C476KE5L	846-7369	490-10531-1-ND
2	C5, C6	EMK212B7475KG-T	4.7uF, 50V, 0805, ±10%, Cer. X7R	2112 841	963-EMK212B7475KG-T		587-2394-1-ND
3	C11, C12, C13	0402N390J500CT	39pF, 50V, 0402, ±5%, COG/NP0	2524 781	791-0402N390J500CT		1292-1312-1-ND
2	C14, C17	CC0402JRNPO9BN101	100pF, 50V, 0402, ±5%, COG/NP0	3019 226	603-CC0402JRNPO9BN101		311-1024-1-ND
2	C15, C16	885012205018	100nF, 10V, 0402, ±10%, Cer. X7R	2495 164	710-885012205018	839-1452	732-7513-1-ND
1	C21	C0603C621J5GACTU	620pF, 50V, 0603, ±5%, COG/NP0	2819 768	80-C0603C621J5G		399-3288-1-ND
1	C22	GCM21BR71H105KA03L	1uF/50V/0805/±10% /Cer. X7R	2688 485	81-GCM21BR71H105KA3L		490-10675-1-ND
1	C24	GRM1885C1H103JA01D	10nF, 50V, 0603, ±5%, Cer. COG	2456 107	81-GRM1885C1H103JA1D	815-1430	490-9666-1-ND
2	C25, C26	CC0603JRNPO9BN300	30pF, 50V, 0603, ±5%, COG/NP0	3369 210	603-CC603JRNPO9BN300		311-3357-1-ND
1	C37	CGA3E1X7R1V474K080AC	470nF/35V/0603/±10%/Cer. X7R	2346 964	810-CGA3E1X7R1V474K	915-5679	445-6928-1-ND
1	C38	CGA4J1X7R1V335K125AC	3.3uF, 35V, 0805, ±10%, Cer. X7R	3490 076	810-CGA4J1X7R1V335KC		445-175569-1-ND
1	C39	GRM31CR71H475KA12L	4.7uF, 50V, ±10%, Cer. X7R, 1206	1735 545	81-GRM31CR71H475KA2L		490-6521-1-ND
1	C40	EEEFK1C101P	100uF/16V/D/±20%/Elect. Aluminium	9695 672	667-EEE-FK1C101P	537-0433	PCE3783CT-ND

1	C41	C3216X5R1A686M 160AC	68uF, 10V, ±20%, Cer. X5R, 1206		810- C3216X5R1A68 6M1C	916- 2908	445-14673- 1-ND
1	C42	MC0603B221K500 CT	220pF, 50V, ±10%, Cer. X7R, 0603	1759 070			
1	C43	CC0603JRNPO9BN 100	10pF/50V/0603/±10 %/Cer. COG/NPO	7219 67	603- C0603JRNPO9B N100	616- 9054	311-1058-1- ND
1	C44	C0603C154K5RACT U	150nF, 50V, 0603, ±10%, Cer. X7R	1414 619	80- C0603C154K5R		399-9015-1- ND
2	C45, C46	C1206C226K8RACT U	22uF, 10V, 1206, ±10%, Cer. X7R	2473 548	80- C1206C226K8R	874- 2424	399-11947- 2-ND
1	C85	04025A1R0BAT2A	1pF, 50V, 0402, ±0.1pF, C0G/NP0	2332 496	581- 04025A1R0BAT 2A		478-3659-1- ND
1	C86	GJM1555C1H3R6C B01D	3.6pF, 50V, 0402, ±0.25pF, C0G/NP0	3243 235	81- GJM1555C1H3R 6CB1D		490-3096-1- ND
2	D1, D7	VLMG1300-GS08	2V, 20mA, Green, 35 mcd, 0603	2251 461	78-VLMG1300- GS08		VLMG1300- GS08CT-ND
1	D3	VLMY1300-GS08	2V, 20mA, Yellow, 180 mcd, 0603	2251 505	78-VLMY1300- GS08		VLMY1300- GS08CT-ND
1	D4	VLMO1300-GS08	2V, 20mA, Orange, 90 mcd, 0603	2251 473	78-VLMO1300- GS08		VLMO1300- GS08CT-ND
1	D5	VLMS1300-GS08	2V, 20mA, Red, 54 mcd, 0603	2251 484	78-VLMS1300- GS08		VLMS1300- GS08CT-ND
1	D6	B360A-13-F	Schottky, 60V, 3A, 0.7V, DO-214AC (SMA)	1858 600	621-B360A-F	738- 4796	B360A- FDICT-ND
1	FB1	BLM15AX121SN1D	FB 120R @ 100MHz, 700mA, 0402	2470 352	81- BLM15AX121S N1D		490-5439-1- ND
1	HW5	1028C	AA Battery cover (ref. 1028C)	2027 288	534-1028C		36-1028C- ND
2	J1, J2	SMA CONNECTOR	RF Connector, SMA, Vertical	2096 223	471- SMACONNECT OR		
1	J3	SIM8055-6-1-14- 00-A	Nano SIM Card Connector, Push-Pull Type, with Switch 6 Pin, SMT, Profile 1.35mm	2924 868			2073- SIM8055-6- 1-14-00- ACT-ND
1	J6	2211S-06G	1x6pins/2,54mm/0- Vertical	1593 415			
1	J7	3221-10-0100-00	IDC BOX HEADER, 1.27MM, 10 Pos, Straight				1175-1733- ND
1	J8	1028	AA battery holder (ref. 1028)	1650 669	534-1028	172- 4325	36-1028-ND
1	J9	1751248	2x1/90- Horizontal/3,5mm		651-1751248	802- 4849	277-5719-ND
1	L1	MLG1005S1N5BT0 00	1.5nH, ±0.1nH, 1A, 100m, 0402	3386 606	810- MLG1005S1N5 BT000		445-16264- 1-ND
1	L2	LQW15AN22NH00 D	22nH, 310mA, ±3%, 0R3, 0402	2840 152	81- LQW15AN22N H00D		490-6788-1- ND
1	L3	LQW15AN68NG00 D	68nH, 140mA, ±5%, 1R96, 0402	1762 629	81- LQW15AN68N G00D	792- 6463	490-6837-1- ND
1	L4	LQW15AN9N1G00 D	9.1nH, ±2%, 540mA, 140m, 0402	1762 616	81- LQW15AN9N1 G00D	106- 599	490-6854-1- ND

1	L6	XAL4030-682MEC	6-8uH, ±20%, 3.6A, 74.1m, Other SMD	2289 057	994-XAL4030-682MEC		
1	L7	XFL4015-471MEC	470nH, ±20%, 9.1A, 8.36m, Other SMD	2289 211	994-XFL4015-471MEC		
2	Q1, Q2	NX3020NAKVYL	MOSFET Canal N, Dual Channel, 30V, 200mA, SOT-666	3439 882	771-NX3020NAKVYL		NX3020NAKVYL-ND
2	Q3, Q4	SIS447DN-T1-GE3	MOSFET Canal P, -20V, -18A, PowerPAK 1212-8N		78-SIS447DN-T1-GE3		SIS447DN-T1-GE3CT-ND
1	Q5	PMN100EPAX	MOSFET Canal P, -60V, -2.5A, SOT-457-6	3438 535	771-PMN100EPAX		1727-PMN100EPAXCT-ND
2	Q6, Q8	NX7002AKVL	MOSFET Canal N, 60V, 300mA, TO-236AB (SOT23)	3439 885	771-NX7002AKVL		
1	Q7	NX3008PBKW	MOSFET Canal P/-30V/-200mA/SOT23	2069 551	771-NX3008PBKW,1 15	816-0582	568-10503-1-ND
13	R1, R9, R36, R37, R44, R45, R47, R48, R49, R50, R53, R54, R55	CRCW060310K0FK EA	10k, 100mW, ±1%, 75V, 0603	1469 748	71-CRCW0603-10K-E3	678-9667	541-10.0KHCT-ND
4	R2, R28, R32, R41	CRCW06031K00FK EA	1K, ±1%, 100mW, 0603	1469 740	71-CRCW0603-1.0K-E3	678-9875	541-3949-1-ND
4	R3, R22, R26, R33	CRCW0603100KFK EA	100K, ±1%, 100mW, 0603	1469 649	71-CRCW0603-100K-E3	820-6745	541-100KHCT-ND
1	R7	MCS04020C1004FE 000	1M, ±1%, 100mW, 0402	2614 408	594-MCS04020C1004FE0		MCS0402-1.00M-CFCT-ND
1	R8	CRCW0603475RFK EA	475R, ±1%, 100mW, 0603	1469 816	71-CRCW0603-475-E3		541-5439-1-ND
4	R10, R11, R12, R13	CRCW06031M00FK EA	1M, ±1%, 100mW, 0603	1469 746	71-CRCW0603-1.0M-E3		541-1.00MHCT-ND
4	R14, R15, R16, R17	CR0603-FX-1000ELF	100R/±1%/125mW/0603	2008 332	CR0603-FX-1000ELF	740-8798	CR0603-FX-1000ELFCT-ND
2	R23, R24	ERJ3EKF4990V	499R/±1%/100mW/75V/0603	2059 311	667-ERJ-3EKF4990V	808-5236	P499HCT-ND
1	R25	NTCLE300E3103SB	10K, ±2%, 100mW, Other	1888 081	594-NTCLE300E3103SB	819-9186	BC2462-ND
1	R34	CRCW06034K99FK EA	4K99, ±1%, 100mW, 0603	1652 885	71-CRCW0603-4.99K-E3		541-4.99KHCT-ND
1	R35	CRCW0603200RFK EA	200R, 100mW, ±1%, 75V, 0603	1469 777	71-CRCW0603-200-E3	678-9960	541-200HCT-ND
1	R38	CRCW060315K0FK EA	15k, 100mW, ±1%, 75V, 0603	1469 758	71-CRCW0603-15K-E3	678-9793	

1	R39	MCT06030C5603FP 500	560K, ±1%, 100mW, 0603	3546 635	594- MCT06030C560 3FP5		749-1753-1- ND
1	R40	CRCW060340K2FK EA	40K2, ±1%, 100mW, 0603	1692 527	71-CRCW0603- 40.2K-E3		541- 40.2KHCT- ND
1	R42	ERJ3EKF5113V	511K, ±1%, 100mW, 0603	2059 521	667-ERJ- 3EKF5113V		P511KHCT- ND
1	R43	CR0603-FX- 9102ELF	91K, ±1%, 100mW, 0603	2908 345	652-CR0603FX- 9102ELF		CR0603-FX- 9102ELFTR- ND
2	R46, R51	CRCW060312K0FK EA	12K, 100mW, ±1%, 75V, 0603	1652 834	71-CRCW0603- 12K-E3	678- 9724	541- 12.0KHCT- ND
1	R52	CRCW060327K4FK EA	27K4, ±1%, 100mW, 0603	1652 865	71-CRCW0603- 27.4K-E3		541- 27.4KHCT- ND
2	S2, S3	FSM1LP	Push Button, 24V, 50mA, SMD	3397 775	506-FSM1LP	756- 1353	450-1132-ND
1	S4	JS202011SCQN	Manual Switch, 6V, 300mA, 1 SW, DPDT	2320 019	611- JS202011SCQN	154- 6137	401-2002-1- ND
1	SW1	AS1D-2M-10-Z	Manual Switch, 60V, 50mA, SPDT-2P	2854 819	229-AS1D-2M- 10-Z		563-1579-ND
1	U1	NRF9160-SICA-R7	Low power cellular IoT solution	3289 981	949-NRF9160- SICA-R7	196- 3896	1490-1076- 1-ND
1	U2	SKY65943-11	GNSS Low-Noise Amplifier Front-End Module with Integrated Pre-Filter and Post-Filter		873-SKY65943- 11		863-2032-1- ND
1	U3	LTC4091EDJC#PBF	36V Battery Charger and Power Backup Manager		584- LTC4091EDJCPB F		LTC4091EDJC #PBF-ND
1	U4	TPS22917DBVT	2-A, 80-mOhms, Ultra-Low Leakage Load Switch	3124 067	595- TPS22917DBVT		296-48370- 1-ND
1	U5	ITM-DYUF-B-02	SmartMesh IP PCBA product incorporating the LTC5800-IPM SoC running Dust's embedded SmartMesh IP networking software.				
1	U6	CY62138FV30LL- 45ZXI	2Mbit, Static RAM, TSOP I	2767 910	727- CY138FV30LL45 ZXI	194- 8866	CY62138FV3 0LL-45ZXI- ND
2	U7, U8	74AUP1G58GW,12 5	Configurable multi function gate, SOT- 363-6	2164 684	771- AUP1G58GW12 5		1727-3194- 1-ND
2	U9, U10	74AHC573PW,118	Octal D-type transparent latch, TSSOP-20	2445 112	771- AHC573PW118		1727-5925- 1-ND
1	U11	TS3A24159DGSR	Analog Switch, 3.6V, 300mA, VSSOP-10	1755 571	595- TS3A24159DGS R		296-21913- 6-ND
1	U12	TPS63802DLAT	1.3V-5.5V, Switching Regulator, VSON- HR-10		595- TPS63802DLAT		296- TPS63802DL ATCT-ND
1	U13	EMIF03-SIM02M8	3 line IPAD, EMI filter for SIM card applications	3130 087	511-EMIF03- SIM02M8		497-14751- 1-ND