



Universidad de Valladolid

E.T.S.I. Telecomunicación

Trabajo Fin de Grado

Grado en Ingeniería de Tecnologías Específicas de
Telecomunicación: Mención en Sistemas Electrónicos

Desarrollo e industrialización de un dispositivo Bluetooth de sensorización de bajo consumo

Autor:

José Luis Gay Domenech

Tutores:

Pedro López Martín

Diego R. Llanos Ferraris

11 de Diciembre de 2019



Título: **Desarrollo e industrialización de un dispositivo bluetooth de sensorización de bajo consumo**

Autor: **D. José Luis Gay Domenech**

Tutor: **D. Pedro López Martín**

Departamento: **Departamento de Electrónica**

Tribunal

Presidente: **Dña. M^a Lourdes Pelaz Montes**

Secretario: **D. Pedro López Martín**

Vocal: **D. Jesús M. Hernández-Mangas**

Suplente: **D. José Emiliano Rubio**

Suplente: **D. Luis A. Marqués Cuesta**

Fecha: **20 Diciembre 2019**

Calificación:

Resumen de TFG

Este trabajo parte de un equipo capaz de medir en contenedores de reciclaje de envases los siguientes parámetros:

Llenado, temperatura, humedad y número de usos.

Las medidas realizadas se envían mediante Bluetooth a los teléfonos móviles de los participantes de la iniciativa Sterling, que reciben premios por reciclar así como por hacer de intermediarios del envío de datos.

Palabras Clave

Arduino, ATmega328p, Nordic Semiconductors, Sensores, Bluetooth, Bluetooth de baja energía, IOT, Reciclaje.

Abstract

This work starts from a prototype able to measure in recycling dumpster the following parameters: Load, temperatura, humidity, use count.

The measures are then sent via bluetooth to surrounding mobile phones from the Sterling initiative partakers, who are awarded with prizes for recycling as well as acting as intermediaries for the data transmission.

Keywords

Arduino, ATmega328p, Nordic Semiconductors, Sensors, Bluetooth, Low energy Bluetooth, IOT, Recycling.

Índice general

CAPÍTULO 1	1
INTRODUCCIÓN	1
1.1 MOTIVACIÓN	1
1.2 ESTRUCTURA.....	1
1.3 OBJETIVOS.....	2
1.3.1 <i>Requisitos de diseño</i>	2
1.4 SOFTWARE	3
1.4.1 <i>Eagle CAD</i>	3
1.4.2 <i>Arduino IDE</i>	3
1.4.3 <i>Visual Studio Code</i>	3
1.4.4 <i>Keil MicroVision</i>	3
1.5 PLANIFICACIÓN.....	3
CAPÍTULO 2	5
BLUETOOTH LOW ENERGY (BLE)	5
2.1 INTRODUCCIÓN.....	5
2.2 BLUETOOTH DE BAJA ENERGÍA	5
2.3 DIVISIONES DE FRECUENCIA.....	6
2.4 CONFIGURACIONES.....	6
2.4.1 <i>Según qué especificación soporten</i>	6
2.4.2 <i>Según el número de chips</i>	7
2.4.3 <i>Topología</i>	8
2.5 GAP (GENERIC ACCESS PROFILE).....	9
2.5.1 <i>Roles</i>	9
2.5.2 <i>Modos</i>	9
2.5.3 <i>Procedimientos</i>	10
2.5.4 <i>Paquetes de difusión</i>	10
2.5.5 <i>Seguridad</i>	12
2.6 LIMITACIONES	12
CAPÍTULO 3	13
FASE ATMEGA: MARK I A MARK III	13
3.1 INTRODUCCIÓN.....	13
3.2 PUNTO DE PARTIDA: STERLING MARK I.....	13
3.2.1 <i>Componentes iniciales</i>	13
3.2.2 <i>Placa inicial</i>	14
3.2.3 <i>Bill Of Materials (Listado de componentes) simplificado</i>	16
3.2.4 <i>Esquemático</i>	17
3.2.5 <i>Diseño PCB</i>	18
3.2.6 <i>Problemas presentados</i>	19
3.3 STERLING MARK II	20
3.3.1 <i>Cambios realizados</i>	20
3.3.2 <i>Consumos</i>	20
3.3.3 <i>Imágenes</i>	21

3.3.4	Problemas presentados	22
3.4	STERLING MARK III	23
3.4.1	Introducción.....	23
3.4.2	Cambios realizados.....	23
3.4.3	Consumos	24
3.4.5	Problemas presentados	25
3.4.6	Sterling Mark III ex.....	26
3.4.7	Bill Of Materials simplificado.....	27
3.4.8	Esquemáticos, diseño PCB	28
CAPÍTULO 4		31
PROBLEMAS ENCONTRADOS Y SOLUCIONES ASOCIADAS		31
4.1	ALTO CONSUMO	31
4.2	SENSOR DE ULTRASONIDOS DE 5V	32
4.3	CONDENSADOR DE 10 μ F.....	32
4.4	PINES DE INTERRUPCIÓN.....	33
4.5	CAÍDA DE VOLTAJE.....	33
4.6	ALIMENTACIÓN LIMITADA.....	33
4.7	VÍAS DE ALIMENTACIÓN ESTRECHAS	34
4.8	PUERTO ICSP.....	34
4.9	PLANO DE MASA BAJO LA ANTENA BLUETOOTH	34
4.10	CALIDAD DE FABRICACIÓN EN LOS NRF24L01	34
4.11	SOBREVOLTAJE EN MÓDULOS	35
4.12	ATMEGA328 VS ATMEGA328P.....	36
4.13	TRANSISTOR SIN RESISTENCIA EN LA PUERTA	36
4.14	MÓDULOS NO BLE.....	36
4.15	CANALES DE EMISIÓN.....	37
4.16	IMPERMEABILIZACIÓN DEL SISTEMA	38
4.17	MEDIDAS Y MARCO, DISEÑO IRREGULAR	39
CAPÍTULO 5		41
TESTEO Y PUESTA EN PRODUCCIÓN		41
5.1	PUNTO DE PARTIDA	41
5.2	GRABADO DE PLACA	42
5.1	PREPARADO DE LOS COMPONENTES.....	44
5.1.1	Caja.....	44
5.1.2	Tapa.....	44
5.1	TESTEO DE LA PLACA	46
5.2	ERRORES COMUNES.....	48
5.2.1	<i>JSN-SR04: ERROR: Too close or transducer not connected</i>	48
5.2.2	<i>DHT11: ERROR. Can't get data.</i>	48
5.2.3	<i>Los dos anteriores a la vez</i>	48
5.2.4	<i>Sólo aparece un punto como paquete enviado y se cuelga</i>	48
5.2.5	<i>Contador de aperturas siempre a 1 (Integrado se reinicia)</i>	48
5.2.6	<i>El monitor serie muestra texto incoherente</i>	48
5.3	PUESTA EN PRODUCCIÓN	49
5.4	MANTENIMIENTO	49
CAPÍTULO 6		51
FASE NORDIC: PROTOTIPO MARK IV		51
6.1	INTRODUCCIÓN.....	51
6.2	ELECCIÓN DE COMPONENTES.....	51
6.3	DISEÑO DE PCB.....	52
6.3.1	Alimentación.....	52
6.3.2	Adaptación de impedancias.....	52

6.3.3	<i>Conexión a la antena</i>	52
6.3.4	<i>Osciladores</i>	53
6.3.5	<i>Antena</i>	53
6.3.6	<i>Plano de tierra</i>	54
6.4	BILL OF MATERIALS SIMPLIFICADO	55
6.5	ESQUEMÁTICOS, DISEÑO PCB	56
CAPÍTULO 7		59
CONCLUSIONES		59
7.1	RESULTADOS	59
7.2	PRESUPUESTO	59
7.3	CONCLUSIONES	60
7.4	TRABAJO FUTURO	61
CAPÍTULO 8		63
BIBLIOGRAFÍA		63

Índice de Ilustraciones

Ilustración 1. Canales BLE.....	6
Ilustración 2. Especificaciones Bluetooth y sus capas correspondientes	7
Ilustración 3. Posibles implementaciones Hardware	8
Ilustración 4. Topologías Bluetooth.....	9
Ilustración 5. Intervalos de ventana de escucha BLE.....	10
Ilustración 6. Formato de una trama Bluetooth.....	11
Ilustración 7. Sterling Mark I.....	15
Ilustración 8. Esquemático Sterling Mark I	17
Ilustración 9. Diseño PCB Sterling Mark I	18
Ilustración 10. Parte trasera del Sterling Mark II.....	21
Ilustración 11. Parte delantera del Sterling Mark II.....	21
Ilustración 12. Sterling Mark III.....	24
Ilustración 13. Sterling Mark III ex.....	26
Ilustración 14. Página 1 del esquemático de Sterling Mark III	28
Ilustración 15. Página 2 del esquemático de Sterling Mk.III	29
Ilustración 16. Diseño de PCB Sterling Mk.III.....	30
Ilustración 17. Resultados bajo 5V de HC-SR04.....	32
Ilustración 18. Resultados bajo 3.3V de HC-SR04.....	32
Ilustración 19. Detalle del condensador añadido en el nRF24L01.....	35
Ilustración 20. Módulo JSN-SR04T	38
Ilustración 21. Caja de conexiones ABB 00846.....	41
Ilustración 22. Programador ICSP USBasp.....	42
Ilustración 23. Conexión ICSP	42
Ilustración 24. Configuración IDE Arduino.....	43
Ilustración 25. Quemado de Bootloader.....	43
Ilustración 26. Programación del integrado	43
Ilustración 27. pasante para el cable del interruptor Reed	44
Ilustración 28. Taladrado para el módulo de ultrasonidos	44
Ilustración 29. Tapa con el módulo de ultrasonidos fijado	45
Ilustración 30. Tapa montada completamente	45
Ilustración 31. Interfaz web mostrando último uso de cada contenedor	49
Ilustración 32. Fallo por DHT11	49
Ilustración 33. Fallo por FDN304P	49
Ilustración 34. nRF51822.....	51
Ilustración 35. Indicaciones sobre la colocación de la antena en su datasheet	54
Ilustración 36. Hoja 1 del esquemático Sterling Mark IV	56
Ilustración 37. Hoja 2 del esquemático del Sterling Mark IV.....	57
Ilustración 38. Diseño PCB Sterling Mark IV	58

Índice de Tablas

Tabla 1. Tipos de formatos de la trama de difusión BLE	11
Tabla 2. BOM Sterling Mark I.....	16
Tabla 3. Consumos Sterling Mark II.....	20
Tabla 4. BOM Sterling Mark III.....	27
Tabla 5. Consumo de los módulos.....	33
Tabla 6. Parámetros de funcionamiento del nRF24L01 [5]	35
Tabla 7. BOM Sterling Mark IV	55

Capítulo 1

Introducción

1.1 Motivación

Este trabajo abarca el marco de desarrollo de un sistema de sensorización de contenedores de residuos. Se ha tomado la dirección del proyecto una vez pasado el primer prototipo creado por la empresa, RDNest, que ha trabajado en calidad de cliente para Ecoembes.

El prototipo sobre el que hemos trabajado se desarrolló inicialmente con el objetivo de cubrir un rango de mercado claramente definido: aparatos de medida de llenado de contenedores.

Desde el inicio del desarrollo se produjo un reto de diseño, la transmisión de la información del dispositivo hasta el gestor del contenedor.

Durante las primeras variaciones del producto se optó por el envío de la información a través de un módulo GPRS (General Packet Radio Service), pero dicha implementación daba lugar a unos gastos periódicos añadidos que encarecían el producto, dificultando su apertura al mercado. Debido a esto se optó por otro medio de comunicación con el sensor: los propios usuarios de los contenedores y sus teléfonos móviles con acceso a Internet. Esto fue viable gracias a un sistema de incentivos a los usuarios basados en su nivel de participación.

El sistema de premios fue un éxito a la hora de incentivar el reciclaje, pero el dispositivo no se encontraba en un estado capaz de producir beneficios: requería un elevado mantenimiento, tenía una tasa de fallo pronunciada y no era compatible con muchos de los teléfonos de los usuarios.

Este trabajo ha sido enfocado a solucionar estos problemas, que iremos tratando más en detalle. También buscaremos facilitar el mantenimiento de los equipos, simplificar el diagnóstico de fallos y rediseñarlos con el fin de permitir su producción en grandes volúmenes.

1.2 Estructura

En este trabajo se abordará exclusivamente la parte hardware del proceso, aunque también esté compuesto por una aplicación móvil, una base de datos y una interfaz de usuario para el gestor.

Se ha partido del prototipo original diseñado por la empresa, el Mark I, a partir del cual se ha iterado en este trabajo hasta obtener la versión actual en producción, el Mark III.

La memoria consta de 6 capítulos que se van a separar en cuatro partes claramente diferenciadas:

1. Preámbulo

Desarrollo del protocolo Bluetooth de baja energía, BLE, su funcionamiento, y nuestro acercamiento a él en el capítulo 2.

2. Fase ATmega

Centrada en mejorar el prototipo inicial, esta fase comienza por un Arduino como prototipo inicial y termina por el microcontrolador de éste integrado en placa añadiendo variaciones como GPS (Global Positioning System) o GPRS. Se alcanza la versión en producción actual funcional, el Mark III Abarcará los capítulos 3 y 4.

3. Puesta en producción

En esta parte se explicará cómo se ha puesto en producción el proyecto gracias a la placa Mark.III desarrollada en la fase ATmega. Abarcará el capítulo 5

4. Fase Nordic

En esta fase se plantea un diseño más robusto y profesional, empleando un chip que además integre las funciones del microcontrolador y la comunicación bluetooth en un mismo integrado. Es una prueba de concepto no orientada a uso en producción. Abarcará el capítulo 6.

Tras estas tres fases tendremos un capítulo en el que ahondaremos en las conclusiones que hemos sacado del proyecto.

1.3 Objetivos

El prototipo inicial presentado cumplía sólo parte de los requisitos de diseño presentados, y no de forma fiable o estable.

Este proyecto está orientado en garantizar que el sistema sea capaz de cumplir con dichas exigencias y además de forma fiable y robusta, así como poder ofrecer mejoras añadidas que permitan abarcar un mayor campo de mercado. Todo esto manteniendo un compromiso económico.

1.3.1 Requisitos de diseño

Buscamos que el sistema:

1. Realice una medida del llenado del contenedor en el que se encuentre.
2. Obtenga una medida de la temperatura y la humedad.
3. Sea capaz de enviar información mediante el protocolo Bluetooth.
4. Cuento el número de aperturas realizadas en el contenedor.
5. Mida la batería restante en el sistema.
6. Posea un código identificador individual para poder ser reconocido y categorizado.
7. Requiera un mantenimiento mínimo, es decir: una vida útil de la batería longeva.

8. Posea una alta robustez, para alargar la vida útil del dispositivo.

1.4 Software

Para el desarrollo del proyecto se han empleado los siguientes programas:

1.4.1 Eagle CAD

1.4.2 Arduino IDE

1.4.3 Visual Studio Code

1.4.4 Keil MicroVision

1.5 Planificación

El desarrollo de este proyecto no se ha realizado en un marco de tiempo determinado con fechas límite asociadas, sino que se ha ido iterando progresivamente en espacios de tiempo irregulares, debido a que ha sido llevado a cabo simultáneamente en un entorno laboral junto a otros proyectos no relacionados.

Por lo tanto, pese a que no procede una planificación Gant sobre el progreso de este, sí que se ha realizado una estimación de las horas hombre empleadas para cada fase en el presupuesto final.

Capítulo 2

Bluetooth Low Energy (BLE)

2.1 Introducción

El protocolo Bluetooth, desarrollado en 1994, perseguía el envío de archivos a un coste y consumo reducidos. Presentaba en su momento con una sorprendente velocidad de 720kbs, aunque ahora podemos alcanzar hasta 24Mbs. La conexión se transmite en el rango de frecuencia 2400-2483.5 MHz, que está separado en 79 canales. [1]

Originalmente se diseñó con la intención de transmisión de información continuada en un corto rango, pero con la versión 4 del protocolo llegó un nuevo enfoque del protocolo: apartarse de la evolución de la velocidad de envío hacia una disminución del consumo. Para ello, entre otras cosas, el transceptor está diseñado para estar dormido hasta que tenga que realizar un intercambio de información, a diferencia del modelo clásico.

En el modelo clásico una conexión de poca información requeriría al menos 100mS, mientras que con esta aproximación se puede reducir a unos pocos mS (Dado que disponemos de una tasa de conexión teórica de hasta ~1Mb/s y el enlace entre dispositivos es más dinámico.)

Podemos hacer una clara diferencia entre el modelo de transmisión clásico al nuevo, el segundo no es el paso siguiente al primero. El primero lo emplearemos para transmisión de elevados volúmenes de información, y el segundo lo orientaremos a otros usos que no requieran transmisión de mucha información continua, como teclados o sensores.

Este proyecto se fundamenta en el segundo modelo de transmisión, llamado Bluetooth de baja energía o BLE en adelante. Esto es debido a que entre otras facilidades que nos brinda este protocolo, nos permite transmitir la información realizando un mucho menor consumo de energía. [2]

2.2 Bluetooth de baja energía

El protocolo conocido como Bluetooth de baja energía es la característica principal de la versión 4.0 del estándar Bluetooth, versión presentada en 2010 por la asociación sin ánimo de lucro Bluetooth SIG (Special Interest Group), de la que forman parte numerosas empresas tales como Ericsson, Microsoft, Nokia...

Los objetivos de dicho protocolo eran buscar el mayor ahorro de consumo de energía, permitiéndose a cambio reducir tanto el ancho de banda como especialmente el alcance, desde unos 100 metros máximos hasta menos de 10, en función de la potencia que queramos emplear.

2.3 Divisiones de frecuencia

El protocolo bluetooth se encuentra distribuido alrededor de los 2.4GHz. El rango se divide en 39 canales no ordenados separados en tramos de 2MHz, yendo desde los 2402 MHz hasta los 2480 MHz.

Dentro de esos 39 canales los tres últimos (situados en 2402MHz, 2426MHz y 2480MHz) son los empleados para difusión, como veremos más adelante.

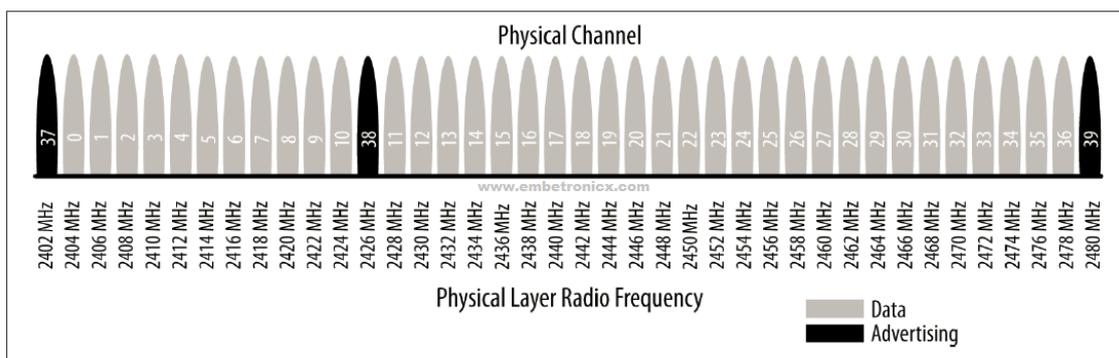


Ilustración 1. Canales BLE

2.4 Configuraciones

Existen todo tipo de implementaciones en el mercado, según las necesidades que tengamos.

Entre los principales factores determinantes nos encontraremos por ejemplo el coste de desarrollo, el coste final de producto, la velocidad requerida... en los siguientes apartados exploraremos las diferencias más notables. [2]

2.4.1 Según qué especificación soporten

Hemos de tener en cuenta de aquí en adelante que el protocolo BLE no tiene compatibilidad con el protocolo Bluetooth clásico. Esto causa que nos encontremos con 3 posibles configuraciones:

1. **Bluetooth clásico (BR/EDR):** Estándar clásico orientado a la transmisión de datos. Es el protocolo original, que ha evolucionado desde la versión 1.0.
2. **Monomodo (BLE, Bluetooth Smart):** Emplea el modo BLE presentado en la versión 4.0, pero no es capaz de conectarse con los módulos que presentan una configuración clásica BR/EDR.
3. **Modo dual: (BR/EDR/LE, Bluetooth Smart Ready):** Es capaz de manejar ambos estándares, nos permite una conexión con los módulos BR/EDR además de con los BLE.

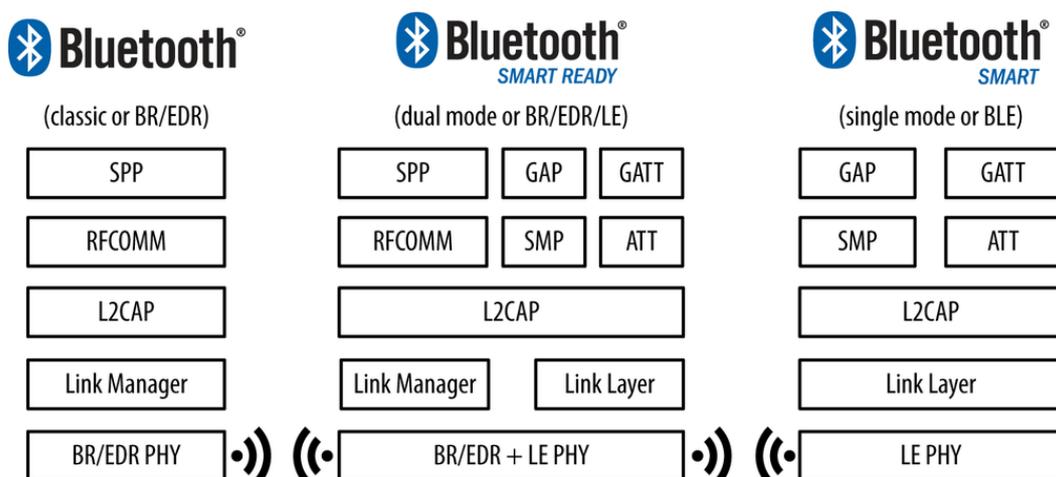


Ilustración 2. Especificaciones Bluetooth y sus capas correspondientes

2.4.2 Según el número de chips

Existen múltiples capas en los dispositivos Bluetooth, y nos vamos a centrar en los tres principales bloques:

1. **Aplicación:** La interfaz con el usuario a la pila Bluetooth
2. **Host:** Las capas superiores de la pila Bluetooth
3. **Controlador:** Las capas bajas de la pila bluetooth, incluyendo la radio.

El *Host* y el *Controlador* se comunican gracias a una HCI, *host-controller interface*, que permite la interconexión de diferentes *Hosts* y *Controladores*. (Está definido por el protocolo Bluetooth).

Todas estas capas se pueden implementar en un solo integrado o entre varios comunicados por una capa de comunicación (ISP, UART...). Las configuraciones que nos acabaremos encontrando son:

1. **SOC (System on Chip)**
Un solo integrado controla las tres capas.
2. **Dual IC over HCI**
Se divide en dos integrados, uno con la aplicación y el host, que se comunica por HCI con el otro, que maneja el controlador. Gracias a que el HCI está definido por el protocolo bluetooth, se pueden intercambiar cualquier host con cualquier Controller.
3. **Dual IC with connectivity device**
Un integrado con la aplicación se conecta a otro con el Host y el Controller.

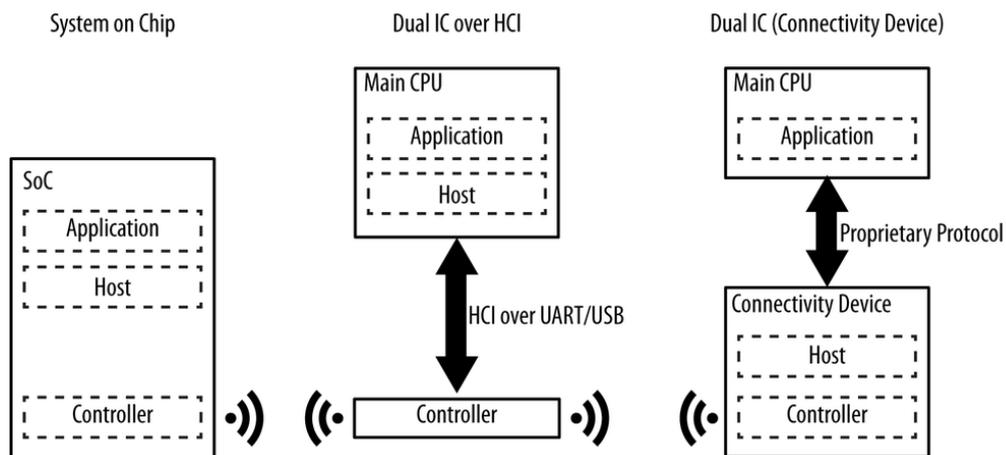


Ilustración 3. Posibles implementaciones Hardware

Generalmente se emplea un SOC a menos que se vaya a realizar un desarrollo mucho más elaborado y complejo, como un teléfono móvil, smartwatch con un procesador especializado, o similares.

2.4.3 Topología

Cada dispositivo Bluetooth se puede conectar al exterior de dos formas: Difusión (Broadcasting a partir de ahora) o enlace (Connection en adelante).

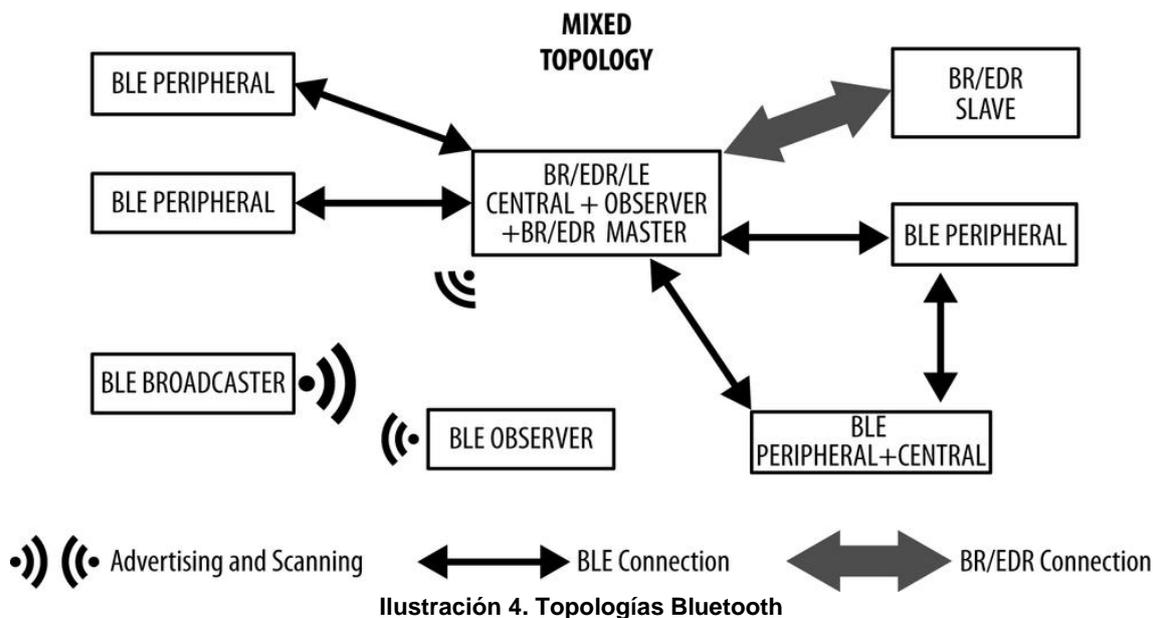
1. Broadcasting:

Tendremos un emisor que emite los paquetes exclusivamente hacia el exterior hacia cualquier otro dispositivo que esté escuchando. Es la única forma de que un sistema envíe información a más de un usuario a la vez.

Cada paquete de Advertising contiene un payload de 31 bytes, en el que transmite desde información propia del emisor hasta información personalizada. En caso de necesitar más espacio para la información, los usuarios pueden pedir una segunda trama de advertising (Mediante un "Scan Request") con otros 31 bytes de payload, llamada "Scan response". Hay que tener en cuenta que con estos tipos de conexión no existe ningún tipo de seguridad.

2. Conexión:

Aquí tendremos un intercambio de datos por ambas partes. Una conexión es "Permanente", intercambiando información periódicamente de manera privada. Este aspecto es muy extenso, pero no se va a detallar debido que en este dispositivo se va a emitir exclusivamente en difusión.



2.5 GAP (Generic Access Profile)

El GAP es la parte más importante de esta tecnología: se encarga de que se puedan comunicar los diferentes sistemas.

Marca unos parámetros a seguir por los diferentes dispositivos, de forma que puedan trabajar correctamente entre ellos. Estos aspectos que seguir se dividen en:

2.5.1 Roles

Cada dispositivo puede adaptar diferentes roles en la comunicación, lo que hace que se tenga que atender a ciertas restricciones.

- **Broadcaster**
Se dedica exclusivamente a emitir información mediante paquetes de difusión, llamados de "Advertising"
- **Observer**
Se dedica exclusivamente a recibir información de Broadcasters.
- **Central**
Se encarga de establecer múltiples conexiones con otros dispositivos, siendo siempre el iniciador de la conexión.
- **Peripheral**
Envía paquetes de broadcast hasta ser descubierto por un Central, para que pueda establecer una conexión con él.

2.5.2 Modos

Un modo es el estado que puede alcanzar un dispositivo durante un tiempo para conseguir un objetivo o que otro dispositivo prosiga un procedimiento.

2.5.3 Procedimientos

Los procedimientos son una secuencia de acciones que permiten alcanzar ciertos objetivos.

2.5.4 Paquetes de difusión

Existen dos tipos de tramas bien diferenciadas: de difusión y de enlace. Vamos a explorar en detalle las de difusión ignorando las de enlace, debido a que las de enlace no serán empleadas.

Las tramas de difusión se emplean para transmitir poca cantidad de información o para buscar establecer conexiones.

Estos paquetes se envían a ciegas sin tener conocimiento del entorno, a una frecuencia de envío fijada por el intervalo de difusión (20ms a 10.24s). Como es de esperar, una mayor frecuencia de envío aumenta el consumo, pero también la posibilidad de que un posible receptor pueda verlos.

Se envían exclusivamente por los tres canales de difusión mencionados previamente, y dado que el receptor sólo puede escuchar por uno de los canales a la vez y no se encuentran sincronizados el emisor con el receptor, ha de coincidir que ambos se encuentren en el mismo canal. Esto se suele solventar intentando enviar paquetes en los tres diferentes canales dentro del plazo de tiempo en el que el receptor analiza uno de ellos. Entrando en detalle:

El dispositivo que escanea el medio tiene dos parámetros para definir cómo hacerlo: El tiempo de intervalo y el tiempo de ventana. Estos dependen del consumo de energía que esté dispuesto a realizar el dispositivo. No son parámetros universales, así que no podremos adecuarlos.

El dispositivo escuchará en cada canal durante el tiempo de intervalo, y al final de cada tiempo de ventana saltará al siguiente canal. El tiempo de intervalo será siempre inferior al de ventana.

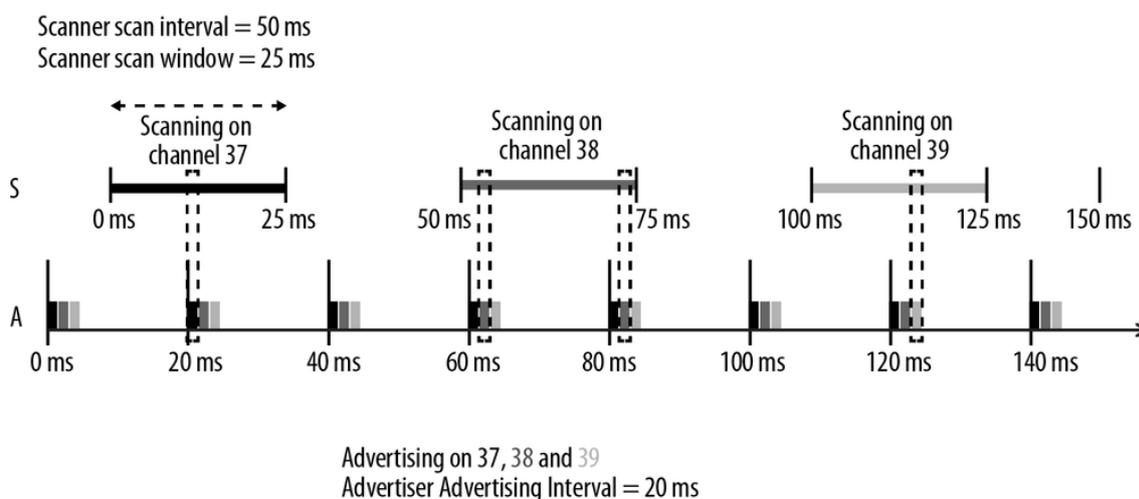


Ilustración 5. Intervalos de ventana de escucha BLE

Los tipos de paquetes de difusión se subdividen según tres propiedades:

1. Conectividad:

El receptor puede iniciar una conexión tras recibir el paquete o no

2. Escaneabilidad:

El receptor puede enviar un "Scan request" al recibir el paquete.

3. Dirección:

O el paquete contiene las direcciones del emisor y el receptor en el payload, siendo orientados a realizar una conexión, o el payload contiene simplemente datos.

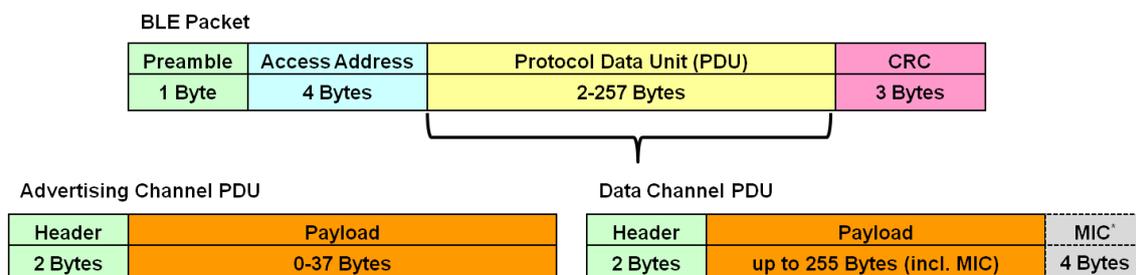
En función de estas propiedades se clasifican en cuatro tipos:

Tipo	Conectividad	Escaneabilidad	Dirección	Nombre GAP
ADV_IND	Yes	Yes	No	Connectable Undirected Advertising
ADV_DIRECT_IND	Yes	No	Yes	Connectable Directed Advertising
ADV_NONCONN_IND	No	No	No	Non-connectable Undirected Advertising
ADV_SCAN_IND	No	Yes	No	Scannable Undirected Advertising

Tabla 1. Tipos de formatos de la trama de difusión BLE

[3]

Sólo existe un formato de trama, para tanto los paquetes de difusión como los de datos:



Ref: BT Specification v4.2, Vol. 6, Part B, Sec. 2.1

*Message Integrity Check: Included as part of Payload if used (for security)

Ilustración 6. Formato de una trama Bluetooth

Los paquetes de difusión BLE tienen un tamaño variable de hasta 39 bytes, que tras restar 2 de la cabecera y 6 de la dirección del dispositivo, se quedan en 31 bytes disponibles.

Dentro de esos 31 bytes hay que tener presente que la información se transmite en campos definidos de la siguiente manera:

- **Byte 1 (Length):** Longitud del campo (AD Type + AD Data en bytes)
- **Byte 2 (AD Type):** Tipo de campo
- **Bytes siguientes (AD Data):** Información del campo

Los tipos de campo más comunes son los siguientes:

1. **Flags (0x01):**

Determinan el tipo de conexión que se va a realizar, es necesario siempre que alguno de sus bits no sea 0 y el emisor busque enlace. Es decir, si el emisor no busca enlace, es omisible. Siempre que no sea BLE, porque en tal caso uno de los bits sería 1 (“BR/EDR Not Supported”). Ocupa 3 bytes..

2. **Incomplete List of 16-bit Service Class UUIDs (0x02):**

Indica una lista parcial de los posibles servicios ofrecidos por el dispositivo

3. **Shortened Local Name (0x08):**

Da un nombre breve al dispositivo

No vamos a cubrir todos los campos existentes ya que existen demasiados para abarcar en este documento. Una lista completa se puede encontrar en [4].

2.5.5 Seguridad

El GAP se encarga de definir el nivel de seguridad necesario, así como de mantener dicho nivel a lo largo de la conexión. En nuestro dispositivo sólo se retransmiten paquetes de difusión, que no conllevan ningún tipo de seguridad, así que no se va a tocar este aspecto.

2.6 Limitaciones

Hay que tener en cuenta que el protocolo BLE no está diseñado para ser una solución universal, por lo que presenta bastantes limitaciones buscando economizar el consumo de energía al máximo. Para entender mejor las limitaciones presentes, vamos a presentar un caso práctico de conexión, con un SoC nRF51822, de Nordic Semiconductors [5].

El estándar Bluetooth define que se realiza el intercambio de datos en ventanas de tiempo configurables de 7.5ms a 4s, y cada paquete de datos enviado puede contener hasta 20 bytes de datos (a menos que se negocie un mayor número).

- El SoC define un máximo de seis paquetes por cada ventana.
- Buscando el mayor número de ventanas por segundo obtenemos 133.
- En cada ventana podremos meter 6 paquetes de 20 bytes, es decir, 120 bytes

$$133 \frac{\text{ventanas}}{\text{segundo}} * 120 \text{ bytes} = 15960 \frac{\text{bytes}}{\text{segundo}} \text{ o } \sim 0.125 \frac{\text{Mbit}}{\text{segundo}}$$

Esto implica que podremos obtener una velocidad máxima de apenas **~125kbit/s**. La especificación BLE nos dice que podremos obtener un máximo de 1Mbit/s, pero como acabamos de ver en la práctica, obtendremos valores próximos a unos 5-10 kbit/s.

¿Esto implica que sea un mal protocolo? No. Como bien hemos explicado previamente, no está orientado a la transmisión de información, sino a la economía de la batería. Organizar las comunicaciones en ventanas implica que al final de cada una de ellas el sistema podrá volver al estado de espera, en el que podrá disminuir su consumo todo lo posible.

Capítulo 3

Fase ATmega: Mark I a Mark III

3.1 Introducción

Esta fase comprende las tres etapas del proyecto que emplean como microcontrolador un ATmega328p.

El trabajo parte de un diseño sencillo (Mark I) sobre el que va iterando, pero manteniendo la restricción del módulo Bluetooth escogido al inicio para evitar en lo posible la modificación de código con el consecuente añadido de horas hombre. También se ha mantenido el módulo de sensorización de temperatura y humedad, pero no resulta tan crucial.

Buscando mejorar dicho diseño se hicieron pequeñas mejoras sobre este prototipo para dejarlo funcionando en producción a un nivel aceptable, definiendo a los equipos modificados como Mark II.

Tras esto, se hizo un rediseño de la placa sustituyendo los Arduinos por una implementación en placa, llamada Mark III. Esto se explorará en la sección 3.4.

Es importante denotar que esta fase se desarrolló manteniendo tanto el módulo Bluetooth como el microcontrolador ATmega328p, pese a tener mejores alternativas, por restricciones de tiempo y económicas.

Una vez sobrepasadas dichas limitaciones se pudo llevar a cabo el Mark IV. Éste será presentado en el capítulo IV.

3.2 Punto de partida: Sterling Mark I

Esta primera iteración la realizó como trabajo de fin de grado por Ester Diego Cristóbal, alumna de la ETSII de la UVA.

Es el diseño hardware inicial previo al inicio de las mejoras, desarrollado buscando ser una prueba de concepto económica.

Por ello, se presenta con unas placas desarrolladas planteando un módulo Arduino Mini Pro como microcontrolador y diversos módulos prediseñados, en busca de un desarrollo económico y accesible. Parte de los componentes fueron comprados en sitios menos fiables, como Aliexpress, de lo cual hablaremos más adelante.

3.2.1 Componentes iniciales

A continuación, se listan los componentes originales:

1. **HC-SR04**
Módulo sensor de distancia (por ultrasonidos). Se encarga de cumplir el primer requisito de diseño: realizar una medida del llenado del contenedor [6].
2. **DHT-11**
Módulo sensor de humedad y temperatura. Cumple el segundo requisito de diseño: realizar medidas tanto de temperatura como de humedad [7].
3. **nRF24L01**
Módulo Bluetooth encargado de la transmisión de la información. Maneja todo lo relacionado con el tercer requisito de diseño: la comunicación Bluetooth [5].
4. **Arduino Pro Mini**
Módulo con un microcontrolador integrado encargado de la parte de procesamiento. Cumple con los requisitos de contar el número de aperturas, de medida de batería, e identificación individual.
5. **Conectores varios**
Nos permiten la conexión de la placa con el exterior.
6. **Interruptor tipo Reed**
Un interruptor tipo Reed es un conmutador que cambia de estado abierto a cerrado al entrar en el campo cercano de un imán. Gracias a este efecto se ha posicionado el imán en la tapa del contenedor y el interruptor fuera de la tapa, pero adyacente y en dentro del contenedor. Esto causa que se abra el circuito al accionarse la tapa, causando un cambio de flanco de voltaje en un pin de interrupción, que despierta el microcontrolador.
7. **Componentes adicionales**
Aquí se engloban las baterías, la caja, el portabaterías, cables adicionales, tornillos y cinta de doble cara.

3.2.2 Placa inicial

La placa inicial se desarrolló buscando integrar estos componentes fácilmente permitiendo su sustitución de forma sencilla gracias al soldado de zócalos sobre ella en los que colocar los componentes.

El trazado de pistas lo realizó el programa de forma automática sin realizar ajuste de tamaño de las pistas según potencia o uso, así como sin optimizar el diseño.

Fue diseñada para disponer de múltiples puertos de comunicación:

1. **Puerto ICSP (In Chip Serial Programming)**
Permite programar tanto el firmware como el bootloader mediante 6 pines.
2. **Puerto serie**
Brinda una conexión serie para facilitar el diagnóstico de funcionamiento del microcontrolador, permitiéndole comunicarse con el PC.
3. **Puerto JST**
Conecta un pin del microcontrolador capaz de manejar interrupciones con el sensor Reed externo para poder despertarse.
4. **Puerto de alimentación**
Permite conectar un portabaterías con pilas 18650 de 3,7V y 2500mAh.

La elección de baterías 18650 a 3,7V se realizó para poder alimentar a un Arduino Pro Mini efectivamente y se mantuvo durante toda la fase ATmega.

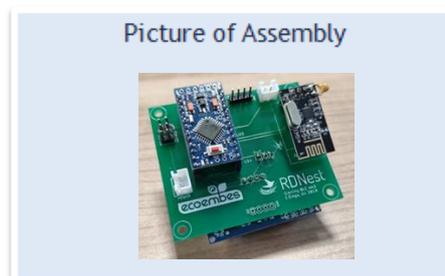
El módulo de ultrasonidos ha sido implementado en la parte inferior de la placa, haciendo que ésta tenga que ser montada sobre unos soportes que permitan que los transductores tengan la postura adecuada.



Ilustración 7. Sterling Mark I

3.2.3 Bill Of Materials (Listado de componentes) simplificado

Assembly Name	Sterling
Assembly Revision	3
Approval Date	05-Dec-18
Part Count	12
Total Cost	\$36,57



Part Name	Description	Qty	Codigo Farnell / Amazon	Unit Cost	Cost
Arduino Pro Mini	Microcontrolador	1	2133073	€ 8,99	8,99 €
nRF24L01	Módulo Bluetooth	1	Link Amazon	€ 1,42	1,42 €
HC-SR04	Sensor Ultrasonidos	1	2801408	€ 4,35	4,35 €
Reed Switch	Sensor de apertura de contenedor Reed	1	Link Amazon	€ 4,99	4,99 €
DHT-11	Sensor Temp&Hum	1	Link Amazon	€ 1,98	1,98 €
Portabaterías	Soporte baterías 18650	1	2674338	€ 2,35	2,35 €
Batería	Batería 18650 3,7V 2,6Ah	1	Link Amazon	€ 9,99	9,99 €
Pin Header	2.54mm 2x3 Contactos	1	1248159	€ 0,57	0,57 €
Pin Header	2.54mm 1x4 Contactos	1	1022251	€ 0,10	0,10 €
Pin Header	Conector JST para batería y sensor Reed	2	1516276	€ 0,06	0,12 €
PCB	Placa (Cálculo con pedido x30)	1		€ 1,71	1,71 €
Total/Placa		12			36,57 €

Tabla 2. BOM Sterling Mark I

3.2.4 Esquemático

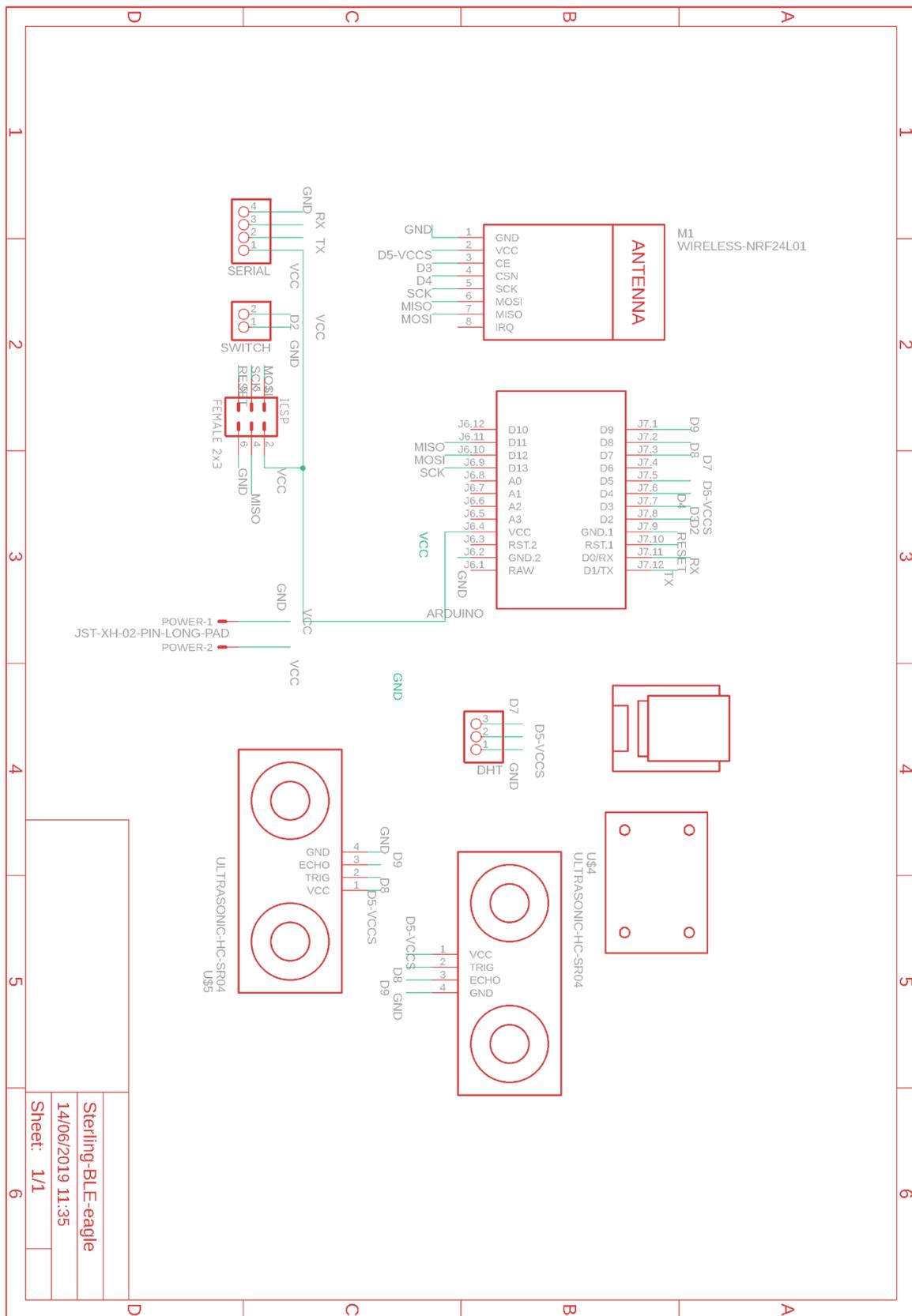


Ilustración 8. Esquemático Sterling Mark I

3.2.5 Diseño PCB

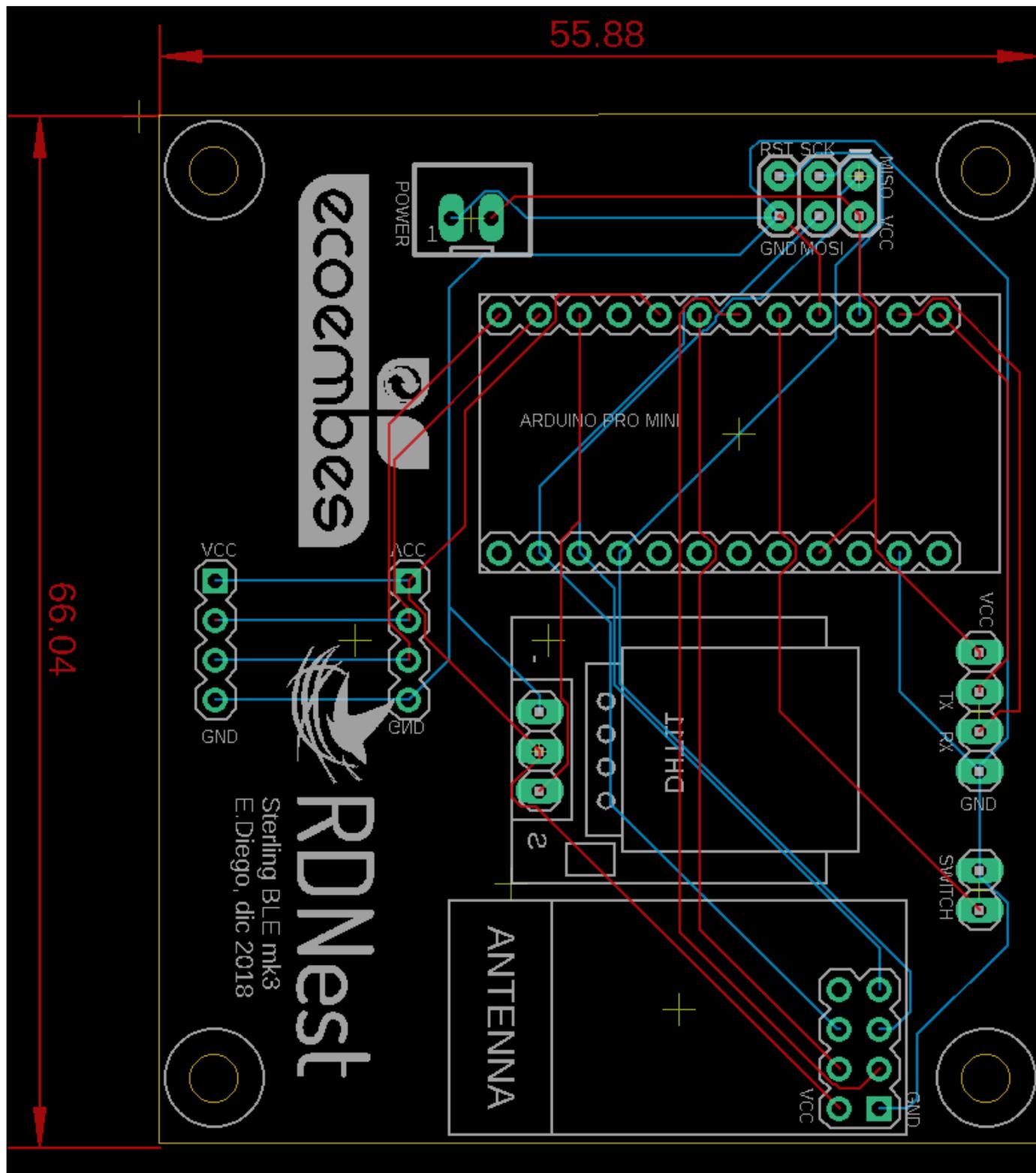


Ilustración 9. Diseño PCB Sterling Mark I

3.2.6 Problemas presentados

Para cada uno de estos problemas se han ido buscando diferentes soluciones que se explorarán en los próximos apartados.

- Pérdida de envíos

Como bien hemos visto antes, se ha de tratar de enviar un paquete por cada canal de difusión dentro de los intervalos de escucha del receptor. El módulo emisor nRF24L01 no es capaz de cambiar de canal en tiempos tan cortos.

- Corta duración de la batería

Tanto el diseño hardware como software inicial acarrea un consumo de energía que reduce la duración de la batería añadida a una media de 2 semanas.

- Baja robustez

Un gran número de los módulos montados no llegan a ser capaces de realizar envío de paquetes o tienden a reiniciarse aleatoriamente debido a una multitud de factores explorados más adelante.

- Poca fiabilidad de los componentes

Parte de los componentes, especialmente el módulo de radiofrecuencia, fueron comprados en Aliexpress. Esto implica que un cierto porcentaje de los componentes resultaron ser defectuosos, haciendo requerir tiempo de diagnóstico y requerir repuestos. Revisar sección 4.10.

- Comportamiento irregular de los componentes

Debido a los dos problemas previos, a la baja robustez del diseño y a la poca fiabilidad de sus componentes.

- Baja escalabilidad

Al haber elegido módulos hobby y no disponibles en tiendas de electrónica, sólo en comercios alternativos, se hace más difícil tanto como su montaje a tantos volúmenes como el conocimiento del stock disponible, además de conllevar un mayor índice de componentes defectuosos.

- Difícil montaje en caja

El sensor de ultrasonidos se situó en la parte inferior de la placa, lo que implica que hubo que alzarla sobre la caja en la que se monta con tornillos. Además, el sensor de distancia inicial requiere dos agujeros individuales para sus transductores.

- No impermeable

- Voltajes de alimentación diferentes

Pese a que el circuito se diseñó para funcionar a 3V, el programador empleado pone dicho circuito a 5V. Esto puede dañar el emisor bluetooth durante el tiempo de programación.

- Puerto ICSP mal planteado

Los pines MOSI y MISO estaban invertidos.

- Plano de masa bajo la antena Bluetooth

En una primera iteración se situó un plano de masa bajo la antena Bluetooth. Esto reduce enormemente el alcance, pero fue corregido antes de hacer un pedido de placas para montaje.

3.3 Sterling Mark II

En esta fase se buscaron realizar mejoras menores sobre el primer dispositivo previas a un rediseño completo del sistema. Se pasó por esta etapa intermedia debido a que el sistema ya había sido desplegado en la UVA y podían hacerse mejoras rápidas. El punto clave crítico de estas mejoras fue el de un aumento de la vida útil de la batería.

Pese a no suponer cambios mayores de diseño se creyó conveniente denominarla con un nuevo nombre para facilitar el estudio del desarrollo del proyecto. Es el primer paso de este proyecto y los primeros cambios y mejoras aplicados por este TFG.

3.3.1 Cambios realizados

La vida útil del prototipo inicial era muy limitada: Requería un cambio de batería aproximadamente cada semana y media. Para reducir en lo posible ese gasto de mantenimiento se realizaron las siguientes mejoras orientadas a un aumento de la vida de la batería, además de intentar mejorar en lo posible la tasa de usuarios que recibiesen correctamente los paquetes:

1. La alimentación de módulos de sensorización y bluetooth fue separada en dos pines diferentes para evitar alimentar innecesariamente a uno a la vez que el otro.
2. Añadidos condensadores de desacoplo y alimentación al módulo bluetooth
3. Aumentada la frecuencia de envío de paquetes en cada activación por interrupción, desde un paquete por segundo sumando un total de 30 a uno cada 30ms sumando un total de 100. (Esto también mejoró la tasa de recepción de paquetes por parte del usuario)
4. Retirados tanto los LEDs como el regulador de tensión para disminuir el consumo. Esto ha supuesto un aumento en la vida de la batería de unas 2 semanas a un año y medio teórico.
Gracias a que añadimos condensadores que estabilizasen la tensión en los componentes y que el rango de voltaje de la batería se encontraba muy cercano al rango de voltajes de operación de casi todos los componentes, compensó la eliminación de este.

3.3.2 Consumos

Las corrientes medidas en la práctica (a una media de voltaje de 3,7V) pasaron a ser:

Durante medida (1.5s)	Durante transmisión (10s)	En reposo
6.8 mA	4.45 mA	80 μ A

Tabla 3. Consumos Sterling Mark II

Con estos datos tenemos un cálculo teórico de la vida útil sin mantenimiento (Con una batería de 2500mAh) de una duración de más de dos años y medio en reposo, y con uso, haciendo cálculos muy conservadores, año y medio. Hay que tener en cuenta que

no se ha valorado la descarga propia de la batería, al no poder realizar un estudio extensivo de esta.

3.3.3 Imágenes

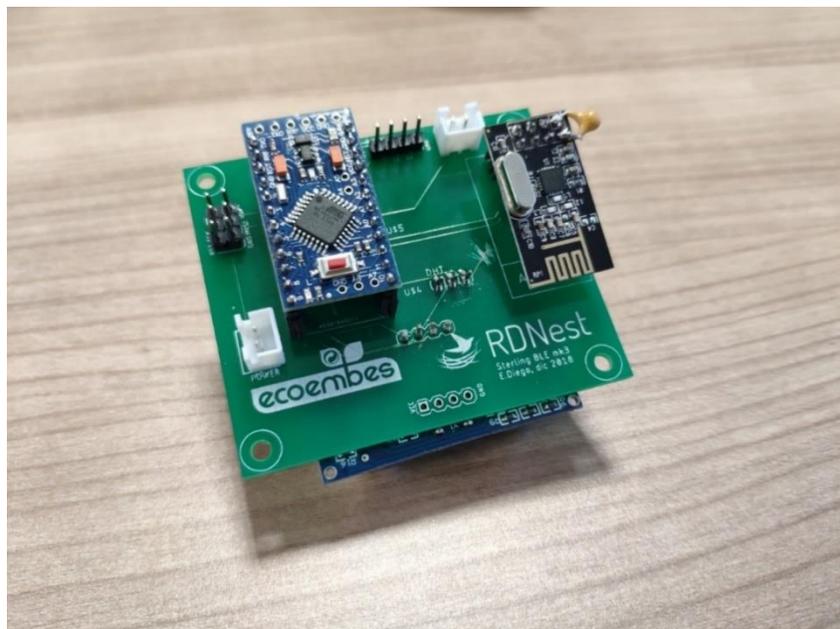


Ilustración 11. Parte delantera del Sterling Mark II

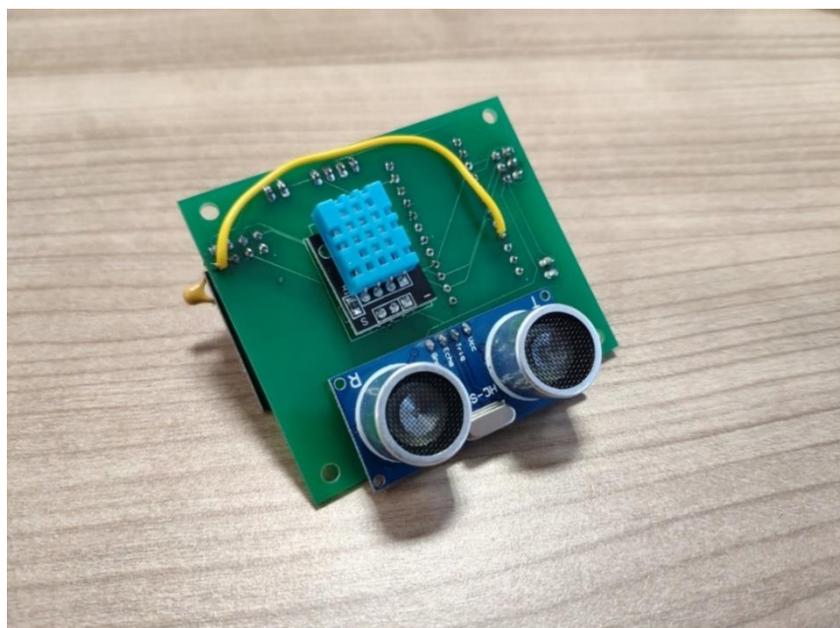


Ilustración 10. Parte trasera del Sterling Mark II

3.3.4 Problemas presentados

Pese a haber resuelto parte de los problemas, seguían presentándose grandes inconvenientes:

- Pérdida de envíos
Pese a haber mejorado la situación previa, el sistema sigue teniendo una elevada pérdida de paquetes.
- Baja robustez
Este problema sigue presente, aunque en menor medida tras dar una alimentación más estable a los componentes
- Poca fiabilidad de los componentes
Muchos de los componentes siguen sin ser de fuentes fiables.
- Comportamiento irregular de los componentes
Al mejorar la robustez hemos mejorado el comportamiento de estos, pero siguen dando problemas no deseados.
- Baja escalabilidad
Menos escalable aún debido a la mano de obra añadida.
- Difícil montaje en caja
Sin cambios.
- No impermeable
- Voltajes de alimentación diferentes
Se ha pasado a montar el módulo Bluetooth en un zócalo para poder ser retirado en el tiempo de programación.
- Puerto ICSP mal planteado
Los pines MOSI y MISO siguen invertidos.

3.4 Sterling Mark III

3.4.1 Introducción

Esta placa es la que se encuentra actualmente en producción y la que ha marcado el objetivo principal del proyecto.

En esta iteración se realizó un rediseño completo de la placa, ocupando la mayor parte del trabajo de este proyecto. Se dejó atrás el Arduino y se avanzó en muchos aspectos, listados a continuación.

3.4.2 Cambios realizados

En esta iteración se buscó principalmente una mayor robustez y capacidad de fabricación en grandes cantidades. Los cambios realizados fueron:

1. Dejado atrás el Arduino completo, se ha implementado el ATmega328p directamente en placa, junto al resto de los componentes necesarios para su correcto funcionamiento. Esto fue más sencillo de lo esperado debido a que los esquemáticos de Arduino son de dominio público.
2. Añadidos múltiples condensadores de desacoplo y alimentación a los sensores en formato SMD, facilitando industrialización.
3. Alimentación de los módulos pasada del microcontrolador a la batería directamente, siendo controlada por un interruptor MOSFET tipo P.
4. Módulo de medida de distancia sustituido de un HC-SR04 a JSN-SR04 [8], haciéndolo impermeable con un montaje adecuado.
5. Añadido un Step-Up opcional que aumenta el voltaje de 3,7v a 5v para poder alimentar apropiadamente al sensor de distancia en caso de que lo necesite. En la práctica los resultados de distancia han sido suficientemente fiables al recibir menos voltaje así que se ha ignorado. Ver sección 4.2.
6. Añadido una huella para añadir un sensor de inclinación CW1300 [9] opcional (integrado o por otro puerto JST añadido) para actuar como activador por interrupción. Pese a funcionar más que correctamente, se acabó decidiendo que el método de despertar más adecuado era el sensor tipo Reed.
7. Añadida compatibilidad con módulo GPRS SIM800L [10].
8. Todos los elementos se han pasado a un solo lado de la placa para facilitar su montaje tanto en placa como posteriormente en caja impermeable.
9. Ahora que se realiza la alimentación de los diferentes módulos desde un transistor, se ha vuelto a conectar la alimentación del transmisor bluetooth a la del sensor de distancia. El sensor de distancia, al disponer de un led integrado, nos permitirá saber cuándo está despierto el sistema a un muy reducido coste energético debido al corto tiempo que se encuentra en dicho estado.

3.4.3 Consumos

Una vez analizada en la práctica obtenemos los siguientes consumos (también a una media de 3.8V):

Activo	En reposo
15 mA	70 μ A

Alimentar a los módulos de sensorización antes para después alimentar al de radiofrecuencia en el Mark II demostró ser una complicación innecesaria que, aunque disminuye ligeramente el consumo durante los periodos de actividad, no produce variaciones realmente apreciables debido a la brevedad de dichos periodos.

En términos teóricos disminuía la duración en términos de un par de meses sobre una media de dos años de vida, y dado que idealmente se va a reemplazar esta versión con la siguiente antes de ese plazo, se optó por alimentar todos los módulos a la vez para disminuir posibles problemas.



Ilustración 12. Sterling Mark III

3.4.5 Problemas presentados

En esta iteración se solucionaron la mayoría de problemas. Pero debido a las limitaciones propias de alguno de los componentes aún se presentaban las siguientes barreras:

- Pérdida de envíos
Pese a haber mejorado la situación previa, el sistema sigue teniendo una elevada pérdida de paquetes.
- Poca fiabilidad de los componentes
Esta iteración sigue restringida al uso del módulo nRF24L01 y el DHT11, que no se pueden obtener de comercios como Farnell, Digikey, o tiendas de electrónica.
- Comportamiento irregular de los componentes
Al mejorar la robustez hemos mejorado el comportamiento de los sensores y el emisor, pero siguen dando problemas no deseados.

Se aspira a superar estos problemas con la siguiente iteración del proyecto: el Mark IV, del que se hablará en el capítulo 6.

3.4.6 Sterling Mark III ex

Buscando ampliar el rango de casos de uso, se realizaron dos añadidos al Mark III: Espacio y conexionado para un módulo GPS, que permita enviar las coordenadas en las que se encuentre, junto a un módulo GPRS SIM800L que sustituyese las conexiones bluetooth.

Ambos módulos se acompañaron de un necesario condensador de $1000\ \mu F$ que estabilizasen los pronunciados picos de corriente creados.

El mayor pedido de placas se ha realizado con esta variante, gracias a que basta con realizar un corte en el lateral que retire el GPS para que la placa vuelva a ser en la práctica la original.

Esta variación sobre el Mark III no se usó en producción, pero su funcionamiento en prototipo fue correcto.

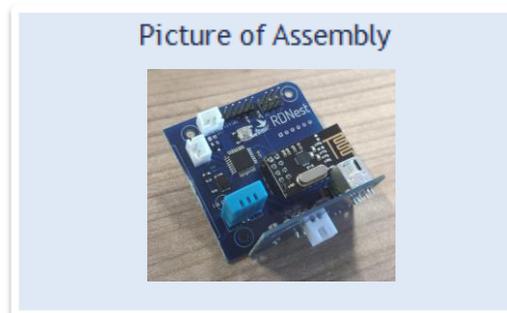
No vamos a desarrollar la documentación sobre este modelo, ya que se limitó a ser una prueba de concepto. El conexionado fue sencillo y el firmware únicamente dedicado a probar la viabilidad del concepto, enviando la información obtenida por el GPS mediante GPRS a un servidor de la empresa satisfactoriamente.



Ilustración 13. Sterling Mark III ex

3.4.7 Bill Of Materials simplificado

Assembly Name	Sterling Mk. III - BLE&GPRS
Assembly Revision	2
Approval Date	28-Jun-19
Part Count	25
Total Cost	\$43,12



Part Name	Description	Qty	Codigo Farnell / Amazon	Unit Cost	Cost
ATmega328P	Microcontrolador	1	715486	€ 1,63	1,63 €
Oscilador	Cristal 8Mhz	1	2101329	€ 0,78	0,78 €
nRF24L01	Módulo Bluetooth	1	Link Amazon	€ 1,43	1,43 €
SIM800L	Módulo GPRS opcional	1	Link Amazon	€ 8,99	8,99 €
JSN-SR04T	Sensor ultrasonidos impermeable	1	Link Amazon	€ 6,99	6,99 €
Reed Switch	Sensor de apertura de contenedor Reed	1	Link Amazon	€ 4,99	4,99 €
DHT-11	Sensor Temp&Hum	1	Link Amazon	€ 1,98	1,98 €
Portabaterías	Soporte baterías 18650	1	2674338	€ 2,35	2,35 €
Batería	Batería 18650 3.6V 2,6 Ah	1	Link Amazon	€ 9,99	9,99 €
FDN304P	Transistor alimentación	1	9846298	€ 0,36	0,36 €
Switch	Interruptor RESET	1	9471880	€ 0,45	0,45 €
C 30pF	Condensadores 30pF para el cristal	2	1865477	€ 0,20	0,40 €
C 0,1pF	Condensadores 0,1pF para el funcionamiento del integrado	2	2990655	€ 0,02	0,04 €
C 0,1uF	Condensadores 0,1uF estabilizadores de tensión	3	2581046	€ 0,04	0,12 €
Resistencia	Resistencia 10kOhm	2	9237755	€ 0,04	0,08 €
Pin Header	2.54mm 2x3 Contactos	1	1248159	€ 0,57	0,57 €
Pin Header	2.54mm 1x4 Contactos	1	1022251	€ 0,10	0,10 €
Pin Header	Conector JST para batería y sensor Reed	2	1516276	€ 0,06	0,12 €
PCB	Placa (Cálculo con pedido x30)	1		€ 1,71	1,71 €
Total/Placa		25			43,06 €

Tabla 4. BOM Sterling Mark III

3.4.8 Esquemáticos, diseño PCB

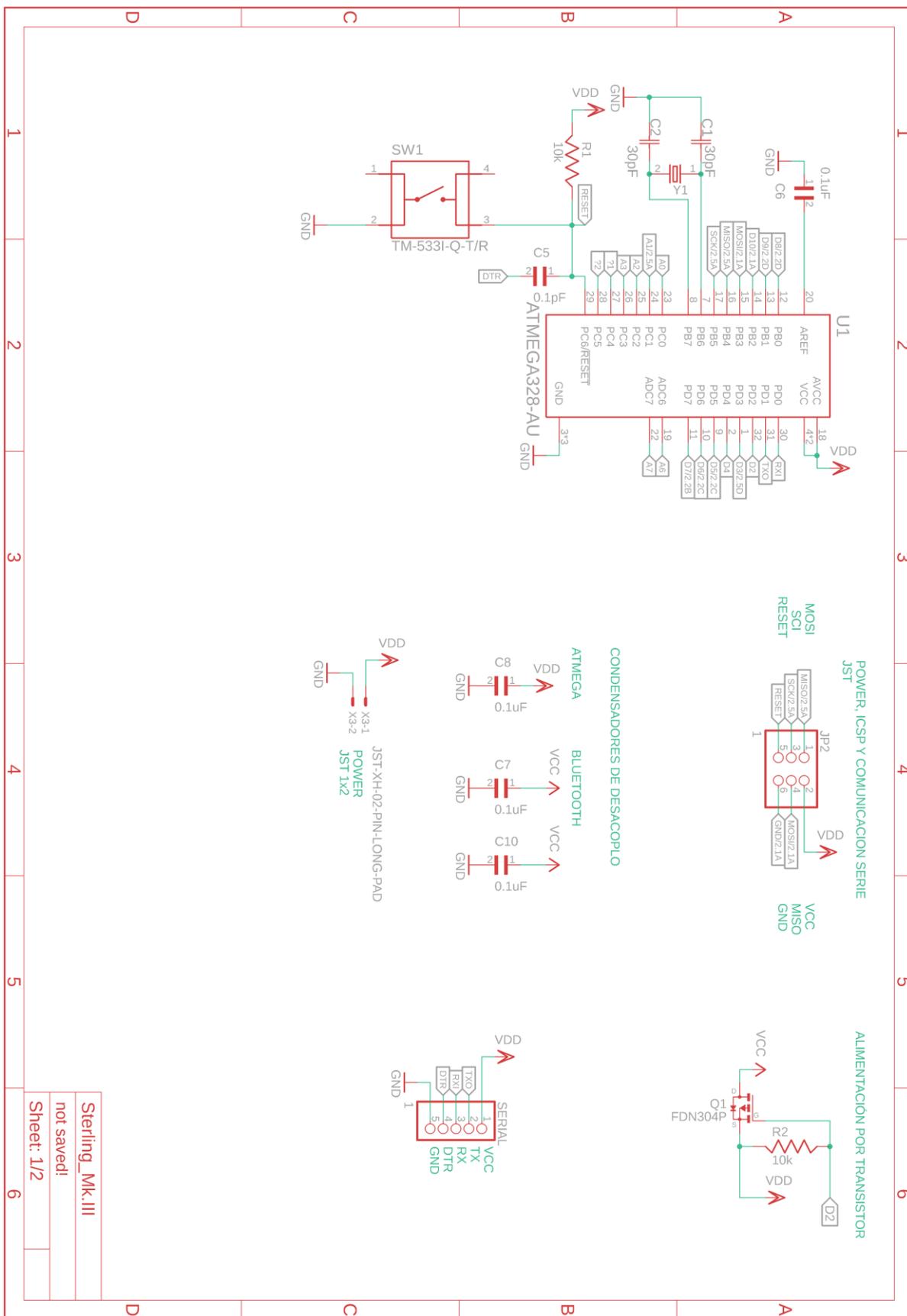


Ilustración 14. Página 1 del esquemático de Sterling Mark III

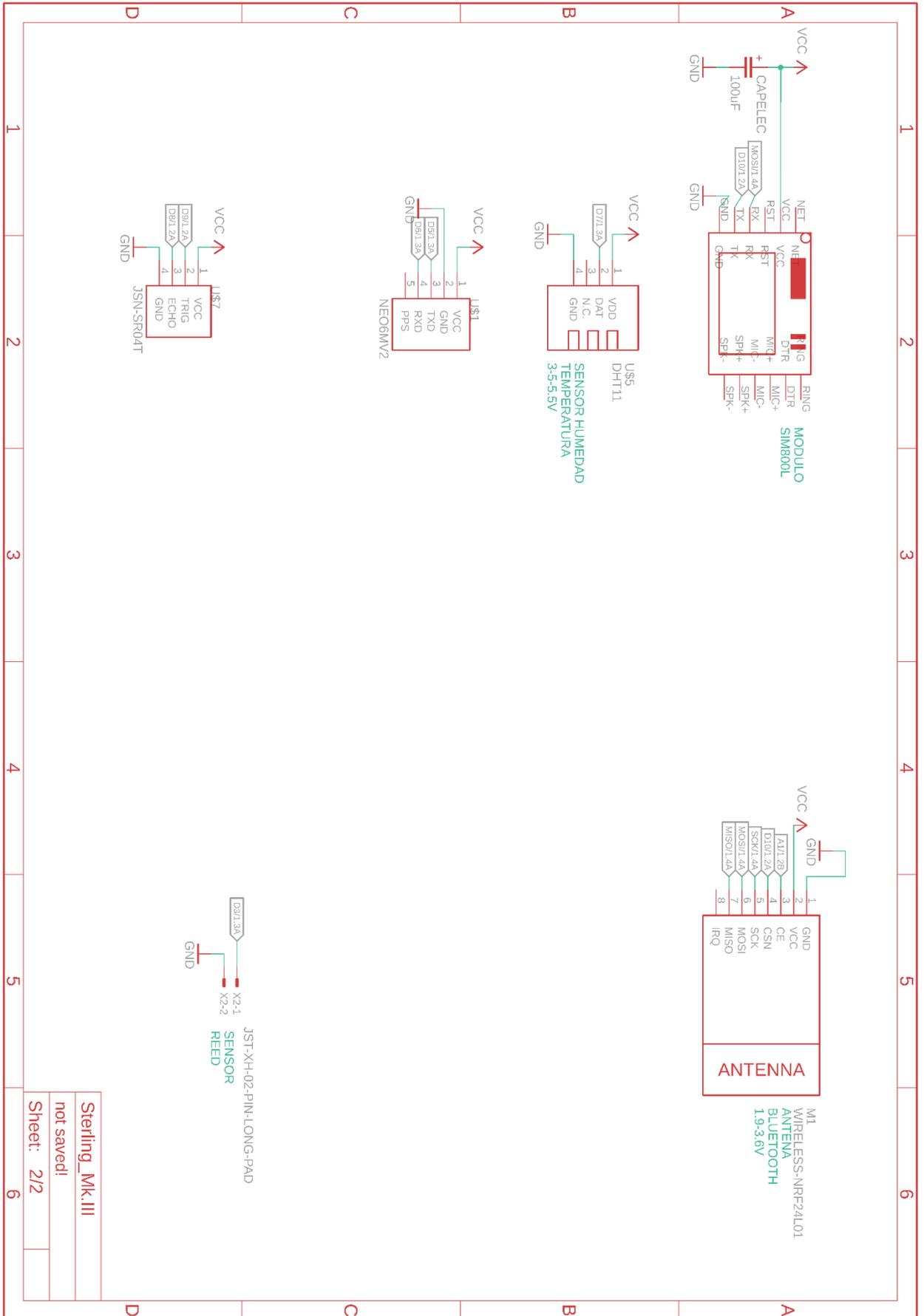


Ilustración 15. Página 2 del esquemático de Sterling Mk.III

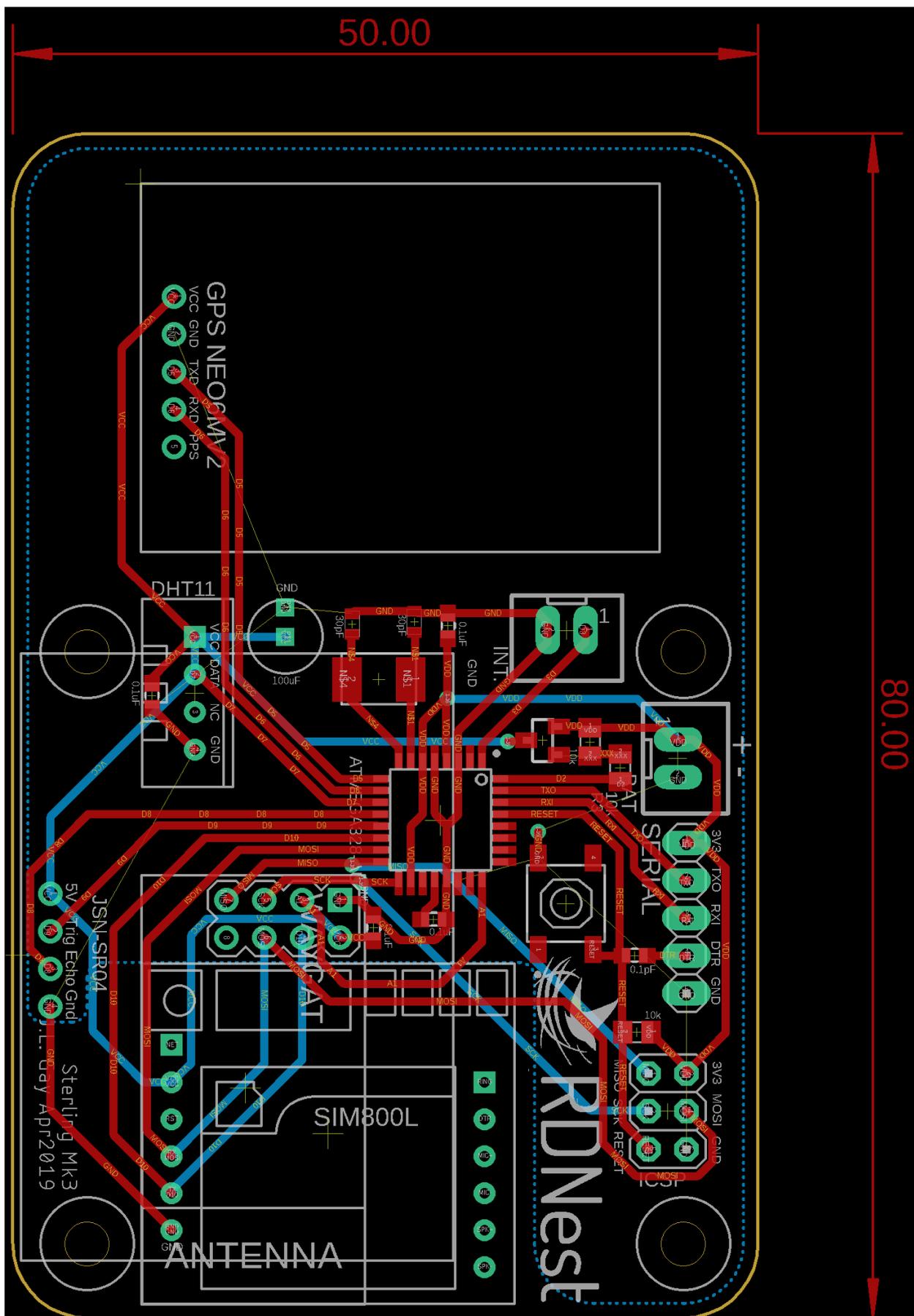


Ilustración 16. Diseño de PCB Sterling Mk.III

Capítulo 4

Problemas encontrados y soluciones asociadas

4.1 Alto consumo

En el diseño de la placa original se había implementado un Arduino Pro Mini con un ATmega328 sobre un zócalo en una placa. Esto implica no sólo un incremento del coste de fabricación, sino también un consumo eléctrico más elevado debido a los componentes añadidos al Arduino además del propio microcontrolador.

En la práctica los sistemas encontrados requerían un cambio de batería aproximadamente cada dos semanas debido sobre todo a los siguientes componentes innecesarios en nuestro circuito:

- **LED integrado del Arduino**
Los Arduinos empleados poseían un LED que se activaba con la alimentación y no era posible desactivar por software. Tras revisar los esquemáticos del circuito se pudo concluir que su presencia no era necesaria para su correcto funcionamiento, así que fue retirado de las placas que estaban ya en el campo de pruebas sin ningún problema.
- **Regulador de 3.3V**
El regulador presentaba el mismo problema: un consumo innecesario, ya que no era requerido para el correcto funcionamiento. También fue retirado sin consecuencias.

Tras un estudio teórico aproximado sobre una placa de uso medio se descubrió que el efecto que tienen estos cambios es de una mejora de la vida de la batería desde dos semanas a un año y medio (una mejora del 3500%). Sin embargo, como era de esperar, debido a otros efectos como las resistencias parásitas de las baterías, los resultados en la práctica no fueron tan positivos.

En los diseños posteriores se pasó a integrar el propio ATmega en la placa sin Arduino, integrando exclusivamente los componentes indispensables. Esto nos ofreció:

- Aumento de la robustez del circuito
- Mayor control sobre la disposición del circuito y el rutado de placa
- Gran reducción de costes.
- Capacidad de producción en gran volumen

4.2 Sensor de ultrasonidos de 5V

Al realizar un estudio inicial de los componentes empleados se pudo observar un claro error de diseño: Siendo el sistema de 3.3V, el módulo de ultrasonidos funcionaba a 5V. Se pueden encontrar pruebas de funcionamiento del módulo en internet bajo una alimentación de 5V frente a una de 3.3V. Se representa, bajo una distancia en la práctica constante, en el eje Y la distancia medida y en el eje X el número de iteración de la prueba.

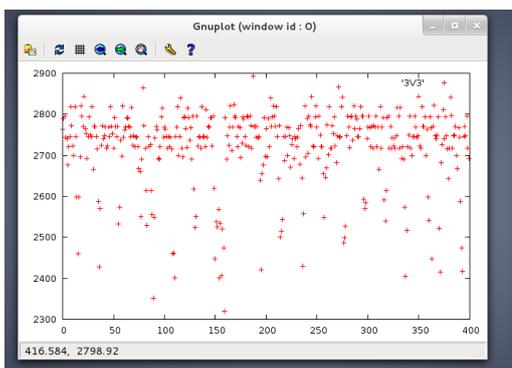


Ilustración 18. Resultados bajo 3.3V de HC-SR04

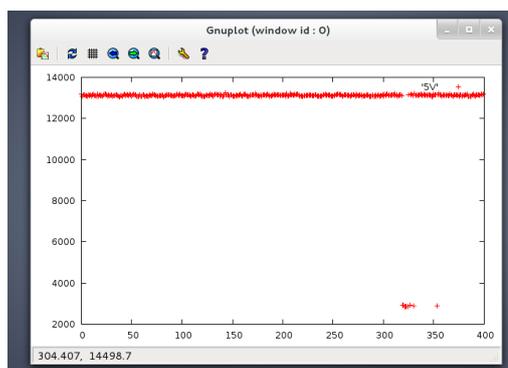


Ilustración 17. Resultados bajo 5V de HC-SR04

[11]

Se puede apreciar que con 5V de alimentación se obtienen unos resultados altamente estables, a diferencia de una alimentación de 3,3V, que genera unos resultados altamente irregulares.

Aumentar el voltaje de funcionamiento del sistema completo a 5V implicaba adaptar otros componentes, y la disponibilidad en mercado de módulos a nuestro voltaje a un precio asequible era inexistente. Por esto se planteó la idea de desarrollar e integrar un circuito propio, pero el coste de desarrollo incrementaba notablemente.

Por lo tanto, la solución más sencilla encontrada estaba basada en simple integrado que aumentase el voltaje de 3.3 a 5V.

Esta solución se puso en práctica y se enfrentó al circuito original, y se probó que otorgando una alimentación previa de al menos 500ms y no dejando que el voltaje de la batería cayese de 3,5V (El máximo de la esta era 5V) los resultados sin el aumento de voltaje eran más que aceptables.

Debido a esto, y que por la planificación del proyecto se iba a instaurar la siguiente fase antes de que la capacidad de las baterías llegase a disminuir el voltaje por debajo de dicho umbral, se consideró no necesario añadir una etapa de subida de voltaje para simplificar desarrollo, montaje, y mantenimiento.

4.3 Condensador de 10 μ F

Desde el primer momento los pines de alimentación de los integrados y módulos no poseían condensadores que estabilizasen la tensión. Dado que por el ahorro de batería algunos de ellos se encendían y apagaban irregularmente debido a los picos de tensión independientemente del regulador, añadir condensadores de estabilización fue necesario.

4.4 Pines de interrupción

En el ATmega328 solamente aceptan interrupciones dos pines. Al realizar uno de los diseños previos del Mark. III no se tuvo en cuenta y esto causó tener una tanda de placas sobre las que hubo que conectar dos pines para suplir el error.

4.5 Caída de voltaje

La alimentación del sistema forma una parte crucial del desarrollo. Al alimentar con baterías es importante tener en cuenta sus rangos de voltaje y que si (en este caso, para disminuir consumo) no se añade un regulador previo que normalice su voltaje habrá cierta caída de voltaje en la batería con el uso que puede causar fallo en componentes no cruciales y el problema pueda pasar desapercibido hasta tiempo después. Esto ya se ha abordado en alguna parte del proyecto.

Se ha visto reflejado con alguno de los sensores, debido a que la placa seguía funcionando y emitiendo correctamente, pero mandando valores erráticos.

Podemos programar el integrado para que al detectar cierta caída del voltaje se apague para evitar este tipo de problemas (Brown-out detection), pero conlleva un mayor consumo de energía que no se desea.

Debido a ello se han tomado varias decisiones de diseño, pero principalmente hemos podido valorar que no sólo cuando una placa deja de emitir, sino además cuando manda valores no válidos, puede tener problemas de batería.

4.6 Alimentación limitada

Una de las cosas que más llamaban la atención al observar el diseño inicial era que todos los sensores, además del módulo Bluetooth, estaban siendo alimentados de un mismo pin del Arduino.

Los datasheets de los componentes alimentados nos dictan que el consumo bajo peor caso de cada uno de los mismos es:

HC-SR04	DHT-11	NRF24L01
15mA	11.3mA	2.5mA

Tabla 5. Consumo de los módulos

Esto nos da un consumo máximo común de 28.8mA.

El regulador del Arduino permite un output máximo de 150mA, pero cada pin individual posee un valor máximo de salida de 40mA. Esto es parcialmente superior al máximo output de cada pin. Por ello, en la teoría no deberíamos tener problemas en este aspecto, pero debido a que los módulos no cumplían correctamente su función se suplió este problema de las siguientes formas:

1. Cable dedicado a la alimentación individual del módulo Bluetooth

Buscando mejorar la capacidad de envío de paquetes del sistema se alimentó el módulo directamente de la batería para poder proporcionar un voltaje más estable, pero el consumo a largo plazo era demasiado pronunciado.

2. Separar la alimentación de los sensores y del módulo Bluetooth

Debido a que en una primera fase se ha de alimentar los sensores para realizar la medida y en una segunda el módulo Bluetooth para realizar la transmisión cabe deducir que no es necesario alimentar en cada una de las fases los elementos de la otra. Por ello, en los prototipos previos cortamos la alimentación del módulo Bluetooth y la conectamos a un pin auxiliar.

3. Implementación de un transistor dedicado a la alimentación.

En el diseño final he optado por la incorporación de un transistor FDN304P que haga de interruptor entre una conexión directa entre la alimentación por batería y los distintos componentes. El consumo se eleva frente a la alimentación por fases previa, pero debido al corto tiempo en el que el sistema está despierto, no supone una pérdida apreciable de vida útil.

4.7 Vías de alimentación estrechas

En el diseño inicial se realizó un ruteo automático de las pistas y con el ancho mínimo que cumpliera los requisitos del programa. Esto puede haber sido causa de alguno de los problemas generados al no permitir una alimentación lo suficientemente estable a todos los componentes.

Se buscó consejo de un ingeniero más experimentado para poder usar unos tamaños más aconsejables, y se acabó optando por un ancho de vía de unos 20 mils para las vías de alimentación y de 12 inicialmente para el resto, acabando por aumentarlo también a gracias a disponer de espacio de sobra.

4.8 Puerto ICSP

A fin de poder facilitar la programación del módulo, fue implementado en el diseño original un puerto ICSP (In-Circuit Serial programming).

El problema con dicho diseño fue que por confusión a la hora de rutado se invirtieron los pines MOSI y MISO, dando problemas a la hora de programar. Esto se corrigió en los posteriores diseños.

4.9 Plano de masa bajo la antena Bluetooth

Desde el principio de desarrollo se implementó un plano de masa para optimizar el rutado y mejorar la estabilidad del voltaje de tierra en los componentes. Esto causó que, al ser situado justo debajo de la antena Bluetooth, disminuyese enormemente el alcance de esta.

En la solución final se ha planteado un plano de masa, pero con una zona sin cobre en el área adyacente a la antena del módulo Bluetooth. Esta solución ha demostrado buenos resultados, afectando perceptiblemente al alcance de la antena.

4.10 Calidad de fabricación en los NRF24L01

La elección de estos módulos Bluetooth se realizó en base a aumentar la rentabilidad del producto y facilitar su desarrollo al ser un módulo muy extendido en el mercado no profesional.

Debido a que no están orientados a un uso industrial, no se pueden encontrar en comercios profesionales. Debido a su elevado volumen de oferta en otras páginas como Aliexpress, nos topamos con que algunos de los módulos pedidos podían ser falsificaciones con una baja fiabilidad.

Cambiando alguno de estos módulos y resoldando el cristal de cuarzo y añadiendo un condensador extra justo en las patas de alimentación en otros en el Mark II pudimos mejorar las capacidades del módulo en términos de potencia.

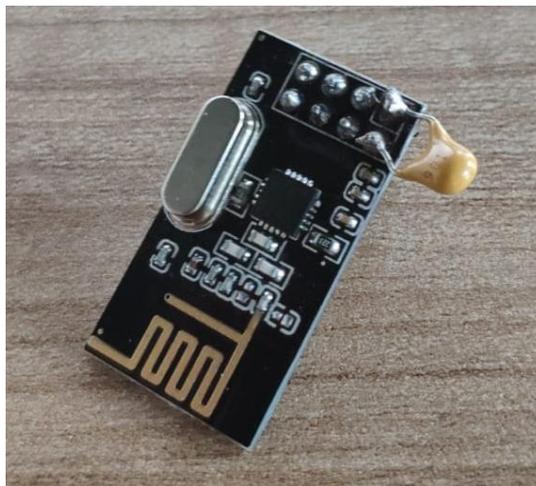


Ilustración 19. Detalle del condensador añadido en el nRF24L01

Gracias a estas pequeñas mejoras hubo un aumento considerable en la potencia de los paquetes enviados (3dBm de media), y con ello, el alcance de envío de los paquetes.

En el Mark II el condensador fue añadido a mano, pero en la siguiente iteración de la PCB se añadió como componente SMD adjunto a los pines como hemos indicado previamente.

4.11 Sobrevoltaje en módulos

Después de varias iteraciones los módulos de radiofrecuencia nRF24L01 fallaban con el uso y el tiempo. No se le dio importancia por los motivos mencionados previamente, pero se acabó observando que el programador empleado (USBasp) ponía el circuito a 5V, voltaje no tolerado por el módulo.

Operating conditions	Minimum	Maximum	Units
Supply voltages			
VDD	-0.3	3.6	V
VSS		0	V
Input voltage			
V _I	-0.3	5.25	V
Output voltage			
V _O	VSS to VDD	VSS to VDD	
Total Power Dissipation			
P _D (T _A =85°C)		60	mW
Temperatures			
Operating Temperature	-40	+85	°C
Storage Temperature	-40	+125	°C

Tabla 6. Parámetros de funcionamiento del nRF24L01 [5]

Podemos observar cómo sus pines de entrada son capaces de manejar voltajes de hasta 5.25V, pero su patilla de alimentación sólo tolera un máximo de 3.6V. Dado que se han sometido dichos módulos a estos voltajes durante periodos no cortos de tiempo, se ha tenido que desechar un alto número de módulos, quedándonos con módulos con rendimientos dispares. Esto se ha solucionado soldando un zócalo para poder retirar los módulos a la hora de programarlos.

4.12 ATmega328 vs ATmega328p

El ámbito comercial está ocupado por el ATmega328p y es el empleado generalmente en los Arduinos. Al comprar los integrados se pidió el ATmega328, un modelo anterior. Éste causaba un mayor consumo y tenía alguna instrucción menos, pero, sobre todo, causaba problemas a la hora de programar. Posee una firma diferente al 328p, por lo que el programador no lo podía reconocer.

Esto retrasó el desarrollo debido a que no es un problema común y costó solucionarlo debido a la falta de experiencia.

Pero exclusivamente fue necesario modificar un archivo de configuración para que interpretase que el 328 que teníamos fuese un 328p.

4.13 Transistor sin resistencia en la puerta

En un primer diseño del transistor no se consideró necesario implementar una resistencia en la conexión entre el transistor y el pin GPIO del integrado.

Esto acabó causando reinicios al sobrecargar el pin debido a que al ponerlo a 0 hacía corto entre la alimentación y tierra momentáneamente, reiniciando el integrado.

Al añadir una resistencia de bajo valor se solucionó. Típicamente se añade una resistencia de valor

$$R_{gate} = \frac{V_{gate}}{I_{max}}$$

Pero en este caso, al no necesitar emplear el transistor en ámbitos de alta frecuencia, una resistencia cualquiera podría valer.

4.14 Módulos no BLE

Debido a fallo de diseño se eligió incorrectamente el módulo pensando que era compatible con el protocolo Bluetooth de baja energía. Sin embargo, alterando las librerías y código, se pueden simular las tramas *advertising* de BLE, aunque con limitaciones de tamaño.

Fue planteada la sustitución de los módulos por otros compatibles con el protocolo BLE, pero no había módulos de tan fácil acceso y el código estaba preparado para este módulo y el cambio de módulo causaría un incremento en el coste desarrollo no deseado. El módulo en cuestión cumplía con las exigencias de diseño así que fue dado por válido en esta etapa de desarrollo.

La emulación del protocolo se consiguió siguiendo los siguientes pasos:

1. Cálculo de CRC

Un emisor BTLE debe ser capaz de realizar cálculos de CRC de 3 bytes, y el NRF24L01 no es capaz de hacerlo. Pese a ello nos permite desactivar el campo dedicado al CRC de una trama Bluetooth convencional, permitiendo al microcontrolador simular este campo, realizar este cálculo y suministrarlo como si fuese parte de la información.

2. Saltos de frecuencia

El protocolo BTLE implica saltos de frecuencia en espacios de tiempo menores de 150 μs , mientras que nuestro módulo no puede mantener tal ritmo. Hemos ignorado dicho requisito, causando muchas dificultades en el envío de los paquetes. Esto está explicado más extensamente a continuación en 5.2.2.

3. Longitud de paquetes

Este módulo, no orientado al protocolo BTLE, puede manejar un tamaño de paquetes limitado a 32 bytes. Esto implica que no podremos establecer conexiones debido a que, como hemos visto anteriormente, los paquetes necesarios para establecer una conexión (CONNECT_REQ) son de 34 bytes de payload, mas 3 de CRC. Pese a no poder establecer conexiones sí podremos emitir paquetes ADV_NONCONN_ID, en los que tras invertir 3 de nuestros 32 bytes en el CRC, 2 en la cabecera que indica el tipo de paquete y 6 en la MAC, nos quedaremos con 21 bytes que podremos emplear en nuestra información (Payload).

4.15 Canales de emisión

Tras realizar numerosas pruebas se obtuvieron resultados muy inestables: Algunos móviles recibían hasta un tercio de los paquetes, otros menos, y otros ninguno.

Esto se veía asociado con bastante seguridad al emisor, así que listando y analizando los conocimientos que tenemos sobre lo que sucede:

- Emitimos saltando entre los 3 canales (o frecuencias) de advertising.
- Se recibe hasta un máximo de 1/3 de los paquetes totales enviados.
- El módulo realiza saltos de frecuencia a un ritmo mucho menor del requerido por el protocolo.
- Si emitimos por un solo canal todos los mensajes, alguno de los móviles puede llegar a recibir hasta más de un tercio de los mensajes.

Con esta información hemos podido deducir que los receptores BTLE de los móviles están limitados a escuchar por uno sólo de los canales, el cual van variando en rangos de tiempo que denominaremos “ventanas”. Esto se ha desarrollado más detalladamente en el apartado 2.5.4.

El protocolo ha de encargarse de enviar los paquetes saltando entre las tres frecuencias posibles lo más rápido posible, esperando poder enviar un paquete por cada una de ellas en cada ventana del receptor para asegurar su recepción. Debido a que nuestro módulo es más lento de lo necesario, nos topamos con que en ocasiones podíamos recibir muchos menos mensajes, o incluso ninguno.

Ejemplo:

Denominando a los tres canales A, B y C y a las ventanas también A, B y C:

En circunstancias normales el receptor abriría una ventana A, y el emisor emitiría un paquete por los canales A, B y C, asegurando una recepción en el destinatario.

En nuestro caso, el receptor abriría una ventana A, el emisor puede emitir por el canal C y para cuando quiera emitir por el A la ventana A estaría cerrada y se abriría la ventana B, imposibilitando la recepción. Esto se puede llegar a repetir durante toda la transmisión.

Este problema se ve reforzado debido a que en la programación inicial se emitía un solo paquete por segundo, en un solo canal, durante 30 segundos (30 paquetes). Para mejorar las posibilidades de que un envío se realizase correctamente se pasó a emitir un paquete cada 30ms durante aproximadamente 9 segundos (unos 300 paquetes en total).

Esto además disminuyó el tiempo que el sistema estaba despierto, aumentando la duración de la batería.

4.16 Impermeabilización del sistema

Se planteó un requisito a mayores para una mejora del sistema: la posibilidad de impermeabilizarlo. El sistema original se encontraba contenido ya en cajas impermeables IP65, por lo que la única preocupación se convirtió en impermeabilizar la interfaz con el exterior: el sensor de ultrasonidos. Tras mucha búsqueda se encontró que la solución más accesible y coste-efectiva era reemplazar el sensor actual (HC-SR04) por un sensor impermeable cuyo uso más común era para sensores de aparcamiento en automóviles. Era la mejor solución coste-resultado, debido a su gran volumen de oferta. Dicho módulo es el JSN-SR04 [12]:



Ilustración 20. Módulo JSN-SR04T

4.17 Medidas y marco, diseño irregular

El diseño original partía de unas placas previas con medidas irregulares: 66.08mm x 55.88mm. En los últimos diseños se optó por un tamaño de 50mm x 50mm y tornillos equidistantes en las esquinas a 40mm.

Esto facilitó enormemente su montaje en la caja IP65, así como un mejor transporte y manipulación. Hay que tener en cuenta que para una mayor versatilidad se pidieron parte de las placas con el formato Mark.III ex, lo cual añade un espacio extra a la izquierda para el módulo GPS. Pero la placa fue diseñada para poder cortar dicha prolongación para volver al formato 50mm x 50mm.

Capítulo 5

Testeo y puesta en producción

5.1 Punto de partida

Debido al bajo volumen de producción en el que se encuentra el producto actualmente se ha decidido implementar la placa en una caja de conexiones prefabricada estanca de 100 x 100 x 50mm, valorada en 4,5€.

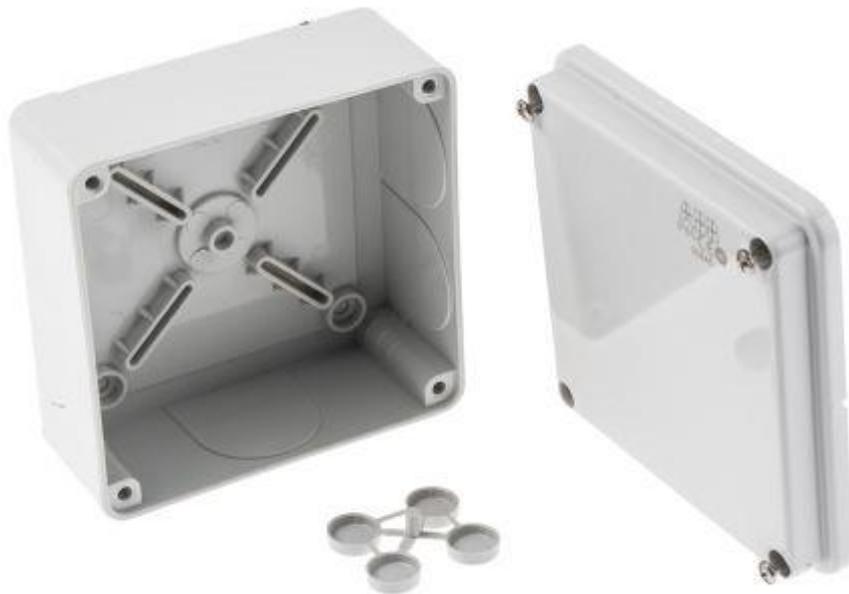


Ilustración 21. Caja de conexiones ABB 00846

5.2 Grabado de placa

El chip permite programación mediante el puerto serie, pero no grabado del bootloader, así que nos vemos obligados a emplear el puerto ICSP que hemos integrado.

Procedemos a desconectar temporalmente el módulo Bluetooth (Ver apartado 4.9) para evitar quemarlo. Tras esto conectamos el puerto ICSP a través de una interfaz adecuada. En nuestro caso emplearemos un dispositivo llamado USBasp debidamente configurado.

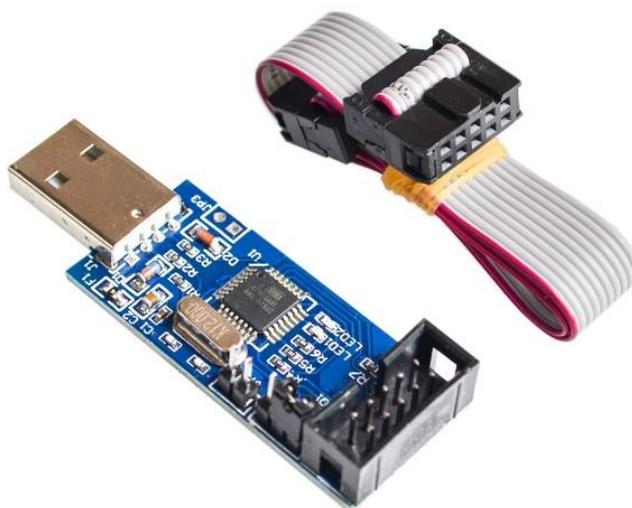


Ilustración 22. Programador ICSP USBasp

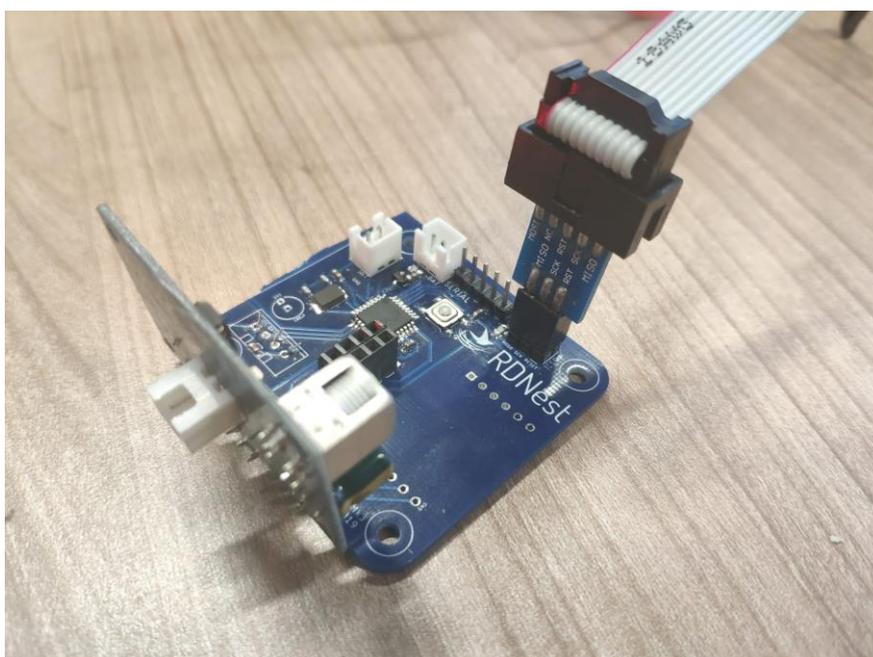


Ilustración 23. Conexión ICSP

Al principio del código nos encontraremos con múltiples variables a definir con las que definiremos tanto cuántos paquetes queremos mandar como qué ID queremos que tenga el contenedor. Tras definir todo esto, en el IDE de Arduino nos cercioramos de indicar que la placa ante la que se encuentra es equivalente a un “Arduino Pro o Pro Mini” y que queremos programarlo para que funcione a 3.3V y 8Mhz.

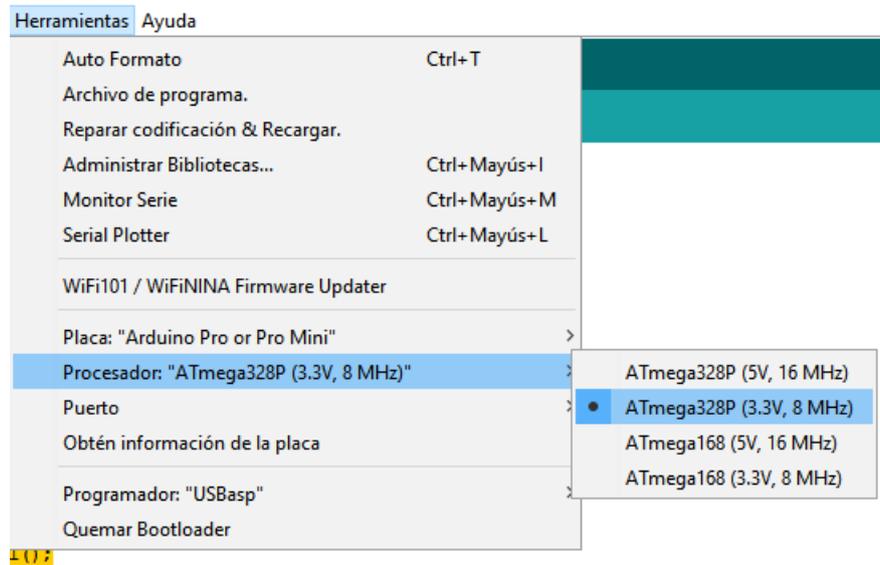


Ilustración 24. Configuración IDE Arduino

Ahora ya podremos pasar a quemar el bootloader (debido a que los integrados vienen sin uno incorporado, a diferencia de los Arduinos) y grabar el firmware. El proceso de grabado de bootloader lo realiza el propio IDE al indicárselo, no requiere más atención. El grabado de firmware requiere que el código esté correcto, especialmente que tengamos las versiones correctas de las librerías.

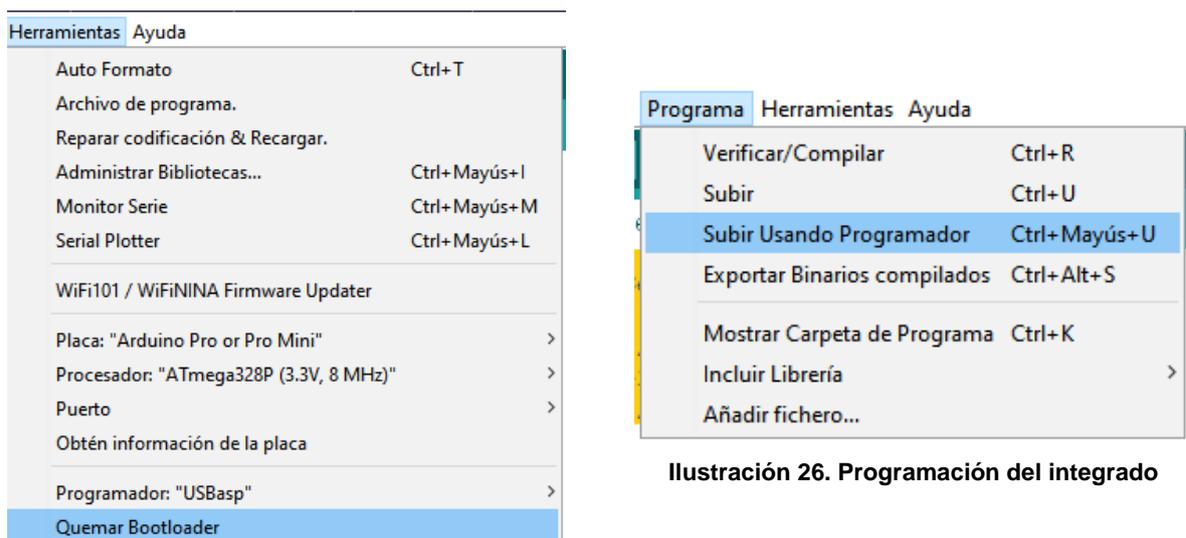


Ilustración 25. Quemado de Bootloader

Ilustración 26. Programación del integrado

Si no se ha indicado lo contrario por el IDE, el programa debería estar correctamente grabado y la placa preparada para funcionar. Reconectamos el módulo bluetooth y pasamos al siguiente paso.

5.1 Preparado de los componentes

Se divide el proceso en dos partes: la caja y la tapa.

5.1.1 Caja



Ilustración 27. pasante para el cable del interruptor Reed

La caja se taladrará por un lateral, por el que se pasará gracias a un pasante aislante el cable que conecta el sensor tipo Reed montado en la tapa del contenedor con el interior de la caja.

Esta caja sin tapa se instalará fija en el interior del contenedor apuntando hacia abajo. La tapa contendrá los componentes permitiendo realizar un mantenimiento de los contenedores sin requerir cambiar la caja completa, simplemente intercambiando la tapa.

5.1.2 Tapa

La tapa ha de albergar el sistema completo. Para ello lo primero es estructurar la posición final del sistema.

Una vez decidida la posición y orientación de los componentes se procede a horadar con un taladro de diámetro 22 un agujero que permitirá el posicionamiento del sensor de ultrasonidos.

Este proceso se realizará siguiendo las medidas de seguridad adecuadas, fijando la tapa y empleando gafas de seguridad.

Tras esto se pasa y fija el transceptor del módulo de ultrasonidos.



Ilustración 28. Taladrado para el módulo de ultrasonidos



Ilustración 29. Tapa con el módulo de ultrasonidos fijado

Tras esto se fija la placa con cinta de doble cara, así como el portabaterías al que previamente se le ha añadido un conector JST.

Después se conecta tanto el transductor del sensor de ultrasonidos al módulo de ultrasonidos como el portabaterías al puerto correspondiente.



Ilustración 30. Tapa montada completamente

5.1 Testeo de la placa

Pasamos a usar una interfaz que nos permita leer el puerto serie de las placas. Existen infinidad de dispositivos en el mercado para ello, nosotros emplearemos un CH340G [13].

Pasamos a conectar los pines TX y GND de la placa con los RX y GND del dispositivo. Abrimos el monitor serie del IDE de Arduino y pulsamos el botón de RESET de la placa.

Dado que disponemos de tiempo de procesamiento no apretado, el código está orientado a facilitar un fácil diagnóstico de posibles problemas.

Esto lo hace gastando tiempo de instrucciones en reflejar por el puerto serie claramente en qué parte del código se encuentra, el estado de los sensores, y el paquete Bluetooth completo generado antes de enviarlo. Además, al comenzar el envío imprimirá por pantalla un punto "." por cada paquete enviado correctamente.

Si todo ha ido bien, nos deberíamos encontrar con lo siguiente:

```
#####
# Setup begins #
# Version:3     #
# Setup ends.  #
#####

-> Loop begins <-

%%%%%%%%%%
%   MODULE STATUS   %
%%%%%%%%%%
% -> JSN-SR04: CORRECT %
% -> DHT11:    CORRECT %
%%%%%%%%%%

*****
RAW: 1B A FFFF 3 19 24 74 5E 1
-> UUID: 1B
-> ID Contenedor: 10
-> Código Sterling: FFFF
-> Version: 3
-> Temperatura:25
-> Humedad: 36
-> Distancia: 116
-> Voltaje: 94
-> Aperturas:1
*****

-COMIENZO DE TRANSMISIÓN-

-PUNTO POR PAQUETE ENVIADO CORRECTAMENTE-

-FIN DE TRANSMISIÓN-

-> Loop ends <-

#####
# Sleeps #
#####
```

Si causamos una interrupción que despierte el sistema, continuará con:

```
!!!!!!!!!!!!!!
!!Interrupt!!
!!!!!!!!!!!!!!
```

Y reiniciará el bucle. Es importante hacerlo esperando al menos 30 segundos y comprobar que incrementa el contador de aperturas

5.2 Errores comunes

Los siguientes apartados implican que se ha realizado un montaje correcto de la placa. Se mostrarán por el monitor serie.

5.2.1 JSN-SR04: ERROR: Too close or transducer not connected

Autodescriptivo, o el transductor está apuntando a algo muy cerca, no está conectado, o es defectuoso.

5.2.2 DHT11: ERROR. Can't get data.

El sensor de humedad y temperatura está defectuoso. Pasa con demasiada regularidad, como se ha podido comprobar gracias a

5.2.3 Los dos anteriores a la vez

Falla la alimentación por parte del transistor FDN304P, o defectuosos. Reemplazar, comprobar conexionado con el integrado.

5.2.4 Sólo aparece un punto como paquete enviado y se cuelga

Módulo Bluetooth desconectado, sin alimentación, o defectuoso.

5.2.5 Contador de aperturas siempre a 1 (Integrado se reinicia)

Si además de estar el contador siempre a 1 con cada despertar emite el texto:

```
#####
# Setup begins #
# Version:3    #
# Setup ends.  #
#####
```

Quiere decir que el integrado se está reiniciando constantemente con cada interrupción. Está diseñado para funcionar igualmente, pero mandará erróneamente el contador de aperturas a 1.

Esto se debe a que la patilla del integrado que controla al transistor hace corto durante un instante a tierra debido al cambio de estado de esta, y al incrementar la corriente que pasa por ella de golpe el integrado se apaga.

Este error se producía a menudo al inicio al no haber calculado correctamente la resistencia que une la puerta del transistor y el integrado, así como la que une puerta y drenador.

Revisar conexionado, si este error persiste, reemplazar transistor, integrado, o resistencias.

5.2.6 El monitor serie muestra texto incoherente

El código está diseñado para enviar por el puerto serie a 38400 baudios. Hay que configurar el monitor para que lea a esa velocidad. Si lee correctamente a otra, es posible que se haya programado el chip para funcionar a 16MHz.

5.3 Puesta en producción

Si todo ha ido bien en este punto tendremos la tapa montada y funcionando, tras lo cual sólo tendremos que cerciorarnos de conectar el conector JST del interruptor tipo Reed que entra en la caja antes de atornillarla a la misma.

5.4 Mantenimiento

A la hora de realizar un seguimiento del contenedor emplearemos la interfaz web.

Dirección	Último timestamp
Facultad de ciencias	2019-12-10 15:39:39
ElInf	2019-12-10 16:51:29
Biblioteca ciencias	2019-12-10 10:44:55
Facultad Economicas	2019-12-10 10:26:04
Desarrollo	2019-12-04 18:21:51

Ilustración 31. Interfaz web mostrando último uso de cada contenedor

Conociendo el último uso de cada contenedor sabremos cuándo falla alguno de ellos así como si necesita mantenimiento cuando transmite una temperatura o humedad de 0, o si el contador se finja a 1. (Ver 5.2)

Última temperatura	Última humedad	Último contador
23	29	37
0	0	20
		1

Ilustración 32. Fallo por DHT11

Ilustración 33. Fallo por FDN304P

Capítulo 6

Fase Nordic: Prototipo Mark IV

6.1 Introducción

La evolución natural del diseño del sistema hizo pasar del prototipo previo que nos encontramos con Arduino y un módulo Bluetooth hobby a algo más profesional: Un integrado con un procesador y capacidades bluetooth.

Tras presentar de qué maneras se puede realizar la implementación bluetooth a nuestro sistema en el capítulo Bluetooth, es natural inclinarse hacia esta, SoC. Esto es debido que requiere un menor nivel de desarrollo al incluir todo el protocolo en un solo integrado, y una mayor facilidad de integración.

Es importante destacar que el trabajo acometido en esta fase no ha dado lugar a un prototipo funcional, sino que se trata de una prueba de concepto que busca comprobar si se pueden evitar todos los problemas detectados en el Mark III a la par que se permite su fabricación en grandes volúmenes. Para ello nos hemos centrado en el diseño de la placa, dejando para una fase posterior (no incluida en este trabajo) el desarrollo software asociado.

Por lo tanto, actualmente se ha llegado a desarrollar de una placa, junto a un código que nos permite simular el comportamiento de un contenedor, y que se ha probado en un kit de desarrollo del chip.

6.2 Elección de componentes

Debido a su gran desarrollo en el ámbito comercial, y su extensa área de soporte y foros, se ha decidido optar por el chip Nordic nRF51822 [14].

Hemos comenzado el desarrollo partiendo de su kit de desarrollo nRF51 SDK. Este incluye una placa con el propio chip integrado junto a todos los componentes necesarios, así como otras muchas características, como el programador necesario, leds, botones, o pasar cada pin del chip a una entrada hembra de pin.

Este kit además servirá para programar las primeras placas prototipo.

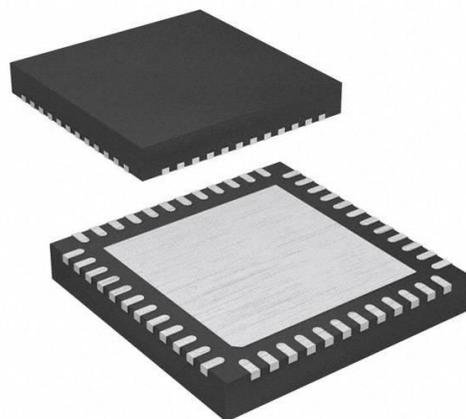


Ilustración 34. nRF51822

6.3 Diseño de PCB

Una vez teniendo programado un prototipo inicial que nos asegure que podremos alcanzar los requisitos de diseño necesarios, se pasa al diseño de PCB.

Para comenzar con el diseño se ha partido de un circuito de referencia proporcionado por Nordic para Altium, que adaptamos a Eagle para realizar nuestro propio diseño.

6.3.1 Alimentación

Existen tres alternativas a la hora de alimentar el chip:

1. LDO (Regulador de baja caída)

El regulador DC/DC interno se desactiva y salta, generando la energía del sistema exclusivamente a partir de la fuente de voltaje externa. Acepta voltajes entre 1.8 y 3.6 voltios.

2. DC/DC

El chip contiene un conversor tipo Buck de voltaje, que baja el voltaje con pérdida mínima de potencia, tras lo que entra en el regulador lineal que alimenta el chip. Aumenta la eficiencia en cuanto a consumo se refiere si el sistema consume corrientes elevadas, como al transmitir por radiofrecuencia. (El LDO es más eficiente ante bajas corrientes). Cuando está activado este modo el sistema es capaz de saltar entre éste y el LDO para aumentar la eficiencia. Acepta voltajes entre 2.1 y 3.6 voltios.

3. Low Voltage

Disponiendo de una fuente de 1.8V se puede usar este modo. Acepta una entrada exclusivamente entre 1.75 y 1.95 voltios.

6.3.2 Adaptación de impedancias

Para que la antena reciba la máxima cantidad de potencia posible han de igualarse la impedancia de entrada vista desde la antena y desde el chip. Esto es necesario para que no se refleje potencia en la antena y permita un correcto funcionamiento de ésta. Para este proceso de adaptación de impedancias se han de añadir componentes tales como baluns, redes Pi o T que modifiquen dicha impedancia.

En nuestro caso hemos podido valernos del circuito de referencia del chip, que ya incluye una red pi para adaptar la impedancia a 50Ω . Es importante tener en cuenta que dicha red pi se ha calculado para una placa de referencia prediseñada, y aquí se ha tenido que modificar la PCB para añadir nuevos componentes. Esto habrá causado un desequilibrio que haga que la impedancia deje de ser 50Ω , pero la pérdida generada no compensa el trabajo necesario para compensarla.

En este caso también puede ser necesario una red que transforme la impedancia de la antena además de la del chip, por lo que se ha añadido a la placa espacio para la posibilidad de añadir estos componentes. [15]

6.3.3 Conexión a la antena

Es necesario que las pistas que llevan la señal a la antena estén adaptadas también a 50Ω . Para ello existen múltiples calculadoras para hallar la anchura de pista necesaria. Hay que tener en cuenta que no vale una calculadora estándar, ya que la pista dejará a cada lado tierra, convirtiéndose en una guía de onda coplanar. Por lo tanto, nos puede valer una calculadora como:

<https://chemandy.com/calculators/coplanar-waveguide-with-ground-calculator.htm>

Nos pedirá la constante dieléctrica relativa (~4.2), el ancho de la pista (que se variará buscando 50Ω en el resultado), la distancia a la tierra (que se regula en el diseño de PCB) y la anchura del dieléctrico (que se indica al fabricante, el valor estándar 1.6mm). Tras todos estos valores se concluye que el ancho de pista que buscaremos será de ~2.5mm.

6.3.4 Osciladores

El chip incluye un oscilador LC integrado y la posibilidad de añadir externamente un oscilador de 16 MHz y otro de 32kHz.

El primero, de 16Mhz, es obligatorio debido a la necesidad de tener una alta precisión en frecuencia para el protocolo BLE.

El segundo, de 32kHz, es opcional y evitarlo ahorraría tanto coste económico como espacio en placa. Pero hay que tener en cuenta que un oscilador RC conlleva una tolerancia peor (de unas 250ppm) que un cristal externo (unas 20ppm). Esto implica un recalibrado cada aproximadamente 4 segundos, lo que conlleva un consumo de unos $10\mu A$ que no queremos.

6.3.5 Antena

Al implementar una antena en nuestro diseño podremos hacerlo de dos formas: mediante un chip, o en la propia PCB. El comportamiento de ambas bajo circunstancias óptimas sería el mismo, pero no tiende a ser el caso. Diseñar una antena en la propia PCB será mucho más complicado, pero ahorrará costes. Implementar una antena chip será más sencillo, pero aumentará costes.

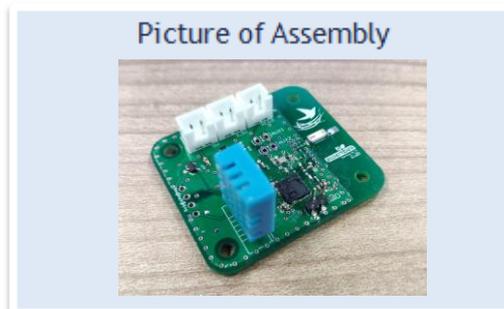
Se ha elegido el uso de una antena en chip debido a la ya elevada complicación del resto del desarrollo a fin de equilibrar la complicación del diseño. La elección, así como su colocación, puede llegar a ser complicada, por lo que los fabricantes tienden a ofrecer amplios manuales y soporte técnico. En este caso se ha elegido una antena de Johanson Technology. Para facilitar la selección de antenas nos presentan una clara guía en:

<https://www.johansontechnology.com/chip-antenna-selection>

Se ha elegido montarla en el borde de la placa para aumentar todo lo posible la fiabilidad de esta. Una vez elegida la 2450AT42A100, en el datasheet se puede ver cómo implementarla en la PCB de forma correcta:

6.4 Bill of Materials simplificado

Assembly Name	Sterling Mk. IV
Approval Date	05-Dec-19
Part Count	38
Total Cost	\$34,16



Part Name	Description	Qty	Codigo Farnell / Amazon	Unit Cost	Cost
NRF51422	Microcontrolador	1	2842312	€ 1,63	1,63 €
Oscilador	Cristal 16MHz	1	3228464	€ 0,61	0,61 €
Oscilador	Cristal 32kHz	1	2101347	€ 0,65	0,65 €
Antena	2450AT42A100	1	2148522	€ 0,80	0,80 €
JSN-SR04T	Sensor ultrasonidos impermeable	1	Link Amazon	€ 6,99	6,99 €
Reed Switch	Sensor de apertura de contenedor Reed	1	Link Amazon	€ 4,99	4,99 €
DHT-11	Sensor Temp&Hum	1	Link Amazon	€ 1,98	1,98 €
Portabaterías	Soporte baterías 18650	1	2674338	€ 2,35	2,35 €
Batería	Batería 18650 3.6V 2,6 Ah	1	Link Amazon	€ 9,99	9,99 €
FDN304P	Transistor alimentación	1	9846298	€ 0,36	0,36 €
C 12pF	Condensador	4	1758943	€ 0,20	0,80 €
C 1nF	Condensador	1	1414573	€ 0,04	0,04 €
C 2,2nF	Condensador	1	1758995	€ 0,01	0,01 €
C 2,2pF	Condensador	1	2672094	€ 0,05	0,05 €
C 1pF	Condensador	1	2896335	€ 0,01	0,01 €
C 1,5pF	Condensador	1	3019111	€ 0,02	0,02 €
C 4,7uF	Condensador	1	2525098	€ 0,01	0,01 €
C 100nF	Condensador	2	2496788	€ 0,01	0,02 €
C 1nF	Condensador	1	1758911	€ 0,01	0,01 €
C 47nF	Condensador	1	3013359	€ 0,01	0,01 €
C 1uF	Condensador	1	9227776	€ 0,03	0,03 €
C 220nF	Condensador	1	2896443	€ 0,02	0,02 €
C 2,2uF	Condensador	2	3013394	€ 0,02	0,04 €
L 4,7nH	Inductor	1	1711735	€ 0,01	0,01 €
L 10nH	Inductor	1	1343081	€ 0,04	0,04 €
L 3,3nH	Inductor	1	1711734	€ 0,02	0,02 €
L 10uH	Inductor	1	2215635	€ 0,13	0,13 €
L 15nH	Inductor	1	2840112	€ 0,04	0,04 €
Pin Header	2.54mm 2x3 Contactos	1	1248159	€ 0,57	0,57 €
Pin Header	2.54mm 1x4 Contactos	1	1022251	€ 0,10	0,10 €
Pin Header	Conector JST para batería y sensor Reed	2	1516276	€ 0,06	0,12 €
PCB	Placa (Cálculo con pedido x30)	1		€ 1,71	1,71 €
Total/Placa		38			34,16 €

Tabla 7. BOM Sterling Mark IV

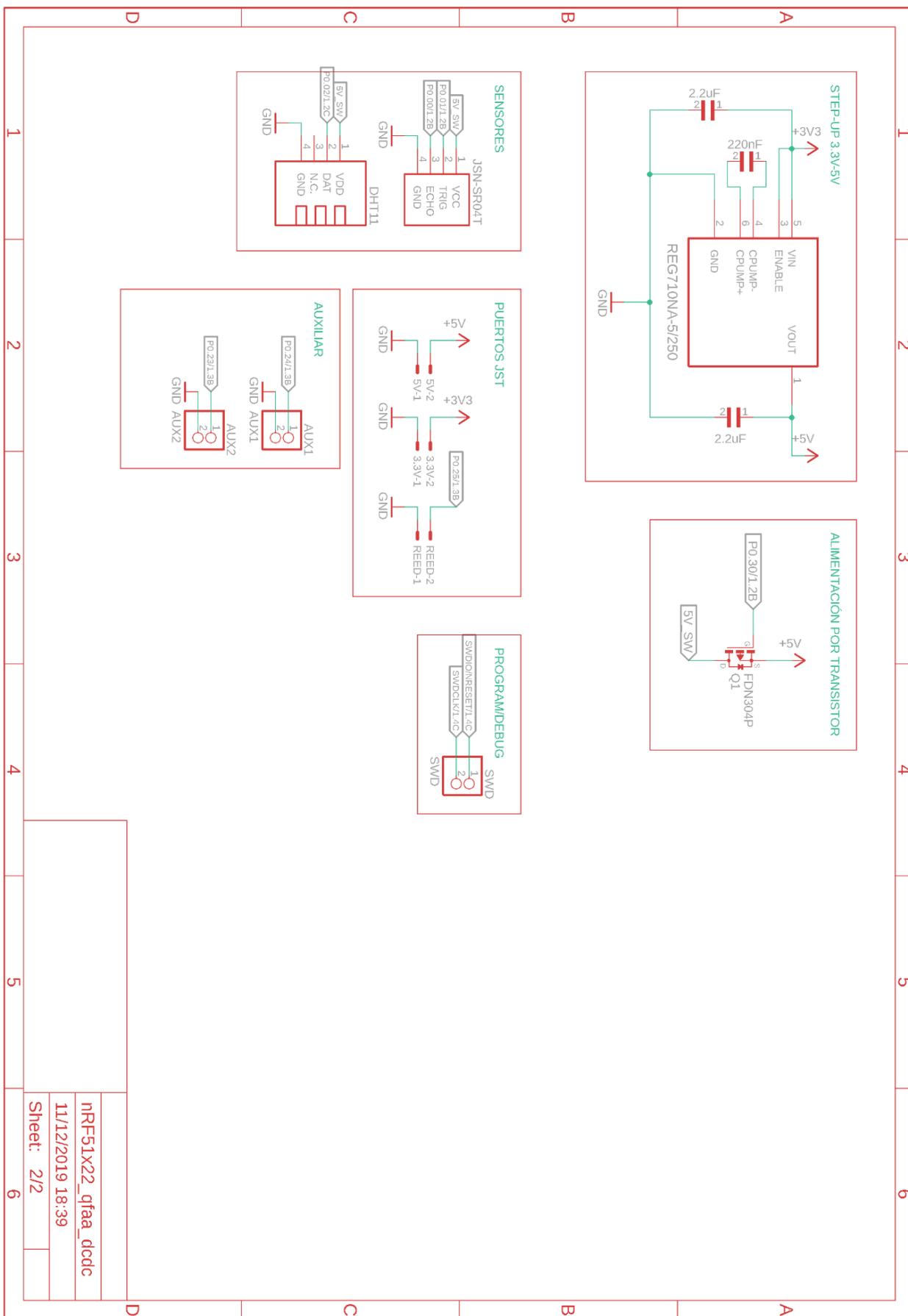


Ilustración 37. Hoja 2 del esquemático del Sterling Mark IV

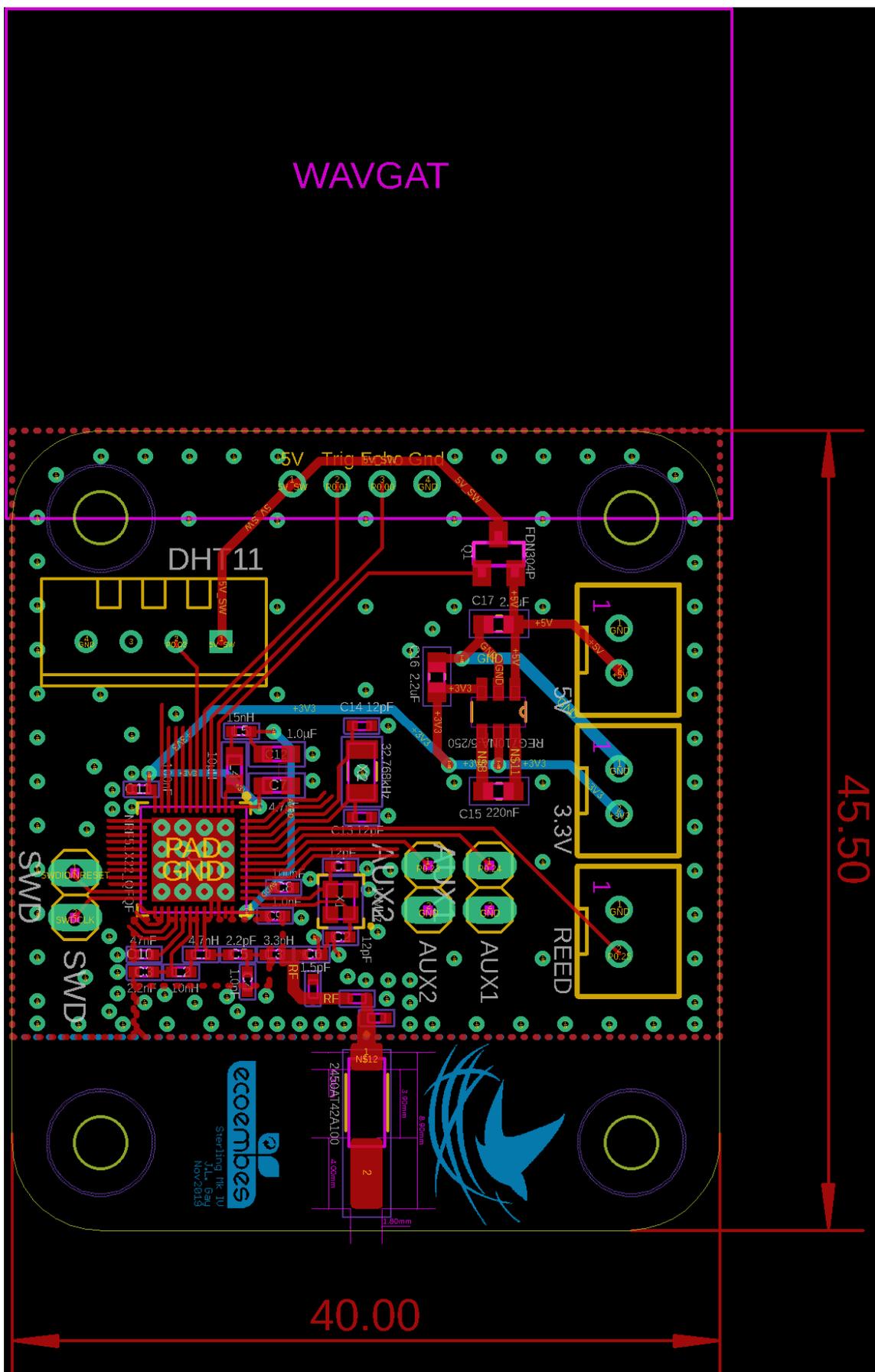


Ilustración 38. Diseño PCB Sterling Mark IV

Capítulo 7

Conclusiones

7.1 Resultados

En el desarrollo de este TFG se ha logrado iterar exitosamente sobre una prueba de concepto hasta lograr un equipo que cumple con los requisitos de diseño planteados y resuelve 9 problemas graves de diseño y expone el sistema a su producción en gran volumen e industrialización.

En la actualidad se encuentra en uso exitosamente el modelo Mark III en 9 contendores de la UVa, y en desarrollo un prototipo funcional que ha probado un mejor rendimiento, sobre el que aún se ha de trabajar para plantear como producto.

Un servidor se encarga de recibir las tramas enviadas por los equipos a través de los teléfonos móviles de los usuarios, y gracias a una interfaz web es posible monitorizar el funcionamiento del sistema de forma no presencial. Esto permite realizar un mantenimiento de las placas mucho más sencillo, así como monitorizar el funcionamiento de las estas.

Gracias a esto se ha podido valorar diversos puntos del proyecto ya abordados como el aumento de la vida útil de la batería de dos semanas a más de seis meses actualmente, u otros que se tratarán en el futuro como la falta de robustez del módulo DHT-11.

7.2 Presupuesto

El proyecto se ha desarrollado en una empresa con las herramientas y equipo previamente disponibles, por lo que no vamos a tener en cuenta ninguno de estos factores ni la amortización de las instalaciones.

Además del tiempo de desarrollo tendremos que añadir tiempo dedicado al mantenimiento, y retirar otros costes previos del proyecto como los contenedores (500€/ud).

Comprenderemos el mantenimiento y el coste de desarrollo en el sueldo de un ingeniero junior, de 18.000€ anuales, que con un 33% de costes sociales asciende a 24.000€. Teniendo en cuenta que el proyecto ha abarcado un marco de unos 6 meses.

$$24.000€ * 6/12 = 12.000€$$

Esto se suma al precio que se estipula en el BOM para el Mark III y unas 30 placas, es decir, 34.13€ (una vez restado el módulo GPS).

$$34.13€ * 30 = 1023.90€$$

Por lo tanto, se puede estimar que el coste por placa hasta ahora será de unos:

$$\frac{12.000\text{€} + 1023.90\text{€}}{30} = 434,13\text{€}$$

El coste por cada placa nueva (sin ensamblar) del Mark III será de 34,13€, como especifica el BOM. A un tiempo estimado de montaje de placa y caja de unas dos horas con un sueldo base a coste de 6,91€ da un coste por placa ideal de:

$$34,13\text{€} + 6,91\text{€} * 2 = 47,95\text{€}$$

Este producto no está planteado para ser vendido como producto final, sino como servicio.

Por ello, sin tener en cuenta costes de desarrollo, para amortizar el coste de una placa, una caja (+5€), y un contenedor (+500€) en un año, y sin tener en cuenta el servidor, tendríamos que ofrecer el coste mínimo ideal por contenedor y mes aproximado a:

$$\frac{555\text{€}}{12 \text{ meses}} = 46,25 \frac{\text{€}}{\text{mes}}$$

7.3 Conclusiones

El trabajo desarrollado en el marco de este proyecto y sus conclusiones pueden resumirse como sigue:

- Se ha partido de un primer prototipo que nos ha ayudado a definir el rumbo del proyecto, así como nos ha facilitado un código funcional que nos permita un salto inicial considerable.
- Se ha realizado un estudio en profundidad del protocolo Bluetooth de baja energía. Se ha explorado extensamente el ámbito de difusión de paquetes desde la capa física hasta la de aplicación.
- Se ha realizado un estudio sobre el prototipo encontrado y una búsqueda sistematizada de errores, así como seguido un proceso en cascada de subsanación de estos y mejora del sistema.
- Se ha sistematizado y documentado el montaje de los equipos, eliminando gran parte de los posibles problemas que se podían presentar y permitiendo un fácil diagnóstico de errores, así como un manual de montaje de estos.
- Hemos comprendido la importancia de controlar la calidad y origen de los componentes, permitiendo conseguir reducir costes a largo plazo, fallos en cascada, y además facilitar el montaje y diagnóstico de los sistemas.
- Al avanzar en el desarrollo del proyecto se ha ido haciendo más notable cada vez la necesidad de mantener un control de versiones, documentarlo correctamente y realizar una correcta gestión de librerías.
- Este control de versiones también ha resultado ser crucial en la gestión de placas para poder manejarlas adecuada y fácilmente.

- La disposición de un campo de pruebas ha probado ser extremadamente útil para mejorar las prestaciones del sistema y descubrir los puntos de fallo. Gracias a ello se han podido buscar formas más eficientes de montaje en caja y mantenimiento que se pondrán en práctica posteriormente.
- El desarrollo de una placa ha demostrado ser radicalmente diferente al de una placa electrónica estándar, requiriendo unos conocimientos más avanzados, así como más tiempo de desarrollo y ajustes finos posteriores que se van a llevar a cabo en el futuro.

7.4 Trabajo futuro

Como trabajo futuro, se pretende completar el desarrollo del Mark IV, tanto en lo que respecta al hardware y el encapsulado como al software correspondiente, con vistas a su futura comercialización y a su posible implantación en grandes volúmenes.

Los mayores retos presentes actualmente son:

- Realizar una lectura correcta de los sensores con el Mark IV
- Programación exitosa de la nueva placa
- Ajuste de la antena en la placa con las redes de adaptación de impedancias
- Reprogramación del firmware vía Bluetooth

Capítulo 8

Bibliografía

- [1] L. Garijo Gutiérrez, «<http://academica-e.unavarra.es/xmlui/bitstream/handle/2454/21877/TFG%20Protocolo%20de%20Enrutamiento%20-%20Luis%20Garijo.pdf>».
- [2] A. C. C. K. T. Robert Davidson, «<https://www.oreilly.com/library/view/getting-started-with/9781491900550/ch01.html>», [En línea].
- [3] «Microchip Developer», [En línea]. Available: <https://microchipdeveloper.com/wireless:ble-introduction>.
- [4] Bluetooth SIG, «Bluetooth AD types», [En línea]. Available: <https://www.bluetooth.com/specifications/assigned-numbers/generic-access-profile/>.
- [5] Nordic Semiconductor, «Datasheet nRF24L01», [En línea]. Available: https://www.sparkfun.com/datasheets/Components/SMD/nRF24L01Pluss_Preliminary_Product_Specification_v1_0.pdf.
- [6] ElecFreaks, «Datasheet HC-SR04», [En línea]. Available: <https://www.electroschematics.com/wp-content/uploads/2013/07/HCSR04-datasheet-version-1.pdf>.
- [7] Mouser, «Datasheet DHT-11», [En línea]. Available: <https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>.
- [8] MakerGuides, «Datasheet JSN-SR04T», [En línea]. Available: <https://www.makerguides.com/wp-content/uploads/2019/02/JSN-SR04T-Datasheet.pdf>.
- [9] COMUS Group, «Datasheet CW1300», [En línea]. Available: <https://www.farnell.com/datasheets/1825428.pdf>.
- [1] SIMCom, «Datasheet SIM800L», [En línea]. Available: https://www.elecrow.com/wiki/images/2/20/SIM800_Series_AT_Command_Manual_V1.09.pdf.
- [1] «Stackexchange», [En línea]. Available: <https://raspberrypi.stackexchange.com/questions/32135/can-i-use-the-3-3v-pin-instead-of-the-5v-pin>.
- [1] MakerGuides, «Datasheet JSN-SR04T», [En línea]. Available: <https://www.makerguides.com/wp-content/uploads/2019/02/JSN-SR04T-Datasheet.pdf>.
- [1] SparkFun, «Datasheet CH340G», [En línea]. Available: <https://cdn.sparkfun.com/datasheets/Dev/Arduino/Other/CH340DS1.PDF>.

8. Bibliografía

- [1] Nordic Semiconductor, «nRF51822 Datasheet,» [En línea]. Available: https://infocenter.nordicsemi.com/pdf/nRF51822_PS_v3.1.pdf.
- [1] «Developer zone - Nordic Semiconductors,» [En línea]. Available: <https://devzone.nordicsemi.com/nordic/short-range-guides/b/hardware-and-layout/posts/general-pcb-design-guidelines-for-nrf51-series>.

