

Universidad de Alcalá

Escuela Politécnica Superior

**Grado en Ingeniería Electrónica y Automática
Industrial**

Trabajo Fin de Grado

Desarrollo de un Sistema Inteligente de Buzones Autónomos
(SIBA) en sistema empotrado de bajo consumo

ESCUELA POLITECNICA
SUPERIOR

Autor: Juan José Pérez Ruiz

Tutor: Marta Marrón Romera

2022

UNIVERSIDAD DE ALCALÁ
ESCUELA POLITÉCNICA SUPERIOR

Trabajo Fin de Grado

**Desarrollo de un Sistema Inteligente de Buzones Autónomos
(SIBA) en sistema empotrado de bajo consumo**

Autor: Juan José Pérez Ruiz

Director: Marta Marrón Romera

Tribunal:

Presidente: Julio Pastor Mendoza

Vocal 1: Carlos Andrés Luna Vázquez

Vocal 2: Marta Marrón Romera

Calificación:

Fecha:

**A todas las personas que me acompañaron
y apoyaron durante esta etapa**

Agradecimientos

”Las cosas no se hacen siguiendo caminos distintos para que no sean iguales, sino para que sean mejores”

Elon Musk 21st

Llegando al final de esta etapa, alcanzando una de las metas más importantes que me he propuesto, me gustaría agradecer el resultado a todas las personas que han hecho posible este logro.

En primer lugar me gustaría agradecer a mis padres toda la confianza que han depositado en mí a lo largo de estos años, así como el apoyo incondicional que me han brindado a pesar de todas las adversidades. A ellos quisiera agradecerles también los esfuerzos realizados para que pueda lograr mi objetivo.

En segundo lugar quisiera acordarme de aquellas personas que con su esfuerzo han contribuido a que pueda hacer realidad mi vocación y que jamás podrán ver el fruto de tanto esfuerzo.

Finalmente, quiero agradecer el resultado a todos los compañeros que han contribuido a alcanzar la meta. A todos aquellos que día tras día a lo largo de estos cuatro años han compartido dudas, soluciones, éxitos y fracasos y cuya ayuda ha sido fundamental para que todos avancemos en la misma dirección. En especial quiero acordarme de mi compañero y amigo Pablo, una de las mejores personas que me ha regalado esta etapa, con la que he trabajado en muchas ocasiones más de lo que nos hubiese gustado a ambos y que ha ayudado a superar cada una de las dificultades a las que me he enfrentado.

En definitiva, quiero agradecerle el resultado obtenido a todas esas personas que con su trabajo han contribuido y siguen contribuyendo a alcanzar mi vocación para que, si logro dedicarme profesionalmente a ella, sea yo el que no tenga que trabajar ni un solo día más.

Autor

Juan José Pérez Ruiz

Resumen

El propósito de este trabajo es desarrollar un sistema capaz de monitorizar el tráfico del correo de los buzones ubicados en España, así como transmitir dicho tráfico de forma inalámbrica.

El sistema monitorizará tanto la entrada de correspondencia en cada buzón como la salida de este. Además, medirá diferentes variables de interés para conocer el estado de la electrónica y poder realizar el mantenimiento oportuno.

La información se procesará mediante un microcontrolador, mientras que las comunicaciones se implementarán mediante una red IoT.

Finalmente, el sistema debe de optimizar al máximo el consumo con un coste reducido.

Keywords: Internet of Things, Ultra bajo consumo, Correo.

Abstract

The purpose of this work is to develop a system capable of monitoring the mail traffic of the mailboxes located in Spain, as well as transmitting said traffic wirelessly.

The system will monitor both the entry of correspondence in each mailbox and its output. In addition, it will measure different variables of interest to know the state of the electronics and be able to carry out the appropriate maintenance.

The information will be processed by a microcontroller, while the communications will be implemented by an IoT network.

Finally, the system must maximize consumption at a reduced cost.

Keywords: Internet of things, Ultra Low Power, Mail.

Índice general

Resumen	ix
Abstract	xi
Índice general	xiii
Índice de figuras	xvii
Índice de tablas	xix
Lista de acrónimos	xxii
1 Introducción	1
1.1 Contexto del trabajo	1
1.2 Definición de objetivos	2
1.3 Plan de trabajo	3
1.4 Organización de la memoria	3
2 Núcleo de procesamiento y comunicaciones	5
2.1 Introducción	5
2.2 <i>Sigfox</i>	5
2.2.1 Flujo de información a través del servicio <i>Sigfox</i> en España	7
2.2.2 Paquetes de datos	8
2.2.3 Seguridad en los mensajes	10
2.2.4 Estructura temporal de los mensajes en <i>Sigfox</i>	10
2.2.4.1 <i>uplink</i>	10
2.2.4.2 <i>downlink</i>	12
2.3 Núcleo de procesamiento	12
2.3.1 Selección del microcontrolador	13
2.3.2 STM32-L432KC	14

3	Diseño del sistema electrónico	15
3.1	Introducción	15
3.2	Estudio de alternativas para la implementación del sistema	17
3.2.1	Análisis de posibles soluciones para la detección de la entrada del correo	17
3.2.1.1	Detección por medio de fotocélulas	17
3.2.1.2	Detección de apertura mediante interruptores magnéticos	18
3.2.1.3	Detección de apertura mediante un sistema mecánico	18
3.2.1.4	Solución final	19
3.2.2	Solución adoptada para la detección de la retirada del correo	19
3.2.3	Monitorización de la temperatura	20
3.2.4	Monitorización de la tensión de alimentación	20
3.3	Sensores y electrónica	20
3.3.1	Introducción	20
3.3.2	Interruptores	20
3.3.3	Módulo de comunicaciones	21
3.3.4	Antena	22
3.4	Esquema eléctrico	22
3.4.1	Introducción	22
3.4.2	Alimentación del microcontrolador	23
3.4.2.1	Alimentación a través del USB-B	23
3.4.2.2	Alimentación a través de los pines <i>V_{in}</i> y <i>GND</i>	24
3.4.2.3	Solución adoptada al problema de consumo	25
3.4.2.4	Posibles fuentes de alimentación	26
3.4.2.5	Conexión de los sensores	28
3.5	Programación del microcontrolador	33
3.5.1	Estructura del código	34
3.5.2	Rutinas de atención a la interrupción	34
3.5.3	Máquina de estados	37
3.5.3.1	Mensajes	41
3.5.3.2	Decodificación del <i>downlink</i>	43
4	Diseño del sistema mecánico	45
4.1	Introducción	45
4.2	Diseño 1: sistema de detección	46
4.2.1	Pieza fija	46
4.2.2	Deslizadera	47
4.2.3	Pestaña	48

4.2.4	Sistema completo. Limitaciones	48
4.3	Diseño 2: caja para la electrónica	50
4.4	Piezas finales	52
5	Resultados	55
5.1	Introducción	55
5.2	<i>Backend</i> de <i>Sigfox</i>	55
5.2.1	Configuración de los nodos	55
5.3	Pruebas y estadísticas	58
5.3.1	Cobertura del sistema	59
5.3.2	Autonomía del sistema	60
5.4	Problemas encontrados	63
6	Conclusiones y trabajos futuros	67
6.1	Conclusiones	67
6.2	Trabajos futuros	67
7	Pliego de condiciones	69
7.1	Elementos físicos	69
7.2	Software	70
8	Presupuesto	71
8.1	Recursos hardware	71
8.2	Recursos de desarrollo y pruebas	71
8.3	Recursos humanos	72
8.4	Presupuesto de ejecución material	72
8.5	Importe de la ejecución por contrata	73
8.6	Honorarios facultativos	73
8.7	Presupuesto Total	73
	Bibliografía	75

Índice de figuras

1.1	Buzón mural	2
2.1	Representación del mayor mensaje <i>Sigfox</i>	9
2.2	Formatos de <i>payload</i> en función de los datos definidos por el usuario	9
2.3	Inicio de transacción <i>Sigfox</i> por parte del dispositivo del usuario, sin sincronización ni escucha previa del canal. Comparativa con un acceso convencional sincronizado por la estación base.	11
2.4	Diversidad, en tiempo y frecuencia, de las tres réplicas de un mensaje <i>uplink</i> en <i>Sigfox</i> . . .	11
2.5	Estructura temporal de una transacción con mensaje <i>downlink</i> en <i>Sigfox</i> [1]	12
2.6	Vista superior de la tarjeta <i>NUCLEO-L432KC</i>	14
3.1	Buzones originales	16
3.2	Esquema del interior del buzón	16
3.3	Esquema del buzón con fotocélula	18
3.4	Esquema del buzón con el sistema mecánico	19
3.5	Sensor magnético y <i>microswitch</i>	21
3.6	Interruptor <i>reed</i> [2]	21
3.7	Módulo DVK-SIP-SFEU-1-GEVK [3]	22
3.8	Antena <i>ANT-PUKDB</i> y ROE [4]	23
3.9	Esquema de alimentación del microcontrolador	25
3.10	Esquema del módulo 0362A	27
3.11	Conexión de los diferentes dispositivos electrónicos	33
3.12	Ejecución del código	35
3.13	Máquina de estados	38
3.14	Máquina de estados del módulo de comunicaciones	41
4.1	Perspectiva de la pieza fija	47
4.2	Vistas de la pieza fija	47
4.3	Perspectiva de la deslizadera	48
4.4	Vistas de la deslizadera	48

4.5	Perspectiva de la pestaña	49
4.6	Perspectiva de la deslizadera	49
4.7	Perspectiva del diseño 1 completo	50
4.8	Perspectiva de la caja y la tapa	51
4.9	Vistas del cuerpo de la caja	51
4.10	Vistas de la tapa	52
4.11	Pieza fija junto con la deslizadera	53
4.12	Sistema de detección de correo completo	53
4.13	Abertura máxima del sistema de detección de correo	54
4.14	Caja diseñada para albergar la electrónica	54
5.1	Tarjeta empleada para conectar el nodo mediante USB	56
5.2	Conexión de los diferentes componentes necesaria para activar el nodo	56
5.3	Pestaña <i>Device</i> en el <i>backend</i> de <i>Sigfox</i>	57
5.4	Extracto del historial de mensajes	58
5.5	Estadísticas en el <i>backend</i> de <i>Sigfox</i>	59
5.6	Configuración del <i>downlink</i> y la decodificación de los mensajes en el <i>backend</i> de <i>Sigfox</i> . .	60
5.7	Cronograma del consumo estimado del sistema	62
5.8	Evolución real de la carga de las baterías	62
5.9	Perspectiva de las piezas partidas. La zona verde indica la zona de unión	63
5.10	Dependencia de la frecuencia con la temperatura	64

Índice de tablas

2.1	<i>Sigfox</i> vs <i>Global System for Mobile communication</i> (GSM)	6
2.2	Marco legal de la banda ISM, en donde se ubica el servicio <i>Sigfox</i> en España	7
2.3	Parámetros generales de las comunicaciones <i>uplink</i> en <i>Sigfox</i> , zona RC1.	7
2.4	Parámetros generales de las comunicaciones <i>downlink</i> en <i>Sigfox</i> , zona RC1.	7
2.5	Duración de un mensaje <i>uplink</i> en la zona RC1, en función del tamaño de los datos de usuario	9
2.6	Microcontroladores seleccionados	13
3.1	Medidas de los diferentes buzones	17
3.2	Consumo de los diferentes componentes	24
3.3	Prioridades de las interrupciones	37
3.4	Mensajes que puede enviar la UART	42
3.5	Codificación del mensaje periódico	42
5.1	Cobertura de <i>Sigfox</i> en España	60
5.2	Consumo de los sensores y del módulo de comunicaciones	61
5.3	Consumo de los mensajes	61
5.4	Estimación de la vida útil del sistema	62
8.1	Recursos hardware usados	71
8.2	Recursos de desarrollo y pruebas	72
8.3	Recursos humanos	72
8.4	Presupuesto de ejecución material	72
8.5	Importe de ejecución por contrata	73
8.6	Importe de los honorarios facultativos	73
8.7	Importe del presupuesto total del proyecto	73

Lista de acrónimos

ABS	<i>Acrylonitrile Butadiene Styrene.</i>
ADC	<i>Analog to Digital Converter.</i>
BCD	<i>Binary-Coded Decimal.</i>
EXTI	<i>External Interrupt.</i>
GPIO	<i>General Purpose Input Output.</i>
GSM	<i>Global System for Mobile communication.</i>
IoT	<i>Internet of Things.</i>
ISM	<i>Industrial, Scientific and Medical.</i>
LBT	<i>Listen Before Talk.</i>
LPWAN	<i>Low Power Wide Area Network.</i>
MAC	<i>Message Authentication Code.</i>
NA	Normalmente Abierto.
NC	Normalmente Cerrado.
NTC	<i>Negative Temperature Coefficient.</i>
PC	<i>Personal Computer.</i>
PCB	<i>Printed Circuit Board.</i>
RC	<i>Radio Configuration zone.</i>
RF	Radio Frecuencia.
ROE	Relación de Onda Estacionaria.
RSSI	<i>Received Signal Strength Indicator.</i>
RTC	<i>Real Time Clock.</i>
SIBA	Sistema Inteligente de Buzones Autónomos.
SMS	<i>Short Message Service.</i>
TFG	Trabajo de Fin de Grado.

UART	<i>Universal Asynchronous Receiver-Transmitter.</i>
UBC	Ultra Bajo Consumo.
UNB	<i>Ultra Narrow Band.</i>
URL	<i>Uniform Resource Locator.</i>
USB	<i>Universal Serial Bus.</i>

Capítulo 1

Introducción

1.1 Contexto del trabajo

Actualmente existe una gran tendencia a interconectar cualquiera de los dispositivos eléctricos o electrónicos que se usan en nuestro día a día con el fin de lograr una mayor comodidad y control. En este contexto, el concepto de redes *Internet of Things* (IoT) juega un gran papel. La rápida evolución de las comunicaciones ha permitido que esta nueva tecnología no se vea como una idea de futuro, sino que se haya convertido en una realidad.

Este tipo de redes se ha extendido notablemente en el ámbito de la domótica, permitiendo interconectar la mayoría de los electrodomésticos que usamos en nuestra vida cotidiana, desde frigorífico hasta alarmas inteligentes, dando lugar al nuevo concepto de *Smart Homes*. Sin embargo, esta nueva tecnología no solo restringe su campo de acción a buscar soluciones en el ámbito de la domótica, sino que se ha extendido a otros muchos sectores como puede ser la monitorización del tráfico y del transporte público en muchas de nuestras ciudades, dando lugar al concepto de *Smart Cities*.

Es en este nuevo concepto donde se engloba el trabajo que se trata de realizar. En el caso que ocupa al proyecto, se incluirá esta nueva tecnología en los buzones de correo postal con el fin de monitorizar su tráfico al igual que cualquier otro de los involucrados en una ciudad, permitiendo optimizar el servicio que la empresa de gestión postal presta a sus usuarios.

La idea de monitorizar el tráfico del correo postal, nace en las poblaciones más pequeñas de España, donde su cantidad y periodicidad es prácticamente inexistente. A pesar de ello, estas pequeñas poblaciones han de poseer un buzón en el que poder depositar sus cartas, y la empresa tiene la obligación legal de entregarlas a su destinatario en un plazo limitado, lo que obliga a la empresa a acudir diariamente a cada una de las ubicaciones donde haya un buzón para poder cumplir su cometido. Dada la densidad de población de alguna de estas ubicaciones, un gran porcentaje de los días no hay cartas en los buzones, lo que hace que el trayecto hasta algunas localidades sea en vano. Es entonces cuando surge la idea de monitorizar el tráfico postal de cada buzón, pudiendo diseñar las rutas de recogida de forma óptima.

Teniendo en cuenta todo lo anterior, para ofrecer una solución al problema se plantea en este proyecto monitorizar tanto la entrada como la salida del correo postal del buzón, así como otras variables que permitan conocer el estado de la electrónica de monitorización, enviando todos los datos recogidos a través de una red *Internet of Things* (IoT), y permitiendo a partir de ellos, optimizar las rutas de recogida, reparto y mantenimiento del correo postal.

1.2 Definición de objetivos

El trabajo en cuestión tiene como principal objetivo monitorizar aquellos buzones cuyo tráfico es reducido y saber cuando es necesario realizar la recogida del correo postal en ellos, para así tratar de optimizar las rutas de recogida.

Todas las características citadas previamente son las que definen a Sistema Inteligente de Buzones Autónomos (SIBA), cuyo desarrollo es el objetivo central de este TFG.

Es por esto que el trabajo se centra en la automatización de los buzones denominados “buzones murales” según la empresa “Correos” [5], los cuales tienen las siguientes características (ver Figura 1.1), con objeto de convertirlos en “buzones inteligentes”.

- DESCRIPCIÓN: Buzón para la admisión de correspondencia ordinaria en zonas rurales o pequeñas localidades. Se coloca fijándose a la pared. Incorpora placas descriptivas del horario de recogida con texto en castellano y en las oficiales de las diferentes comunidades autónomas.
- DIMENSIONES del buzón: Alto 509 mm; Largo 415 mm; Ancho 201 mm; Ancho de la boca 230 mm; Alto de la boca 40 mm; Peso: 7 kg.
- MATERIAL DE FABRICACIÓN: Chapa de hierro al carbono de 1 mm de espesor. Acabado con pintura RAL1021 y RAL5010.



Figura 1.1: Buzón mural

El diseño realizado en el trabajo para automatizarlo deberá poder instalarse en unos 20.000 buzones ya existentes, lo que implica que deberá de ser tan universal como sea posible.

Además, el coste unitario del sistema tendrá un gran impacto a la hora de su desarrollo, por lo que será otro factor importante a considerar.

Por último, para que el sistema constituya una solución realista, su consumo deberá ser tan reducido como sea posible, de tal forma que ofrezca autonomía suficiente (meses o años) como para minimizar el mantenimiento.

Por otro lado, el sistema deberá soportar ligeras modificaciones remotas. Actualmente, cada uno de los buzones lleva asociado un identificador que permite hacer un seguimiento de la gestión asociada a cada buzón concreto. Dado que el mismo sistema inteligente a diseñar en este trabajo debe poder instalarse en cualquiera de los buzones, es importante poder incorporarle el identificador del buzón en el que se ha instalado, de tal forma que si un sistema inteligente cambia de buzón, su identificador pueda ser actualizado de forma remota. Además deberá poder modificarse la hora a la que se desea remitir los

datos del tráfico diario al sistema central de la empresa de servicios (Correos), de tal forma que pueda ajustarse a las necesidades de cada situación.

A continuación, el siguiente objetivo subyace bajo la necesidad de que su instalación y mantenimiento sea sencilla, de forma que no sea necesaria la intervención de personal cualificado para ello. De hecho, la instalación del sistema deberá poder realizarse *in situ*, de tal forma que no sea necesario el traslado del buzón para dotarle de inteligencia.

El siguiente requisito recoge la necesidad de realizar un diseño de hardware distribuido, de modo que tanto el microcontrolador como los sensores y el sistema de comunicación, sean tan independientes como sea posible, de modo que no esté sujeta a la política de un cierto operador o fabricante, que puede ser conveniente cambiar en un futuro.

Finalmente, el último de los objetivos a tener en cuenta en el diseño del sistema empotrado inteligente a incluir en los buzones, es su robustez de tal forma que resulte difícil de vandalizar, contribuyendo a la durabilidad del sistema.

1.3 Plan de trabajo

A continuación, se describen las diferentes tareas necesarias para completar el trabajo propuesto.

- En primer lugar se valorarán las diferentes alternativas en referente a la detección de cartas insertadas en el buzón , evaluando diferentes alternativas, atendiendo al coste asociado a la electrónica necesaria, su consumo y su facilidad de integración en el sistema de buzón inteligente completo.
- En segundo lugar se valorarán las posibles fuentes de alimentación a emplearse , teniendo en cuenta la relación coste - autonomía que ofrecen al sistema.
- En tercer lugar se llevará a cabo el diseño de las piezas a mecanizar para poder ubicar el sistema electrónico diseñado en el buzón, tratando de que la mecanización de éstos sea la menor posible.
- A continuación, se deberán montar los componentes electrónicos elegidos para llevar a cabo las tareas descritas anteriormente de forma eficaz.
- Se constituirá el código encargado de gobernar el comportamiento del sistema, haciendo de intermediario entre los sensores y el módulo de comunicaciones.
- Se deberá después realizar el montaje de los diversos prototipos, confirmando la validez del sistema en diferentes buzones reales.
- Finalmente se probará el sistema en emplazamientos reales, permitiendo conocer datos realistas del consumo, la cobertura de la red *Internet of Things* (IoT) y el funcionamiento general del sistema.

1.4 Organización de la memoria

A lo largo de este trabajo se va a explicar el funcionamiento global del sistema de buzones inteligentes.

En el capítulo 2 se describen las diferencias entre los diferentes tipos de redes de comunicaciones que pueden emplearse para desempeñar la tarea requerida, justificando el motivo por el que se selecciona una red concreta y analizando como se establece la conexión entre el nodo y el operador de la red. Además, este capítulo aborda a su vez la elección del núcleo de procesamiento de datos, pues esta es la pieza clave

de todo el sistema electrónico. Al igual que ocurre con la red, en este capítulo también se abordan las diferencias entre algunas de las tarjetas que podrían emplearse para crear el sistema.

A continuación, en el capítulo 3 se explica el procedimiento seguido para llevar a cabo el diseño hardware y software del sistema electrónico desarrollado en el trabajo. En este capítulo se comienza explicando las diferentes opciones barajadas para construir la estructura del sistema, los diferentes dispositivos electrónicos propuestos, así como su ubicación en el buzón para que el sistema cumpla con los requerimientos de diseño. En este capítulo también se concretan las diferentes variables o magnitudes físicas que deben de ser monitorizadas así como los dispositivos empleados para realizar esta tarea. Por su parte este capítulo incluye también el estudio y diseño del sistema de alimentación del circuito electrónico. Finalmente en el mismo capítulo se aborda el diseño software del sistema microprocesador diseñado.

Posteriormente, en el capítulo 4, se describe el diseño mecánico empleado para dar soporte a toda la electrónica desarrollada, así como para garantizar su buen funcionamiento. En este apartado se describen con detalle las piezas fabricadas, así como sus dimensiones. Por su parte, en este capítulo se hace especial hincapié en la versatilidad de las piezas, comentando los parámetros que permiten ubicarlas en cualquiera de los buzones analizados.

Acto seguido se evalúan los resultados obtenidos en el capítulo 5. Estos resultados concretan el entorno del que dispone el gestor del buzón inteligente para parametrizar el sistema, así como los resultados obtenidos en cuanto a consumo y cobertura del sistema se refiere. Estos datos conforman la base para determinar tanto la autonomía como la materialización de la instalación del sistema, en un amplio número de buzones, ofreciendo, por tanto, una visión global de la viabilidad de el sistema de buzones inteligentes.

Por último este documento finaliza con el capítulo 6 en el que se describen las conclusiones obtenidas a lo largo del desarrollo del sistema , así como los trabajos futuros a los que éste da pie.

Capítulo 2

Núcleo de procesamiento y comunicaciones

2.1 Introducción

El principal objetivo del desarrollo es lograr que el sistema inteligente SIBA, informe de forma remota del estado de cada buzón. En este desarrollo se prevé que el tráfico de datos medio entre los distintos nodos SIBA sea muy reducido, pero la comunicación entre ellos será de gran alcance.

Se trata, por tanto, de una comunicación en la que existen numerosos nodos situados a kilómetros del *Getaway* que deben ser capaces de transmitir un tráfico reducido de datos con un consumo de potencia muy reducido. Para ello lo más indicado es emplear una red *Low Power Wide Area Network* (LPWAN), diseñada específicamente para aplicaciones de este tipo, como las IoT.

Estas redes operan en lo que se conocen como bandas *Industrial, Scientific and Medical* (ISM). Son bandas del espectro de RF reguladas internacionalmente en las que se puede operar sin licencia, siempre y cuando se respeten las restricciones en lo que a potencia de emisión se refiere.

Existen diferentes bandas aplicables a las redes LPWAN, sin embargo la más empleada en Europa es la banda que comprende las frecuencias situadas entre 863 MHz y los 870MHz, por tratarse de la de un radio de acción más amplio con un consumo menor.

Al margen de las especificaciones de comunicación comentadas, es necesario un sistema de procesamiento, que sea capaz de monitorizar, medir y controlar las diferentes variables de estado, de interés en SIBA.

A lo largo de este capítulo se especifican las características que debe tener el procesador a utilizar (en función de los sensores el resto de especificaciones electrónicas impuestas), así como forma de adaptar este sistema de procesamiento con la red de comunicaciones elegida.

2.2 Sigfox

Sigfox es el operador de redes IoT más extendido en España cubriendo con su cobertura la mayor parte del territorio nacional. Este operador ofrece soporte a su red IoT, la cual opera a 868 MHz, incluyendo no solo la red en sí y su mantenimiento, sino también diferentes módulos hardware que pueden conectarse a ella, así como la posibilidad de que sean los usuarios los que adopten su tecnología para implementarla en

dispositivos propios. La mayor restricción que presenta es la limitación en cuanto al número de mensajes diarios como así como en cuanto a la carga de dichos mensajes [6].

Tabla 2.1: *Sigfox vs Global System for Mobile communication (GSM)*

Características	<i>Sigfox</i>	GSM
Densidad de antenas para cubrir una ciudad (1M habitantes)	3	60
Densidad de antenas para cubrir 1000km ² (área rural)	1 - 3	10 – 20
Densidad de nodos por cada estación base	Alta	Baja
Enlace bidireccional	Baja	Media / Alta
<i>Bitrate</i> (kb/s)	0.1	9.6 (<i>Short Message Service (SMS)</i>)
Penetración de la señal en edificios	Alta	Media
Coste del módulo emisor (chip básico)	1€	10€
Coste de la suscripción al servicio (anual + tráfico generado)	5€	30€

Tal y como se resume en la tabla 2.1 [7], en la que se comparan las características de comunicación de *Sigfox* con las de GSM, *Sigfox* ofrece la mejor solución para el sistema que se desea desarrollar en este TFG en la relación prestaciones/coste.

Además, si bien existen más operadores de IoT con características similares a las de *Sigfox*, éste último es el más indicado para los intereses de este trabajo, por la gran infraestructura que ha desplegado en España.

Varios estudios [8] [9] [10] señalan las ventajas de optar por *SigFox* frente a sistemas similares, como LoRa o soluciones NB-IoT ad hoc.

La elección de este operador se justifica, por tanto, si tenemos en cuenta los siguientes aspectos:

- Ventajas:

- Las características de coste, prestaciones, alcance y consumo energético de este servicio son idóneas y, actualmente, mejores que las posibles alternativas.
- Existe un amplio repertorio de módulos de Radio Frecuencia (RF) comerciales, ya homologados para poder conectarse al servicio *Sigfox* de forma inmediata.

- Inconvenientes:

- Es un servicio dependiente de una empresa privada y, por tanto, sujeto a cambios en su política comercial e incluso su desaparición o sustitución por otras tecnologías.
- Los módulos de RF propios están sujetos a procesos de evaluación y homologación para ser autorizados a conectarse a este servicio. Esto fuerza a usar módulos de segundas fuentes o bien homologar nuestros propios módulos RF.
- Las zonas de cobertura del servicio *Sigfox* dependen de la política comercial.

Por todo ello el sistema SIBA desarrollado se hará usando tecnología *Sigfox*, pero deberá permitir el cambio de tecnología en caso de ser necesario, con objeto de no quedar anclado a un distribuidor específico, lo que obliga a realizar el diseño del sistema completo desacoplando la parte de comunicaciones de la de procesamiento, como se explica más adelante en esta memoria.

2.2.1 Flujo de información a través del servicio *Sigfox* en España

El tráfico de información que puede enviarse a través de *Sigfox* está condicionado por la legislación a la que debe ajustarse la propia red de *Sigfox*. Los parámetros a los que debe ajustarse la configuración de RF se resumen en la siguiente tabla 2.2 [11].

Tabla 2.2: Marco legal de la banda ISM, en donde se ubica el servicio *Sigfox* en España

Parámetro o descripción	Valor
Identificación de la zona (<i>Radio Configuration zone</i> (RC)) de <i>Sigfox</i> para la zona europea	RC1
Banda de frecuencias del servicio (MHz)	862 - 876
Potencia de emisión máxima admisible, en antena, para los usuarios	16 dBm
Limitación de ocupación de la banda (porcentaje por hora), por usuario	1 %

Esta normativa establece diferentes regulaciones que afectan al tráfico dentro de la red de *Sigfox* en función del perfil del ocupante de la banda ISM, dependiendo de si el ocupante opera como usuario o como gestor de la red. En el caso del sistema de interés en este TFG, el rol que adopta se enmarca dentro del perfil de usuario, dentro del cual, a su vez, aparecen diferentes restricciones dependiendo de la dirección de la comunicación.

Concretamente, las restricciones del tráfico en ambos sentidos se resumen en las tablas 2.3 y 2.4 para las comunicaciones usuario-gestor (*uplink*) y gestor-usuario (*downlink*) respectivamente [12].

Tabla 2.3: Parámetros generales de las comunicaciones *uplink* en *Sigfox*, zona RC1.

Parámetro o descripción	Valor
Velocidad binaria para el usuario en la zona RC1 (Europa)	100 bits/s
Tamaño de los mensajes de usuario hacia <i>Sigfox</i> (variable)	De 0 a 12 bytes
Tiempo máximo de ocupación de canal por mensaje de usuario	2,08 s
Uso máximo de la banda ISM, por hora	36 segundos
Mensajes por hora máximos, disponibles para el usuario	6
Mensajes por día máximos, disponibles para el usuario	140

Tabla 2.4: Parámetros generales de las comunicaciones *downlink* en *Sigfox*, zona RC1.

Parámetro o descripción	Valor
Velocidad binaria para el usuario en la zona RC1 (Europa)	400 bits/s
Tamaño de los mensajes de usuario hacia <i>Sigfox</i> (valor fijo)	4bytes
Uso máximo de la banda ISM, por hora	360 segundos
Mensajes por día máximos, disponibles para el usuario	4

De cualquier modo, y de forma resumida, es necesario presentar las prestaciones que el servicio *Sigfox* España ofrece [13], y que justifican su uso en este TFG:

- Muy bajo consumo: gracias al uso de la tecnología *Ultra Narrow Band* (UNB) y una estrategia de

acceso al medio *Listen Before Talk* (LBT) no sincronizada con las estaciones base. Esta estrategia minimiza la carga de datos de señalización adicionales, propias de otros protocolos.

- Comunicaciones bidireccionales: aún en las condiciones restrictivas de la banda ISM, se puede intercambiar datos en los dos sentidos:
 - *uplink*: desde el nodo del usuario hasta los servidores *Sigfox*. Limitado a un máximo de 1.680 bytes de información por día.
 - *downlink*: desde los servidores *Sigfox* hasta los nodos del usuario. Limitado a un máximo de 32 bytes de información por día (bastante poco pero suficiente para este desarrollo).
- Muy largo alcance: gracias al uso de tecnología UNB, en las estaciones base *Sigfox* el umbral de recepción de mensajes *uplink* está en -142dBm. En las mejores condiciones, usando el límite superior de potencia de emisión en los nodos de usuario 14dBm (25mW), las pérdidas de enlace admisibles hasta las estaciones base rondan los 163dB, ya incluidas las ganancias de inserción de las antenas, lo cual justifica su gran radio de acción.
- Gran área de cobertura: gracias a su largo alcance, el servicio abarca gran parte del territorio nacional con una densidad de estaciones base menor que otros con similares funciones.

En consecuencia, el perfil del sistema *Sigfox* es idóneo para su aplicación a Sistema Inteligente de Buzones Autónomos (SIBA), pues el tráfico de datos requerido por SIBA, en ambos sentidos, se estima muy inferior a los límites del servicio *Sigfox*:

- Para el envío de información del buzón hacia la nube (estado del buzón y/o alarmas varias), se dispone de 1.680 bytes diarios en sentido *uplink*.
- Para que el buzón reciba información de identificación, parámetros temporales o cualquier otra indicación de interés, se dispone de 32 bytes diarios en sentido *downlink*.

En el apartado siguiente se analizan en detalle los paquetes de datos listados y comentados en los párrafos anteriores.

2.2.2 Paquetes de datos

Los información transmisible en sentido *uplink*, como se resume en la Figura 2.1, puede ir desde los 0 hasta los 12 bytes. Esto constituye el *payload* del mensaje. Además, para facilitar la identificación, seguridad y gestión del mensaje, *Sigfox* añade un cierto número de bytes al mismo denominado (*overhead*), por lo que, en total, la máxima longitud de un mensaje *Sigfox* es de 26 bytes, tal y como se puede apreciar en la Figura2.1.

Los bytes de datos de *payload* son totalmente accesibles para el usuario: no hay restricciones ni caracteres reservados, por lo que el usuario es libre de codificar la información como desee, teniendo en cuenta las siguientes condicionantes:

- El mensaje del usuario puede contener cualquier número de bytes, desde 0 hasta 12, codificados en binario/hexadecimal, con cualquier contenido. Nótese que:
 - El gasto energético depende del número de bytes transmitido: a menor número de bytes, menor consumo energético del nodo emisor.

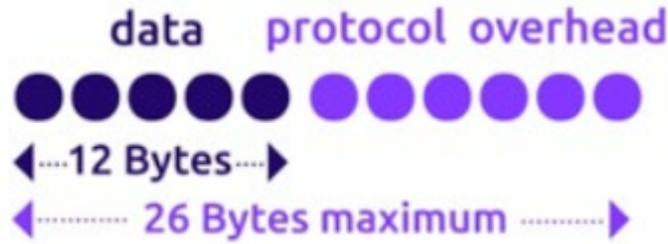


Figura 2.1: Representación del mayor mensaje *Sigfox*

- Incluso 0 bytes de *payload* puede “transmitir” información, pues la mera existencia de un mensaje implica una indicación del tipo “nodo activo”.
- En todo caso, cada mensaje *Sigfox* responde a un conjunto predefinido de bytes de *payload*, correspondiente a longitudes de 0, 1, 4, 8 y 12 bytes. El conjunto correspondiente se determina redondeando por arriba el número de bytes definidos por el mensaje del usuario. Estos bytes “de relleno” (*padding bytes*) son completados automáticamente en el proceso de emisión de *Sigfox*.

La Figura 2.2 aclara el concepto de los bytes de relleno (*padding bytes*). Por ejemplo, si deseáramos transmitir 9 bytes de información, el emisor *Sigfox* añadiría tres *padding bytes* para llegar a los 12 bytes correspondientes al siguiente tamaño superior a 9 establecido para el *payload*.

Bytes of data	0	1	2	3	4	5	6	7	8	9	10	11	12
Payload size in bytes	0	1	4		8				12				

Figura 2.2: Formatos de *payload* en función de los datos definidos por el usuario

Incluyendo los bytes de *overhead* del protocolo, y teniendo en cuenta la velocidad de datos en la zona RC1 (100bps, la más baja de la cobertura *Sigfox*), la duración de la transmisión de un mensaje *uplink* tiene cinco posibles valores, en función del tamaño del *payload* ajustado. En la tabla 2.5 se resume esta información para cada uno de los *payload*.

Tabla 2.5: Duración de un mensaje *uplink* en la zona RC1, en función del tamaño de los datos de usuario

Datos de usuario	<i>payload</i> ajustado (bytes)	Tiempo de emisión (segundos)
Inferior a 1 bit	0	1,1
De 1 bit a 1 byte	1	1,2
De 2 a 4 bytes	4	1,45
De 5 a 8 bytes	8	1,75
De 9 a 12 bytes	12	2

Es importante tener en cuenta los datos mostrados en la Tabla 2.5, a la hora de definir y/o especificar los datos de usuario en el sistema, pues, como se ha explicado, determinarán el tiempo de su transmisión. Nótese además, que un mayor tiempo de transmisión conlleva un mayor gasto energético.

2.2.3 Seguridad en los mensajes

Como ya se ha explicado, los 12 bytes de datos máximos de transmisión en la red *Sigfox* son de carga útil real, pues es en los bytes de *overhead* en los que el protocolo transmite el ID único del dispositivo físico *Sigfox*, que permiten identificar es dispositivo emisor en la nube de recepción (en el servidor *backend* de *Sigfox*). El ID del dispositivo consta de 4 bytes (32 bits) y es único para cada dispositivo emisor homologado [14].

Adicionalmente, a la hora de procesar los mensajes *Sigfox* realiza una serie de comprobaciones previas sobre los mismos[14], principalmente:

- Verificación *Message Authentication Code* (MAC). Cada dispositivo dispone de una llave simétrica de autenticación que se le otorga a la hora de fabricar el dispositivo, y cada mensaje que se envía contiene una parte cifrada basada en esa clave. De esta forma el *backend*, conociendo el ID del dispositivo emisor, es capaz de descifrar esa parte de verificación del mensaje utilizando la clave simétrica, y así comprobar la autenticidad del dispositivo que lo ha emitido.
- Número de secuencia (*sequence number*). Este número ayuda a evitar la duplicación de mensajes. Es un contador entero de 9 bits (va, por tanto, de 0 a 4095), que se incrementa en una unidad, automáticamente, con cada mensaje transmitido desde el nodo emisor. El *backend* de *Sigfox* comprueba este número cuando se valida un mensaje. Si este número no es consistente, el *backend* reacciona de diversas maneras, en función de la severidad de la discrepancias (ver [14]). Igualmente, el número de secuencia depende de capacidad del nodo emisor de encriptar o no la información del *payload* [14].

2.2.4 Estructura temporal de los mensajes en *Sigfox*

Los mensajes *uplink* son iniciados por el dispositivo del usuario y son retransmitidos por una o varias estaciones base hacia los servidores *backend*. Por otro lado, los mensajes *downlink* son enviados por el servidor *backend* de la red hacia un único dispositivo final a través de una única estación base.

Ambos procesos son muy diferentes e implican una secuencia temporal diferente y determinada por cada caso. Esto es así por la concepción de muy bajo consumo para los nodos de usuario del servicio, pues Los nodos de usuario *Sigfox* están, por defecto, en estado de bajo consumo (dormidos) y no monitorizan el canal RF.

En los siguientes subapartados, se describen, por tanto, los procesos de *uplink* y *downlink*.

2.2.4.1 *uplink*

En el caso del *uplink*, se emplea la técnica mostrada en la Figura 2.3. A la izquierda de ésta se muestra el inicio de este tipo de transacción en *Sigfox*. Comparado con un acceso convencional (derecha, en la misma Figura 2.3), existe menos ocupación temporal del canal debido a la ausencia del proceso de sincronización con una estación base determinada. Además, el tiempo de ocupación del canal puede ser inferior del mostrado a la derecha, en caso de no necesitar de respuesta alguna desde la estación base.

La estrategia de acceso al canal del emisor *Sigfox* se conoce como “escuchar antes de hablar” (*Listen Before Talk*, LBT). Esto es: antes de seleccionar una frecuencia de emisión dentro de su banda, analiza durante un tiempo dado si hay alguna otra emisión en esa misma frecuencia, y si la frecuencia está libre, emite su mensaje en ella. No obstante, esto no garantiza, por sí solo, que el mensaje llegue correctamente a alguna estación base *Sigfox*.

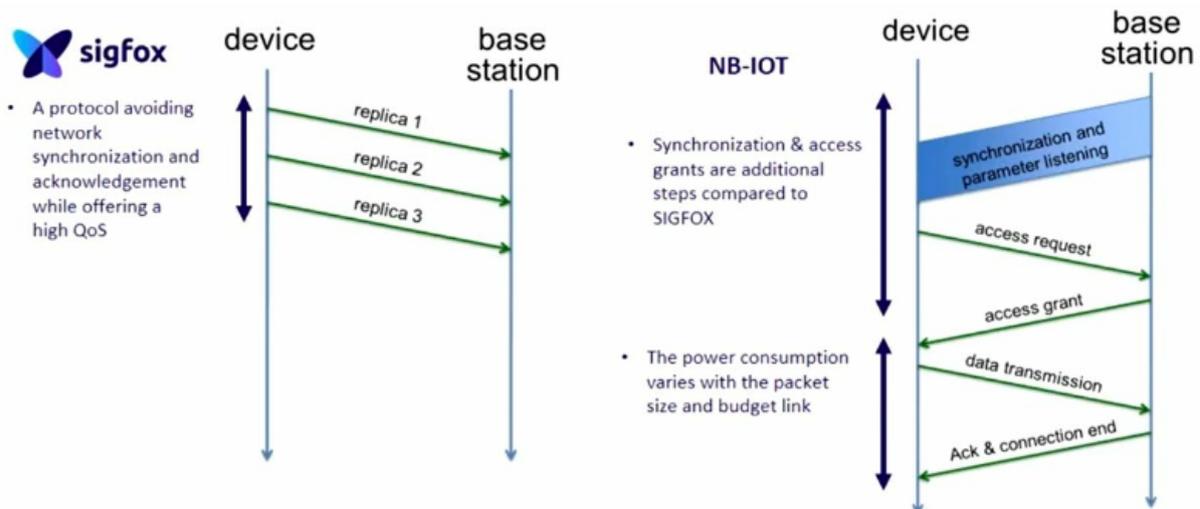


Figura 2.3: Inicio de transacción *Sigfox* por parte del dispositivo del usuario, sin sincronización ni escucha previa del canal. Comparativa con un acceso convencional sincronizado por la estación base.

Como no existe confirmación por parte de la estación base por defecto (sí, bajo petición expresa), *Sigfox* emplea una técnica de redundancia conocida como “diversidad en tiempo y frecuencia”. La Figura 2.4 ilustra esta técnica: una primera réplica del mensaje se transmite en la *Frame1* a una frecuencia $F1$, siguiendo a ella otras dos réplicas del mismo mensaje (*Frame2* y *Frame3*), transmitida en tiempos consecuentes (t_2 y t_3) en frecuencias $F2$ y $F3$, todas diferentes entre sí. Se pretende así garantizar la entrega del mensaje, evitando posibles interferencias en una o varias de dichas frecuencias.

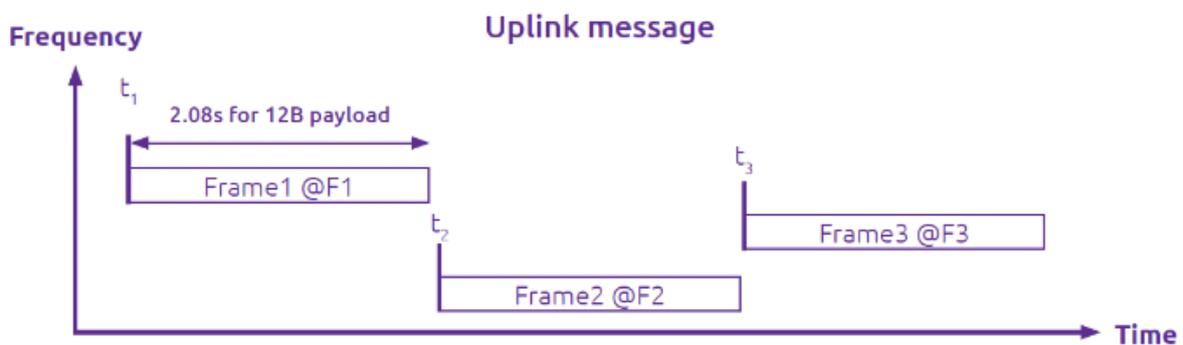


Figura 2.4: Diversidad, en tiempo y frecuencia, de las tres réplicas de un mensaje *uplink* en *Sigfox*.

Esta redundancia y “diversidad en tiempo y frecuencia”, facilita además que los mensajes puedan ser recibidos a la vez por varias estaciones base, lo que contribuye a disminuir la probabilidad de perder mensajes, y es uno de los factores clave detrás de la calidad de servicio en la red *Sigfox*.

Un detalle, no menor, es que se trata siempre del mismo mensaje, lo que implica que el número de secuencia es el mismo en todas las réplicas. Esto permite a *Sigfox* descartar las réplicas no necesarias. Sin embargo, esta estrategia hace que el tiempo de ocupación del canal sea tres veces mayor al de un mensaje simple, ocupando alrededor de 6 segundos por mensaje de 12 bytes. Por esta razón, el número de mensajes por hora estaba limitado a los 6 indicados en la Tabla 2.3: 1% de $3600s = 36s$, a 6 segundos por mensaje, resulta el límite de 6 mensajes/hora dado.

2.2.4.2 *downlink*

El envío de un mensaje en sentido servidor-usuario (*downlink*), siempre es iniciado por el dispositivo del usuario. La secuencia es la siguiente:

- El dispositivo del usuario solicita un mensaje de vuelta (*downlink*) en la emisión de un mensaje de ida (*uplink*). Esto se realiza activando el *downlink request flag* en el emisor (no incluido en el *payload* del mensaje).
- El servidor *Sigfox* solicita al usuario la información a enviar en el *payload* del mensaje de vuelta. El formato admitido en este caso es siempre de ocho bytes.
- Dentro de una ventana temporal y frecuencial prefijada ([14]), la red *Sigfox* envía el mensaje *downlink* especificado al nodo del usuario.
- Si procede, el nodo del usuario confirma la recepción apropiada del mensaje *downlink*, finalizándose así la transacción.

La Figura 2.5 ilustra la secuencia temporal descrita. En ella se observa la existencia de una ventana de recepción fija (*downlink Frame*), en la que el nodo del usuario espera recibir siempre el mensaje *downlink*.

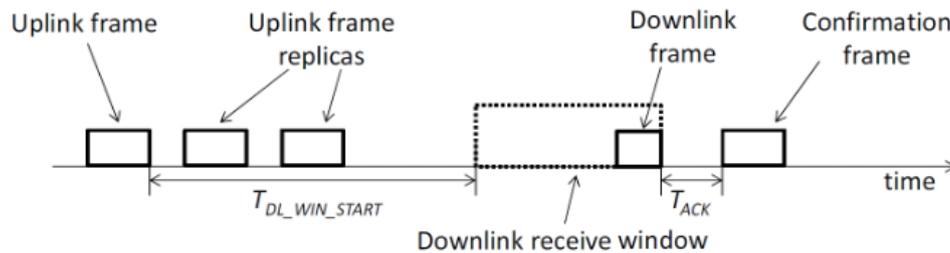


Figura 2.5: Estructura temporal de una transacción con mensaje *downlink* en *Sigfox* [1]

En la zona RC1 (Europa) los parámetros temporales que definen dicha ventana son:

- *TDL_WIN_START*: 20 segundos. Es el tiempo entre el fin de la primera réplica y el comienzo de la ventana de recepción.
- *TWIN_RX*: 25 segundos. Duración de la ventana de recepción. El mensaje *downlink* debe recibirse durante este periodo.
- *TACK*: 1,4 segundos. Tiempo de retardo para el envío de la confirmación de recepción del mensaje *downlink* a la estación base.

Todas las características expuestas validan el servicio *Sigfox* para la aplicación de interés en este TFG, siendo necesario realizar, en los siguientes apartados, un análisis equivalente de la otra parte que constituirá el módulo SIBA: el núcleo de procesamiento.

2.3 Núcleo de procesamiento

Para seleccionar el microcontrolador han de considerarse diferentes aspectos del sistema a implementar. En primer lugar uno de los aspectos más importantes es la capacidad del procesador para ingresar en modo

Ultra Bajo Consumo (UBC). Este aspecto es esencial, ya que de él depende la autonomía del sistema. En segundo lugar otro de los aspectos importantes a tener en cuenta es el número de interrupciones externas de las que dispone así como la capacidad de dichas interrupciones para hacer que el procesador retorne del modo UBC, pues estas interrupciones serán las encargadas de informar al procesador de cualquier evento externo que afecte al desarrollo del sistema. Finalmente otro de los aspectos más importantes es el coste del controlador, ya que dicho elemento supone una de las contribuciones más importantes al coste total del sistema.

2.3.1 Selección del microcontrolador

Dada la actual oferta de microcontroladores, resulta sencillo encontrar múltiples soluciones que pueden desempeñar la tarea requerida en este TFG. Para el proceso de selección se optó por elegir un dispositivo de la familia STM32, ya que se trata de una marca con una gran oferta y stock, y muy extendida en la industria. Tras analizar las diferentes posibilidades que ofrece esta marca, se seleccionaron tres dispositivos, los cuales se resumen en la Tabla 2.6.

Tabla 2.6: Microcontroladores seleccionados

Microcontrolador	Características	Precio (€)
B-L475E-IOT01A1	Diseñada para operar con redes IoT Gran cantidad de sensores Amplia conectividad	27
STM32F103C8T6	Coste y tamaño reducido reducido Necesidad de ST-Link externo	12
STM32-L432KC	Coste y tamaño reducidos Gran variedad de modos ubc Sensor de temperatura	13

Analizando las diferentes opciones resumidas en la Tabla 2.6, se ha determinado que la tarjeta que mejor se adapta a las necesidades del sistema es la STM32-L432KC. Esto es debido a que la tarjeta B-L475E-IOT01A1, a pesar de estar diseñada específicamente para aplicaciones IoT, incorpora una gran cantidad de sensores que no serán utilizados, y que implicarán un coste y consumo innecesario, por lo que es preferible seleccionar una tarjeta más sencilla que abarate los costes de SIBA.

En lo que respecta a la tarjeta STM32F103C8T6, cabe destacar que es una de las tarjetas más famosas de la familia STM y se adapta perfectamente a las necesidades del sistema. Además, esta tarjeta logra abaratar los costes al no incorporar el ST-Link en su circuito. Sin embargo, actualmente esta tarjeta es difícil de encontrar en grandes cantidades, lo que puede comprometer la expansión del sistema diseñado.

Finalmente la tarjeta STM32-L432KC ofrece unas prestaciones muy similares a la tarjeta STM32F103C8T6. Además el procesador que incorpora ofrece una gran variedad de modos de bajo consumo, lo cual es idóneo para optimizar al máximo la autonomía del sistema SIBA. Otra de las ventajas de esta tarjeta, es que incorpora un sensor de temperatura integrado en el circuito, el cual puede emplearse para detectar incidencias en la electrónica.

Para elegir, por tanto, esta última, es necesario contemplar en detalle sus funcionalidades concretas, de modo que sean viables para las necesarias en SIBA. En el siguiente apartado se analizan estos detalles, si bien, a lo largo del capítulo 3 se describe su programación y funcionalidad concreta.

2.3.2 STM32-L432KC

En esta sección se detallan, por tanto, de forma más precisa las prestaciones del microcontrolador elegido. El procesador que incluye esta tarjeta pertenece a la familia STM32, en su versión de bajo consumo. Para su incorporación a SIBA, se eligió el módulo *NUCLEO-L432KC*, que incluye el procesador *STM32L432KC-U6U*, en un formato *Printed Circuit Board* (PCB) compatible con la especificación de *Arduino Nano*, tal y como se puede observar en la Figura 2.6.

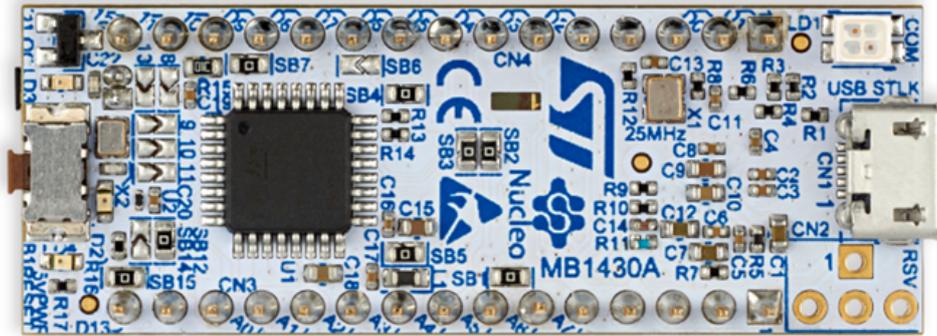


Figura 2.6: Vista superior de la tarjeta *NUCLEO-L432KC*

Esta tarjeta incluye el microcontrolador, el sensor de temperatura, los reguladores de tensión necesarios para su correcta alimentación así como la circuitería necesaria para depurar aplicaciones por ordenador (a través del enlace *ST-Link*), la cual incluye su propio procesador y sus propios reguladores.

Las características más importantes que han sido determinantes para elegir esta tarjeta en el desarrollo de este TFG se listan a continuación:

- Modo de funcionamiento de bajo consumo. El *STM32L432* tiene varios modos de este tipo, incluyendo modo UBC y modo *Stop*, en el que el procesador drena apenas 280nA.
- Activación de la tarjeta de vuelta del modo UBC mediante líneas externas. Esto es esencial para poder conectar los sensores de “reactivación.”^a estas líneas, de modo que la activación de los sensores provoque, a su vez, la del procesamiento.
- Retorno del bajo consumo mediante una interrupción interna de gran temporización (*Real Time Clock* (RTC)).
- Gran variedad de periféricos de bajo consumo.
- Amplio rango de tensiones de funcionamiento, compatible con el uso de pilas o baterías (3,6V - 1,71V).
- Hasta 6 interrupciones externas capaces de despertar al procesador.
- Sensor de temperatura y *Analog to Digital Converter* (ADC) incorporados.
- Tamaño reducido y coste bajo.

Capítulo 3

Diseño del sistema electrónico

En este capítulo se abordarán las diferentes variables que se desean monitorizar, así como las diferentes soluciones aportadas y la solución definitiva que permite conocer el estado de cada variable de entorno.

3.1 Introducción

En primer lugar debe conocerse las variables que resultan de interés, de tal forma que puedan determinarse los sensores que permitan conocer su estado. Como es evidente una de las variables que se desean conocer es la apertura de la pestaña que permite ingresar el correo en el interior del buzón. En segundo lugar, la siguiente variable que se debe monitorizar es la apertura de la puerta del buzón, permitiendo conocer cuando el correo ha sido retirado.

Las variables anteriores se centran en determinar el tráfico del correo, sin embargo, también es necesario conocer las variables que informan del estado de la electrónica. En concreto se monitorizarán la temperatura y la tensión de alimentación del sistema. La temperatura permitirá determinar si alguno de los componentes electrónicos puede destruirse bien como consecuencia de algún cortocircuito, o bien como consecuencia de la temperatura ambiente. Según datos proporcionados por Correos, algunos buzones han llegado a alcanzar los 70°C, lo cual puede suponer un riesgo para el sistema.

Por otro lado la tensión de alimentación proporcionará la información relativa al estado de la fuente, ya que esta será la que sufra un mayor desgaste durante la vida útil del sistema. La evolución de dicha tensión será la que permita anticiparse a los fallos futuros, ofreciendo la oportunidad de realizar el mantenimiento antes de que el cada buzón deje de funcionar.

Para poder implementar el sistema, se cuenta con 4 buzones de tipo "buzón mural". Estos buzones constituyen una muestra representativa de los buzones emplazados en la actualidad, ya que a lo largo de los años el fabricante de los mismo ha ido cambiando y por tanto los buzones no son exactamente iguales aunque a simple vista puedan parecerlo.

Según se aprecia en la figura 3.1 los buzones no son exactamente iguales. Si atendemos a la altura de cada buzón pueden apreciarse ligeras diferencias, aunque la más importante es la variabilidad en el tamaño de las aberturas a través de las cuales se depositan las cartas en el interior, ya que al situarse en la parte superior de la puerta, las pequeñas variaciones de tamaño pueden tener grandes consecuencias en lo que a la ubicación de los sensores se refiere.

Además, si prestamos atención al interior de cada buzón, podremos encontrar más diferencias. En la siguiente figura se muestra la vista frontal de cada buzón con la puerta abierta, así como un corte trans-



Figura 3.1: Buzones originales

versal del buzón visto desde la parte trasera con la puerta cerrada. En dicha figura aparecen representadas las medidas que se han tomado para el diseño del soporte de los sensores.

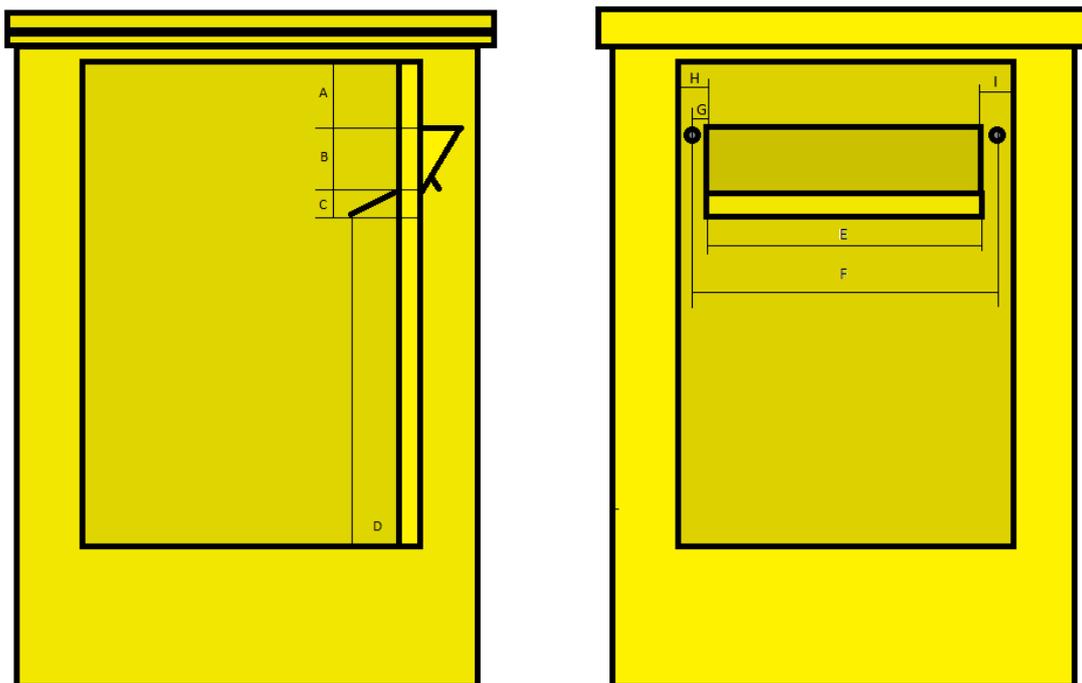


Figura 3.2: Esquema del interior del buzón

Tal y como se resume en la siguiente tabla las medidas reflejadas en la figura 3.2 no coinciden para ninguno de los buzones. A pesar de esto, las variaciones son muy pequeñas, por lo que el sistema se diseñará de tal forma que pueda acomodarse en cualquiera de los buzones.

Las medidas más críticas son la que definen los espacios que quedan alrededor de la abertura, ya que son estas medidas las que delimitan el espacio del que se dispone para ubicar los sensores

Tabla 3.1: Medidas de los diferentes buzones

Medidas(mm)	A	B	C	D	E	F	G	H	I
Buzón 1	11	30	5	30	223	258	15	48	52
Buzón 2	12	51	7	25	220	272	10	43	48
Buzón 3	15	31	24	45	215	264	21	45	58
Buzón 4	9	38	14	17	205	245	20	45	49

3.2 Estudio de alternativas para la implementación del sistema

3.2.1 Análisis de posibles soluciones para la detección de la entrada del correo

El objetivo del diseño consiste en obtener un sistema que pueda adaptarse a cada uno de los buzones, de tal forma que un único diseño pueda amoldarse a las diferentes geometrías. Para lograr la detección de las cartas se sopesaron las siguientes alternativas:

1. Detección por medio de fotocélulas
2. Detección de apertura mediante interruptores magnéticos
3. Detección de apertura mediante un sistema mecánico

A continuación se analizarán las ventajas y los inconvenientes de cada una de estas alternativas.

3.2.1.1 Detección por medio de fotocélulas

Dentro de las fotocélulas existen diversos tipos. El primero de ellos lo constituyen las llamadas fotocélulas de barrera. Este sistema consisten en dos dispositivos, receptor y emisor que se ubican de tal forma que permanezcan enfrentados. Estos dispositivos constituyen la opción más fiable a la hora de detectar objetos siempre y cuando los objetos que se deseen detectar sean opacos, ya que son muy robustos frente a las condiciones del entorno. Sin embargo este tipo de dispositivos obligan a tener necesariamente dos dispositivos, los cuales han de ser alimentados por separado.

El siguiente tipo de fotocélulas se denominan fotocélulas de reflexión difusa. Las fotocélulas de reflexión difusa consisten en un dispositivo capaz de enviar un haz de luz cuando se activan, de tal forma que son capaces de detectar si hay algún objeto próximo a ellas midiendo la cantidad de luz que refleja el objeto. En este caso el emisor y el receptor del haz de luz se sitúan en un mismo encapsulado, lo que hace simplifica la tarea de su alimentación.

Finalmente el último tipo lo constituyen las fotocélulas reflexivas con reflector. Este tipo de fotocélulas se asemeja mucho las fotocélulas de reflexión difusa, con la diferencia de que necesitan un reflector que devuelva el haz de luz al receptor. La principal ventaja con respecto a las fotocélulas de reflexión difusa es que la distancia de detección se ve notablemente incrementada.

Para lograr detectar el correo de forma eficiente, la fotocélula deberá situarse en la parte superior del buzón de tal forma que el haz de luz incida sobre el fondo de este. Conviene situarla lo más cerca posible de la puerta sin que se vea detectada la solapa interior, de tal forma que cuando se introduzca una carta o un paquete, se corte el haz de luz.

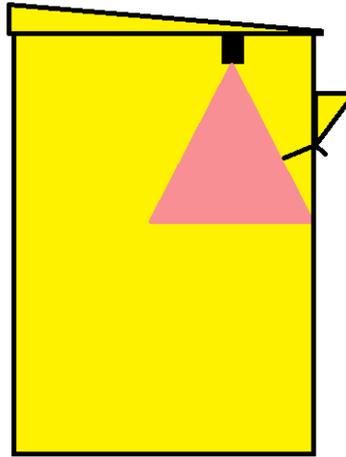


Figura 3.3: Esquema del buzón con fotocélula

Dentro de las diferentes soluciones que ofrecen las fotocélulas, las fotocélulas de barrera son las más precisas tal y como se ha comentado anteriormente, sin embargo obligan a tener dos dispositivos, lo cual no solo complica el diseño, sino que también incrementa el coste y el consumo. Por otra parte las fotocélulas reflexivas con reflector solucionan este problema, ya que solo sería necesario un dispositivo. Sin embargo estas fotocélulas, al igual que las fotocélulas de barrera obligan a mecanizar el buzón tanto en la zona superior como en el fondo del mismo. Además, dadas las dimensiones del buzón, la distancia de detección no tiene por qué ser demasiado amplia, lo que hace que las fotocélulas auto-reflexivas sean la mejor solución.

La solución de las fotocélulas, es una de las soluciones más fiables y prácticas, además de poder instalarse en su totalidad en el interior del buzón lo cual contribuye a incrementar la robustez del sistema. Sin embargo, este tipo de dispositivos tienen dos grandes inconvenientes. En primer lugar tienen un consumo excesivo, así como un coste elevado. En segundo lugar es necesario mecanizar el buzón para poder ubicarlas.

3.2.1.2 Detección de apertura mediante interruptores magnéticos

La siguiente solución la constituye un sistema implementado mediante un interruptor magnético, de tal forma que cuando la pestaña que cubre la abertura por la que se deposita el correo active y desactive el interruptor. Este mecanismo es sumamente sencillo, con la ventaja añadida de que el coste es muy inferior al de las fotocélulas, además de tener un consumo muy reducido. Sin embargo la gran desventaja de este mecanismo radica en que la pestaña que cubre la abertura puede quedarse abierta en algunos de los buzones, de tal forma que la detección del correo puede no ser eficaz. Además el hecho de que el imán encargado de abrir y cerrar en el interruptor sea accesible desde el exterior contribuye negativamente a la robustez del sistema.

3.2.1.3 Detección de apertura mediante un sistema mecánico

En este caso se propone el desarrollo de otra pestaña similar a la que incorpora el propio buzón situada en el interior de dicho buzón. La ventaja que ofrece esta pestaña es que puede ser ajustada más fácilmente, de tal forma que tenga holgura suficiente como para no quedarse abierta. Además esta nueva pestaña logra que el sistema quede completamente oculto desde el exterior, de tal forma que incrementa la robustez del sistema. Por otra parte, esta pestaña puede ser ajustada de tal forma que se amolde a cualquiera

de los buzones independientemente de su geometría, ofreciendo a su vez una gran versatilidad en lo que a la ubicación del interruptor se refiere. Gracias a este dispositivo es mucho más sencillo sustituir el interruptor magnético de la solución anterior por un pulsador convencional, el cual tiene un coste aún más reducido. Otra de las grandes ventajas que presenta este sistema es que puede ser instalado con los propios tornillos que soportan la pestaña exterior, de tal forma que no es necesario mecanizar los buzones existentes.

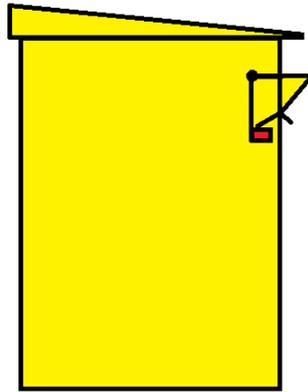


Figura 3.4: Esquema del buzón con el sistema mecánico

3.2.1.4 Solución final

Teniendo en cuenta todas las soluciones aportadas, se ha llegado a la conclusión de que la opción más fiable es la implementación del sistema mecánico. Esta es la única solución capaz de satisfacer los requerimientos de versatilidad, coste, consumo, robustez y simplicidad en la instalación.

3.2.2 Solución adoptada para la detección de la retirada del correo

En este caso la detección de apertura y cierre de la puerta es mucho más sencilla que la detección de entrada del correo, ya que con un simple *switch* podemos detectar el estado de la puerta. Tal y como ocurría a la hora de detectar el tráfico de correo entrante, esta sencilla solución permite reducir el coste y el consumo.

Por su parte en este caso la robustez del sistema está garantizada, ya que las posibilidades para su ubicación son mucho más amplias, pues puede ubicarse en numerosos lugares inaccesibles desde el exterior del buzón.

En este caso se plantean dos posibles soluciones, la primera de ellas consiste en emplear un *microswitch* convencional que pueda activarse con el movimiento del bulón de la cerradura, mientras que la otra solución consiste en emplear un *switch* magnético que se active con el propio movimiento de la puerta.

Dadas las diferentes geometrías de cada buzón la opción del *switch* convencional se descarta rápidamente, pues no todos los buzones disponen de espacio suficiente para ubicarlo con comodidad. Además esta solución tiene un gran inconveniente, ya que el sistema de cierre no es igual en todos los buzones, lo que dificulta el diseño.

El *switch* magnético ofrece la posibilidad de ubicarlo de forma sencilla en casi cualquier punto del perímetro de la puerta, lo que contribuye a facilitar su instalación.

Barajando las opciones, puede concluirse que la mejor solución es el empleo de un interruptor magnético.

3.2.3 Monitorización de la temperatura

La temperatura será la encargada de proporcionar datos sobre la integridad del sistema. No solo sobre la temperatura que puede alcanzar el interior del buzón como consecuencia de las condiciones climáticas, sino que también puede ofrecernos datos de posibles fallos en los circuitos. Es por esto que el sensor empleado deberá situarse lo más cerca posible de la electrónica. Dado que hoy en día existe una gran variedad de microcontroladores que incorporan un sensor en sus circuitos impresos, la mejor opción es seleccionar un microcontrolador que lo incorpore.

Además el hecho de emplear un sensor de temperatura que esté integrado en la tarjeta implica que podrá ser configurado de forma sencilla con los modos de bajo consumo que incorpore la misma, simplificando la tarea de minimizar el consumo del sensor.

3.2.4 Monitorización de la tensión de alimentación

Para monitorizar la tensión de alimentación no es necesario emplear ningún sensor adicional, basta con que el microcontrolador empleado disponga de un conversor analógico-digital, de tal forma que la tensión pueda ser monitorizada y procesada con el fin de poder transmitirla a través de la red IoT.

3.3 Sensores y electrónica

3.3.1 Introducción

En esta sección se abordarán los aspectos relativos a la selección de los diferentes sensores. Una vez hemos concretado la solución que va a desarrollarse, el siguiente paso es elegir los componentes que permitirán construir el sistema.

3.3.2 Interruptores

Tal y como se ha analizado en la sección anterior, para implementar la detección de la apertura y cierre tanto de la puerta como de la pestaña, son necesarios un *microswitch* y un interruptor magnético.

Dado que ambos interruptores se ubicarán en la puerta, la línea que los conecte con el microcontrolador deberá de poder doblarse, permitiendo realizar el juego de la bisagra. Con el paso del tiempo, esta línea puede llegar a deteriorarse, impidiendo que los cierres y aperturas de ambas puertas (puerta y pestaña) sean detectados. Para solucionar este problema cada uno de los interruptores se conectara al núcleo de procesamiento mediante dos líneas, de tal forma que una de ellas ofrezca el nivel lógico complementado de la otra. Esto permitirá detectar la rotura de alguna de las líneas, en cuyo caso deberá realizarse una labor de mantenimiento.

Sabiendo que el nivel lógico de ambas líneas ha de ser complementario, los interruptores empleados deberán tener dos terminales, uno de ellos normalmente abierto y otro normalmente cerrado.

Los interruptores que se muestran en la figura 3.5 serán los que se empleen en el sistema. Tal y como indican las especificaciones ambos tienen los terminales necesarios (Normalmente Abierto (NA) y Normalmente Cerrado (NC)).

Figura 3.5: Sensor magnético y *microswitch*

Cabe destacar que el *switch* convencional incorpora una palanca con una longitud considerable, lo que ayudará activarlo con una fuerza menor, contribuyendo a hacer el sistema más sensible a la apertura y el cierre de la pestaña.

Ambos interruptores serán conectados al microcontrolador mediante resistencias de *pull-up*, las cuales serán las encargadas de disminuir el consumo de ambos dispositivos.

El funcionamiento de estos interruptores se explica de forma más detallada en la sección 3.4.2.5.

Es preciso señalar que el interruptor magnético está compuesto por un interruptor *reed*. Este interruptor consta de 3 pletinas encapsuladas en vidrio. Una de las pletinas está ubicada de tal forma que su extremo se sitúa entre las otras dos permaneciendo en contacto con una de ellas en el estado de reposo. Al aplicar un campo magnético, la pletina central deja de estar en contacto con la pletina anterior para conectar con la tercera pletina, permitiendo así conmutar al circuito. Cuando el campo cesa, se regresa al estado de reposo.

Figura 3.6: Interruptor *reed* [2]

3.3.3 Módulo de comunicaciones

Otra de las especificaciones de Correos indica que es importante que el módulo de comunicaciones sea independiente del microcontrolador empleado, de tal forma que si los servicios ofrecidos por el operador de la red IoT dejan de ser suficientes pueda intercambiarse el módulo de comunicaciones sin necesidad de reponer el microcontrolador al completo.

Siguiendo las recomendaciones de *Sigfox* el módulo empleado es el que puede verse en la figura 3.7.

Este módulo es el encargado de establecer la comunicación entre el nodo (usuario) y la nube de *Sigfox* (gestor). Este módulo es capaz de emitir hasta una potencia máxima de 14 dBm. Con una antena



Figura 3.7: Módulo DVK-SIP-SFEU-1-GEVK [3]

omnidireccional con una ganancia típica de 2dBi, se llega al citado límite legal de 16dBm radiados.

Este módulo ha sido seleccionado debido a las características que se resumen a continuación.

- Controlable por comandos *AT*, vía serie. Esto permite una fácil adaptación del procesador SIBA en caso de cambios en el sistema inalámbrico.
- Dispone de modos de Ultra Bajo Consumo (UBC). En este modo, el módulo drena solo unos 180 nA de la batería, lo que prolonga la duración de las pilas y la autonomía del buzón.
- Amplio margen de tensiones de alimentación. El módulo puede funcionar con alimentaciones que van de los 3,6V hasta los 2,1V, lo que le hace idóneo para ser alimentado con solo dos pilas alcalinas AA, de 1,5V.
- Homologado por *Sigfox* para conectarse a su servicio. Esto ahorra los costes derivados de realizar un nuevo diseño y someterlo a los procesos de homologación exigidos.
- Precio del chip ajustado. En caso de producción en serie, el coste del dispositivo entra dentro de lo admisible.

3.3.4 Antena

La antena empleada ha de ser compatible con la banda ISM establecida en España. Además la antena empleada debe de ser resistente tanto a las inclemencias climáticas como a las posibles actos vandálicos que pueda sufrir. Es por esto por lo que se ha decidido emplear la antena *ANT-PUKDB*, de *RF-solutions*, la cual tiene un diámetro de 55mm y unos 24 mm de altura con respecto a la superficie a la que se adhiere. Esta antena puede fijarse a cualquier superficie gracias a un tornillo de 12 mm de diámetro a través del cual se conecta el cable coaxial. Además, incorpora una junta de goma que permita conservar la estanqueidad del buzón. Por su parte para que la fijación sea lo mas eficiente posible se añade una arandela *growe* que impide que la tuerca que fija la antena en su lugar se afloje con facilidad.

Esta antena se trata de una antena bi-banda, pudiendo operar a 434 o 868 MHz, tal y como puede verse en la Relación de Onda Estacionaria (ROE), con una ganancia de hasta 2dBi que permiten alcanzar la potencia máxima de emisión permitida en esta banda (14 dBm del módulo de comunicaciones sumados a los 2 dBi de la antena completan los 16 dBm máximos permitidos).

3.4 Esquema eléctrico

3.4.1 Introducción

En esta sección se abordaran las posibles soluciones que permiten alimentar al sistema.

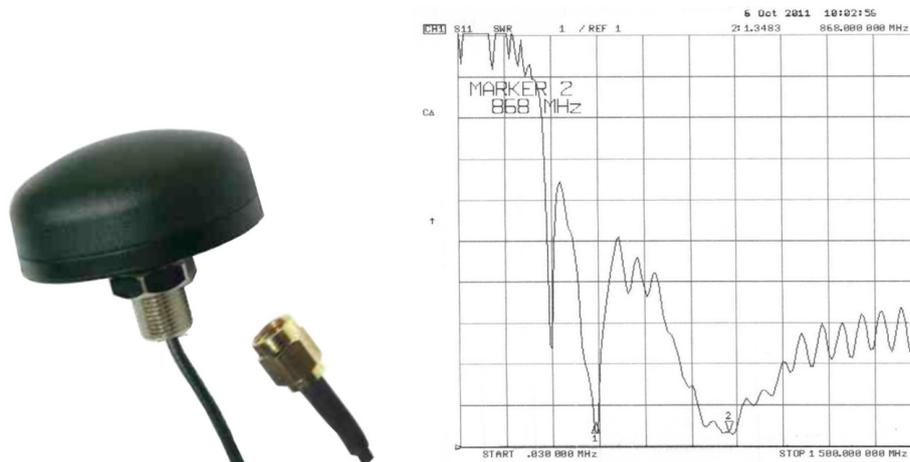


Figura 3.8: Antena *ANT-PUKDB* y ROE [4]

Para resolver esta cuestión se han planteado dos posibles soluciones, una de ellas consiste en alimentar el sistema a partir de baterías recargables, las cuales se recargarían a través de una célula solar mientras que la otra solución pasa por alimentar el sistema a partir de baterías no recargables.

A lo largo del capítulo se desarrollará cada una de las soluciones, comparando las prestaciones de ambas y ofreciendo una solución final al problema del que se parte.

Además en esta sección se abordan las conexiones del microcontrolador con los sensores externos, así como con el módulo de comunicaciones, justificando el uso de cada periférico. En este segundo punto de la sección se abordarán las conexiones de patillas hacia afuera desde el punto de vista del microcontrolador, tratando a este como un modelo de caja negra.

3.4.2 Alimentación del microcontrolador

Después de haber seleccionado el microcontrolador que va a emplearse, se han de analizar las posibles vías que ofrece para poder ser alimentado.

En general todos los microcontroladores disponibles en el mercado ofrecen dos vías fundamentales de alimentación. La primera de ellas permite alimentarlo a partir del *Universal Serial Bus* (USB) que permite cargar el programa, mientras que la otra vía consiste en alimentarlo directamente a través de los pines que ofrecen para ello. A lo largo de esta sección se analizarán sendas vías, prestando especial atención a la viabilidad de cada una de ellas en lo que a consumo y costes se refiere.

3.4.2.1 Alimentación a través del USB-B

Atendiendo al esquema eléctrico del microcontrolador, se ha analizado el circuito que permite alimentar el sistema empleando el USB.

Para alimentar el circuito se emplea la tensión de 5 voltios que ofrece el propio USB. Dado que el USB se emplea a su vez para cargar la aplicación, esta tensión no solo se emplea para alimentar el microprocesador, sino que también alimenta el ST-Link, dispositivo que permite cargar la aplicación. Tanto el microprocesador como el ST-Link emplean tensiones de alimentación de 3,3 voltios, por lo que esta tensión debe de regularse. En lo que respecta al ST-Link basta con regular la tensión, pero en el caso del microprocesador además debe de limitarse la corriente, por lo que son necesarios dos reguladores de

tensión y un limitador de corriente. En la figura 3.9 se muestra un esquema simplificado de la alimentación del sistema.

El hecho de que sean necesarios estos dispositivos implican un mayor consumo de energía. En total se ha estimado que el sistema completo al alimentarlo a través del USB, drena alrededor de los 240 mA. Este consumo es completamente inadmisibles si se desea alimentar con batería ya que la vida del sistema es tan limitada que, incluso empleando baterías recargables mediante energía solar, la autonomía del sistema se vería seriamente comprometida durante la noche.

En la siguiente tabla se resume de forma más detallada el consumo de los diferentes dispositivos, los cuales se referencian con la misma nomenclatura que aparece en la figura 3.9.

Tabla 3.2: Consumo de los diferentes componentes

Componente	Corriente máxima (mA)
U4	25
U3	0,2
U2	0,25
U1	0,25
Microprocesador	0,0014
<i>ST-Link</i>	150
Led 1	5
Led 2	60

Teniendo en cuenta el consumo de los diferentes componentes y suponiendo que la capacidad de las baterías empleadas es de unos 2500 mAh, la autonomía estimada es de unas 10 horas, lo cual compromete seriamente la autonomía independientemente de si se emplean baterías recargables o no.

3.4.2.2 Alimentación a través de los pines V_{in} y GND

La opción alternativa que ofrece la tarjeta para poder ser alimentada consiste en alimentarla a partir de los pines V_{in} y GND . A estos pines puede conectarse una tensión continua comprendida entre 7 y 12 voltios según especifica el fabricante en las hojas de características [15] [16]. Esta tensión, se ve regulada por diferentes integrados que permiten obtener los valores de 5 y 3,3 voltios necesarios para que la tarjeta opere adecuadamente.

La tensión de entrada se regula en una primera etapa (componente U1) que reduce y estabiliza la dicha tensión a 5 voltios. Seguidamente, esta tensión se introduce simultáneamente en otros dos reguladores. Ambos reguladores permiten obtener una tensión de 3,3 voltios a partir de la tensión de 5 voltios previamente obtenida. Lo que diferencia a ambos reguladores, es que el primero de ellos (U3), emplea la tensión de salida tanto para alimentar el terminal de 3,3 voltios de la tarjeta, como para alimentar el procesador, mientras que el segundo de ellos (U2) emplea su tensión de salida para alimentar el *ST-Link*, tal y como ocurría al alimentar el sistema a través del USB.

Cabe destacar que en este caso no es necesario limitar la corriente de entrada. Sin embargo, es necesario emplear un regulador adicional para poder obtener la tensión de 5 voltios.

Atendiendo tanto a los componentes afectados como al consumo de cada uno de ellos (3.2), puede obtenerse el consumo de esta segunda alternativa.

de entre 3,6 y 1.71 voltios para operar correctamente. Además el módulo de comunicaciones seleccionado soporta tensiones de entrada de entre 3,6 y 2,1 voltios por lo que cualquier tensión de alimentación comprendida en este último rango es adecuada para el funcionamiento del sistema completo.

Para llevar a cabo el aislamiento de toda la electrónica prescindible de la tarjeta, la propia tarjeta cuenta con diferentes *jumpers* y puentes de soldadura que permiten modificar las pistas impresas al antojo del diseñador (3.9). Siguiendo las indicaciones de las hojas de características [15] [16] se ha determinado que, para aislar la alimentación, debe de retirarse el puente de soldadura *SB14*, de tal forma que todos los reguladores queden completamente aislados. Por su parte para aislar el *ST-Link* del procesador, el fabricante especifica que debe de retirarse el *SB9*, el cual se encarga de conectar la línea de *reset* de ambos dispositivos. Una vez quedan aislados estos componentes, el procesador puede alimentarse a través del pin 3V3 siempre y cuando se alimente a partir de baterías, las cuales proporcionan una tensión constante y cuya variación en el tiempo es muy lenta.

Con todas estas modificaciones se logra aislar completamente el procesador, de tal forma que ninguna corriente de fuga pueda alimentar componentes indeseados.

La gran contrapartida de aislar toda esta electrónica consiste en el hecho de no poder volver a cargar el programa una vez retirados los puentes, lo cual obliga a sustituir la electrónica en caso de desear sustituir el código. Para solucionar este problema, los terminales de los puentes retirados (*SB14* y *SB9*) se han de soldar a dos parejas de pines de tal forma que con un simple *jumper*, pueda volverse a conectar todos los componentes aislados.

Considerando todo lo anterior puede concluirse que la mejor opción para alimentar el sistema de tal forma que se maximice el consumo de la tarjeta es aislar los componentes retirando los puentes especificados y alimentar al sistema mediante una batería cuya tensión este comprendida entre 3,6 y 1.71 voltios, como pueden ser dos baterías de 1,5 voltios conectadas en serie. Sin embargo, para que el módulo de comunicaciones pueda operar correctamente, la tensión de alimentación mínima ha de ser de 2,1 voltios, por lo que la tensión de alimentación ha de estar comprendida entre 2,1 voltios y 3,6 voltios. Nótese que para este segundo rango, conectar dos baterías en serie de 1,5 voltios, sigue siendo una solución válida.

Aplicando los cambios citados anteriormente, el consumo del núcleo de procesamiento se ha reducido únicamente al propio consumo del microprocesador.

Cabe destacar que este consumo no es el consumo total del sistema ya que en esta sección no se esta considerando ni el impacto de los sensores ni el del módulo de comunicaciones. Sin embargo tras realizar las modificaciones citadas, en núcleo de procesamiento deja de suponer un riesgo para la autonomía del sistema, ya que su consumo se ve sumamente reducido como se explica en la sección 5.3.1.

3.4.2.4 Posibles fuentes de alimentación

Para alimentar el sistema completo, basta con suministrarle una tensión comprendida entre 3,6 y 2,1 voltios. Para lograrlo, la fuente más sencilla y económica consta de 2 baterías de 1,5 voltios conectadas en serie, de tal forma que la tensión de alimentación sea de 3 voltios. Esta es la solución preferente a la hora de alimentar el sistema, pues además de ser la solución más sencilla y económica, garantiza la autonomía del sistema. Sin embargo, se ha estudiado la posibilidad de alimentar el sistema a través de baterías recargables mediante energía solar.

Para llevar a cabo esta alternativa se propone hacer uso del módulo 03962A. Este módulo se encarga de controlar la carga y descarga de una batería. El módulo consta de 3 integrados. Uno de ellos es el *TP4056*, es el encargado de controlar los ciclos de carga y descarga de la batería. Para ello el chip debe

La fuente basada en paneles solares ofrece una buena solución para alimentar el sistema, sin embargo, este sistema está diseñado para cargar baterías de 4 voltios, por lo que sería necesario un regulador para adaptar la tensión de alimentación. Además el coste de este sistema es muy superior al de las baterías convencionales. Adicionalmente, para implementar este sistema es necesario ubicar el panel solar en el exterior, lo que implica mecanizar el buzón y supone un riesgo para la integridad del sistema. Es por esto por lo que, basándose en las estimaciones de consumo realizadas (5.3.2), se ha optado por descartar esta opción, ya que las baterías no recargables ofrecen una mejor relación coste/autonomía.

3.4.2.5 Conexión de los sensores

En este punto se va a comentar la conexión de los diferentes sensores y periféricos con el microcontrolador.

En primer lugar se abordan los aspectos relativos a la conexión de los *switches*, tanto el *switch* convencional; encargado de detectar el tráfico de correo; como el *switch* encargado de detectar la apertura de la puerta. Como se ha comentado en secciones anteriores, estos *switches* cuentan con dos contactos, siendo uno de ellos normalmente cerrado, mientras que el otro se trata de un contacto normalmente abierto.

Como resulta obvio, para detectar la apertura tanto de la pestaña como de la puerta, basta con una única fuente de interrupción para cada *switch*. Sin embargo, ha de considerarse el hecho de que ambos sensores se encuentran en la puerta del buzón, mientras que la electrónica se encuentra en el interior del mismo, lo que implica que la conexión entre ambos nodos debe de resistir a numerosos ciclos de apertura y cierre de la puerta. Para conectar ambos nodos, se cuenta con un cable plano conectado mediante un zócalo. El problema derivado de esta conexión consiste en el hecho de que las líneas de conexión pueden deteriorarse pasados un cierto número de operaciones de apertura y cierre. Considerando estas premisas, los contactos de los sensores ofrecen la posibilidad de detectar el deterioro de algunas de las líneas de conexión, puesto que, al contar con dos fuentes de interrupción por cada *switch* es posible detectar fallos en la continuidad de alguna de líneas sin que afecte a la detección del movimiento de la puerta o la pestaña.

Cabe destacar que este sistema no es infalible, ya que si fallasen todas las líneas de un mismo *switch*, el sistema se vería comprometido. Sin embargo, dada la baja probabilidad de que todas las líneas dejen de funcionar simultáneamente, lo que se obtiene es un buen indicador del estado del cable plano pues presumiblemente será este el elemento *hardware* que sufra un deterioro más rápido.

Los sensores empleados cuentan, además de los dos terminales previamente citados, con un contacto común. La estrategia a seguir para detectar la rotura de la línea así como la apertura y cierre de la puerta o la pestaña consiste en conectar los dos contactos a la tensión de alimentación (V_{cc}) mediante sendas resistencias de *pull-up*, mientras que el terminal común se conecta a la masa del sistema. Por su parte los contactos se han de conectar a diferentes líneas de interrupción. Dado que los contactos son complementarios, siempre que la línea se encuentre en buenas condiciones, la tensión leída en los pines conectados a interrupciones externas, deben de ser complementarias, en caso contrario, la conclusión obtenida es que una de las líneas no tiene continuidad. Nótese que para que el sistema sea eficaz no basta con conectar una de las líneas a una interrupción externa, sino que deben de conectarse ambas, ya que si una de ellas se deteriora, sea la otra de las líneas la encargada de avisar al microcontrolador del mal estado de la misma. Esta estrategia será la empleada para ambos *switches*.

Nótese que uno de los contactos de cada *switch* se encuentra la mayor parte del tiempo cerrado y por tanto drenando corriente. La corriente que drena cada uno de estos contactos puede calcularse como el cociente entre la tensión de alimentación (3 voltios) y el valor de la resistencia de *pull-up* (40.000 ohmios en caso de emplear la resistencia interna del microcontrolador). Atendiendo a las hojas de características,

el fabricante indica que el mínimo nivel de tensión que puede reconocerse como un nivel alto es del 70 % de la tensión de alimentación, por lo que cualquier tensión superior a 2,1 voltios será reconocida como un nivel alto. Para minimizar el consumo se ha optado por emplear sendas resistencias externas de 390.000 ohmios para los contactos normalmente cerrados, lo cual implica que la corriente drenada por cada uno de estos contactos es casi diez veces menor. Según especifica el fabricante, la corriente que drenan los pines puede ser de hasta 650nA, por lo que la caída de tensión en la resistencia se sitúa en torno a los 0,25 voltios, tensión más que aceptable para reconocer un nivel alto. Esta sencilla medida ayuda enormemente a reducir el consumo, incrementando notablemente la autonomía del sistema.

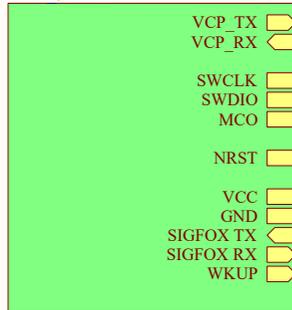
Por su parte el modulo de comunicaciones cuenta con 10 pines que permiten conectar dicho módulo. De estos pines, dos de ellos se emplean para conectar la masa del sistema, un tercero se emplea para alimentar el módulo, otros dos pines se emplean para establecer la comunicación serie, el sexto se emplea para despertar al módulo del modo de bajo consumo, el siguiente esta conectado al *reset* del módulo y los tres pines restantes se emplean para llevar a cabo la depuración. En el caso que nos ocupe solo sera necesario emplear seis de estos pines, los tres dedicados a la alimentación, los dos dedicados a las comunicaciones y el pin encargado de despertar al módulo del modo de bajo consumo.

Atendiendo a la descripción anterior es sencillo establecer las conexiones. Las masas se han de conectar a la misma masa del micro, la tensión de alimentación se conecta al pin de de 3,3 voltios disponible en el microcontrolador, los pines de comunicaciones se conectan a los pines dedicados a la comunicación serie del microcontrolador y el pin destinado a despertar al módulo del modo de bajo consumo se conecta a un pin de propósito general, de tal forma que el micro tenga el control sobre dicha línea, despertando el módulo según corresponda.

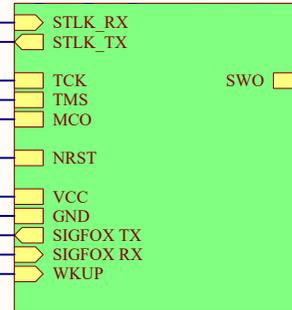
Finalmente, el último elemento que debe conectarse es el zumbador piezoeléctrico. Este sencillo componente únicamente dispone de dos terminales, masa y alimentación. Dado que se trata de un zumbador activo basta con alimentarlo a 3,3 voltios para que emita un tono. Teniendo en cuenta estas especificaciones, la línea de masa se conecta a la masa del microcontrolador y el terminal de alimentación a un pin de propósito general, permitiendo al micro que lo active cuando sea preciso.

Para entender mejor las conexiones de todos los dispositivos citados a lo largo de esta sección se incluye en las páginas sucesivas el esquema eléctrico del microcontrolador en el cual se pueden apreciar las conexiones modificadas, así como las conexiones del mismo con los diferentes dispositivos (*switches*, zumbador y módulo de comunicaciones). Especialmente, la atención debe centrarse en el esquema del microcontrolador, atendiendo a los *jumpers* JP2 y JP3, pues son estos los que sustituyen a los puentes de soldadura que incorporaba la tarjeta de serie. Además en este mismo esquema se incluye el conexionado de los *switches* a la tarjeta a través del zócalo empleado para conectar el cable plano.

U_MCU_32
MCU_32.SchDoc

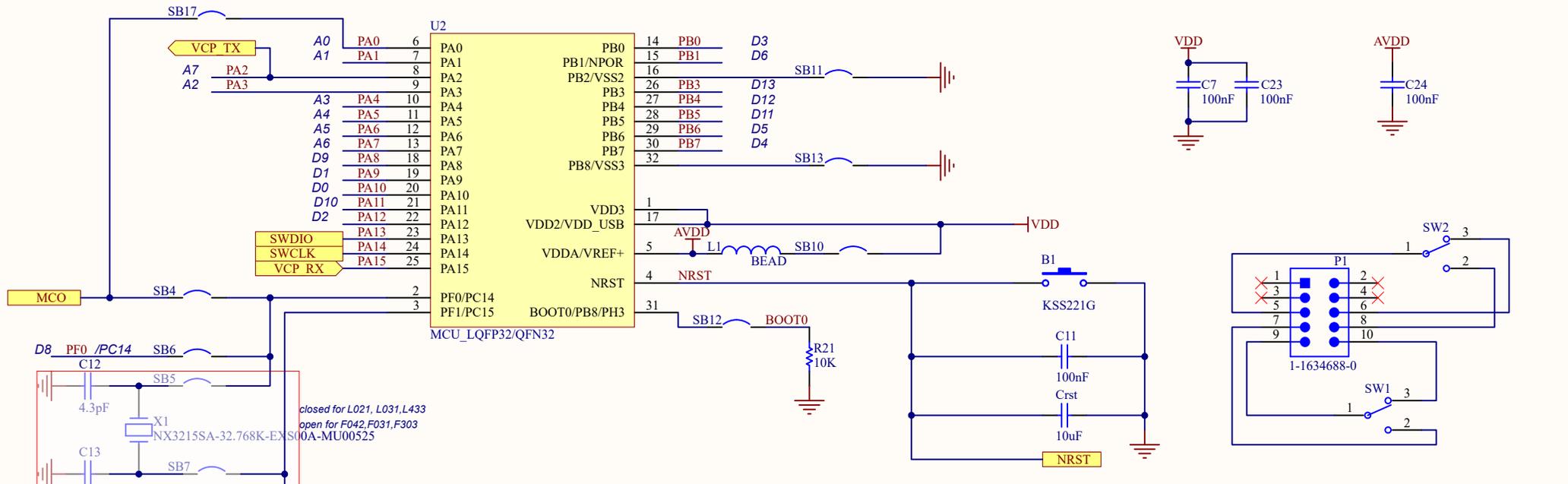


U_ST_LINK_V2-1 / SIGFOX
ST LINK V2-1 SIGFOX.SCHDOC

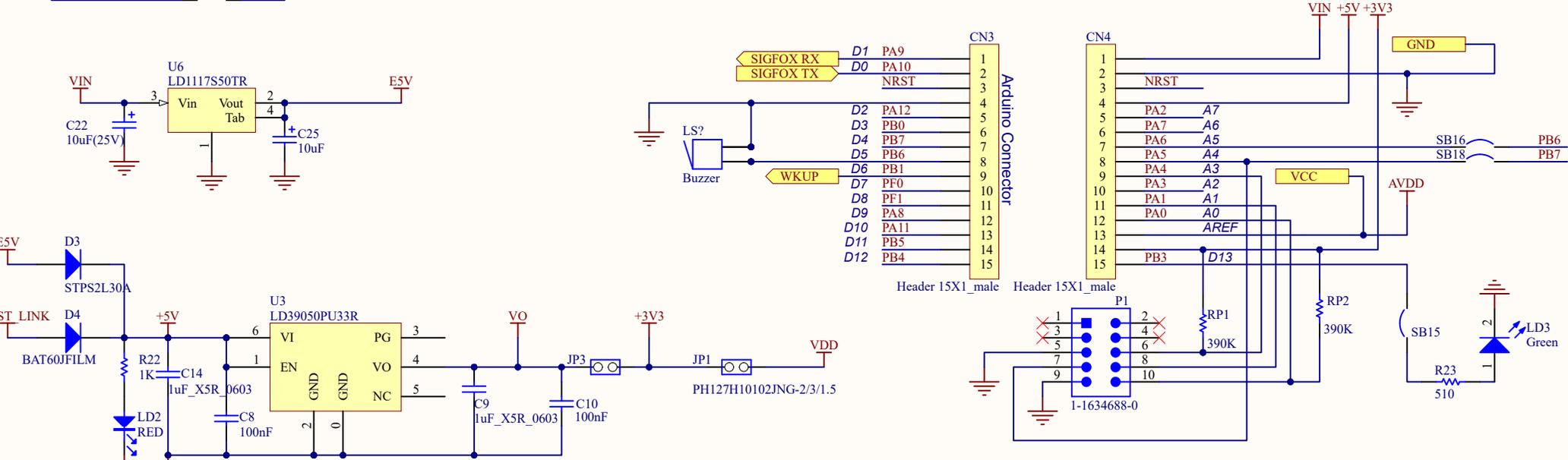


REV B: SB14 changed to JP1 Jumper for easy IDD measurement, and enlarge board length; CN1 USB PN changed to Micro-B for Device.
REV C: Add SB18/SB16 for connecting D4/D5 to A4/A5

Title: TOP			 life.augmented
Project: NUCLEO32			
Size: A4	Reference: MB1180	Revision: C.1	
Date: 17/04/2022	Sheet: 1 of 3		



Extension connectors



Title: MCU		
Project: NUCLEO32		
Size: A4	Reference: MB1180	Revision: C.1
Date: 09/06/2022	Sheet: 2 of 3	



En la figura 3.11 se muestra el conexionado de los diferentes componentes de forma más gráfica y sencilla, con el fin de aclarar las conexiones entre los diferentes componentes. En esta figura se muestran las conexiones de los diferentes dispositivos entre sí, considerando al microcontrolador y al módulo de comunicaciones como cajas negras. Cabe destacar que de los pines conectados del microcontrolador, los pines Ax representan entradas de interrupciones externas, mientras que los pines Dx representan salidas de propósito general.

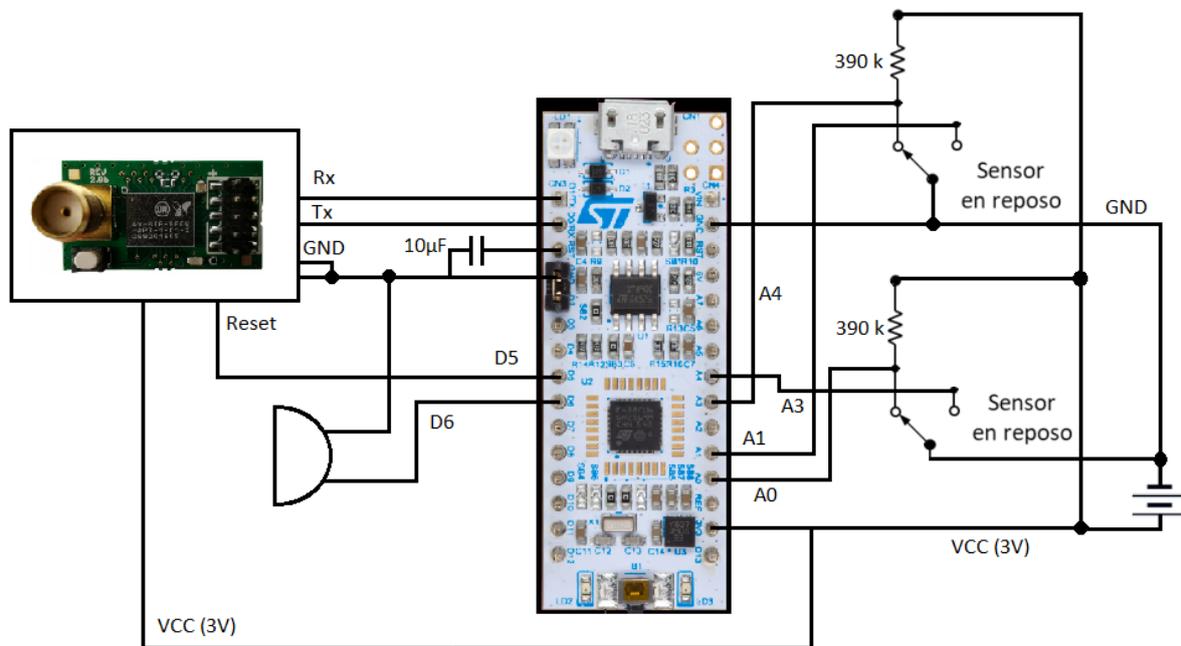


Figura 3.11: Conexionado de los diferentes dispositivos electrónicos

Es preciso señalar que al haber modificado el circuito, el sistema no se reinicia al conectar la alimentación, por lo que, en principio es necesario accionar el pulsador de *reset*. En lo que respecta al microcontrolador, este dispone de una célula *RC* para que se realice un *reset* al conectar la alimentación. Sin embargo el *reset* del microprocesador es mucho más rápido que el del módulo de comunicaciones, lo que implica que el módulo de comunicaciones no se reinicie adecuadamente. Para solventar este problema se ha añadido un condensador de 10µF en paralelo con el condensador ya existente para ralentizar el tiempo de subida y retrasar así el *reset*, dando tiempo a que la alimentación se estabilice. El hecho de que el tiempo de subida se vea incrementado no es un problema, ya que ambos dispositivos, microprocesador y módulo de comunicaciones disponen de un comparador con histéresis en los pines de entrada, incluido el pín de *reset*.

3.5 Programación del microcontrolador

Esta sección se centra en explicar el código empleado para implementar la funcionalidad del sistema. Para que el sistema pueda llevar a cabo su tarea, se van a emplear algunos de los periféricos que incorpora el microcontrolador, como son el RTC, los *timers* de bajo consumo, la *Universal Asynchronous Receiver-Transmitter* (UART), el sensor de temperatura y algunos de los puertos *General Purpose Input Output* (GPIO).

El funcionamiento del código es el siguiente. En primer lugar el sistema debe ser capaz de reaccionar

ante la apertura de la pestaña que cubre la abertura por la que se depositan el correo. En este caso el sistema únicamente debe contabilizar las aperturas, tratando de eliminar los rebotes mecánicos de la propia pestaña así como los rebotes eléctricos del pulsador. En segundo lugar debe ser capaz de detectar la apertura de la puerta del buzón, tratando de eliminar los rebotes del pulsador al igual que ocurría con el sensor de la pestaña, con la diferencia de que en este caso debe enviarse un mensaje vía *Sigfox* informando de la operación así como emitir un tono, tanto al abrir como al cerrar la puerta, que indique al cartero que la operación se ha llevado a cabo de forma satisfactoria. Cabe destacar que durante estas operaciones, el sistema deberá ser capaz de detectar posibles errores en las líneas que comunican dichos sensores con el microcontrolador, informando vía *Sigfox* en caso de detectarse algún error.

Finalmente el sistema debe ser capaz de activarse una vez al día a la hora especificada por el usuario desde el *backend* de *Sigfox*, momento en el cual ajustará el reloj interno y configurará la alarma para enviar el mensaje al día siguiente. Será en este instante cuando el sistema envíe la información referente al correo recibido y envíe la información sobre el estado del sistema en lo que a tensión de alimentación y temperatura se refiere.

Durante el tiempo en el que el sistema permanezca a la espera de que se produzca alguno de los eventos anteriores, el sistema deberá permanecer en modo UBC, limitando al máximo el consumo.

3.5.1 Estructura del código

El código se ha estructurado siguiendo la estructura típica a la hora de programar un microcontrolador. Al comienzo del programa se inicializan todos los periféricos necesarios para el desarrollo del sistema (fase de *setup*). Una vez inicializados el código finaliza con un bucle infinito en el que se implementa una máquina de estados capaz de atender a las necesidades del sistema.

El código aparece dividido en diferentes funciones y *handlers*, los cuales realizan tareas muy específicas. A estas funciones y *handlers* se les hace referencia desde la función *main*, encargada de establecer el orden de ejecución del código.

El código debe ser capaz de responder a diferentes eventos además de permanecer en estado UBC, durante el tiempo en el que no hay necesidades que atender. Las necesidades que han de atenderse son las siguientes:

- Interrupción periódica.
- Introducción de correo: Apertura de la pestaña.
- Retirada de correo: Apertura de la puerta.

Para entender la ejecución del código se ha desarrollado el esquema de las figuras 3.12 y 3.13, en la cual aparece de forma resumida el hilo de ejecución del código. En la figura 3.12 se muestra el hilo de ejecución de la función *main*. Por su parte la máquina de estados implementada en el bucle infinito al final de la función *main* se resume en el grafo de la figura 3.13.

3.5.2 Rutinas de atención a la interrupción

Según se comenta anteriormente, Al ejecutarse un *reset* en el código, la primera tarea que se realiza es la configuración de los diferentes periféricos (fase de *setup*). En esta configuración se fija la frecuencia de reloj a la que va a operar el sistema (32 MHz), además de configurar los diferentes periféricos.

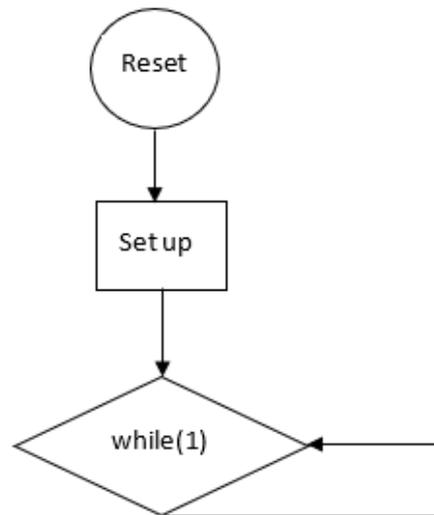


Figura 3.12: Ejecución del código

En esta fase, comenzamos configurando el modo de bajo consumo. Este modo denominado *Stop mode 2*, permite reducir el consumo del procesador a $1.4\mu\text{A}$, permitiendo que tanto el RTC como las interrupciones externas puedan permitir que el microprocesador abandone el modo de bajo consumo. Cabe destacar que en esta fase no se duerme al microcontrolador, tan solo se configuran los registros para que en el momento en que se haga efectiva la orden que le permita entrar en el modo de UBC, el microprocesador ingrese en dicho modo.

A continuación se inicializan los dos *timers* de bajo consumo. Estos *timers* se sincronizan con la frecuencia de 32 MHz y permiten despertar al procesador del modo de bajo consumo. Para la aplicación que se pretende desarrollar estos *timers* se configuran en modo de auto-carga. Este modo permite al *timer* contar un determinado número de ciclos de reloj e interrumpir una vez haya concluido el conteo. En este caso concreto los *timers* se han configurado para contar un total de 5 ms y 15 ms.

Seguidamente se inicializa el ADC. Este periférico se inicializa a una frecuencia de 8 MHz, para que el ADC pueda cumplir con la frecuencia de muestreo. La configuración establecida permite que opere con su resolución máxima (12 bits) de forma que opere en régimen de conversión continua. Este modo permite al ADC convertir los canales habilitados de uno en uno cada vez que se solicite una conversión siguiendo el orden establecido, es decir, cada vez que se inicie una conversión el ADC convertirá la tensión de uno de sus canales, convirtiendo el siguiente canal cuando se vuelva a iniciar una conversión. En el caso del sistema implementado, el ADC debe de convertir dos canales.

En primer lugar debe de convertir la tensión de referencia analógica que recibe el microprocesador. Esta tensión deriva de la tensión de las baterías, por lo que es necesario medirla diariamente para calcular tanto la propia tensión como la alimentación del sensor de temperatura con precisión, de tal forma que el sistema pueda auto-calibrarse a la hora de calcular la temperatura.

El siguiente paso consiste en configurar los diferentes puertos. En total deben de configurarse un total de 6 puertos. Cuatro de estos puertos serán los encargados de conectar los interruptores que informan sobre el estado de la puerta y la pestaña, el quinto es el encargado de activar el zumbador piezoeléctrico y el sexto tiene como función despertar al módulo de comunicaciones. Los 4 pines conectados a los interruptores han sido configurados como pines de interrupción externa para que los sensores puedan despertar al sistema del modo UBC. De estos 4 pines, dos de ellos se encuentran conectados a dos

contactos NA, mientras que los otros dos pines se encuentran conectados a contactos NC, siguiendo el esquema eléctrico comentado en secciones anteriores 3.4.2.5.

Por su parte los otros dos pines, los cuales no han de estar conectados a interrupciones externas permanecerán configurados como salidas en modo drenador abierto, de tal forma que no consuman energía mientras no se sea necesario su uso. Cuando la máquina de estados les requiera, su configuración cambiará fijando un nivel alto o bajo según corresponda.

Finalmente se configuran la UART y un *buffer* circular. En el caso del *buffer*, este se ha configurado para que cuente con una capacidad de 64 posiciones de 8 bits, de tal forma que permite almacenar la cola de mensajes enviados o recibidos para poder disponer de ellos en caso necesario. Por su parte la UART, se ha configurado según las especificaciones del módulo de comunicaciones, las cuales establecen que la velocidad es de 9600 *bauds* con un tamaño de *frame* de 8 bits sin bit de paridad y con un único bit de *Stop*.

Para controlar el orden de ejecución del código se ha implementado la máquina de estados comentada previamente. Para entender la máquina de estados previamente han de conocerse las rutinas de atención a las interrupciones. En primer lugar, cabe destacar que las interrupciones son eventos completamente asíncronos que, como su nombre indican, interrumpen la ejecución normal del código. Para que el sistema sea lo más robusto posible se ha prestado especial atención a la latencia de estas rutinas, ya que debe ser tan pequeña como sea posible para no generar errores en el desarrollo normal del código.

Dentro de estas rutinas podemos diferenciar 4 grandes grupos. En primer lugar hemos de considerar las rutinas de las interrupciones externas. Estas rutinas tienen como única función reconocer la interrupción que ha permitido despertarse al microcontrolador y lanzar un *timer* para eliminar los posibles rebotes. Esta tarea cuya latencia es extremadamente corta permite que las interrupciones externas no queden pendientes a causa de los rebotes.

El siguiente grupo de rutinas es la asociada a los *timer*. Dentro de este tipo de rutinas, existen diferencias en las rutinas de cada uno de los *timers*. En el caso del *timer* 1 su única función es limpiar los rebotes del interruptor asociado a la pestaña. Para ello su rutina consiste en registrar el valor de cada uno de los pines asociados al interruptor asociado a la pestaña e incrementar un contador en caso de que el valor actual coincida con el valor registrado. Con este sencillo algoritmo puede contabilizarse de forma sencilla el tiempo que la tensión en cada uno de los pines permanece constante. En función del valor del contador el *timer* se relanzará o no. Este *timer* interrumpe cada 5 ms y el contador debe llegar hasta un valor 30 para validar la tensión de los pines, lo que implica que la tensión debe permanecer estable durante 150 ms. Cuando se detecta que la tensión es estable, se modifica el valor de un *flag* que informará a la máquina de estados sobre las operaciones realizadas en la pestaña.

Este mismo algoritmo se ha implementado de forma análoga en el *timer* 2 el cual está asociado a la puerta del buzón. Por el contrario, en este caso al emplear un *switch* diferente, se ha determinado que el tiempo necesario para que la tensión se estabilice es mayor que en el caso de la pestaña, por lo que el periodo de interrupción de este *timer* se ha fijado a 15 ms, lo que supone que en este otro caso la tensión debe permanecer estable durante 450 ms, ya que el contador asociado a este evento también se inicializa en 30.

Además, este *timer* es el encargado de contabilizar el tiempo que permanece activo el zumbador al modificarse el estado de la puerta. Para contabilizar este tiempo el *timer* en cuestión cuenta con un segundo contador que permite activar el zumbador. Al igual que ocurre con el otro *timer* cuando la tensión de ambos pines se estabiliza, se configuran los *flags* correspondientes para transmitir esta información a la máquina de estados, de tal forma que sea esta la que toma las decisiones sucesivas.

Cabe destacar que el zumbador solo se activa en caso de que las líneas que conectan el *switch* estén

en perfecto estado, por lo que para que este segundo contador solo incrementa su cuenta y activa el zumbador si la maquina de estados se lo indica.

A continuación, el siguiente grupo de rutinas lo compone la rutina del *Analog to Digital Converter* (ADC). Esta rutina la componen únicamente 2 instrucciones. La primera almacena el valor de digital de la tensión convertida y la segunda habilita un *flag* que informará a la máquina de estados de que la conversión se ha completado.

Por su parte la última de las rutinas es la del RTC. Esta rutina tiene como única función configurar su *flag* correspondiente para informar a la máquina de estados del evento acaecido para que sea esta la que tome las decisiones correspondientes.

Nótese que todas las rutinas de atención a la interrupción tienen una latencia extremadamente corta, contribuyendo a que la ejecución del código sea lo más fluida posible.

Cada una de las interrupciones tiene asociada una prioridad tal y como se resume en la tabla 3.3.

Tabla 3.3: Prioridades de las interrupciones

Interrupción	Prioridad
RTC	0
UART	0
ADC	1
<i>timers</i>	2
<i>External Interrupt</i> (EXTI)	3

Como se refleja en la tabla 3.3, la prioridad más alta se asocia al RTC y la UART. El RTC tiene la prioridad más elevada por presentar la latencia más corta de todas las interrupciones, lo que permite atenderla sin que entre en conflicto con ninguna de las otras prioridades. Por su parte la UART tiene una prioridad elevada para evitar que la recepción o la transmisión de datos se vean interrumpidas evitando que los datos aparezcan corruptos. La siguiente interrupción con mayor prioridad la constituye el ADC, ya que al igual que el RTC tiene una latencia muy corta. Seguidamente, las interrupciones con mayor prioridad las constituyen los *timers*, Estos tienen una latencia más elevada y por tanto pueden ocupar la CPU durante tiempo más elevado. Sin embargo la temporización que llevan a cabo no es crítica, ya que únicamente se emplean para eliminar los rebotes, por lo que este nivel de prioridad permite a otras interrupciones más prioritarias desalojarla sin que su función se vea entorpecida. Finalmente las interrupciones con la prioridad más baja la constituyen las interrupciones externas. Estas interrupciones pueden ser desalojadas por cualquier otra interrupción, lo que permite que los posibles rebotes generados en los interruptores no ocupen la CPU. Además, estas interrupciones pueden retrasarse tanto como sea necesario, ya que la información que recaban solo es transmitida una vez al día por lo que disponen de un largo plazo de tiempo para ser atendidas sin que afecte al funcionamiento global del sistema.

3.5.3 Máquina de estados

La máquina de estados es la encargada de tomar las decisiones oportunas en función de la información suministrada por los diferentes periféricos y sensores. El funcionamiento de la misma se resume en el grafo de la figura 3.13.

En el grafo puede verse como el estado por defecto es el en que el sistema permanece en modo de UBC. En este estado, el microcontrolador permanece en el modo *Stop 2*, en el cual solo drena un total de 1.4 mA.

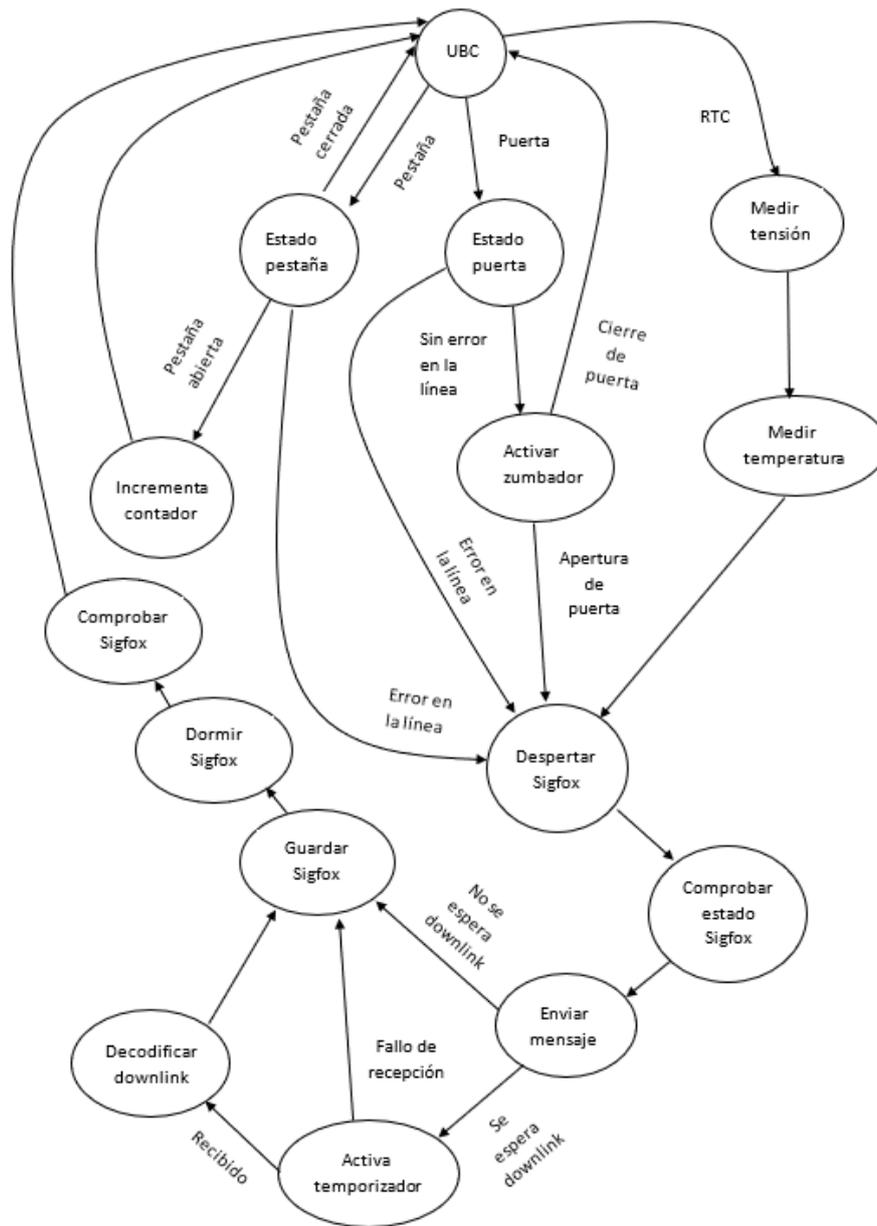


Figura 3.13: Máquina de estados

Además, el módulo de comunicaciones permanece en el modo de bajo consumo, drenando una corriente de 180 nA. Por su parte los pines analógicos encargados de despertar el módulo de comunicaciones y de activar el zumbador permanecen en drenador abierto, lo cual evita que la corriente se consuma a través de dichos pines.

A partir de este estado puede verse como nacen tres grandes ramas. Estas se analizarán individualmente en lo sucesivo. Estas tres ramas comienzan y finalizan en el estado de reposo (modo UBC). Para que el funcionamiento de la máquina sea el correcto una vez que se ingresa en alguna de las ramas debe de regresarse al estado UBC antes de poder ingresar en una rama nueva. Este mecanismo garantiza que la ejecución de cada una de las tareas asociadas a cada rama se concluya satisfactoriamente. En caso de que una de las ramas requiera activarse durante el desarrollo de alguna de las otras ramas, quedará a la espera de que la rama en curso finalice su ejecución antes de atender al evento nuevo.

La primera de las ramas (rama izquierda de la figura 3.13), es la encargada de gestionar la pestaña.

Como puede verse, cuando se detecta que la pestaña se ha activado, tras eliminar los posibles rebotes que esta pueda tener gracias a los *timers*, se evalúa el estado de la pestaña. En caso de que se determine que la pestaña se ha cerrado correctamente, el sistema regresa al estado UBC. Si por el contrario se determina que la pestaña se ha abierto se incrementa el contador de correo antes de regresar al modo UBC. Si lo que ha ocurrido es que se ha detectado un fallo en las líneas que conectan el sensor con el microcontrolador, se inicia el algoritmo de envío de mensajes que se analizará posteriormente.

Para el caso de la puerta el proceso a seguir para su tratamiento es completamente análogo al de la pestaña, con la diferencia de que en este caso aparece un estado adicional vinculado a la activación del zumbador. Si se detecta la apertura o cierre de la puerta sin que se produzcan fallos en las líneas, se activa y se desactiva el zumbador. Después, si se ha detectado que la puerta se ha abierto o se ha detectado algún fallo en las líneas se comienza el algoritmo de envío de mensajes. En caso de detectar el cierre de la puerta se regresa al modo UBC. Además, al detectar la apertura de la puerta, el sistema reinicia el contador de cartas, así como las posibles alarmas de tensión y/o temperatura que hayan podido detectarse.

Finalmente la última de las ramas es la encargada de atender a la interrupción periódica generada por la interrupción del RTC. Una vez se accede a la rama, la máquina de estados se encarga de lanzar dos veces el ADC, para convertir los canales conectados a la tensión de alimentación analógica del microprocesador y al sensor de temperatura consecutivamente.

Al alimentar el microprocesador a través de baterías la tensión de alimentación analógica depende directamente de la tensión proporcionada por las baterías. Esto supone un problema a la hora de trabajar con el ADC, ya que la tensión de referencia del mismo varía con el tiempo. Para solventar este problema, cada microcontrolador almacena en memoria no volátil un valor propio para cada microprocesador que permite calibrar el ADC. Para calibrar el ADC las hojas de características [15] [16] indican que debe de aplicarse la ecuación 3.1. Esta ecuación pondera el valor nominal de alimentación en función del valor que lea el ADC en el canal conectado a su tensión de referencia y el valor de calibración propio del ADC.

$$V_{DDA} = 3 * \frac{V_{calibración}}{V_{referencia}} \quad (3.1)$$

Esta tensión de alimentación no solo se corresponde con la tensión de las baterías, sino que además permitirá obtener una lectura correcta de la tensión proporcionada por el sensor de temperatura.

Una vez se conoce la tensión de referencia del ADC, se puede convertir la tensión proporcionada por el sensor de temperatura. Esta tensión puede calcularse con la expresión 3.2

$$D_{temp} = 4096 * \frac{V_{temp}}{V_{DDA}} \quad (3.2)$$

$$V_{temp} = V_{DDA} * \frac{D_{temp}}{4096} \quad (3.3)$$

Tras conocer el valor de la tensión del sensor de temperatura, puede calcularse la temperatura haciendo uso de la función de transferencia del sensor proporcionada por las hojas de características [15] [16]. Esta ecuación sigue la expresión mostrada en la ecuación 3.4.

$$temperatura = \frac{V_{temp} - V_{30C}}{Pendiente} + temp_{referencia} \quad (3.4)$$

$$temperatura = V_{temp} * 400 - 274 \quad (3.5)$$

Además de calcular la temperatura y la tensión de alimentación cada uno de estos estados evalúa si la tensión y la temperatura son adecuadas para el buen funcionamiento del sistema. En el caso de la tensión de alimentación se ha establecido que en el mensaje periódico informe de si existe riesgo de que se agoten las baterías cuando la tensión detectada sea inferior 2,7 voltios, momento en el cual las baterías comenzarán a descargarse mucho más rápido.

Por su parte los dispositivos electrónicos seleccionados aguantan un rango de temperaturas comprendido entre los -40°C y los 80°C . El estado asociado al cálculo de la temperatura, evalúa si la temperatura supera los 65°C o cae por debajo de los -25° , indicando en el mensaje diario que la temperatura se aproxima al límite de la electrónica.

Como puede verse ambos umbrales son demasiado conservadores, con el fin de garantizar la protección del sistema, aunque pueden cambiarse fácilmente en caso de que así lo especifique el cliente.

Una vez se han concretado todos los datos anteriores se esta en condición de iniciar el algoritmo de envío de mensaje. Este algoritmo cuenta con diferente estados contenidos dentro de la máquina de estados y son análogos para cualquier mensaje que se desee enviar.

El primero de los estados que inicia el algoritmo de envío de mensaje tiene como propósito despertar al módulo de comunicaciones, para lo cual ha de introducirse un nivel alto a través del GPIO9. Una vez se establece este pin a nivel alto, se debe de comprobar que el módulo esta operativo antes de enviar el mensaje, ya que requiere de cierto tiempo para abandonar el modo de bajo consumo. Para llevar a cabo esta operación se le envía el comando *AT*, al cual debe de responder con el mensaje *OK*. Tras establecer el pin correspondiente a nivel alto se envía dicho mensaje. En este primer mensaje se corre el riesgo de que se haya enviado demasiado pronto, sin que el módulo haya tenido tiempo suficiente como para despertarse, por lo que no llegará respuesta alguna. Para evitar que este escenario bloquee la ejecución del código permaneciendo a la espera de una respuesta que jamás llegará, la maquina de estados continua su ejecución durante un ciclo más del bucle infinito, de tal forma que en la siguiente iteración regrese al mismo punto en el que se reenviará el comando. Este proceso se repetirá iterativamente hasta que se reciba una respuesta por parte del módulo de comunicaciones. En este momento ya se tiene seguridad de que puede enviarse el mensaje periódico con garantías de que será enviado correctamente por lo que la maquina de estados ingresa en el siguiente estado, el cual se encarga de realizar esta tarea.

Una vez enviado el mensaje el siguiente estado dependerá de si se ha solicitado un *downlink* o no. En caso de haber solicitado al operador el envío de un *downlink*, se pasará al estado de recepción de *downlink*, en el que se decodifica la información recibida.

Tras realizar algunas pruebas, se determinó que en algunas ubicaciones, especialmente cuando el sistema se ubica en el exterior, el *downlink* no se recibía correctamente, lo cual impedía que la máquina de estados pudiese seguir su curso. Sin embargo, según especifica el fabricante, y según se ha podido determinar experimentalmente, desde que se envía el mensaje con petición de *downlink* hasta que se recibe dicho *downlink*, transcurre unos 40 segundos, aunque este tiempo puede variar ligeramente en función de la cobertura disponible. Considerando esto, se ha optado por temporizar el minuto inmediatamente posterior al envío del mensaje que solicita el *downlink*. De tal forma que si el *downlink* en cuestión no se recibe en un plazo máximo de un minuto tome por válida la información que conoce en lugar de actualizar dicha información a partir del *downlink* esperado. Esta temporización se realiza con el propio RTC, ya que este temporizador permite gestionar interrupciones tan largas como se deseen lo que evita que la interrupción del propio temporizador corrompa el contenido del *downlink*.

Nótese que, en caso de no recibir el *downlink*, podría haberse implementado un algoritmo que reenviase de nuevo otro mensaje para volver a recibir un nuevo *downlink*. Sin embargo, esta solución quedó descartada ya que si un *downlink* no se logra recibir de forma adecuada no hay nada que garantice que un

segundo *downlink* vaya a ser recibido con éxito. En caso de producirse un fallo en la red, enviar un mensaje cada minuto provocaría la descarga de las baterías en un plazo inferior a un día, lo que implicaría llevar a cabo una tarea de mantenimiento para reemplazar las baterías. Por el contrario si hubiese algún problema en la red, como puede ser el deterioro de una antena, el algoritmo implementado es completamente inmune ya que, una vez que se inicie, en caso de error el sistema seguirá operando sin problema. En caso de que el *downlink* no se reciba correctamente, como resulta obvio, en caso de actualizar la información del propio *downlink* en el *backend*, está no será tomada en cuenta ya que el nodo no es capaz de recibirlo. En cualquier caso, la información que alberga el *downlink* únicamente contiene la hora exacta empleada para calibrar el RTC, la hora a la que debe de despertarse el sistema al día siguiente y el identificador del buzón, por lo que en caso de no recibirse el *downlink* el nodo continuará operando de la misma forma que venía haciéndolo hasta el momento. Además, tanto si el *downlink* se ha recibido correctamente como si no, el sistema lo registrará en el mensaje diario para que pueda ser enviado al día siguiente.

Por el contrario, si no se espera *downlink* alguno, si se ha finalizado la decodificación del mismo o si se ha determinado que el *downlink* no ha podido recibirse, se pasa al estado en el que se duerme el módulo de comunicaciones. Para dormir este módulo se ha de configurar el pin conectado al GPIO9 del módulo en modo drenador abierto y enviar el comando *AT\$WR*, el cual se encarga de almacenar la configuración del módulo antes de dormirlo. Una vez se ha guardado la configuración del módulo se ha de enviar el comando *AT\$P=2* para dormir el módulo. Tras esta operación la maquina de estados comprueba que el mensaje encargado de dormir al módulo se haya transmitido correctamente y regresa al modo UBC, permitiendo que la totalidad del sistema regrese al modo de bajo consumo.

Toda la información referente a los modos de bajo consumo del módulo de comunicaciones se resumen en la figura 3.14.

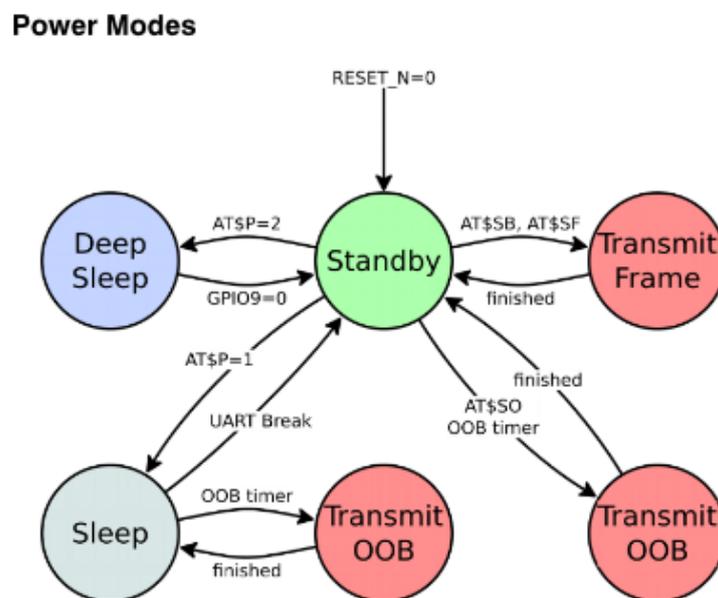


Figura 3.14: Máquina de estados del módulo de comunicaciones

3.5.3.1 Mensajes

En la tabla 3.4 se resumen los diferentes mensajes pre-configurados que puede enviar el sistema, así como la utilidad de cada uno de ellos.

De entre todos ellos, cabe destacar la información que alberga el primero de ellos, ya que es el único

Tabla 3.4: Mensajes que puede enviar la UART

Mensaje	Utilidad
$AT\$SF=xxxxxxxxxxxxxxxxxxxx,1$	Contiene la información recabada a lo largo de un día
$AT\$SF=01$	Informa de un error en la conexión del sensor de la pestaña
$AT\$SF=10$	Informa de un error en la conexión del sensor de la puerta
$AT\$SF=BA$	Informa de que se ha abierto la puerta del buzón
$AT\$WR$	Guarda la información del módulo antes de dormirse
$AT\$P=2$	Duerme al módulo
AT	Comprueba la conexión

mensaje cuyo contenido es completamente variable. En primer lugar, debe de considerarse que el *backend* de *Sigfox* tiene la capacidad de decodificar la información recibida, pero no puede decodificar caracteres sueltos (datos de 4 bits), sino que debe de decodificarla según los tipos de datos uint8 y uint16. No obstante, parte de la información que se transmite puede almacenarse en un único carácter ASCII, por lo que, a pesar de poderse almacenar con un único carácter se han destinado dos para que sea posible su decodificación. Tras haber considerado lo anterior, la información del mensaje se organiza según la tabla 3.5.

Tabla 3.5: Codificación del mensaje periódico

Carácter	Información
1-2	Alerta de temperatura
3-4	Indicador de temperatura positiva
5-6	Valor de la temperatura
7-8	Número de cartas
9-10	Fallo en el <i>downlink</i>
11-12	Alerta de tensión
13-16	Tensión de alimentación en centivoltios
17-20	Identificador del nodo

Cabe destacar el hecho de que hay algunos datos booleanos y algunos datos numéricos. Como ya se ha comentado los datos booleanos se representan mediante el carácter ASCII de los números 0 o 1. Sin embargo los datos numéricos deben de enviarse mediante caracteres ASCII formando un número entero hexadecimal. Este es el caso de los valores de tensión, temperatura, el número de cartas y el identificador del buzón. En el caso de la temperatura se codifica en un entero y del número de cartas se codifican en un entero de 16 bits, cuyo rango comprende valores entre 0 y 255, rango mas que suficiente para registrar el número de cartas que almacena el buzón y la temperatura. Además, la temperatura se complementa con su bit de signo almacenado en los caracteres 3 y 4, por lo que su rango varía entre 255 y -255 grados. Por su parte la tensión de alimentación se representa en centivoltios, ya que el *backend* solo puede decodificar enteros y se considera que las centésimas de voltio ofrecen una precisión más que suficiente. Teniendo esto en cuenta, puede deducirse que el rango de tensiones puede variar entre 0 y 40,95 voltios. Nótese que este rango es más que suficiente ya que según las características eléctricas del microcontrolador la tensión de alimentación no puede superar los 3,6 voltios. Finalmente, el identificador asociado al nodo se

representa según entero de 16 bits lo cual permite contabilizar hasta 65.536 buzones. Según la información proporcionada por Correos el sistema debe de poder implantarse en unos 23.000 buzones, por lo que el rango, de nuevo, es más que suficiente.

Nótese que para que el de identificadores sea completo, el identificador debe de codificarse en hexadecimal a la hora de configurar el *downlink*. De no hacerse así, el rango se verá reducido, además de decodificarse los datos de forma errónea.

3.5.3.2 Decodificación del *downlink*

A través de *downlink* se obtienen tres parámetros que condicionan el funcionamiento del sistema. El primero de ellos es la hora a la que el sistema se conecta con la nube, la cual permite al sistema auto-corrregir la hora del RTC en caso de que este se desfase. El segundo parámetro contiene la hora a la que debe despertarse el sistema al día siguiente para enviar el mensaje periódico, y el último de los parámetros contiene la información del identificador del buzón, lo cual permite desligar la electrónica del buzón en el que se instale. En lo que respecta al identificador del buzón la decodificación es directa, ya que el identificador sencillamente se reenvía con el mensaje diario. Sin embargo para poder establecer tanto la hora actual como la hora a la que se desea que se active la alarma que despierta al sistema deben de ser introducidas en formato *Binary-Coded Decimal* (BCD) pero los caracteres recibidos por el módulo se reciben en formato ASCII, lo que obliga a realizar la conversión. En el caso de la alarma, la decodificación es directa ya que el formato en el que se envía la hora es *hh:mm*. Sin embargo, la hora actual es un valor que el operador envía de forma automática y esta expresada en formato *epoch*. Este formato expresa la hora actual como el número de segundos que han transcurrido desde el 1 de enero de 1970 a las 00:00. Este inmenso número se envía codificado como los caracteres ASCII que expresan dicho valor en hexadecimal. Para decodificar este valor, la primera tarea consiste en extraer el valor y codificarlo en hexadecimal. A continuación este valor se codifica en decimal gracias a una segunda función y se obtienen el número de años transcurridos. El siguiente paso consiste en determinar el número de años bisiestos que han transcurrido desde entonces y determinar si el año actual es bisiesto o no. Una vez se obtiene esta información el sistema está en condiciones de determinar el mes y el día en el que se encuentra. Para llevar a cabo esta tarea se han implementado dos calendarios, uno de ellos que permite obtener el mes y el día actual si el año es bisiesto y otro si no lo es. Tras averiguar el mes y el día en cuestión solo falta por obtener la hora, los minutos y los segundos realizando sucesivas divisiones. El último paso consiste en codificar cada uno de los parámetros recientemente obtenidos que expresan la fecha en formato BCD, para poder introducirlos finalmente en los registros del RTC.

Este sistema a pesar de ser farragoso permite que el RTC siga con precisión la hora actual, ya que en caso de producirse algún desfase automáticamente se restablece la hora al día siguiente. Además cabe destacar que la hora introducida en el *downlink* para que configurar la alarma es siempre la hora a la que se despertará el sistema un día más tarde. Esta medida se ha tomado como protección ya que si por error se establece una hora para que se active la alarma anterior al momento al que debe de activarse el día en que se establece, el sistema jamás volverá a despertarse, lo que implicaría que debe realizarse el *reset* del sistema y obliga a desplazar a un operario hasta la posición del nodo. Al establecerse la alarma para el día siguiente este problema jamás ocurrirá ya que se asegura que la hora establecida para la alarma esta siempre por llegar.

Capítulo 4

Diseño del sistema mecánico

4.1 Introducción

En este capítulo se abordarán los aspectos relacionados con el diseño del sistema mecánico encargado de detectar la entrada de cartas en el buzón donde SIBA se instale.

Son dos los sistemas mecánicos a diseñar:

- Diseño 1: sistema de detección de la entrada de cartas dentro del buzón.
- Diseño 2: caja para la electrónica de procesamiento y comunicación del buzón.

Antes de diseñar las piezas, ha de tenerse en cuenta que serán diseñadas en plástico haciendo uso de una impresora 3D, lo que implica que hemos de tener en cuenta las propiedades del material de fabricación de las impresoras. En este TFG el plástico empleado para el diseño de la pieza es *Acrylonitrile Butadiene Styrene* (ABS). Esto implica que las piezas diseñadas deberán tener un cierto grosor para que pueda ser suficientemente robusta como para tener suficiente durabilidad en la manipulación y la instalación.

Para el diseño 1 será necesario buscar la forma de tal forma que la pieza pueda acomodarse en cualquiera de los buzones en los que SIBA se va a instalar.

Al igual que el del sistema de detección, el diseño 2 de la caja que contendrá la electrónica deberá acomodarse a cualquier buzón de tal forma que la mecanización del buzón sea mínima. La caja en cuestión deberá contener el sistema de proceso, con su microcontrolador, sistema de alimentación y módulo de comunicaciones.

Además, los sensores de la puerta deberán poder ubicarse en el diseño 1 y conectarse de forma discreta con el diseño 2, que contiene microcontrolador, pudiéndose además acceder a éste con facilidad.

Por su parte la antena deberá ser instalada en hacia el exterior del buzón, para lograr la transmisión y recepción de los mensajes de forma efectiva.

Por último la caja del diseño 2 deberá de poder abrirse y cerrarse con facilidad, de tal forma que no sea necesario desmontarla para acceder a su interior.

Con todas estas especificaciones, se aborda en los siguientes apartados la creación de los dos diseños indicados.

4.2 Diseño 1: sistema de detección

A la hora de realizar el diseño 1 en primer lugar se plantea la forma que permita fijar la pieza a la puerta del buzón. Todos los buzones cuentan con dos varillas roscadas soldadas a ambos lados de la abertura, en el interior de la puerta. Estas varillas son las encargadas de dar soporte a la pestaña que cubre la abertura del buzón y serán las encargadas de dar soporte también a la pieza diseñada. Es por esto que para diseñar esta primera pieza se crearán dos aberturas rasgadas que puedan acomodarse a las diferentes distancias que separan ambas varillas en los distintos tipos de buzones en los que SIBA se instale.

Lo siguiente que debemos garantizar, es que las cartas y los paquetes que se inserten por la ranura del buzón se vean forzados a activar la pestaña a diseñar sea cual sea la forma en que estas se inserten en el buzón.

A la entrada del buzón hay siempre una superficie por la que han de deslizarse las cartas antes de precipitarse al fondo del mismo. Esta superficie no tiene el mismo tamaño ni la misma inclinación en todos los buzones, por lo que la pieza a diseñar deberá de ajustarse a las características de dicha superficie en todos los casos.

El gran inconveniente que tiene esta superficie por la que las cartas se deslizan es la gran variabilidad de su longitud, llegando a existir una diferencia de hasta 28mm entre la más corta y la más larga. Por esta razón, para garantizar que las cartas introducidas en el buzón activan la pestaña en cualquier caso, lo ideal sería diseñar una pieza adaptable. Teniendo en cuenta las características del material, esta opción queda descartada.

Considerando lo anterior, se ha llegado a la conclusión de que la pestaña de activación de “entrada de carta” debe situarse en un emplazamiento fijo, de tal forma que quede en el punto más interno del buzón de la superficie por la que se desliza el correo.

En cualquier caso, para evitar que alguna de las cartas entre en el buzón sin activar la pestaña se amplía la superficie por la que desliza el correo de tal forma que el borde de esta nueva superficie quede enrasada con la vertical de la pestaña diseñada.

El siguiente aspecto a tener en cuenta es el sistema de sujeción de la pestaña. Esta se sostiene gracias a tres ejes, dos de ellos colocados en los extremos de la pestaña y uno tercero situado en el centro de la misma. Además, se cubre el eje centra con una superficie cilíndrica adherida a la pestaña, de tal forma que las cartas activen el mecanismo al entrar y permitan a su vez el giro de la pestaña. El diseño tiene además que asegurar que la pestaña pueda girar un ángulo ligeramente inferior a 90° para que la abertura no interfiera con el resto de elementos que situaremos en el interior del buzón (cables, sensores y electrónica).

Atendiendo a la descripción anterior se ha realizado el diseño 1, compuesto por tres piezas más pequeñas. En primer lugar se ha diseñado la denominada pieza fija, la cual da soporte a las otras dos piezas. En segundo lugar se ha diseñado la deslizadera, la cual tiene como función ajustar el diseño al buzón en el que vaya a instalarse. Finalmente se diseña la pestaña, la cual sera la encargada de detectar el correo entrante en el buzón.

4.2.1 Pieza fija

En la Figura 4.1 se muestra el aspecto de la pieza fija diseñada. Como se puede ver, la pieza cuenta con los tres orificios comentados que dan soporte al eje. A su vez, aparecen los orificios alargados que permiten ubicar la pieza haciendo uso de las varillas soldadas. Además en los laterales de la pieza pueden apreciarse las guías que evitan que la deslizadera se descuadre.

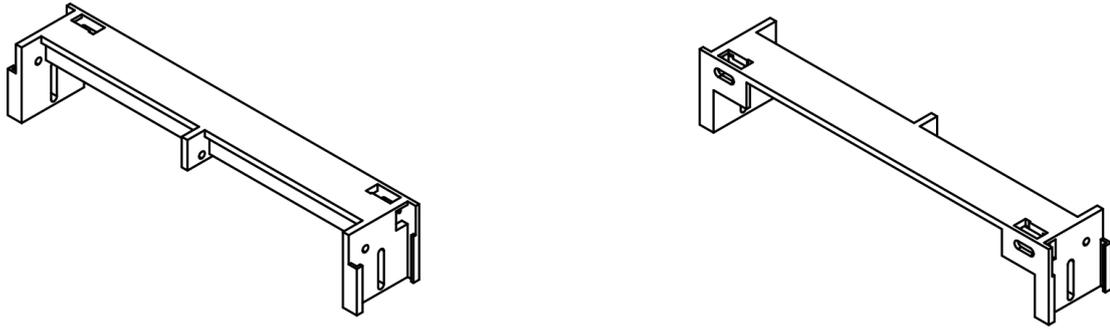


Figura 4.1: Perspectiva de la pieza fija

Cabe destacar la ranura en ambos laterales, que permite fijar la deslizadera en su posición final, así como los orificios cuadrados que aparecen en la parte superior y, en los laterales de la pieza, los cuales permiten introducir las herramientas necesarias para poder atornillar la pieza a las varillas.

Además, tal y como puede verse en la Figura 4.1, la pieza incorpora un nervio que da consistencia a la estructura, y que ha sido diseñado de forma que no limite el espacio por el que puede introducirse el correo.

Esta pieza es la más complicada de las tres que componen el diseño 1, pues no solo da soporte al resto de piezas sino que es la que tiene unas restricciones de espacio más ajustadas.

En la siguiente Figura (4.2) se muestran las vistas de la pieza, con todas las medidas necesarias para poder construirla. Todas las cotas de la figura se han expresado en milímetros.

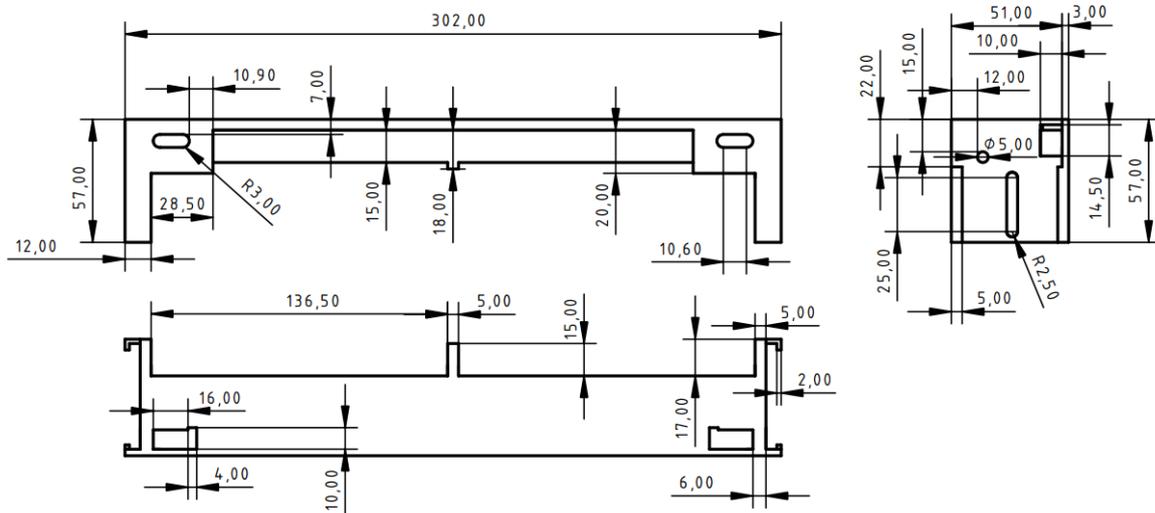


Figura 4.2: Vistas de la pieza fija

4.2.2 Deslizadera

En segundo lugar se muestra la pieza denominada deslizadera. Tal y como puede verse en las siguientes figuras (4.3 y 4.4, con sus vistas) esta pieza es sumamente sencilla, ya que su única función es proporcionar una base sobre la que puedan deslizarse las cartas dentro del buzón hasta una distancia que permita a la pestaña enrasar con el borde de la deslizadera.

Los orificios que aparecen en los laterales de la deslizadera son los que permitirán fijar su posición a la pieza fija.

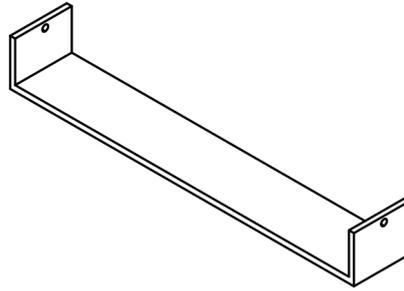


Figura 4.3: Perspectiva de la deslizadera

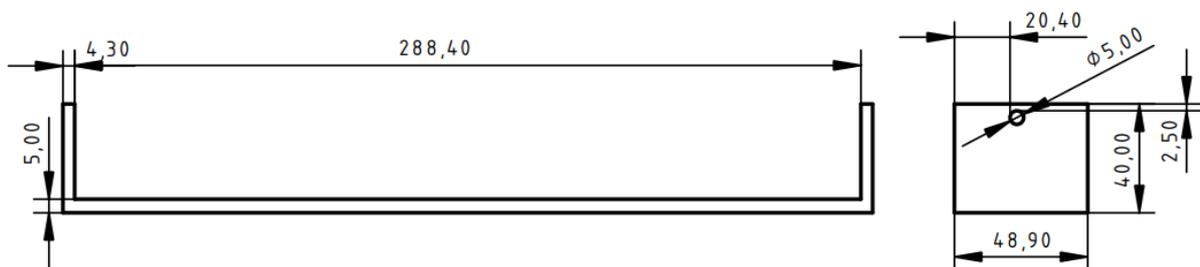


Figura 4.4: Vistas de la deslizadera

4.2.3 Pestaña

Finalmente, se muestra la última de las piezas que completa el diseño 1. Esta pieza, denominada pestaña, es la verdadera encargada de implementar la detección del correo entrando en el buzón. En la Figura 4.5 se muestra la perspectiva de la pieza.

Tal y como se aprecia en esta figura, la pieza incorpora una serie de ranuras con objeto de disminuir la tensión de fabricación y el peso final de la pieza, de modo que se disminuya además el precio de las piezas así como el tiempo de fabricación.

Además se ha incorporado un pequeño saliente en el lateral derecho de la pieza, de tal forma que se facilite la pulsación del switch al abrirse la pestaña.

Según se comentó anteriormente el eje aparece cubierto por una superficie semicilíndrica que permite el giro de la pestaña.

A continuación, en la Figura 4.6 se muestran las vistas de la pieza en cuestión.

4.2.4 Sistema completo. Limitaciones

Si combinamos todas las piezas explicadas en las secciones anteriores se obtiene el modelo de la Figura 4.7, en el cual se puede apreciar como la apertura de la pestaña deja espacio suficiente como para permitir que entre cualquier carta en el buzón, así como la deslizadera.

Cabe destacar que las piezas han sido diseñadas de tal forma que aparezca una única forma de montaje entre ellas, lo que permitirá que puedan ser encajadas con facilidad. Si atendemos a la distancia que separa los soporte de los ejes en la pieza fija puede verse que es de 136,5mm, mientras que la distancia entre

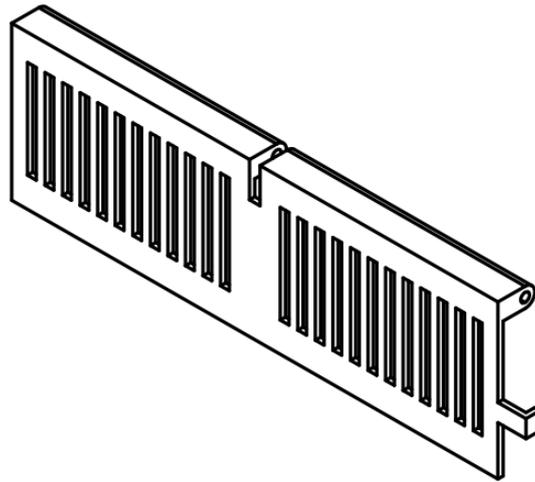


Figura 4.5: Perspectiva de la pestaña

ejes de la pestaña es de 135,5mm. Por su parte el espesor de la deslizadera es de 4,3mm, mientras que el espesor de la vía por la que ha de deslizarse la carta es de 5mm. Gracias a la precisión de la impresora, se han logrado respetar estas holguras de tal forma que las piezas encajan perfectamente entre sí.

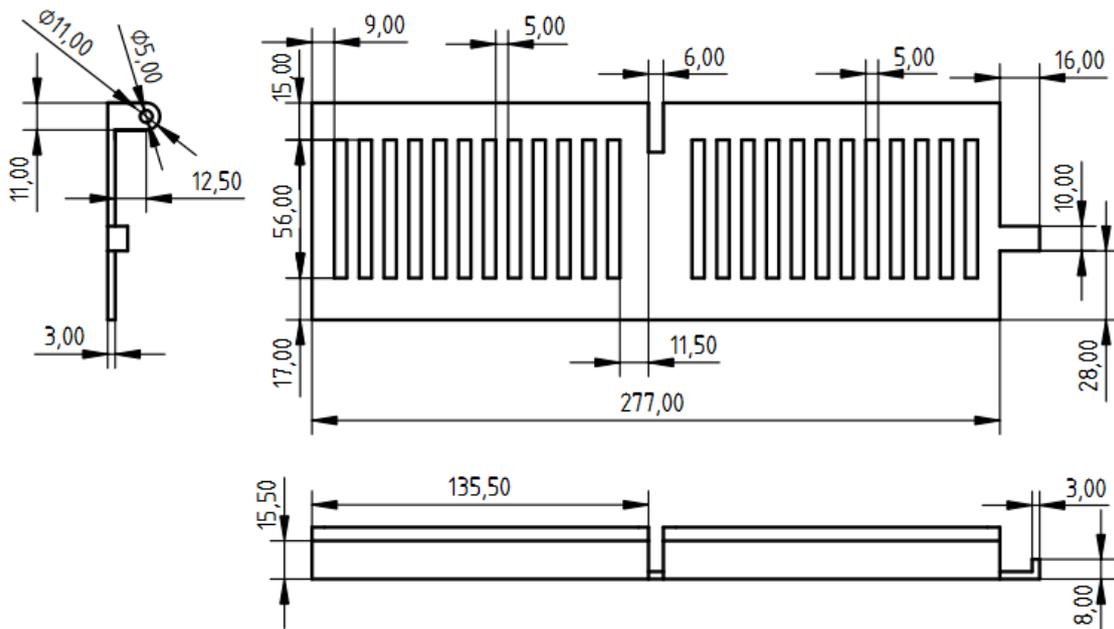


Figura 4.6: Perspectiva de la deslizadera

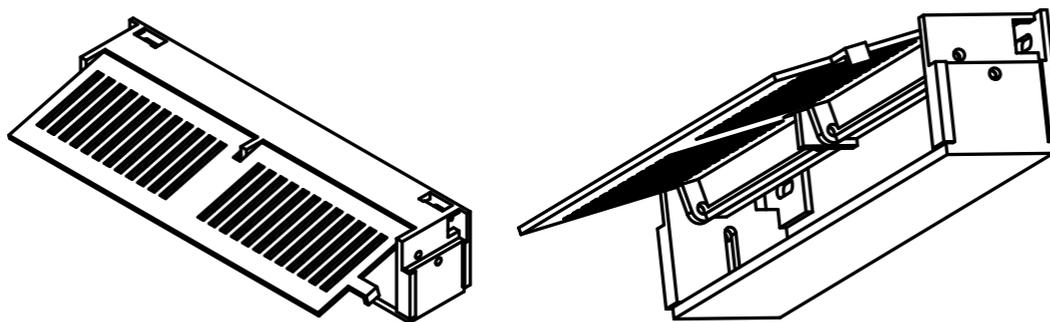


Figura 4.7: Perspectiva del diseño 1 completo

4.3 Diseño 2: caja para la electrónica

Como ya se ha comentado el capítulo 1, la caja para la electrónica deberá contener el sistema de alimentación global, el módulo de comunicaciones *Sigfox*, el microcontrolador, la salida de la antena y los posibles periféricos necesarios para conecten los sensores al microcontrolador.

El primer aspecto que debe tenerse en cuenta a la hora de diseñar de la caja es su ubicación en el interior del buzón: para tener un fácil acceso a la misma, sin que entorpezca el funcionamiento del resto del sistema, esta se ubicará en la parte superior del interior del buzón, adherida a su techo.

Además, con objeto de minimizar la mecanización del buzón y considerando que es necesario que la antena esté ubicada en el exterior del mismo para que pueda emitir con normalidad, se aprovechará el propio tornillo de fijación de la antena para fijar también la caja al techo del buzón.

De hecho, para facilitar el acceso a la electrónica, se ha diseñado una caja en la que, la abertura por la que se ha de introducir el tornillo de la antena, esté centrada en la base de la caja, de tal forma que el cuerpo de la caja quede suspendido en el interior del buzón una vez fijada la antena.

Por ello, todos los componentes electrónicos indicados para incluir dentro de la caja se colocarán en su tapa, de tal forma que al abrir la caja toda la electrónica se extraiga junto con la tapa de modo que se pueda trabajar con el sistema cómodamente.

Finalmente, para fijar la tapa en su lugar, esta cuenta con dos pestañas que encajan en los laterales del cuerpo de la caja. Además, una abertura en la tapa permitirá colocar un tornillo en el lado opuesto al que se han de encajar las pestañas que permitirá cerrar la caja de forma robusta.

Con este mecanismo incluido en el diseño 2 pueden realizarse operaciones de mantenimiento en la electrónica de forma cómoda, sin necesidad de retirar la caja al completo del buzón, pues con tan solo abrir la tapa ya se dispone de un cómodo acceso a toda la electrónica que constituye el cerebro de SIBA.

A su vez la tapa ha sido diseñada de tal forma que al cerrarla el interior de la caja quede dividido en dos compartimentos: uno para ubicar las baterías, y otro para ubicar la electrónica, mientras que el saliente que divide ambos compartimentos se emplea para ubicar el módulo de comunicaciones.

Finalmente, ha de señalarse que el cuerpo de la caja cuenta con una última abertura a través de la cual se puede acceder al conector USB de la tarjeta, para tareas de depuración en el procesador, sin necesidad de abrir la caja.

En las siguientes figuras 4.8, 4.9 y 4.10, se muestran la perspectiva y las vistas tanto del cuerpo de la caja como de la tapa.

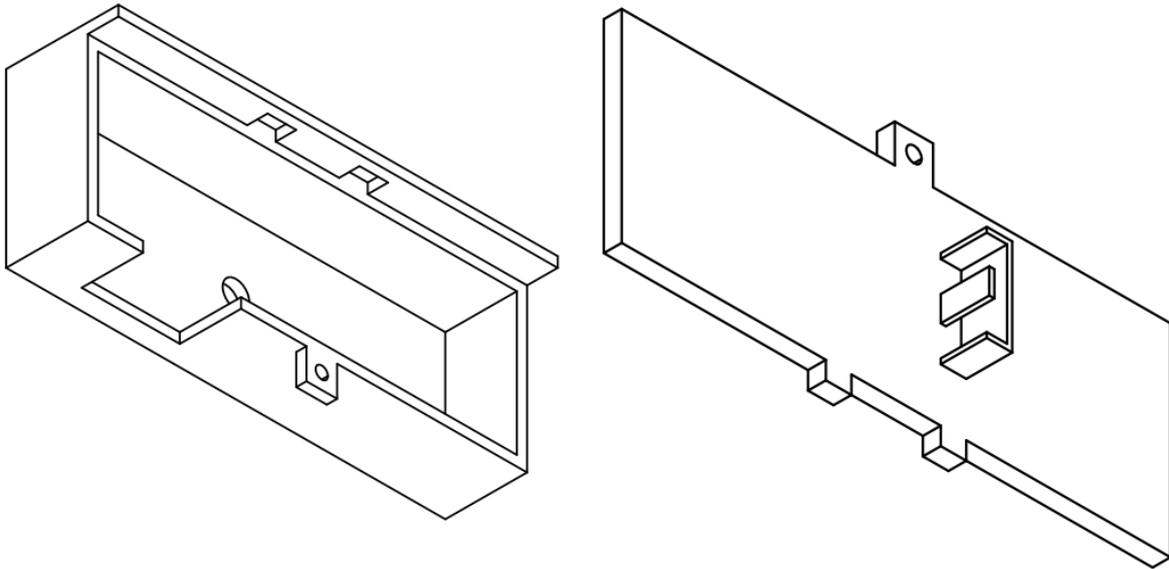


Figura 4.8: Perspectiva de la caja y la tapa

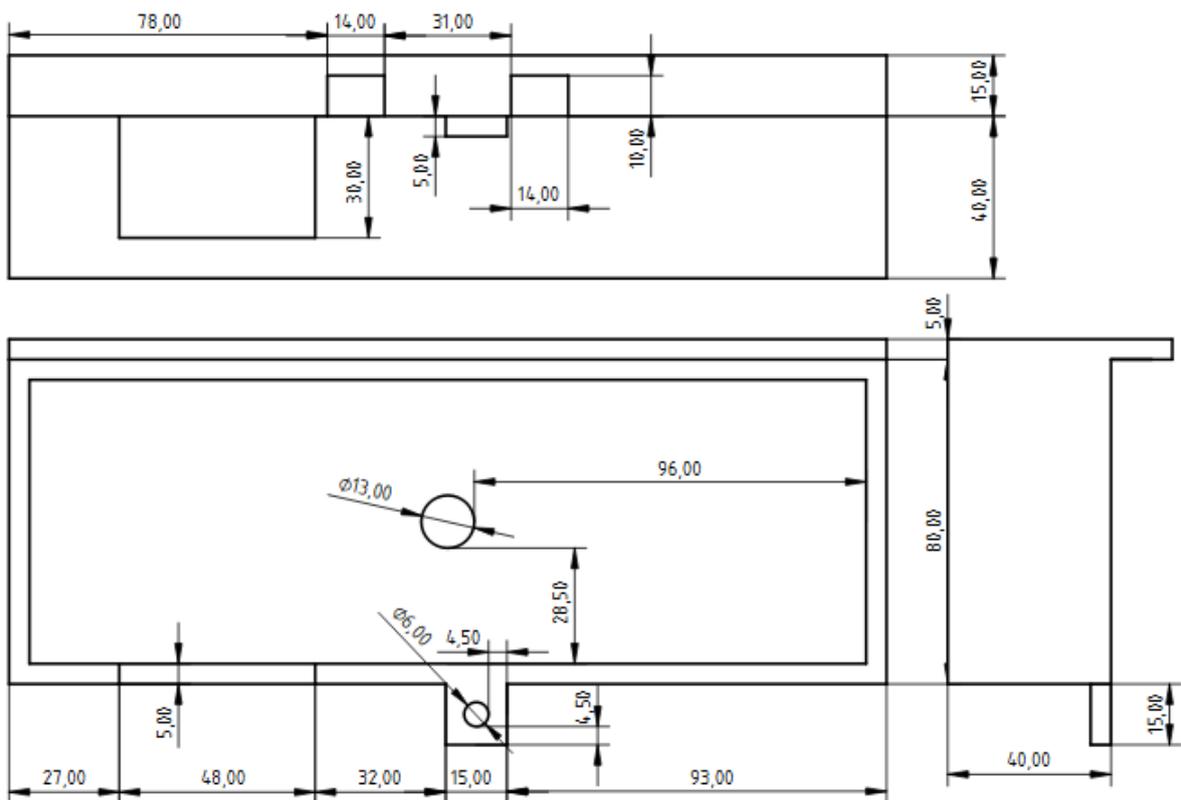


Figura 4.9: Vistas del cuerpo de la caja

Atendiendo a la Figura 4.10 se puede apreciar como la tapa queda dividida en dos partes, tal y como explica anteriormente. La mitad derecha, con un área menor se empleará para fijar el porta-pilas,

mientras que la mitad izquierda albergará la tarjeta perforada con toda la electrónica soldada. Por su parte el elemento en forma de C que separa ambas mitades albergará el módulo de comunicaciones *Sigfox* colocado en posición vertical.

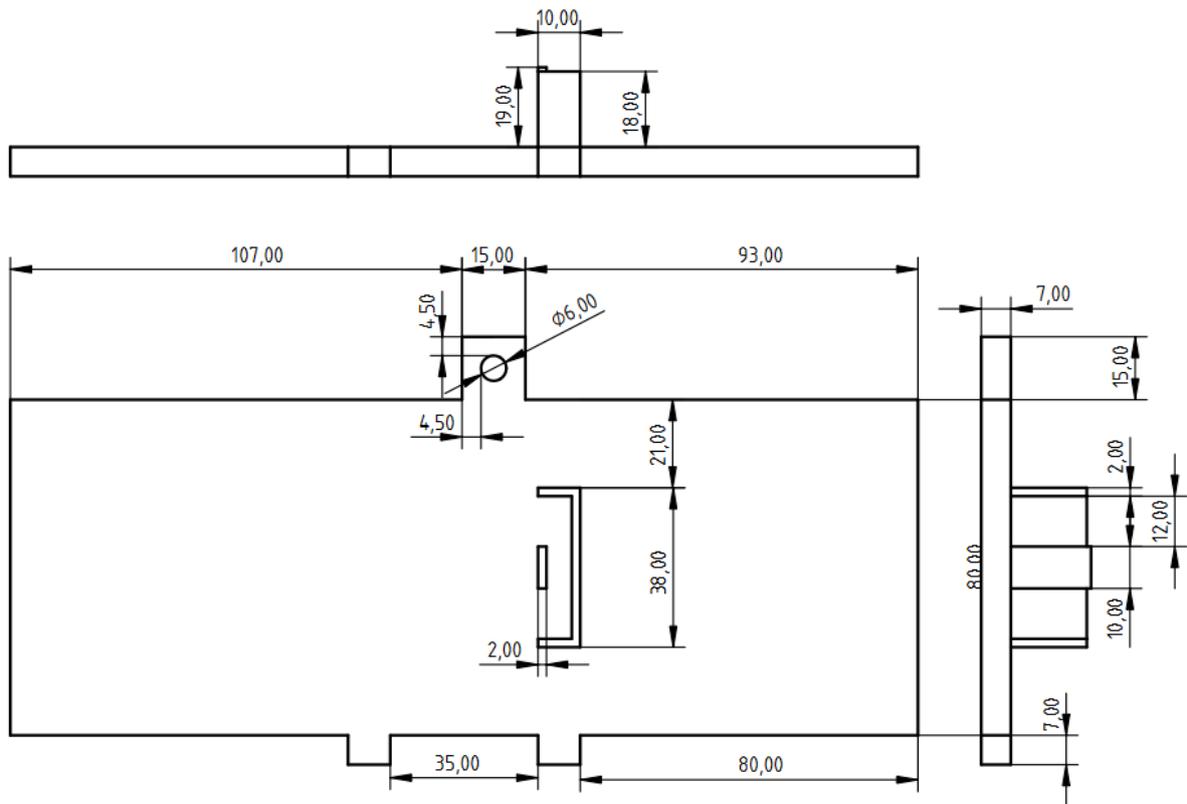


Figura 4.10: Vistas de la tapa

4.4 Piezas finales

En esta sección se muestran los resultados obtenidos tras realizar la impresión 3D de los dos diseños mecánicos descritos en este capítulo y ubicar las piezas resultantes, correctamente ensambladas en el buzón.

En lo que respecta a las piezas usadas para detectar la entrada de cartas en el interior del buzón (diseño 1, descrito en la sección 4.2), el resultado es el que se muestra en las siguientes imágenes:

- En primer lugar, tanto la pieza fija como la deslizadora se acomodan perfectamente al interior de todos los buzones, quedando ubicadas como aparece reflejado en la Figura 4.11.
- Por su parte, la Figura 4.12 muestra la pestaña ubicada en su lugar, la cual encaja a la perfección, al mismo tiempo que oscila en torno a su eje sin problema, facilitando la detección de cualquier entrada de correo.
- En la Figura 4.13, se muestra la pestaña encargada de detectar el correo abierta al máximo. Como se aprecia en la imagen, la abertura resultante es suficiente como para dejar entrar a cualquier paquete que quepa por la abertura exterior del buzón. Además, nótese como la abertura no alcanza los 90° , lo cual garantiza que el movimiento de la pestaña no se verá entorpecido por la caja que contiene el sistema electrónico ubicada en la parte superior en el interior del buzón.



Figura 4.11: Pieza fija junto con la deslizadera



Figura 4.12: Sistema de detección de correo completo

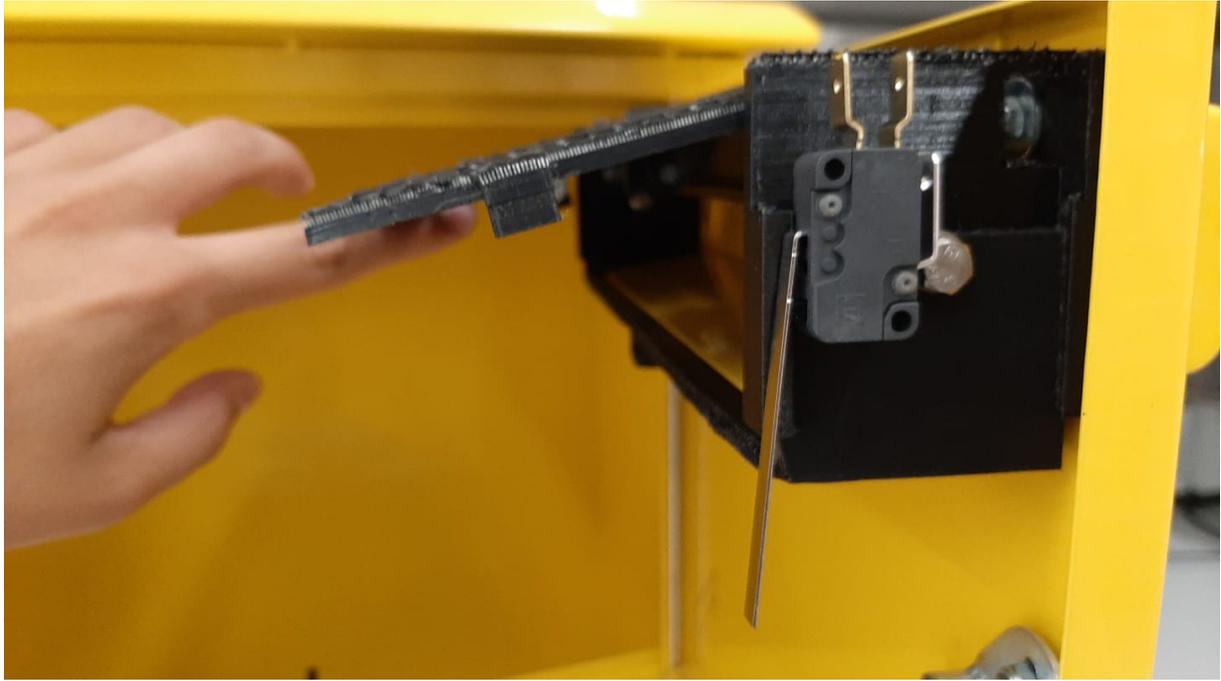


Figura 4.13: Abertura máxima del sistema de detección de correo

En lo que respecta a la caja que contiene la electrónica (diseño 2, descrito en la sección 4.3), el resultado puede apreciarse en la Figura 4.14.

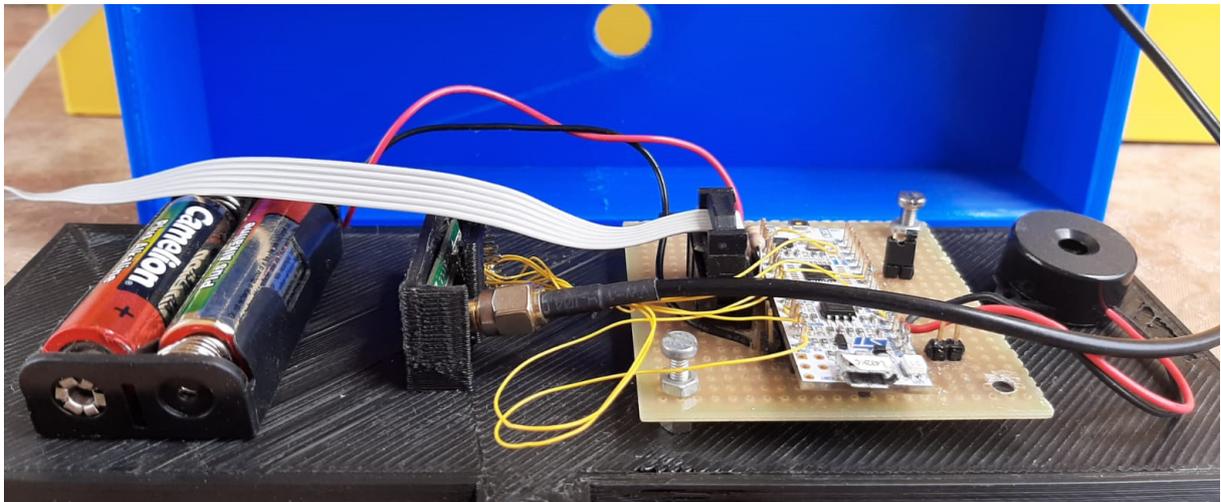


Figura 4.14: Caja diseñada para albergar la electrónica

Finalmente, cabe destacar que ha sido necesario imprimir todas las piezas en dos mitades, ya que su longitud supera las capacidades de las impresoras 3D disponibles. Para facilitar su posterior ensamblaje, se han adherido a las piezas diferentes nervios que permitan unir ambas mitades con tornillos. En cualquier caso, estas modificaciones no afectan a la funcionalidad de las piezas diseñadas, y pueden ser cambiadas si se emplean dispositivos de impresión 3D profesionales.

Es importante concluir indicando que en algunas de las piezas construidas puede verse una fina capa rugosa, defecto adicional introducido por la impresora. Esta capa es difícil de retirar sin dañar las piezas, por lo que se ha optado por mantenerla, ya que no afecta al funcionamiento del sistema, y de nuevo puede evitarse con el uso de sistemas de fabricación de mayor calidad.

Capítulo 5

Resultados

5.1 Introducción

En este capítulo se han intentado recopilar todos los resultados del desarrollo de SIBA, explicados a lo largo del documento, si bien los relacionados con el diseño final de la electrónica y la mecánica se han expuesto en capítulos anteriores.

Se mostrarán por tanto en las diferentes secciones, pruebas de instalación, configuración y puesta en marcha, con características concretas de interés, como la cobertura o el consumo del sistema final.

5.2 *Backend* de *Sigfox*

En esta sección se explican las posibilidades que ofrece el *backend* de *Sigfox*, el cual permite realizar el tratamiento de los mensajes, así como las configuraciones del módulo.

Concretamente, este capítulo se centra en aquellos aspectos de la aplicación que resultan realmente útiles para el proyecto SIBA.

5.2.1 Configuración de los nodos

Para poder establecer la comunicación con el operador, en primer lugar es necesario registrarse para obtener una cuenta de usuario en *Sigfox*. Esta cuenta es imprescindible para poder hacer uso de los servicios de los dispositivos, ya que será a esta cuenta a la que se vinculen los diferentes dispositivos.

Para vincular un dispositivo, es necesario conocer su ID así como su PAC. Estos códigos son los encargados de identificar el nodo así como de activarlo por primera vez respectivamente.

Estos códigos están almacenados en la propia memoria del módulo, por lo que para obtenerlos es necesario conectar el módulo mediante un terminal serie. Para facilitar esta tarea, los módulos incorporan una tarjeta como la que se muestra en la Figura 5.1. Esta tarjeta hace las veces de intermediario entre el módulo de comunicaciones y un puerto USB, de tal forma que se pueda establecer la conexión con el módulo de forma sencilla.

Tal y como se aprecia en la Figura 5.1, la tarjeta incorpora un terminal con 10 pines análogo al del propio nodo. Este terminal permite conectar el nodo y a tarjeta, pudiendo conectarse la tarjeta a un *Personal Computer* (PC) mediante USB. Además para que la conexión se establezca correctamente es necesario conectar también la antena. La Figura 5.2 muestra la conexión de los diferentes dispositivos.



Figura 5.1: Tarjeta empleada para conectar el nodo mediante USB



Figura 5.2: Conexión de los diferentes componentes necesaria para activar el nodo

Para obtener la información relativa a los códigos ID y PAC, ha de conectarse un terminal serie al puerto *COM* empleado para conectar el nodo. Después de conectarlo debe de configurarse el terminal serie. En el caso de este módulo la comunicación se establece según los siguientes parámetros.

- 9600 baud
- 8 bits de tamaño de palabra
- Sin bit de paridad
- Un bit de paridad
- Sin control de flujo

Tras haber establecido la conexión, para comprobar si la comunicación se ha establecido correctamente se envía el comando *AT*, al cual responderá el módulo con un mensaje *OK*, en caso de haberse establecido correctamente la comunicación. Finalmente, para obtener los parámetros ID y PAC que permitirán registrar el dispositivo es necesario enviar los comandos *AT\$I=10*; el cual devuelve el ID del dispositivo, y el comando *AT\$I=11*; el cual devuelve el PAC del dispositivo.

Después de haber obtenido la información, puede registrarse el dispositivo en cuestión para poder hacer uso de este. Para llevar a cabo esta tarea, se debe iniciar sesión en el portal de *Sigfox*, donde aparecerá la información que se muestra en la Figura 5.3.

Esta pestaña permite gestionar cada uno de los dispositivos vinculados a la cuenta. Para agregar un nuevo dispositivo, es necesario acceder a la opción *New*, en la que ha de indicarse el ID y el PAC obtenidos. Además de esto, el *backend* ofrece la posibilidad de poder asociar las coordenadas de la ubicación del nodo así como un nombre para dicho nodo.

El *backend* ofrece diferentes pestañas en las que se puede consultar y configurar el tráfico de mensajes de cada nodo así como las características del mismo. En primer lugar cuenta con una pestaña que resume la información del nodo, en la que aparecen datos relativos a la ID y el PAC del nodo, su ubicación,

Device - List

Count: 5 / 5

Communication status	Device type	Group	Id	Last seen	Name	Token state
<input type="radio"/>	Arduino_DevKit_1	Universidad de Alcalá	1D988C	2021-12-22 10:57:57	Arduino_DevKit_1-device	<input checked="" type="checkbox"/>
<input type="radio"/>	SIBA-Villalbilla	Universidad de Alcalá	9E21F3	2021-12-19 09:47:32	Villalbilla_ON-Semi_DevKit_4-device	<input checked="" type="checkbox"/>
<input type="radio"/>	Laboratorio	Universidad de Alcalá	9E21F8	2022-03-04 13:28:52	Labo_ON-Semi_DevKit_5-device	<input checked="" type="checkbox"/>
<input type="radio"/>	SIBA-Iriepal	Universidad de Alcalá	9E2278	2022-02-17 05:29:14	Iriepal	<input checked="" type="checkbox"/>
<input type="radio"/>	SNOC_DevKit_3	Universidad de Alcalá	C2F92F	2021-08-20 17:52:24	SNOC_DevKit_3-device	<input checked="" type="checkbox"/>

Figura 5.3: Pestaña *Device* en el *backend* de *Sigfox*

el último mensaje emitido, junto con la su *Received Signal Strength Indicator* (RSSI); su número de secuencia y su fecha asociados, la cuenta a la que está vinculado, etc.

La siguiente pestaña (5.4) permite consultar el historial de mensajes enviados por el nodo. En esta ventana aparecen los mensajes enviados, la fecha y la hora a la que han sido enviados, el número de secuencia, el indicador de fuerza de la señal recibida (RSSI), la ubicación del nodo emisor y la información referente al *uplink* y el *downlink* asociados al mensaje concreto.

A continuación, la siguiente pestaña de importancia reseñable dentro del menú *Display* que aparece en la Figura 5.3, es la que hace referencia a la gestión de los eventos. En este menú pueden configurarse diferentes opciones automáticas para notificar la recepción de un nuevo mensaje. Este menú permite reenviar cualquiera de los datos que se muestran en la pestaña del historial de mensajes a una dirección bien sea de correo electrónico o *Uniform Resource Locator* (URL), pudiéndose agregar además el texto que desee el usuario.

Finalmente la última de las pestañas dentro del menú "Display", es la denominada como Estadísticas. Como su nombre indica en esta pestaña pueden consultarse el número de mensajes, tanto *uplink* como *downlink* que han sido enviados y/o recibidos por el nodo a lo largo del tiempo, siguiendo el estilo que se muestra en la Figura 5.5.

Tal y como se aprecia en la Figura 5.5, no solo se resume la información de los mensajes, sino que también se resume la referente a los *Callbacks*, los cuales se explican en lo sucesivo, indicando el número de *Callbacks* enviados y el tiempo necesario para llevar a cabo esta tarea.

Por su parte si se accede al menú *Device Type*, pueden configurarse la información de los *downlink* así como a los *Callback*.

En lo que respecta a los *downlink*, el *backend* ofrece la posibilidad de configurar la información que recibirá el nodo. Según se ha descrito en capítulos anteriores esta información ha de tener una longitud fija de 8 bytes, y ha de estar codificado en hexadecimal. Sin embargo, esta información no tiene por que estar compuesta por datos fijos, sino que puede contener información referente a las condiciones en que se ha recibido el mensaje. Esta información es muy diversa y puede contener la hora y fecha a la que se ha recibido o la RSSI con la que se ha enviado entre algunos otros parámetros.

Además en esta pestaña el *backend* ofrece la posibilidad de decodificar la información de los mensajes recibidos de tal forma que la información se muestre como aparece en la Figura 5.4. Según se explica en

2022-02-25 10:02:53	1258	ba				
2022-02-25 07:29:39	1256	00011100000001410004 Temp_extrema: 0 Temp_pos: 1 Temp_valor: 17 Num_cartas: 0 Tension_extrema: 0 Tension_V/100: 321 Identificador: 4				
2022-02-24 13:50:27	1255	ba				
2022-02-24 10:06:29	1254	ba				
2022-02-24 07:29:46	1252	00011003000001430004 Temp_extrema: 0 Temp_pos: 1 Temp_valor: 16 Num_cartas: 3 Tension_extrema: 0 Tension_V/100: 323 Identificador: 4				

Figura 5.4: Extracto del historial de mensajes

los capítulos anteriores los mensajes han de ir codificados en hexadecimal o binario, lo cual dificulta la interpretación de los mismo una vez recibidos en el *backend*. Esta funcionalidad logra facilitar sumamente esta tarea. En la Figura 5.6 se muestra la vista de la pestaña comentada.

Cabe destacar la posibilidad que ofrece el *backend* en esta pestaña para informar por correo electrónico en caso de detectar alguna incidencia a la hora de transmitir el *downlink*, la cual, a su vez, se vera reflejada en el historial de mensajes (Figura 5.4).

Por último, la ultima pestaña reseñable, es la encargada de configurar los *Callback*. Los *Callback* permiten enviar cualquier información de los mensajes a un servidor externo, de tal forma que puedan ser tratados por otro software. Entre la información enviada, no solo es la referente a la hora, el consumo o la RSSI de cada mensaje sino que también pueden enviarse diferentes campos de la información contenida en el propio mensaje.

Estos campos deben coincidir con algunos de los campos empleados para decodificar la información de los mensajes. Esta funcionalidad es la que permite a esta aplicación poder seguir creciendo en un futuro cercano, de tal forma que el sistema creado en este proyecto pueda llegar a sincronizarse con los servidores de Correos permitiendo que el proyecto SIBA quede completamente integrado con los sistemas de la empresa.

5.3 Pruebas y estadísticas

Esta sección se centra principalmente en analizar tanto la cobertura como la autonomía del sistema. En el caso de la cobertura de la red se hará uso del mapa de cobertura proporcionado por el operador, mientras



Figura 5.5: Estadísticas en el *backend* de *Sigfox*

que para el análisis de la autonomía, se hará uso de los datos extraídos del sistema experimentalmente.

5.3.1 Cobertura del sistema

En este punto se va a analizar la cobertura que ofrece la red en el territorio de España aplicado a la ubicación de los diferentes buzones. Para analizarla se emplea una lista con todos los buzones de España, de la cual se extrae una muestra representativa suministrada por Correo. A partir de esta lista se obtiene la cobertura de su ubicación según el mapa de cobertura ofrecido por Sigfox. Este mapa categoriza la fortaleza de la señal en cada ubicación según el número de estaciones que dan cobertura a la misma, estando dicho número comprendido entre 0 y 3. A partir de los datos de cada buzón de la muestra se extraerá la cobertura media del territorio.

Realizando el estudio sobre 100 muestras, se han obtenido los resultados mostrados en la Tabla 5.1. Como se puede ver en ella, casi el 70 % del territorio nacional tiene una cobertura que puede calificarse de excelente. Además el 80 % del territorio tiene una cobertura calificada como buena.

The screenshot shows a configuration interface for a Sigfox device. It is divided into three main sections:

- Device type information:** Includes fields for Name (SIBA-Villaibilla), Description (DevKit 4 (ON Semiconductor)), Keep-alive (in minutes) (0), Subscription automatic renewal (checked), Contracts (universi_ad7d_18c65), and Alert email.
- Downlink data:** Includes Downlink mode (DIRECT), a note about documentation, and Downlink data in hexa (time)09300010.
- Payload display:** Includes a note about parsing modes, Payload parsing (Custom grammar), and Custom configuration (Temp_extrema:uint:8 Temp_pos:uint:8 Temp_valor:int:8 Num_c).

Buttons for 'Ok' and 'Cancel' are located at the bottom of the interface.

Figura 5.6: Configuración del *downlink* y la decodificación de los mensajes en el *backend* de *Sigfox*

Tabla 5.1: Cobertura de Sigfox en España

Numero de estaciones	Porcentaje del territorio
0	10 %
1	10 %
2	11 %
3	69 %

En conclusión, según los datos anteriores, puede decirse que la elección de esta red IoT es una buena, ya que se logra cubrir la inmensa mayoría del territorio con un coste muy inferior a una red GSM.

5.3.2 Autonomía del sistema

Para estimar la vida útil de las baterías se va a emplear la herramienta CubeMX [17], proporcionada por el fabricante del microcontrolador empleado. Esta herramienta permite seleccionar las baterías empleadas así como la secuencia entre los diferentes modos de consumo en los que ingresa el propio controlador. Además ofrece la posibilidad de añadir consumos adicionales para cada uno de los tramos de la secuencia. En estos consumos se incluye el consumo estimado de los diferentes periféricos empleados, pudiendo obtener de esta forma una estimación del consumo del sistema completo.

Dado que la herramienta contempla el consumo del microcontrolador, solo es necesario estimar de forma aproximada el consumo de los diferentes periféricos. Para ello hemos de diferenciar entre dos consumos diferentes, el consumo durante el tiempo en el que la tarjeta se encuentra en modo de bajo consumo y el consumo durante el tiempo en el que la tarjeta no se encuentra en dicho modo durante un día entero, ya que este es el periodo mínimo del ciclo de consumo pues puede suponerse que la potencia consumida es equivalente en días diferentes.

Para realizar esta simulación de la forma más verosímil posible, se ha simulado asumiendo que el consumo es el del microprocesador, ya que tras realizar las modificaciones pertinentes en el esquema eléctrico, tanto los reguladores como el *ST-Link* quedan aislados

Excluyendo el microcontrolador, las diferentes fuentes de consumo se resumen en la tabla 5.2, en la que se incluyen los tramos durante los cuales consumen potencia.

Tabla 5.2: Consumo de los sensores y del módulo de comunicaciones

Fuente de consumo	Tramo	Consumo
Switches	Todos los tramos	15 μ A
Módulo de comunicaciones	Modo Run	*
Zumbador	Modo Run al abrir la puerta	10 mA

* Ver tabla 5.3

En el caso del módulo de comunicaciones la casuística es más amplia que en el caso de las otras fuentes de consumo, ya que dependiendo del mensaje que vaya a enviarse, de la longitud del mismo y de si se espera recibir *downlink* o no el consumo es diferente. En este caso concreto solo existen dos tipos de mensajes. Un mensaje con una longitud máxima en el que se solicita *downlink* y un mensaje de un byte sin recepción de *downlink*. La información proporcionada por el fabricante para estos dos tipos de mensaje se resume en la tabla 5.3.

Tabla 5.3: Consumo de los mensajes

Tipo de mensaje	Consumo
Mensaje completo con <i>downlink</i>	25.5 mA
Mensaje buzón abierto	13.4 mA

Considerando todos estos consumos anteriores, puede determinarse que el consumo adicional al del propio controlador durante la emisión del mensaje diario es de 25.5 mA. El consumo durante la emisión de el mensaje que indica la apertura de la puerta es de 13.4 mA y, finalmente, el consumo adicional durante el tiempo que permanece en modo de bajo consumo es de 15 μ A.

Cabe destacar que durante el tramo en el que el sistema permanece dormido no se tienen en cuenta el consumo del zumbador ni el del módulo de comunicaciones. Esto es debido a que la línea que conecta el zumbador se encuentra en drenador abierto durante este tramo por lo que puede despreciarse.

Por su parte, el módulo de comunicaciones entra en modo UBC durante este tramo de funcionamiento, y tal y como indica el fabricante, el consumo de dicho módulo es de 180 nA por lo que también puede despreciarse frente a los 15 μ A que consumen los switches. Por todo esto se ha establecido el *timeline* que se muestra en la Figura 5.7.

Como se puede ver en la Figura 5.7 se muestra un cronograma en el cual aparecen 4 picos de consumo. El primero de ellos representa el arranque del sistema en el cual se configuran todos los periféricos antes de que ingrese en el modo UBC. El segundo de los flancos representa el consumo derivado del envío del mensaje periódico. El siguiente representa la corriente consumida al retirar el correo del interior del buzón y el último de los flancos concentra el consumo de corriente que se produciría al introducir 10 cartas en el buzón ya que este se estima que puede ser el número de cartas medio que reciba diariamente.

Es preciso indicar que en el último de los flancos no intervienen ni el módulo de comunicaciones ni el zumbador, por lo que el pico de consumo se debe al hecho de despertarse para llevar a cabo la detección del correo.

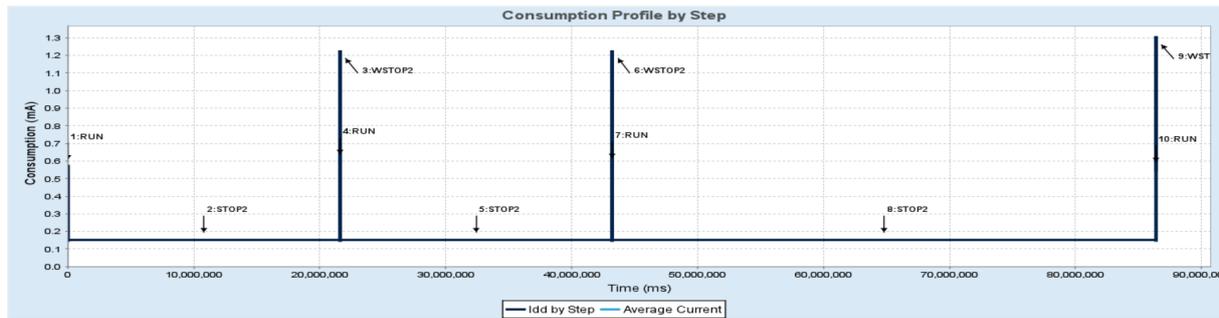


Figura 5.7: Cronograma del consumo estimado del sistema

Tras establecer los datos del consumo estimado, se ha calculado gracias a esta herramienta la información que se resume en la tabla 5.4

Tabla 5.4: Estimación de la vida útil del sistema

Tensión de alimentación	3 V
Batería	Alcalina AAA (1250 mAh)
Consumo medio	152.13 μ A
Vida útil	10 meses, 26 días 20 horas

Como se desprende de la tabla anterior, la autonomía lograda es idónea para esta aplicación, ya que el sistema puede sobrevivir durante casi 11 meses, un lapso de tiempo mucho más que aceptable incluso si la vida útil se ve levemente reducida debido a las condiciones climáticas a las que se enfrente.

Durante las pruebas realizadas se han recabado los datos diarios que se reflejan en la gráfica de la Figura 5.8.

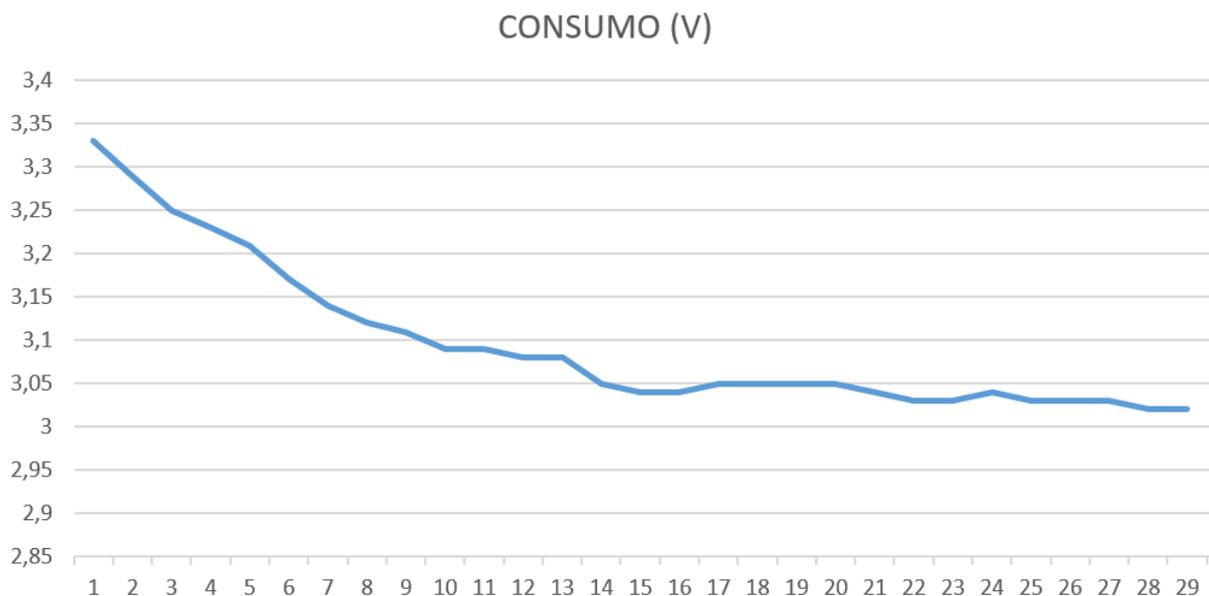


Figura 5.8: Evolución real de la carga de las baterías

Como se puede ver en la Figura 5.8 las baterías han sido capaces de mantener al sistema operativo durante un mes sin durante el tiempo en el cual la curva de descarga tiene una mayor pendiente, lo

que indica que las predicciones realizadas con la herramienta son un buen reflejo de la realidad. Si se asume que la pendiente de descarga es prácticamente constante entre los 3 voltios y los 2.2 voltios, puede estimarse que la autonomía real del sistema según los datos experimentales es de unos 8 meses, lo cual se aproxima mucho a las predicciones previamente realizadas empleando la herramienta CubeMX.

5.4 Problemas encontrados

En esta sección se van a comentar los problemas encontrados durante el desarrollo del sistema así como las soluciones adoptadas para paliarlos.

En primer lugar el primero de los problemas que fue necesario solventar, fue el hecho de que las piezas diseñadas para dar forma al sistema mecánico eran demasiado voluminosas para poder imprimirlas con las impresoras disponibles.

Para solucionar este problema, se optó por dividir las piezas en dos mitades e incluir en su diseño algunos bordes adicionales para que pudiesen ser atornilladas. Estos elementos adicionales no intervienen en absoluto en la mecánica del sistema, por lo que sigue cumpliendo su función. En concreto, estos elementos adicionales se introdujeron en la pieza fija y la deslizadera, pues son estas las piezas que permitían su anexión sin afectar a su función al mismo tiempo que se hacía posible su impresión.

En el caso de la pestaña, no es posible incluir estas modificaciones, ya que si se introduce el soporte para atornillar ambas mitades o bien entorpece la entrada del correo, o bien impide obtener una cara de la pieza completamente plana sobre la que poder comenzar la impresión. Considerando estas premisas se optó por dividir la pieza en dos mitades y unir las después con adhesivo. La contrapartida de este tipo de unión es que no es desmontable y el adhesivo debe de ser compatible con el tipo de plástico empleado en la impresión, aunque se logra que la pieza cumpla su función sin mayores inconvenientes.

En la Figura 5.9 se muestran las perspectivas de una de las mitades de la deslizadera y de la pieza fija, pues son estas las piezas en las que fue necesario introducir los soportes para atornillarlas. En ambas figuras aparecen marcadas en verde las superficies de unión adicionales donde deben de colocarse los tornillos.

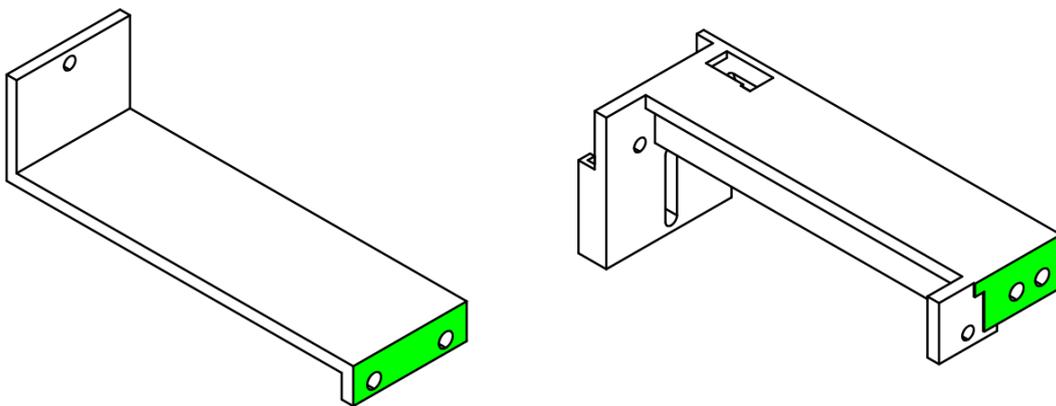


Figura 5.9: Perspectiva de las piezas partidas. La zona verde indica la zona de unión

El segundo problema encontrado es mucho más sofisticado que el anterior. El problema se halló al realizar las pruebas de campo en ubicaciones reales. El prototipo se probó en el laboratorio y operó correctamente, sin embargo, al realizar las pruebas de campo el dispositivo comenzó a funcionar de forma anómala, ya que tras despertarse para enviar el mensaje diario la aplicación colapsaba sin motivo

aparente. Tras investigar el problema se determinó que se trataba de un problema en la comunicación serie del microcontrolador con el módulo de comunicaciones. Dado que la fuente de reloj del microcontrolador no es demasiado precisa, sumado al error que introduce la temperatura en el sistema, la comunicación entre ambos dispositivos se corrompía.

Para evitar el problema, la primera solución consistió en compensar el desfase de velocidad de la fuente de reloj modificando la velocidad del puerto serie. Según los datos del fabricante, la frecuencia de la fuente de reloj empleada podía verse reducida en un 2% con respecto a su valor nominal como consecuencia de la temperatura. Tras analizar el problema la velocidad del puerto serie se fijó a 9700 bauds, lo cual dista de ser la mejor solución, pero permitió al sistema operar correctamente sin incrementar el consumo, lo que además sirvió para modelar la dependencia de la frecuencia con la temperatura. Para modelar dicha dependencia se han estudiado los datos de temperatura leídos por el sensor que incorpora el microcontrolador, así como el desfase entre la hora a la que debía despertarse el sistema y la hora real a la que el sistema se despertaba del modo UBC.

Para ilustrar la dependencia se muestra la siguiente gráfica (5.10). En esta Figura se muestra el retraso medio en minutos para una misma temperatura leída por el sensor. Los datos reflejados en la gráfica no deben ser tomados como referencia, ya que la resolución de la medida de temperatura es de 1°C. Además, únicamente se ha tomado una medida de temperatura al día sin considerar la evolución de la temperatura a lo largo del día, y la muestra de temperaturas no es representativa de las temperaturas a lo largo del tiempo ya que únicamente se han tomado las muestras de 25 días.



Figura 5.10: Dependencia de la frecuencia con la temperatura

Sin embargo la temperatura en cuestión coincide aproximadamente con la temperatura mínima de cada día y, aunque no pueda tomarse como una referencia, da una idea de si las temperaturas tomadas en el día en cuestión han sido más altas o más bajas. Esta gráfica tiene como único fin ilustrar de forma orientativa que existe una fuerte dependencia entre la temperatura y la frecuencia del reloj, lo que corrobora las hipótesis de las que se partía.

Una vez identificado el problema, la siguiente solución considerada consiste en analizar las fuentes de reloj disponibles en el microcontrolador y seleccionar la más apropiada. En el caso del microcontrolador seleccionado, se dispone de un oscilador de cuarzo, el cual tiene una frecuencia muy baja, aunque más que suficiente para temporizar el puerto serie y el RTC, con la ventaja de ofrecer una gran estabilidad frente a cambios de temperatura y de tensión de alimentación.

Esta fuente puede sustituir al resonador empleado en algunos periféricos, de tal forma que se puede lograr una mayor precisión. Sin embargo, esta fuente de reloj no puede reemplazar por completo a la fuente empleada actualmente, ya que solo puede emplearse para sincronizar ciertos periféricos, lo cual implica que el sistema ha de operar con ambas fuentes de reloj. Además, el oscilador tiene un consumo mucho mayor que las otras fuentes de reloj, lo cual implica que en caso de emplear esta fuente, la autonomía del sistema se vería comprometida.

Finalmente, dado que la temporización del RTC no tiene por que ser demasiado precisa, se ha optado por mantener la primera solución, ya que experimentalmente se ha demostrado que tolera un amplio margen de temperaturas y no incrementa el consumo de energía.

Capítulo 6

Conclusiones y trabajos futuros

6.1 Conclusiones

En este trabajo se fijó por objetivo la creación de un sistema capaz de monitorizar el tráfico de correspondencia postal de un buzón mural real, instalado por la empresa Correos (C), con la especificación fundamental de optimizar en la medida de lo posible los recursos existentes de un sistema autónomo, de bajo mantenimiento y para su instalación en masa a lo largo de toda la red postal de la geografía española: el consumo y el coste del sistema.

Una vez construido el prototipo se puede concluir que el sistema cumple con las expectativas planteadas:

- En lo que respecta a la mínima modificación de los buzones existentes en la red postal, se considera que se ha logrado cumplir el objetivo, pues el sistema electrónico solo requiere de un orificio en la parte superior de cada buzón para poder ser instalado.
- En lo que respecta al consumo, se ha logrado que la autonomía del sistema propuesto sea de varios meses lo cual se considera un plazo suficiente como para que sea viable su mantenimiento.
- Además, la red de comunicaciones IoT sobre la que funciona el sistema ofrece una amplia cobertura en la mayoría del territorio español analizado, por un coste muy reducido, lo que contribuye a hacer el sistema aún más viable.
- Finalmente, el sistema ha logrado implementarse mediante componentes electrónicos sencillos, lo que no solo contribuye a su robustez, sino que implica un gran reducción en los costes de su producción, y facilita su fabricación en masa y su reemplazo.

Todos estos factores permiten concluir que los requerimientos iniciales han sido satisfechos, haciendo de la idea inicial una realidad.

6.2 Trabajos futuros

El proyecto desarrollado en esta primera fase se centra en obtener un prototipo que permita evaluar la viabilidad del sistema propuesto. Una vez analizados los resultados, este proyecto puede dar lugar a los siguientes trabajos:

- Fabricación de un sistema mecánico más robusto, en el que las piezas fabricadas sean de metal, lo que dotaría al sistema de una mayor robustez.
- Fabricación del sistema electrónico periferia en PCB, de tal forma que una única tarjeta permita incluir toda la electrónica salvo los sensores y el módulo de comunicaciones. Además esta tarjeta podría incluir conectores estándar que permitieran acceder tanto a los sensores como al módulo de comunicaciones de forma más cómoda.
- Implementación de una aplicación que permita compartir los datos recabados con un sistema informático de gestión de la correspondencia postal, lo que podría permitir automatizar varias tareas de la empresa de servicios encargada de la recepción y envío a destino, como la gestión de las rutas de recogida en función de los datos proporcionados por cada nodo.
- Implantación del sistema en otros tipos de buzones diferentes al usado.

Capítulo 7

Pliego de condiciones

A continuación se detallan de forma concreta los elementos empleados para llevar a cabo la realización de un sistema prototipo desarrollado, enumerando tanto los componentes electrónicos, como el resto de recursos hardware y software necesarios.

7.1 Elementos físicos

- Componentes electrónicos
 - Tarjeta perforada 25x25 cm.
 - Microcontrolador STM32L432KC.
 - Zócalo para conectar cable plano de 10 líneas.
 - Cable plano de 10 líneas.
 - Dos baterías alcalinas 1.5 V y 2500 mA h.
 - Sensor de proximidad magnético.
 - Imán.
 - Microswitch activado por palanca.
 - Porta-pilas para dos baterías de 1.5 V.
 - Modulo de comunicaciones Sigfox.
 - Antena para exteriores de 868 Mhz.
 - Zumbador piezoeléctrico activo.
 - Resistories variados.
 - Pines de inserción.

- Instrumental
 - Impresora 3D Zortrax.
 - Estación de soldadura y bomba para desoldar.
 - Taladro eléctrico.
 - Caja de herramientas básicas.

- Osciloscopio, polímetro y fuente de alimentación.
- Monitor de bus UART.

- Portátil MSI Katana, para depuración, pruebas y documentación.

7.2 Software

- Modelado de piezas en 3D: FreeCAD [\[18\]](#)
- Paquetes de diseño de circuitos electrónicos: Altium.
- Procesador de textos: Word, OverLeaf (Latex).
- Programación del microcontrolador: Keil μ vision 5, Cube MX.
- Sistema Operativo: Windows 11.

Capítulo 8

Presupuesto

En este capítulo se incluye el coste de los diferentes recursos empleados para llevar a cabo el proyecto, clasificándolos según el tipo de recurso.

8.1 Recursos hardware

Todos los elementos hardware así como su precio y las unidades empleadas se incluyen en la tabla 8.1.

Concepto	Precio por Unidad	Cantidad	Subtotal
Módulo MCU STM32L432KC	11,01 €	1	11,01 €
Módulo SIGFOX: ON-SEmi DVK-SFEU-1-GEVK	98,30 €	1	98,30 €
Antena Anti-Vándalos PUKDB	30,93 €	1	30,93 €
Microswitch dos contactos NC-NO	5,48 €	1	5,48 €
Switch magnético dos contactos NC-NO	6,86 €	1	6,86 €
Zumbador	4,86 €	1	4,86 €
Conectores (IDC)	3,30 €	2	6,60 €
Cables 6-10 HILOS (1M)	1,67 €	1	1,67 €
Filamentos para 3D-PRINTER (mecanizado)	34,99 €	1	34,99 €
Tornillería y similares	2,00 €	1	2,00 €
Portapilas 2*AA	2,88 €	1	2,88 €
Pila alcalina AA	0,50 €	2	1,00 €
TOTAL			207,58 €

Tabla 8.1: Recursos hardware usados

8.2 Recursos de desarrollo y pruebas

Para poder llevar a cabo el desarrollo del proyecto son necesarios diferentes recursos software, instalados en un ordenador portátil. El coste de estos recursos se ha calculado como la parte proporcional a su tiempo de uso, con respecto al coste del propio recurso especificado para 2 años. Los recursos empleados se resumen en la siguiente lista.

- Altium Designer para la implementación del diseño de circuitos electrónicos.

- Microsoft office para la redacción de la documentación técnica.
- μ Vision de Keil para programación de microcontroladores.
- FreeCAD. Este trata de un software libre y gratuito empleado para realizar el diseño de piezas en tres dimensiones.
- Ordenador portátil (MSI Katana) de prestaciones medias para la instalación de los tres paquetes software anteriores.

En la tabla 8.2 se incluyen dichos recursos con sus respectivos costes ponderados.

Concepto	Precio por Unidad	Coficiente	Subtotal
μ Vision	400,00 €	0,0833	33,32 €
Microsoft office	122,4 €	0,0833	10,20 €
FreeCAD	0,00 €	0,0833	0,00 €
Altium Designer	7140,72 €	0,0833	595,06 €
Portatil MSI Katana	1300 €	0,0833	108,33 €
TOTAL			746,91 €

Tabla 8.2: Recursos de desarrollo y pruebas

8.3 Recursos humanos

A continuación se incluyen los costes de recursos humanos necesarios. El trabajo se desarrolla por un ingeniero técnico trabajando a tiempo completo.

Concepto	Precio por hora	Cantidad de horas	Subtotal
Ingeniero Técnico	50,00 €	500	25000,00 €
TOTAL			25000,00 €

Tabla 8.3: Recursos humanos

8.4 Presupuesto de ejecución material

El presupuesto de ejecución material se resume como la suma de los subtotales anteriores, la cual se resume en la tabla 8.4.

Concepto	Subtotal
Recursos hardware	207,58 €
Recursos desarrollo	746,91 €
Recursos humanos	25000,00 €
TOTAL	25954,49 €

Tabla 8.4: Presupuesto de ejecución material

8.5 Importe de la ejecución por contrata

Los costes de la ejecución por contrata deben incluir los gastos derivados del uso de las instalaciones donde se ha llevado a cabo el trabajo, las cargas fiscales, los gastos financiero, las tasas administrativas y las obligaciones de control del proyecto.

Dicho gasto se asume estableciendo un recargo sobre el coste del importe del presupuesto de ejecución material. Dicho recargo se propone equivaler al 22 % de dicho importe.

Concepto	Subtotal
22 % del coste total de ejecución de material	5707,74 €

Tabla 8.5: Importe de ejecución por contrata

8.6 Honorarios facultativos

Se fija en este proyecto un porcentaje del 7 % sobre el coste total de ejecución por contrata.

Concepto	Subtotal
7 % del coste de ejecución por contrata	399.54 €

Tabla 8.6: Importe de los honorarios facultativos

8.7 Presupuesto Total

A continuación, en la tabla 8.7, se realiza la suma de todos los conceptos del presupuesto tenidos en cuenta en los anteriores apartados.

Concepto Precio	Subtotal
Presupuesto de ejecución material	25954,49 €
Importe de la ejecución contratada	5707,74 €
Horarios facultativos	399,54 €
TOTAL (sin IVA)	32051,57 €
IVA (22 %)	7051,35€
TOTAL	39113,11 €

Tabla 8.7: Importe del presupuesto total del proyecto

Bibliografía

- [1] “Gomez c, veras jc, vidal r, casals l, paradells j. “a sigfox energy consumption model”. sensors. 2019; 19(3):681.” <https://www.mdpi.com/1424-8220/19/3/681>.
- [2] “Sensores reed.” <https://www.eleinmec.com/article.asp?23> (accedido el 04/06/2022).
- [3] “Ax-sip-sfeu, data sheet.” <https://www.farnell.com/datasheets/2688496.pdf> (accedido el 10/06/2022).
- [4] “Antena ant-pukdb.” https://www.mouser.es/datasheet/2/975/RFSO_S_A0009835151_1-2576152.pdf (accedido el 04/06/2022).
- [5] Correos, “Página web de sociedad estatal correos y telégrafos,” <https://www.correos.es/es/es/empresas>.
- [6] “What is sigfox?” <https://build.sigfox.com/sigfox> (accedido el 03/06/2022).
- [7] “Iot global network. sigfox m2m network access, 2016,” <https://www.iotglobalnetwork.com/products/single/id/571/sigfox-m2m-network-access> (accedido el 03/06/2022).
- [8] “Kais mekki, eddy bajic, frederic chaxel, fernand meyer. “a comparative study of lpwan technologies for large-scale iot deployment”. ict express, volume 5, issue 1, 2019, pages 1-7, issn 2405-9595.” <https://doi.org/10.1016/j.ict.2017.12.005>.
- [9] “U. raza, p. kulkarni and m. sooriyabandara, ”low power wide area networks: An overview,in iee communications surveys tutorials, vol. 19, no. 2, pp. 855-873. 2017.” <https://doi.org/10.1109/COMST.2017.2652320>.
- [10] “Modelado y evaluación de la tecnología sigfox para ns3. adrián munera sánchez. tfg, escola tècnica superior d’enginyeria informàtica, universitat politècnica de valència. julio de 2018.” <https://riunet.upv.es/bitstream/handle/10251/107791/MUNERA%20-%20Modelado%20y%20evaluaci%C3%B3n%20de%20la%20tecnolog%C3%ADa%20SigFox%20para%20NS3..pdf?sequence=1&isAllowed=y> (accedido el 26/01/2022).
- [11] “Sigfox radio configuration.” <https://support.sigfox.com/docs/radio-configuration> (accedido el 26/01/2022).
- [12] “Get to know sigfox.” <https://build.sigfox.com/study> (accedido el 26/01/2022).
- [13] “Sigfox technical overview. report january 2018.” <https://www.avnet.com/wps/wcm/connect/onesite/03aebfe2-98f7-4c28-be5f-90638c898009/sigfox-technical-overview.pdf?MOD=AJPERES&CVID=magVa.N&CVID=magVa.N&CVID=magVa.N> (accedido el 26/01/2022).
- [14] “Sigfox connected objects: Radio specifications. february 2020.” <https://build.sigfox.com/sigfox-device-radio-specifications> (accedido el 26/01/2022).

-
- [15] “Hojas de características del microcontrolador: Stm32l432kc,” <https://www.st.com/en/microcontrollers-microprocessors/stm32l432kc.html#documentation> (accedido el 05/06/2022).
- [16] “Manual de usuario del microcontrolador: Stm32l432kc,” https://www.st.com/resource/en/user_manual/dm00231744-stm32-nucleo32-boards-mb1180-stmicroelectronics.pdf (accedido el 05/06/2022).
- [17] “Herramienta de programación cube mx,” <https://www.st.com/en/development-tools/stm32cubemx.html> (accedido el 04/06/2022).
- [18] “Herramienta de diseño de piezas 3d: Freecad,” <https://www.freecadweb.org> (accedido el 04/06/2022).

Universidad de Alcalá
Escuela Politécnica Superior



ESCUELA POLITECNICA
SUPERIOR



Universidad
de Alcalá