

UNIVERSIDAD DE MÁLAGA

**ESCUELA TÉCNICA SUPERIOR DE
INGENIERÍA DE TELECOMUNICACIÓN**



TRABAJO FIN DE MÁSTER

LUCES: Sistema inalámbrico de control de luminarias

**MÁSTER EN SISTEMAS ELECTRÓNICOS
PARA ENTORNOS INTELIGENTES**

MÁLAGA, 2021

JESUS MUÑOZ MARTÍNEZ

UNIVERSIDAD DE MÁLAGA

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN

LUCES: Sistema inalámbrico de control de luminarias

REALIZADO POR:

Jesús Muñoz Martínez

DIRIGIDO POR:

Francisco Javier Vizcaíno Martín

DEPARTAMENTO DE TECNOLOGÍA ELECTRÓNICA

TITULACIÓN: Máster en Sistemas Electrónicos para Entornos Inteligentes

PALABRAS CLAVE: LUCES, ILUMINACIÓN, MESH, PCB, RED, SENSORES.

RESUMEN:

Este proyecto consiste en el desarrollo hardware, software y documentación de un sistema de control de iluminación inalámbrico. Este sistema permite integrar una luminaria convencional dentro de un sistema de control inteligente conectado a Internet. Su uso está orientado a la iluminación de espacios públicos como calles y parques. El control de encendido y de los niveles lumínicos de las luminarias integradas en el sistema, está basado en temporizaciones horarias, luminosidad ambiente o solar y presencia de usuarios en la vía iluminada. El sistema avisa por correo electrónico en caso de fallo de luminaria y proporciona información del consumo eléctrico, luminosidad ambiente, uso de la vía, etc. La arquitectura está basada en una red Wifi mallada con motas que controlan las luminarias y un nodo central que hace de puente con internet y proporciona el acceso a la gestión del sistema. La verificación de los requisitos se ha realizado sobre los prototipos funcionales desarrollados. Concluir que este sistema destaca por proporcionar la reducción del consumo eléctrico e inicia la posibilidad de integración de sistemas de iluminación en las “Ciudades inteligentes”.

Málaga, octubre 2021

TITLE: "LIGHTS: Wireless luminaire control system"

KEYWORDS: LIGHTS, LIGHTING, MESH, PCB, NETWORK, SENSORS.

ABSTRACT:

This project consists of the development of hardware, software, and documentation of a wireless lighting control system. This system allows the integration of a conventional luminaire into an intelligent control system connected to the Internet. Its use is oriented to the lighting of public spaces such as streets and parks. The control of switching on and lighting levels of the luminaires integrated in the system is based on hourly timings, ambient or solar luminosity and the presence of users on the illuminated road. The system alerts by e-mail in case of luminaire failure and provides information on power consumption, ambient brightness, road use, etc. The architecture is based on a meshed WiFi network with motes that control the luminaires and a central node that bridges the Internet and provides access to system management. The verification of the requirements has been performed on the developed functional prototypes. To conclude that this system stands out for providing the reduction of electricity consumption and initiates the possibility of integration of lighting systems in "Smart Cities".

Málaga, October 2021

Agradecimientos

A Esther, mi mujer, por su apoyo durante todos estos años, a María, mi hija, por su paciencia y comprensión cuando quiere que su padre tenga más tiempo para jugar con ella. A mi madre Joaquina por infinita paciencia y mi padre Evelio, a mis hermanos que siempre me han animado y valorado y a los amigos, que muchos son entusiastas de esta parte de la ingeniería. A Francisco Javier, mi tutor, por su tiempo y guía en este trabajo, y al resto de profesores por el tiempo que hemos compartido en estos años de formación académica. Y por último a Coque, mi perro, por su compañía y cariño mientras realizaba este trabajo fin de máster.

Tabla de contenidos

SECCIÓN I: Especificación de requisitos y casos de uso

SECCIÓN II: Diseño e implementación

SECCIÓN III: Pruebas

SECCIÓN IV: Conclusiones y cierre

ANEXOS:

I: Esquemáticos y planos de fabricación de la PCB

II: Firmwares y código del panel de control.

III: Manual de usuario.

IV: Manual de instalación.

CD-ROM

Github: <https://github.com/jesussmunozmartinez/LUCES>

Especificación de requisitos y casos de uso

RESUMEN DE DOCUMENTO

Especificación de los requisitos y casos de uso del proyecto LUCES (Sistema inalámbrico de control de luminarias). Esta especificación incluye también los objetivos, alcance, directivas del proyecto, descripción de los interesados y usuarios finales del producto/sistema generado.

TÍTULO DOCUMENTO	Especificación de requisitos y casos de uso.
FECHA DE ENTREGA	30/09/2021
AUTOR/ES	Jesús Muñoz Martínez

COMENTARIOS	VERSIÓN	FECHA
Versión Inicial	1.0	06/07/2021

Índice

1	INTRODUCCIÓN	6
1.1	OBJETIVOS	6
2	DIRECTIVAS DEL PROYECTO	7
2.1	OPORTUNIDAD DE NEGOCIO	7
2.2	DESCRIPCIÓN DEL PROBLEMA	7
2.3	DESCRIPCIÓN DEL PRODUCTO	7
3	DESCRIPCIÓN DE PARTICIPANTES Y USUARIOS	8
3.1	PRELIMINAR	8
3.2	RESUMEN DE LOS PARTICIPANTES	8
3.3	RESUMEN Y ENTORNO DE LOS USUARIOS	8
3.3.1	Entorno de los usuarios	9
3.4	PERFILES DE LOS PARTICIPANTES	9
3.4.1	Director empresa cliente	9
3.4.2	Instalador	9
3.4.3	Comercial	10
3.4.4	Jefe de proyecto	10
3.4.5	Desarrolladores	10
3.5	PERFILES DE USUARIO	10
3.5.1	Comprador del producto	10
3.5.2	Personal de mantenimiento	11
3.5.3	Usuario de la zona iluminada por el sistema	11
3.6	ALTERNATIVAS Y COMPETENCIA	11
3.6.1	Interact City / IoT lighting	11
3.6.2	Lutron Caséta Wireless	11
4	REQUISITOS	12
4.1	DIAGRAMA GENERAL	12
4.2	PRECEDENCIA Y PRIORIDAD	13
4.3	REQUISITOS FUNCIONALES	14
4.3.1	R3.1 NodoCentral	14
4.3.2	MotaWireless	14
4.4	REQUISITOS NO FUNCIONALES	15
4.4.1	R1.1 Esquemáticos	15
4.4.2	R1.2 PCB	15
4.4.3	R1.3 ManualInstalación	15
4.4.4	R1.4 ManualUsuario	15
4.4.5	R2 Firmware	15

5	CASOS DE USO	16
5.1	ACTORES DEL SISTEMA	16
5.2	DIAGRAMA GENERAL	16
5.3	DESCRIPCIÓN TEXTUAL DE LOS CASOS DE USO	17
5.3.1	C1 Configuración	17
5.3.2	C2 Información	17
5.3.3	C3 Actualización firmware	18
5.3.4	C4 Control luminaria	19
5.3.5	C5 Medir	20
6	REFERENCIAS	22

Índice de figuras.

<i>Fig. 1. Diagrama general de requisitos</i>	12
<i>Fig. 2. Diagrama general de casos de uso.....</i>	16

Índice de tablas.

<i>Tabla 1.Descripción del problema.</i>	7
<i>Tabla 2.Descripción del producto.</i>	7
<i>Tabla 3.Resumen de los participantes.</i>	8
<i>Tabla 4.Resumen y entorno de los usuarios.</i>	9
<i>Tabla 5.Perfil director.</i>	9
<i>Tabla 6.Perfil instalador.</i>	9
<i>Tabla 7.Perfil comercial.</i>	10
<i>Tabla 8.Perfil jefe de proyecto.</i>	10
<i>Tabla 9.Perfil desarrollador.</i>	10
<i>Tabla 10.Perfil comprador.</i>	10
<i>Tabla 11.Perfil personal de mantenimiento.</i>	11
<i>Tabla 12.Perfil usuario zona iluminada.</i>	11
<i>Tabla 13.Requisitos: Precedencia y prioridad.</i>	13
<i>Tabla 14.Requisitos funcionales del nodo central.</i>	14
<i>Tabla 15.Requisitos funcionales de la mota.</i>	14
<i>Tabla 16.Casos de uso.</i>	16
<i>Tabla 17.Caso de uso: configuración.</i>	17
<i>Tabla 18.Caso de uso: información.</i>	18
<i>Tabla 19.Caso de uso: actualización de firmware.</i>	18
<i>Tabla 20.Caso de uso: control de luminaria.</i>	19
<i>Tabla 21.Caso de uso: medir.</i>	20

Definiciones, acrónimos y abreviaturas

LED	Light Emitting Diode (Diodo emisor de luz).
DALI	Digital Addressable Lighting Interface (Interfaz de iluminación digital y direccionable).
MESH	Topología de red mallada.
NTP	Network Time Protocol..
PWM	Pulse Wide Modulation (Modulación por ancho de pulso).

1 Introducción

Cada vez hay una mayor preocupación por el gasto energético y el medio ambiente, tanto por el coste que supone en una sociedad fuertemente dependiente de la energía para su mantenimiento, como por la importancia que la producción y consumo de energía tiene en el deterioro de nuestro medio natural.

Una de las partidas ciertamente importantes en ese consumo energético va asociada a las necesidades de iluminación de nuestros entornos, desde las calles y avenidas de nuestras ciudades y medios terrestres de comunicación (autopistas y autovías), hasta nuestros hogares, pasando por las empresas y zonas de trabajo.

De ahí que diferentes organismos públicos, encabezados por los ayuntamientos, empresas, instituciones y los ciudadanos en general estén demandando sistemas que permitan un uso más eficiente de la energía eléctrica con fines de iluminación, de tal manera que se puedan conseguir unos niveles lumínicos óptimos en función del entorno, y para cada necesidad en los momentos adecuados, sin un derroche de energía ni molestia para el medio natural.

El objetivo de este documento es capturar, analizar y definir las características y necesidades de alto nivel del sistema de control de iluminación inteligente, llamado *LUCES*. Se centrará en describir las expectativas de cada una de las partes del proyecto y de los usuarios finales y por qué estas necesidades existen. Los detalles de cómo *LUCES* cumplen estas necesidades se detallan en los casos de uso y en las especificaciones adicionales.

1.1 Objetivos

Deseamos realizar un sistema de iluminación inteligente y conectado para zonas urbanas que permita convertir luminarias existentes o nuevas en luminarias inteligentes para su control y automatización y obtener un sistema de iluminación con al menos las siguientes ventajas:

- Más eficiente con el consumo eléctrico.
- Mas eficiente en la gestión de la iluminación y por tanto menor contaminación lumínica.
- Flexibilidad en la disposición y cantidad de las luminarias.
- Gestión del sistema desde un único punto.
- Detecta la ocupación de la vía a iluminar.
- Mayor confort para el usuario de la vía.
- Mejor mantenimiento, avisando en caso de fallo de la luminaria.

Aparte de las ventajas descritas, el desarrollo de este sistema tiene la motivación extra de que es un producto que aporta beneficios para el medio ambiente, es decir para toda la sociedad ya que indirectamente nos beneficiamos todos, y directamente para los usuarios finales que tendrán beneficios tanto económicos y de confort, además, les ayudará a cumplir con las nuevas regulaciones anticontaminación y protección del medio natural.

2 Directivas del Proyecto

2.1 Oportunidad de negocio

El creciente interés de los organismos públicos, empresas y ciudadanos en general en reducir el gasto energético y en el cumplimiento de las nuevas normativas medioambientales, ya sean relativas al producir energía eléctrica como el uso inadecuado de niveles de iluminación, abre un nicho de negocio en el desarrollo de sistemas de iluminación inteligentes y conectados.

Además, el auge de un mundo cada vez más digitalizado y conectado, conduce a que la mayoría de los ciudadanos estén más interesados en sistemas conectados y llamados inteligentes que en sistemas estándares.

2.2 Descripción del problema

El problema de	El gasto energético y contaminación lumínica generados en las necesidades de iluminación de nuestros entornos.
Afecta a	Organismos públicos, empresas y ciudadanos en general
Lo cual tiene como impacto	Mayor gasto energético y contaminación lumínica.
Una solución satisfactoria sería	Sistemas de iluminación eficientes en el uso de la energía eléctrica que proporcione niveles lumínicos óptimos en función del entorno y aplicación.

Tabla 1. Descripción del problema.

2.3 Descripción del producto

Para	Ciudades y negocios.
Los cuales	Necesitan controlar la eficiencia y estado de sus sistemas de iluminación.
LUCES	Es un sistema hardware y software.
Que	Permite la integración de luminarias en redes de comunicaciones y crear sistemas de iluminación conectados, más eficientes energéticamente, con una mejor gestión, diagnóstico y mantenimiento.
Frente a	Interact IoT lighting o Caséta Wireless
Nuestro producto	Implementa un hardware que permite reutilizar luminarias ya existentes o nuevas, para crear y gestionar sistemas de iluminación conectados, de forma flexible y económica.

Tabla 2. Descripción del producto.

3 Descripción de participantes y usuarios

3.1 Preliminar

Una empresa, “NewLum” orientada a la comercialización e instalación de sistemas de iluminación, nos propone un proyecto consistente en el diseño de un sistema de iluminación inteligente orientado a la eficiencia energética. Este sistema estará indicado para la iluminación de calles, parques, zonas comunitarias, aparcamientos, pasillos, grandes salones, etc., siendo los posibles clientes finales ayuntamientos, hoteles, comunidades de vecinos, etc.

La empresa que nos contrata, solicita una solución software y hardware. Una vez obtenida esa solución y comprobada en un cliente final, esta empresa se encargará de la fabricación, venta e instalación del producto. Su primer cliente de este sistema es el lujoso Hotel Atlántico, que ha aceptado a probar el sistema prototipo a cambio de un buen acuerdo comercial.

3.2 Resumen de los participantes

Nombre	Representa	Rol
Director empresa comercializadora	Representante de la empresa que solicita el proyecto	Es el participante principal. Encargado de establecer la mayoría de los requisitos del sistema. Encargado de aprobar los presupuestos y de validar el resultado del proyecto.
Instaladores	Instaladores de la empresa comercializadora	Son los empleados que instalarán el sistema para el cliente final
Comerciales	Comerciales de la empresa “NewLum”	Comerciales de la empresa comercializadora encargados de vender el producto a usuarios finales como ayuntamientos, hoteles, comunidades de vecinos, etc.
Jefe de proyecto	Empresa que desarrolla el proyecto	Encargado de especificar, diseñar, planificar y gestionar el personal del proyecto.
Desarrolladores	Empleados de la empresa que desarrolla el proyecto	Encargados de desarrollar el producto y su documentación.

Tabla 3.Resumen de los participantes.

3.3 Resumen y entorno de los usuarios

Nombre	Descripción	Participante
Comprador del producto	Cliente que ha adquirido el producto para utilizarlo	Representado por: Responsable de la

		empresa o institución que ha adquirido el producto
Empleados de mantenimiento	Encargado de configurar y mantener el sistema de iluminación.	Representado por: Encargado de mantenimiento de la empresa o institución que ha adquirido el producto
Usuarios de las zonas iluminadas	Personas ajenas al producto que disfrutan de las prestaciones y funcionalidades de este.	Representado por: Usuario o cliente de la empresa o institución que ha adquirido el producto

Tabla 4.Resumen y entorno de los usuarios.

3.3.1 Entorno de los usuarios

El entorno de los **usuarios** es una zona iluminada por el sistema transitada principalmente por peatones, ya sea una calle, pasillos, hall, etc.

Para las tareas de **mantenimiento** los usuarios del sistema son profesionales cualificados. Consistirá en la sustitución de la luminaria o reconfiguración cuando reciban una alerta de fallo. Una sola persona podía realizar esta tarea. Al estar el sistema conectado a un nodo central que avisa en caso de fallo, el responsable de mantenimiento puede estar en remoto y solo asistir a la instalación cuando se necesite.

3.4 Perfiles de los participantes

3.4.1 Director empresa cliente

Representante	Javier Vizcaya
Tipo	Usuario de negocio
Responsabilidades	Revisar los requisitos y el diseño de interfaz de usuario. Validar los requisitos implementados.
Criterio de Éxito	El producto cumpla las especificaciones y esté finalizado en la fecha acordada
Entregables	Documentos con especificaciones iniciales
Comentarios	Será el encargado de aprobar la finalización del proyecto.

Tabla 5.Perfil director.

3.4.2 Instalador

Representante	Joaquín Torres
Tipo	Técnico en instalaciones eléctricas y electrónicas
Responsabilidades	Revisar manual de instalación y aportar los requisitos necesarios para que el sistema se pueda instalar en los clientes finales
Criterio de Éxito	El proyecto finaliza con todos los requisitos satisfechos y verificados.
Entregables	No
Comentarios	Aportará información en base a su formación y experiencia laboral

Tabla 6.Perfil instalador.

3.4.3 Comercial

Representante	Carlos Fernández
Tipo	Comercial empresa comercializadora
Responsabilidades	Informar del tipo de iluminación que solicita el mercado actualmente
Criterio de Éxito	El proyecto finaliza con un producto con todos los requisitos satisfechos y atractivo para los clientes
Entregables	No
Comentarios	Aportará información en base a las consultas de los clientes

Tabla 7. Perfil comercial.

3.4.4 Jefe de proyecto

Representante	J. Muñoz
Tipo	Director de proyectos
Responsabilidades	Encargado de especificar, diseñar, planificar y gestionar el personal del Proyecto.
Criterio de Éxito	El proyecto finaliza con todos los requisitos satisfechos y verificados.
Entregables	Documento de requisitos, Documento de Planificación, Documento de Asignación de Tareas, Presupuesto y acta de constitución.
Comentarios	Ninguno

Tabla 8. Perfil jefe de proyecto.

3.4.5 Desarrolladores

Representante	Jesús Martínez
Tipo	Ingeniero electrónico
Responsabilidades	Encargado de la implementación del sistema y la documentación asociada al mismo
Criterio de Éxito	El proyecto finaliza con todos los requisitos satisfechos y verificados
Entregables	Hardware, software y manuales
Comentarios	Ninguno

Tabla 9. Perfil desarrollador.

3.5 Perfiles de usuario

3.5.1 Comprador del producto

Representante	Manolo Luna
Descripción	Director del Hotel Atlántico
Tipo	Cliente de la empresa comercializadora
Responsabilidades	Informar si el sistema no funciona o no está configurado correctamente.
Criterio de Éxito	La zona está iluminada correctamente y acorde a la presencia de personas. Reducción en los consumos de electricidad.
Comentarios	Ninguno

Tabla 10. Perfil comprador.

3.5.2 Personal de mantenimiento

Representante	Francisco Peña
Descripción	Responsable de mantenimiento del Hotel Atlántico
Tipo	Técnico cualificado.
Responsabilidades	Comprobar que no han saltados avisos de fallos en el sistema y mantener el sistema configurado correctamente
Criterio de Éxito	La zona está iluminada correctamente y acorde a la presencia de personas. Reducción en los consumos de electricidad.
Comentarios	Ninguno

Tabla 11. Perfil personal de mantenimiento.

3.5.3 Usuario de la zona iluminada por el sistema

Representante	Antonio López
Descripción	Cliente del Hotel Atlántico
Tipo	Usuario final
Responsabilidades	Ninguna. Solo informar de si el sistema no funciona si lo desea.
Criterio de Éxito	Pasear por la zona donde está instalado el sistema con los mínimos niveles de luminosidad recomendados
Comentarios	Ninguno

Tabla 12. Perfil usuario zona iluminada.

3.6 Alternativas y competencia

3.6.1 Interact City / IoT lighting

Interact City [1] es una solución software avanzada para la gestión y optimización energética de activos de iluminación para ciudades inteligentes. En la parte hardware se apoya en luminarias inteligentes de fabricantes como Philips, VariLite Color Kinetics y Strand. Es compatible con luminarias con sensores de presencia que cumplen el nuevo estándar de luminarias inteligentes Zhaga-D4i [2].

Al no tener implementación hardware propia, podríamos contemplar esta solución como futuro socio para la gestión y control de grandes proyectos.

3.6.2 Lutron Caséta Wireless

Cáseta Wireless [3] es una solución del fabricante Lutron está orientada para el control de luces y persianas para el hogar, oficinas y otros interiores. Este sistema de control de iluminación se instala sustituyendo los interruptores y pulsadores por un hardware propio y funciona con una variedad de tipos de focos inteligentes, entre los que se incluyen LEDs y CFL (fluorescentes compactos) atenuables, y brindan la comodidad de tener un hogar inteligente, automatizado y conectado.

Aunque implementa un control de iluminación que permite ajustar la intensidad de las luminarias y automatizaciones en el horario de encendido, el campo de aplicación, los sistemas domóticos, distinto al del proyecto LU CES, hace que pudiéramos contemplarlo como complemento para el ámbito del hogar y no como competencia.

4 Requisitos

4.1 Diagrama general

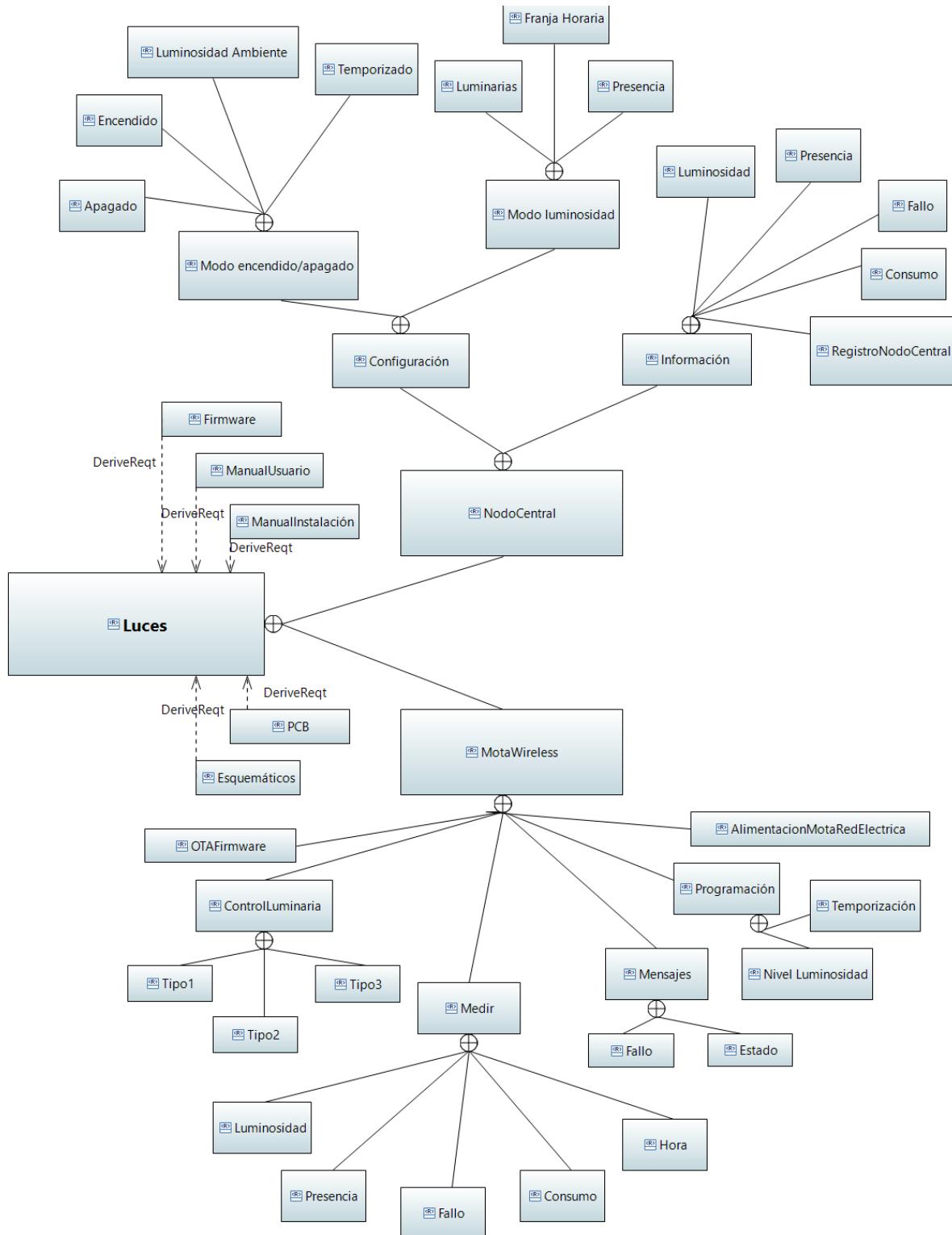


Fig. 1. Diagrama general de requisitos

4.2 Precedencia y prioridad

Id	Nombre	Prioridad	Precedencia
R1.1	Esquemáticos	F	
R1.2	PCB	F	R1.2
R1.3	ManualInstalación	D	R1.2
R1.4	ManualUsuario	D	R4.* - R13.*.
R2	Firmware	F	R4.*-R7, R8.*- R11.*.
R3.1	NodoCentral	F	
R3.2	MotaWireless	F	
Requisitos aplicables al nodo central			
R4.1	NCEncendidoManual	F	R8.*.
R4.2	NCApagadoManual	F	R8.*.
R4.3	NCEncendidoTemporizado	F	R8.* , R9.6, R10.2.
R4.4	NCEncendidoAutoLuminosidadAmbiente	D	R8.* , R9.1, R10.1.
R5.1	NCAjusteLuminosidadLuminarias	F	R8.* , R9.1, R10.1.
R5.2	NCAjusteLuminosidadFranjaHoraria	F	R8.* , R9.1, R9.5
R5.3	NCAjusteLuminosidadPresencia	O	R8.* , R9.1, R9.2.
R6.1	NCInfoLuminosidad	O	R9.1, R11.2.
R6.2	NCInfoPresencia	O	R9.2, R11.2.
R6.3	NCInfoConsumo	O	R9.5, R11.2.
R6.4	NCInfoFallo	F	R9.3, R11.1.
R7	RegistroNodoCentral	F	R11.1, R11.2.
Requisitos aplicables a las motas			
R8.1	ControlLuminariaTipo1	F	
R8.2	ControlLuminariaTipo2	D	
R8.3	ControlLuminariaTipo3	O	
R9.1	MedirLuminosidad	D	
R9.2	DetectarPresencia	O	
R9.3	DetectarFallo	F	R9.1, R9.4
R9.4	MedirConsumo	O	
R9.5	MediciónHoraria	F	
R10.1	NivelLuminosidadProgramable	F	
R10.2	TemporizaciónProgramable	F	R9.5
R11.1	AvisoFallo	F	R4.2, R9.3.
R11.2	EnvioEstado	D	R4.2, R9.*.
R12	OTAFirmware	F	
R13	AlimentacionMotaRedElectrica	F	

Tabla 13.Requisitos: Precedencia y prioridad.

4.3 Requisitos Funcionales

4.3.1 R3.1 NodoCentral

La arquitectura del sistema tiene un nodo central con sus funcionalidades descritas con los siguientes requisitos funcionales:

R4 Modo de encendido o apagado

R4.1 NCEncendidoManual	Encendido manual de las motas desde el nodo central
R4.2 NCApagadoManual	Apagado manual de las motas desde el nodo central
R4.3 NCEncendidoTemporizado	Modo de encendido y apagado temporizados
R4.4 NCEncendidoAutoLuminosidadAmbiente	Automáticamente por niveles de luminosidad ambiente

R5 Modo de ajuste del nivel de luminosidad

R5.1 NCAjusteLuminosidadLuminarias	Manualmente a un valor fijo
R5.2 NCAjusteLuminosidadFranjaHoraria	Mediante valores definidos para cada franja horaria
R5.3 NCAjusteLuminosidadPresencia	Dinámicamente usando la detección de presencia

R6 Mostar información

R6.1 NCInfoLuminosidad	Del nivel lumínico detectado
R6.2 NCInfoPresencia	De la activación o no del detector de presencia
R6.3 NCInfoConsumo	Del consumo actual de la mota
R6.4 NCInfoFallo	De si hay un fallo o avería en la luminaria

R7 RegistroNodoCentral: Registro de datos almacenados en el nodo central

Tabla 14.Requisitos funcionales del nodo central.

4.3.2 MotaWireless

La arquitectura del sistema estará formada por motas comunicadas entre sí por alguna tecnología inalámbrica, con las siguientes capacidades descritas con estos requisitos funcionales:

R8 Control de intensidad lumínica

R8.1 ControlLuminariaTipo1	Tipo LED con intensidad controlada por PWM
R8.2 ControlLuminariaTipo2	Con balastro regulable por el protocolo 0-10V
R8.3 ControlLuminariaTipo3	Intensidad controlada por TRIAC

R9 Medición

R9.1 MedirLuminosidad	Medición de niveles lumínicos en la zona de la luminaria
R9.2 DetectarPresencia	Detección de presencia en la zona de la luminaria
R9.3 DetectarFallo	Detección de fallo de la luminaria
R9.4 MedirConsumo	Medición de consumo de la luminaria
R9.5 MediciónHoraria	Medición de la hora actual, reloj

R10 Programación

R10.1 NivelLuminosidadProgramable	Los niveles de luminosidad se pueden programar
R10.2 TemporizaciónProgramable	La mota se puede programar por franjas horarias

R11 Mensajes y alarmas

R11.1 AvisoFallo	Envío de mensaje de fallo en luminaria
R11.2 EnvioEstado	Envío de mensajes de estado, consumo, etc.

R12 AlimentacionMotaRedElectrica: La mota se alimenta de la red eléctrica

R13 OTAFirmware: Actualización del firmware en remoto

Tabla 15.Requisitos funcionales de la mota.

4.4 Requisitos No Funcionales

4.4.1 R1.1 Esquemáticos

Se deben entregar los planos de diseño eléctrico para una futura producción.

4.4.2 R1.2 PCB

Se deben entregar los planos de diseño de las placas de circuito impreso para una futura producción.

4.4.3 R1.3 ManualInstalación

Se deben entregar un manual de instalación para una futura comercialización.

4.4.4 R1.4 ManualUsuario

Se deben entregar un manual de usuario para una futura comercialización.

4.4.5 R2 Firmware

Se deben entregar el firmware de todos los tipos de nodos para una futura producción.

5 Casos de uso

5.1 Actores del sistema

Nombre	Descripción
Usuario	Grupo de personas que van a hacer uso del sistema de iluminación
Luminaria	Representa a cada una de las luminarias que controlan el sistema de iluminación
Tiempo	Representa al reloj interno del sistema que invoca a un tiempo determinado ciertos eventos
Luminosidad	Representa los niveles de luminosidad que invocan ciertos eventos

Tabla 16. Casos de uso.

5.2 Diagrama general

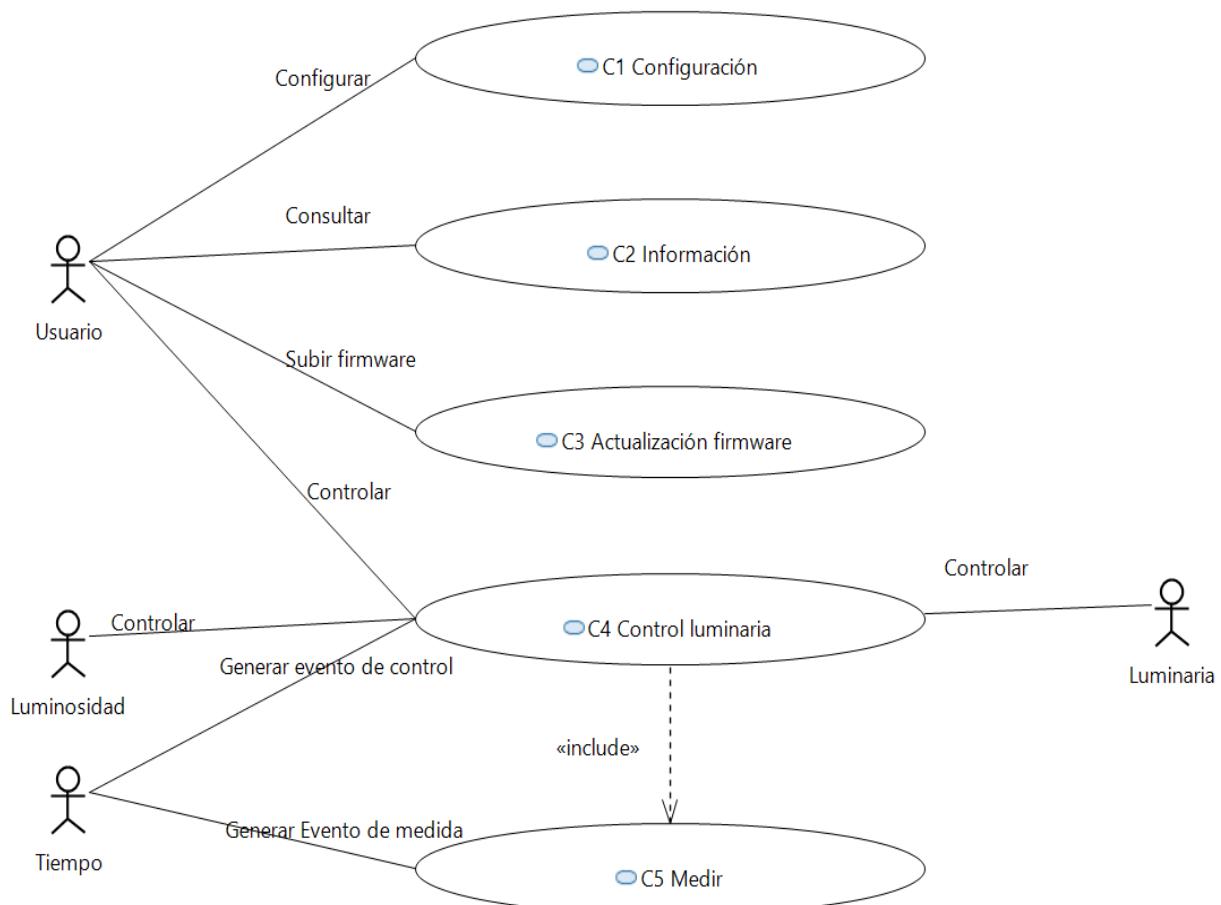


Fig. 2. Diagrama general de casos de uso

5.3 Descripción textual de los casos de uso

5.3.1 C1 Configuración

5.3.1.1 *Contexto de uso:*

Cuando el usuario desea configurar el sistema de iluminación.

5.3.1.2 *Actor Principal:*

Usuario.

5.3.1.3 *Participantes y Objetivos:*

Participante	Objetivo
Usuario	Configurar el sistema de iluminación

Tabla 17.Caso de uso: configuración.

5.3.1.4 *Pre-Condiciones:*

Sistema de iluminación instalado.

5.3.1.5 *Garantías mínimas:*

El Sistema de iluminación queda configurado.

5.3.1.6 *Escenario de éxito principal:*

1. El usuario accede al interfaz de configuración del sistema de iluminación alojado en el nodo central y accede a los controles de “Configuración”.
2. El nodo central se comunica con las motas enviando un mensaje para programarlas según el modo seleccionado.
3. La mota envía un mensaje al nodo central con la información de los parámetros programados.
4. El nodo central almacena en el registro del sistema un evento de configuración exitosa.

5.3.1.7 *Escenario secundario 1:*

- 1a. El usuario accede al interfaz de configuración del sistema de iluminación alojado en el nodo central y accede a los controles de “Configuración”.
- 2a. El nodo central intenta comunicarse con las motas enviando un mensaje para programarlas según el modo seleccionado.
- 3a. Si una mota no devuelve un mensaje al nodo central con la información de los parámetros programados se informa de un error en la interfaz de configuración.
- 4a. El nodo central almacena en el registro del sistema un evento de error en la configuración de la mota.

5.3.2 C2 Información

5.3.2.1 *Contexto de uso:*

Cuando el usuario desea información sobre el estado del sistema e información de los distintos valores de las medidas de las motas.

5.3.2.2 *Actor Principal:*

Usuario

5.3.2.3 *Participantes y Objetivos:*

Participante	Objetivo
Usuario	Consultar el estado del sistema y valores de los sensores de las motas

Tabla 18.Caso de uso: información.

5.3.2.4 *Pre-Condiciones:*

Nodo central instalado.

5.3.2.5 *Garantías mínimas:*

El usuario obtiene información del estado del sistema y de los valores de las medidas de las motas que lo forman.

5.3.2.6 *Escenario de éxito principal:*

1. El usuario accede al interfaz de control del sistema de iluminación alojado en el nodo central y accede a la sección de “*Información*”.
2. El nodo central muestra información del sistema y de las medidas de los sensores de las motas con los últimos valores recibidos de las comunicaciones con las motas.

5.3.2.7 *Escenario secundario 1:*

- 1a. El usuario accede al interfaz de control del sistema de iluminación alojado en el nodo central y accede a la sección de “*Información*”.
- 2a. El nodo central no muestra información de las medidas de los sensores y no se almacenan eventos en el registro del sistema.

5.3.3 C3 Actualización firmware

5.3.3.1 *Contexto de uso:*

Cuando el usuario desea actualizar el firmware de las motas inalámbricas del sistema de iluminación.

5.3.3.2 *Actor Principal:*

Usuario

5.3.3.3 *Participantes y Objetivos:*

Participante	Objetivo
Usuario	Sube un nuevo firmware al sistema de iluminación

Tabla 19.Caso de uso: actualización de firmware.

5.3.3.4 *Pre-Condiciones:*

Sistema de iluminación instalado.

5.3.3.5 Garantías mínimas:

Registro de evento de intento de actualización de firmware.

5.3.3.6 Escenario de éxito principal:

1. El usuario sube a la red del sistema el archivo con el nuevo firmware desde un PC o aplicación móvil.
2. Se comprueba si el archivo es correcto y se envía a los correspondientes nodos.
3. Los nodos reciben el firmware con la actualización y actualizan su versión de firmware.
4. El nodo central almacena en el registro del sistema un evento de la actualización.

5.3.3.7 Escenario secundario 1

- 1a. El usuario sube a la red del sistema el archivo con el nuevo firmware desde un PC o aplicación móvil.
- 2a. Se comprueba que el archivo del firmware no es correcto para ningún tipo de nodo.
- 3a. No se realiza ninguna actualización de firmware.

5.3.4 C4 Control luminaria

5.3.4.1 Contexto de uso:

Cuando se desea encender, ajustar la intensidad lumínica o apagar la luminaria.

5.3.4.2 Actor Principal:

Tiempo.

5.3.4.3 Participantes y Objetivos:

Participante	Objetivo
Tiempo	Encender, apagar la luminaria o ajustar el nivel de intensidad lumínica
Luminosidad	Encender o apagar la luminaria
Usuario	Encender, apagar la luminaria o ajustar el nivel de intensidad lumínica
Luminaria	Encenderse o apagarse o ajustar nivel de intensidad lumínica

Tabla 20. Caso de uso: control de luminaria.

5.3.4.4 Pre-Condiciones:

Sistema de iluminación instalado y configurado.

5.3.4.5 Garantías mínimas:

La luminaria se enciende y se ajusta su intensidad lumínica o se apaga según programación establecida.

5.3.4.6 Escenario de éxito principal:

1. A una frecuencia determinada, la mota consulta los mensajes recibidos, la hora del sistema y las medidas de los sensores.

2. En los mensajes recibidos hay un mensaje del nodo central, controlado por el usuario, con la orden de encender, ajustar la intensidad lumínica o apagar la luminaria.
3. La mota enciende, ajusta o apaga la luminaria según la información recibida en el mensaje.
4. La mota envía un mensaje al nodo central del evento de control.

5.3.4.7 Escenario secundario 1:

- 1a. A una frecuencia determinada, la mota consulta los mensajes recibidos, la hora del sistema y las medidas de los sensores.
- 2a. Como no hay mensaje de control de luminaria desde el nodo central, la mota comprueba que el modo de encendido o apagado de la luminaria es temporizado.
- 3a. La mota comprueba su reloj interno y enciende o apaga la mota acorde a la temporización programada.
- 4a. Si la luminaria está encendida, la mota ajusta la intensidad de esta de acuerdo con el modo de luminosidad programado.
- 5a. La mota envía un mensaje al nodo central del evento de control.

5.3.4.8 Escenario secundario 2

- 1b. A una frecuencia determinada, la mota consulta los mensajes recibidos, la hora del sistema y las medidas de los sensores.
- 2b. Como no hay mensaje de control de luminaria desde el nodo central, la mota comprueba que el modo de encendido o apagado de la luminaria es por nivel de luminosidad.
- 3b. La mota comprueba la medida del sensor de luminosidad y enciende o apaga la mota acorde a la temporización programada.
- 4b. Si la luminaria está encendida ajusta la intensidad de esta de acuerdo con el modo de luminosidad programado.
- 5b. La mota envía un mensaje al nodo central del evento de control.

5.3.5 C5 Medir

5.3.5.1 Contexto de uso:

A una frecuencia dada, el sistema obtiene la medida de sus sensores incorporados.

5.3.5.2 Actor Principal:

Tiempo

5.3.5.3 Participantes y Objetivos:

Participante	Objetivo
Tiempo	Activa la lectura de los datos

Tabla 21.Caso de uso: medir.

5.3.5.4 Pre-Condiciones:

Mota instalada.

5.3.5.5 Garantías mínimas:

Se obtiene el valor de las medidas de los sensores que equipa y se envía mensaje de estado, si se detecta fallo en la luminaria se envía mensaje de fallo.

5.3.5.6 Escenario de éxito principal:

1. El tiempo solicita a una mota del sistema de iluminación que consulte el valor de medida de sus sensores.
2. La mota lee el valor de sus sensores.
3. La mota controla la luminaria acorde al valor de su programación y medida de los sensores.
4. La mota envía un mensaje con la información de las medidas de sus sensores y los nuevos valores de estado de la luminaria.
5. El nodo central almacena el mensaje en el registro del sistema y actualiza su panel de información con las medidas recibidas.

5.3.5.7 Escenario secundario 1

- 1a. El tiempo solicita a una mota del sistema de iluminación que consulte el valor de medida de sus sensores.
- 2a. La mota lee el valor de sus sensores.
- 3a. La mota detecta en las medidas que indican la existencia de fallo en la luminaria.
- 4a. La mota envía un mensaje de fallo de luminaria.
- 5a. La mota envía un mensaje con la información de las medidas de sus sensores y los nuevos valores de estado de la luminaria.
- 6a. El nodo central almacena los mensajes en el registro del sistema y actualiza su panel de información.
- 7a. El nodo central envía un mensaje de fallo de luminaria al usuario.

6 Referencias

- [1] Interact, «Interact City,» Julio 2021. [En línea]. Available: <https://www.interact-lighting.com/global/what-is-possible/interact-city>.
- [2] TodoLux, «TodoLux. Lighting Influencers.,» Julio 2021. [En línea]. Available: <https://todo-lux.com/investigacion-y-desarrollo/zhangad4i-nuevo-estandar-luminarias-inteligentes/>.
- [3] Lutron, «Caséta Wireless,» Julio 2021. [En línea]. Available: <https://www.lutron.com/es-LA/Products/Paginas/SingleRoomControls/CasettaWireless/Overview.aspx>.

Diseño e implementación

RESUMEN DE DOCUMENTO

Descripción la arquitectura (sistema, subsistemas e interfaces) y posterior análisis a más bajo nivel de la implementación de los elementos que la forman.

TÍTULO DOCUMENTO	Diseño e implementación.
FECHA DE ENTREGA	30/09/2021
AUTOR/ES	Jesús Muñoz Martínez

COMENTARIOS	VERSIÓN	FECHA
Versión Inicial	1.0	06/08/2021

Índice

1	ASPECTOS GENERALES DEL SISTEMA	5
2	ARQUITECTURA	7
2.1	ARQUITECTURA LÓGICA	7
2.1.1	Vista general	7
2.1.2	Sistema de iluminación	7
2.1.3	Mota Wireless	7
2.1.4	Nodo Central	8
2.1.5	Panel de control	8
2.1.6	Registro del sistema	8
2.1.7	Email Aviso Fallo	8
2.1.8	Firmware	8
2.2	ARQUITECTURA FÍSICA	8
2.2.1	Vista general	8
2.2.2	Sistema de Iluminación	9
2.2.3	Nodo Central	9
2.2.4	Mota Wireless	9
2.2.5	Red de comunicaciones	9
2.2.6	Control Luminaria	9
2.2.7	Sensor Luminosidad	9
2.2.8	Sensor de Presencia	9
2.2.9	Sensor de Consumo	9
3	IMPLEMENTACIÓN HARDWARE	10
3.1	NODO CENTRAL	10
3.2	MOTA INALÁMBRICA	11
3.2.1	Desarrollo de la PCB.	13
3.2.2	Creación del prototipo mediante grabado con CNC	15
4	IMPLEMENTACIÓN SOFTWARE	19
4.1	FIRMWARES DEL NODO CENTRAL Y MOTAS INALÁMBRICAS.	19
4.2	COMUNICACIONES ENTRE SUBSISTEMAS	21
4.3	PANEL DE CONTROL	22
5	REFERENCIAS	27

Índice de figuras.

<i>Fig. 1. Ejemplo de implementación del sistema LUCES</i>	5
<i>Fig. 2. Red mallada implementada con ESP-WIFI-MESH</i>	6
<i>Fig. 3. Arquitectura Lógica</i>	7
<i>Fig. 4. Arquitectura física</i>	8
<i>Fig. 5. ESP32-DevKitC V4 con el módulo ESP32-WROOM-32</i>	10
<i>Fig. 6. Carcasa para el Nodo Central</i>	11
<i>Fig. 7. Módulo ESP12</i>	11
<i>Fig. 8. Controlador 0-10V de Lutron.</i>	12
<i>Fig. 9. Esquemático de la PCB de la mota wireless.</i>	14
<i>Fig. 10. Pistas de la cara superior de la PCB.</i>	14
<i>Fig. 11. Modelo en 3D de la PCB.</i>	15
<i>Fig. 12. Software Flatcam, para pasar el Gerber a Gcode.</i>	16
<i>Fig. 13. Detalles del procesado del Gerber con Flatcam..</i>	16
<i>Fig. 14. Programa de control de la CNC con el Gcode cargado.</i>	16
<i>Fig. 15. Fresado del cobre no necesario en la PCB.</i>	17
<i>Fig. 16. Vías: Generando código Gcode a partir del Excellon.</i>	17
<i>Fig. 17. Placa PCB</i>	18
<i>Fig. 18. PCB con los componentes soldados.</i>	18
<i>Fig. 19. Programando el firmware con Visual Studio Code + PlatformIO.</i>	19
<i>Fig. 20. Comunicaciones entre los distintos subsistemas.</i>	21
<i>Fig. 21. Tratamiento de mensajes del servidor web según sus tipos.</i>	21
<i>Fig. 22. Tratamiento de mensajes recibidos en la mota.</i>	22
<i>Fig. 23. Aspecto del panel de control.</i>	23
<i>Fig. 24. Comunicación con WebSocket. Imagen traducida de la de Rui Santos [6].</i>	23
<i>Fig. 25. Flujo de configuración de las luminarias.</i>	24
<i>Fig. 26. Modos de configuración del panel de control.</i>	25
<i>Fig. 27. Acceso al modo de configuración desde el panel de control.</i>	25
<i>Fig. 28. Modo de configuración.</i>	26

Índice de tablas.

<i>Tabla 1. Sensores de las motas inalámbrica.</i>	12
<i>Tabla 2. Módulos de control de las motas inalámbrica.</i>	13

Definiciones, acrónimos y abreviaturas

CNC	Control numérico computerizado
GPIO	General purpose input/output.(Entradas/Salidas de propósito general)
LED	Light Emitting Diode (Diodo emisor de luz).
MESH	Topología de red mallada.
NTP	Network Time Protocol.
PCB	Printed Circuit Board (Placa de circuito impreso).
PIR	Passive Infrared [sensor] (Sensor de Infrarrojo pasivo)
PWM	Pulse Width Modulation (Modulación por ancho de pulso).
SoC	System on Chip (Microcontrolador y periféricos en un solo chip)
TRIAC	Es un tipo de semiconductor que trabaja con corriente alterna (AC).

1 Aspectos generales del sistema

El sistema inalámbrico de control de luminarias LUCES, basa su arquitectura en una red mallada de motas sensorizadas gobernadas por un nodo central, todo ello mediante comunicaciones inalámbricas. Un ejemplo de implantación del sistema en un entorno real puede ser el mostrado en la figura 1.

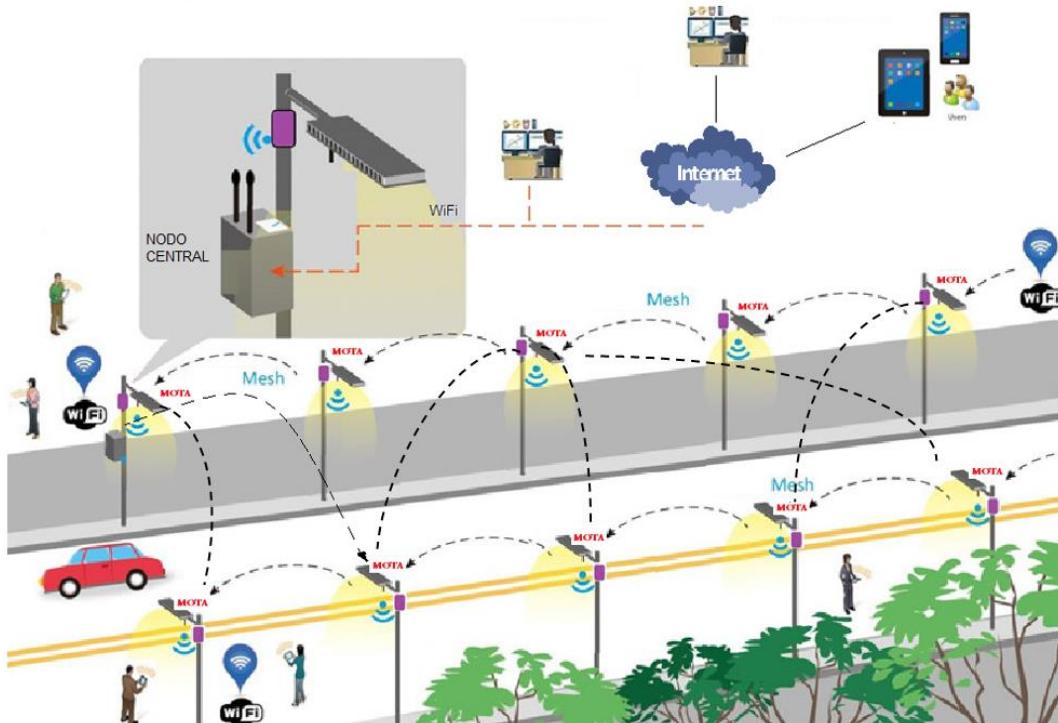


Fig. 1. Ejemplo de implementación del sistema LUCES

Como se puede apreciar, un nodo central gestiona el funcionamiento de la red de luminarias y comunica a la misma con el exterior, hace de puente, ya sea una red LAN e Internet. La red de luminarias está formada por motas que miden determinados factores de la zona a iluminar y según su programación, controlan el funcionamiento de la luminaria.

Por tanto, los aspectos más relevantes del diseño de este sistema son:

- Arquitectura centralizada en un **nodo central** con las siguientes funciones:
 - Crea, organiza y mantiene la red de luminarias.
 - Proporciona el panel de control para la gestión del sistema.
 - Realiza la función de puente entre la red de luminarias (interna) y la red exterior y/o internet.
 - Avisa por correo electrónico en caso de fallo de luminaria.
- Las luminarias están controladas mediante **motas** que se encargan:
 - Medir su entorno mediante sensores en los aspectos de iluminación, ocupación de la vía y estado de la luminaria.
 - Medir el consumo de la luminaria y avisar al nodo central si hay fallo.
 - Controlar el encendido y nivel lumínico de la luminaria guardando la programación recibida del nodo central.
- Comunicaciones inalámbricas utilizando el protocolo WiFi 802.11n.

- Encriptación WPA.
- API ESP-WIFI-MESH para crear la red mallada de las luminarias, figura 2.
- Añade motas automáticamente sin necesidad de reiniciar la red.
- Mensajes entre nodos o mediante *broadcasting*.

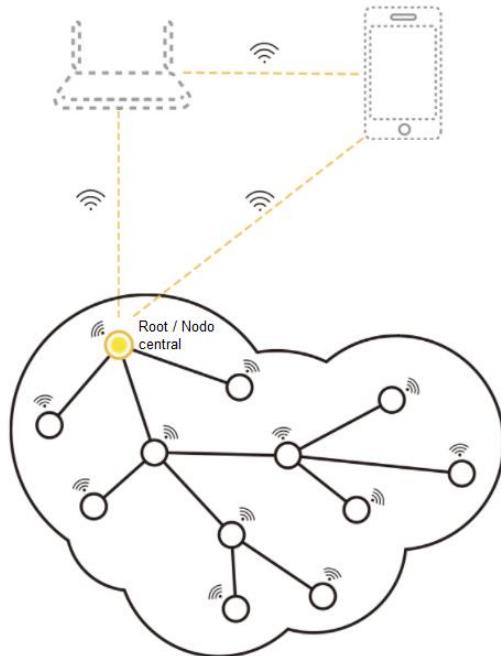


Fig. 2. Red mallada implementada con ESP-WIFI-MESH

Comentar, que la estructura de la red elegida, ESP-WIFI-MESH, es una red de comunicación inalámbrica con nodos organizados en una topología de malla que utiliza la función AP-STA simultánea en los SoC Espressif. Proporciona la implementación de una red autoformada y autorreparable. La topología de red de ESP-WIFI-MESH puede escalar hasta 1000 nodos teóricos en grandes áreas, sin requerir ningún soporte de infraestructura Wi-Fi específico. No utiliza direcciones IP, sino que los nodos se identifican por identificador generado a partir de su MAC.

2 Arquitectura

2.1 Arquitectura Lógica

2.1.1 Vista general

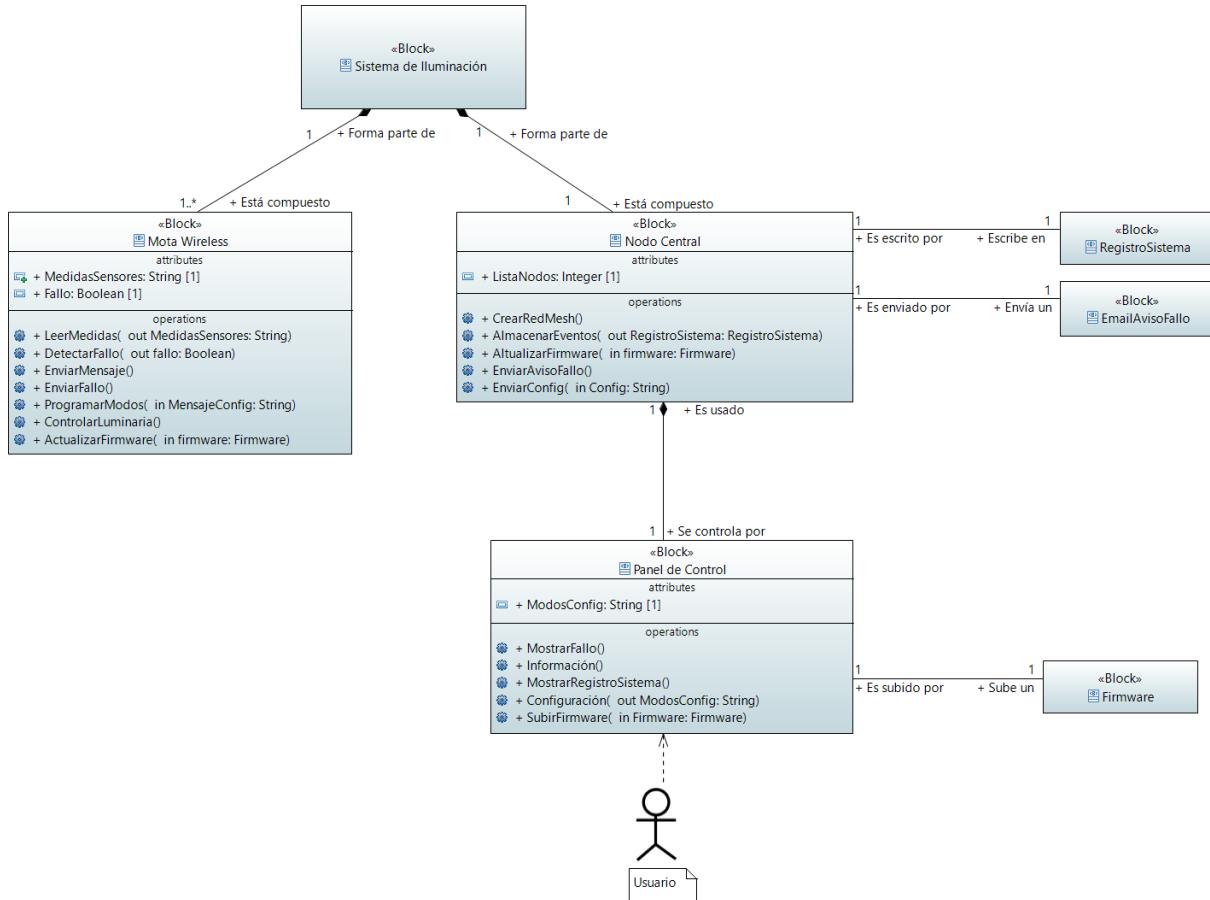


Fig. 3. Arquitectura Lógica

2.1.2 Sistema de iluminación

Modela el conjunto de todo el software. El sistema de iluminación está compuesto por un número finito de motas inalámbricas que lee sus sensores y controla una luminaria conectada a la misma. Es sistema tiene un nodo central que controla la asociación de luminarias a la red y proporciona una interfaz para visualizar el estado del sistema, sus motas conectadas y los valores de sus sensores, configurar los distintos modos de funcionamiento y avisar en caso de fallo de alguna luminaria.

2.1.3 Mota Wireless

Modela las motas inalámbricas. Una mota tiene asociado sensores de los que se encarga de tomar medidas de forma periódica. Con las medidas de estos sensores puede controlar la luminaria instalada según los modos de funcionamiento programado y detectar el fallo de la luminaria. Envía al nodo central mensajes informando de esas medidas y estado de sí

misma.

2.1.4 Nodo Central

Este nodo se encarga de implementar el panel de control, inicializar y mantener la red de comunicaciones, registrar los eventos del sistema, enviar a las motas la configuración establecida en el panel de control y enviar el correo electrónico de fallo al usuario en caso de fallo de luminaria.

2.1.5 Panel de control

Modela la parte gráfica del sistema de iluminación y la única accesible por los usuarios. El panel de control permite mostrar los datos de todos los sensores de las motas inalámbricas, informar con alarmas al usuario y mostrar el registro del sistema. Este interfaz permite también configurar los modos de funcionamiento de las motas y subir los firmwares para los modos.

2.1.6 Registro del sistema

Modela el manejo del registro del sistema en el que se almacenarán los eventos que ocurrán durante el funcionamiento del sistema.

2.1.7 Email Aviso Fallo

Modela el email que se envía en caso de fallo.

2.1.8 Firmware

Modela el firmware que sube el usuario cuando desea actualizar los nodos.

2.2 Arquitectura Física

2.2.1 Vista general

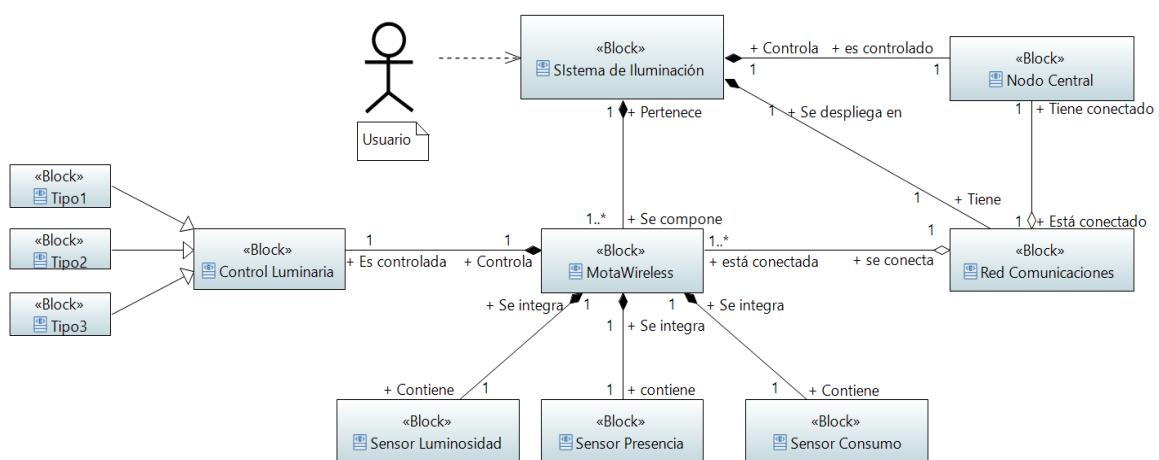


Fig. 4.Arquitectura física

2.2.2 Sistema de Iluminación

Representa el conjunto del todo el hardware. El sistema de iluminación está compuesto por motas inalámbricas, una red de comunicaciones y un nodo central con el panel de control de todo el sistema. El sistema de iluminación es mantenido por el encargado de mantenimiento.

2.2.3 Nodo Central

Representa al Nodo que integra el software de control y configuración del sistema de iluminación. Solo hay uno por cada sistema de iluminación.

2.2.4 Mota Wireless

Representa cada una de las motas inalámbricas que tiene el sistema de iluminación. Cada mota está conectada a la red de comunicaciones y tiene integrados tres sensores y al menos un hardware de control de luminaria.

2.2.5 Red de comunicaciones

Representa la red de comunicaciones del sistema de iluminación. Es una red mallada inalámbrica. Tanto el Nodo Central como las motas inalámbricas se conectan a esta red.

2.2.6 Control Luminaria

Representa la generalización de todos el hardware de control de luminarias que puede existir en las diferentes motas inalámbricas.

2.2.7 Sensor Luminosidad

Representa a los sensores de luminosidad.

2.2.8 Sensor de Presencia

Representa a los sensores de presencia.

2.2.9 Sensor de Consumo

Representa a los sensores de consumo.

3 Implementación hardware

3.1 Nodo central

Para implementar las funcionalidades del nodo central se ha elegido el microcontrolador ESP32 de Espressif, que junto al ESP8266 tiene soporte para ESP-WIFI-MESH. La elección de este SoC frente al ESP8266 es debido a que implementa un hardware más potente a un coste no mucho más superior, unos 3€ más por unidad. El hardware más potente nos proporciona una mayor estabilidad y un número mayor de motas en la red de luminarias.

Resumimos las características más interesantes de este SoC:

- CPU de doble núcleo de 32 bits a 240 Mhz.
- Memoria RAM de 520 Kbyte.
- Memoria flash de 4 Mbytes.
- WiFi certificado: 802.11 b/g/n (802.11n hasta 150 Mbps).
- Bluetooth v4.2 BR/EDR y especificación BLE.
- Interfaces: SD card, UART, SPI, SDIO, I2C, PWM, GPIO, ADC, DAC ...

Como las principales tareas del nodo central se basan en gestionar y proporcionar un sistema de configuración a través de su **interfaz inalámbrica** y no se necesitan otros sensores o usos de GPIOs, se ha optado por utilizar una placa comercial que nos permite reducir los costes de desarrollo y producción. Se ha elegido el módulo ESP32-WROOM-32 montado en la placa Essp32 Devkitc_v4 [1], figura 5.

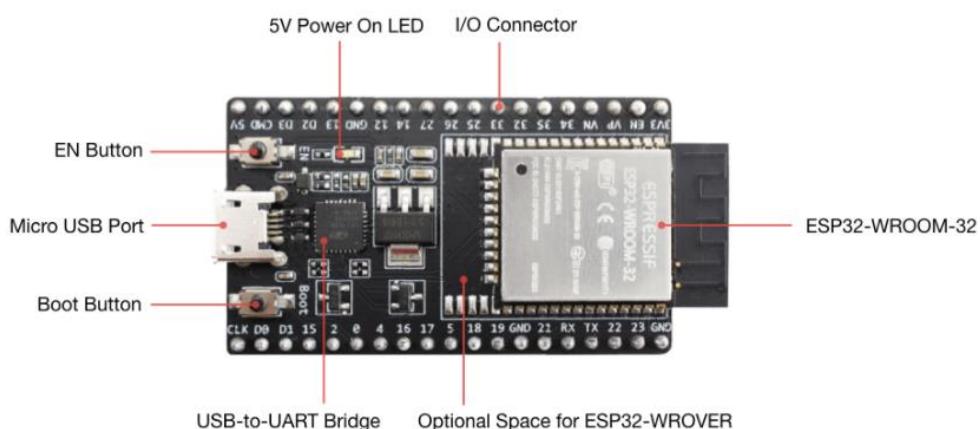


Fig. 5.ESP32-DevKitC V4 con el módulo ESP32-WROOM-32

El nodo central se ha complementado con el diseño de una carcasa para su protección y estética comercial, mostrado en la figura 6. El prototipo de la carcasa se ha creado mediante impresión 3D. El fichero con el modelo 3D está incluido con los entregables.

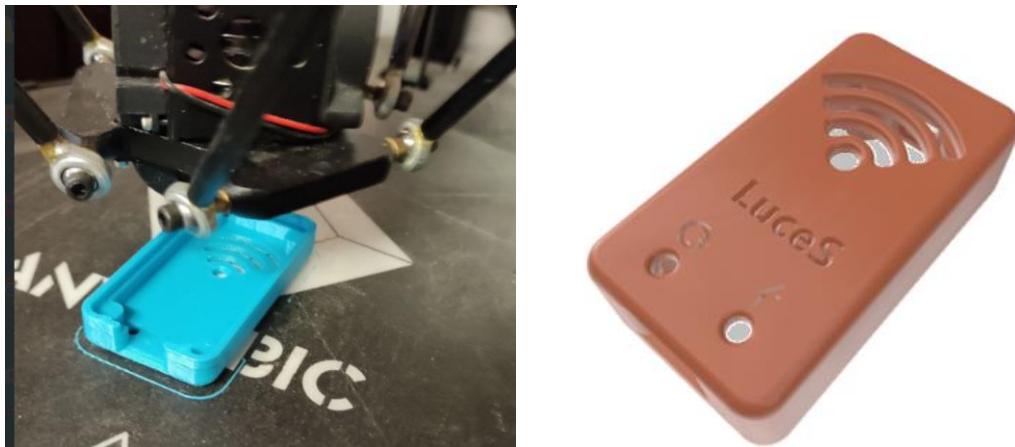


Fig. 6.Carcasa para el Nodo Central

3.2 Mota inalámbrica

Para la implementación hardware de la mota inalámbrica se ha desarrollado una placa PCB para dar soporte a los distintos sensores y actuadores. Como microcontrolador principal se ha elegido en este caso el ESP8266, con menos potencia que el ESP32, pero más que suficiente para integrarse en una red ESP-WIFI-MESH e interactuar con los sensores y el control de las luminarias. Las principales características del ESP8266 son:

- CPU RISC de 32-bit Tensilica Xtensa LX106 a un reloj de 80 MHz.
- RAM de instrucción de 64 KB, RAM de datos de 96 KB.
- Memoria flash externa QSPI - 512 KB a 4 MB.
- IEEE 802.11 b/g/n Wi-Fi con soporte de autenticación WEP y WPA/WPA2.
- UART, ADC (10 bits), SPI, I²C, I²S, GPIO, ...

Para el diseño de la PCB se ha escogido un módulo ESP-12, figura 7, que integra además del ESP8266 una memoria flash de 4 Mbytes y las suficientes GPIOs para dar soporte a los periféricos utilizados en la implementación de la mota inalámbrica, como son los diferentes sensores, así como los módulos de potencia que pueden ser utilizados.

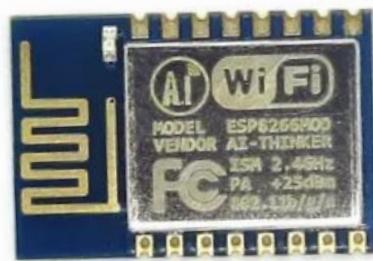


Fig. 7.Módulo ESP12.

Cada mota inalámbrica está dotada de un sensor de luminosidad, un sensor de presencia y un sensor de corriente AD/DC. Con ellos puede detectar si hay un fallo de la luminaria, usuarios en la vía o si está anocheciendo o amaneciendo.

En la tabla 1, se listan los sensores que llevan todas las configuraciones de las motas inalámbricas.

Sensor/ actuador	Descripción	Interfaz
		ESP8266
BH1750	Sensor de luminosidad (1- 65535 lx) con interfaz I2C	Bus I2C
		A blue printed circuit board (PCB) with a white BH1750 component and several surface-mount components.
HC-SR501	Sensor de PIR, con alcance de 7 metros y 110º de campo de visión	Salida GPIO
		A green PCB with a white PIR sensor and various electronic components.
ACS712-05	Sensor de corriente de 5 amperios y 2.1kV de aislamiento con salida analógica	ADC de 10 bits
		A blue PCB with a black ACS712 component and other electronic parts.

Tabla 1.Sensores de las motas inalámbrica.

La PCB desarrollada para la mota inalámbrica, proporciona la interfaz y la lógica para controlar los módulos de control de la potencia eléctrica de las luminarias. Se ha implementado el uso de 4 módulos de control, que se instalarán en la mota según esté basada la luminaria a controlar:

- Corriente DC: Una salida PWM del ESP8266 controla el módulo que regula la corriente que alimenta a la luminaria.
- Voltaje DC: Una salida PWM del ESP8266 controla el módulo que regula el voltaje que alimenta a la luminaria.
- Protocolo 0-10V: Una salida PWM que actúa sobre un MOSFET que controla una tensión DC de 10 voltios obtenida a partir de un BOOST cuya entrada son los 5V de alimentación. El protocolo 0-10V se usa en controladores comerciales como el de la figura 8, del fabricante Lutron.



Fig. 8.Controlador 0-10V de Lutron.

- Alimentación 220 AC: Para estas luminarias se utiliza un módulo de control basado en un TRIAC. Se utiliza una salida GPIO que controla el disparo del TRIAC y una interrupción que detecta el cruce por cero de la señal alterna para así calcular el momento del disparo.

En la tabla 2, presentamos los actuadores o módulos de control con los que se puede configurar la mota inalámbrica.

Sensor/ actuador	Descripción	Interfaz
Modulo PWM-I- LED	Controlador de corriente LED ajustable por PWM	ESP8266 Salida PWM
Modulo PWM-DC- LED	Controlador de voltaje ajustable por PWM	Salida PWM
Módulo. 0- 10v	MOSFET IRF520N + BOOST	Salida PWM
Módulo TRIAC	Módulo de atenuación con TRIAC.	Entrada GPIO con interrupción y salida GPIO

Tabla 2. Módulos de control de las motas inalámbrica.

3.2.1 Desarrollo de la PCB.

Para el desarrollo de la PCB de la mota inalámbrica se utilizó el software KiCad versión 5.1.1.10. En el Anexo I se pueden encontrar los esquemáticos del diseño y los planos Gerber y Excellon necesarios para la fabricación de la placa.

A continuación, se muestra a modo de ejemplo del proceso de diseño algunas capturas. En la figura 9, se corresponde con el esquemático de la placa principal de la PCB.

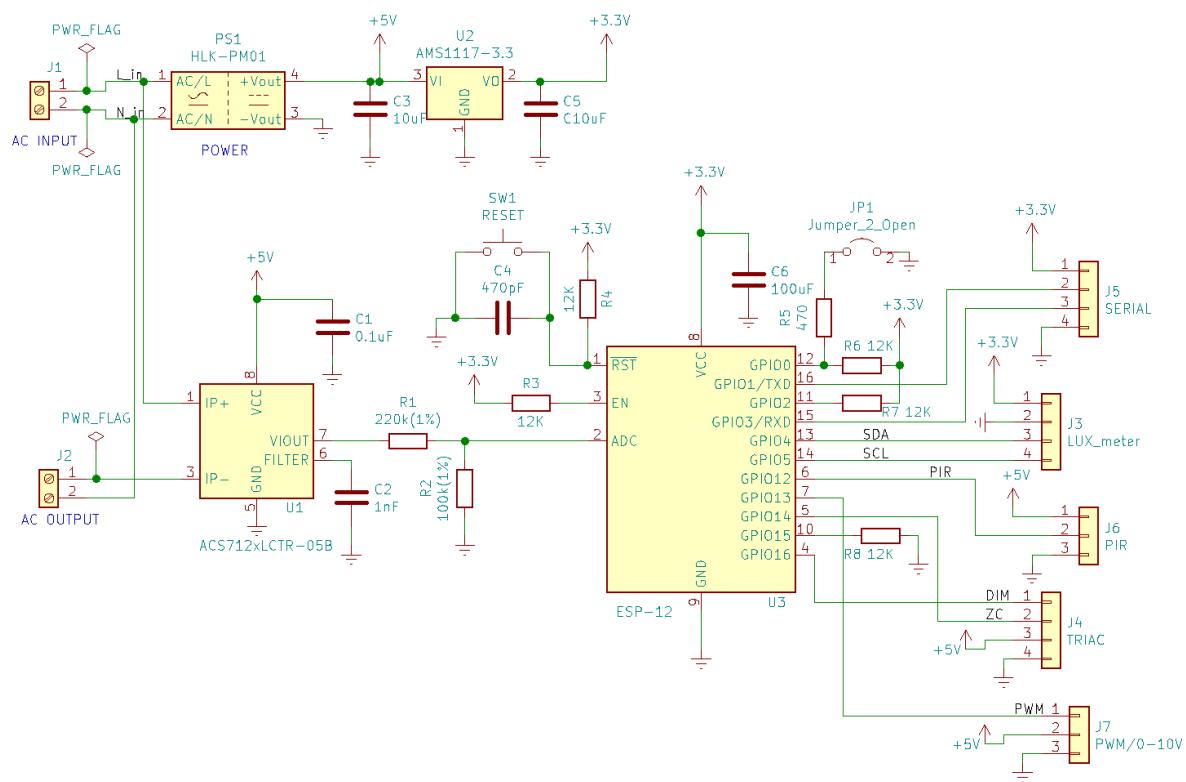


Fig. 9. Esquemático de la PCB de la moto inalámbrica.

En la figura 10, se muestra el trazado de las pistas de la cara superior de la PCB principal.

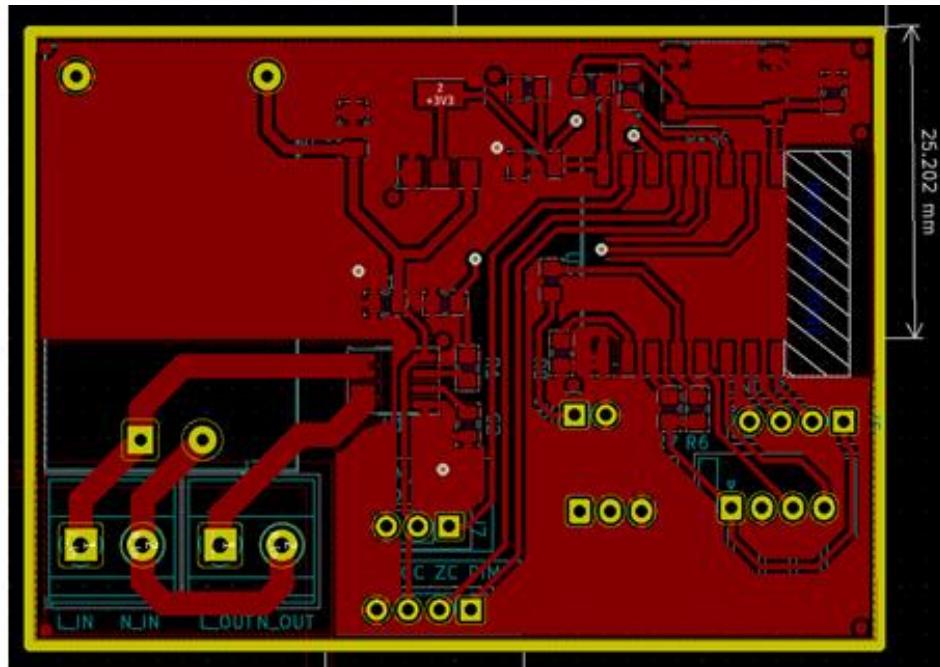


Fig. 10. Pistas de la cara superior de la PCB.

Modelo en 3D de la placa desarrollada, figura 11.

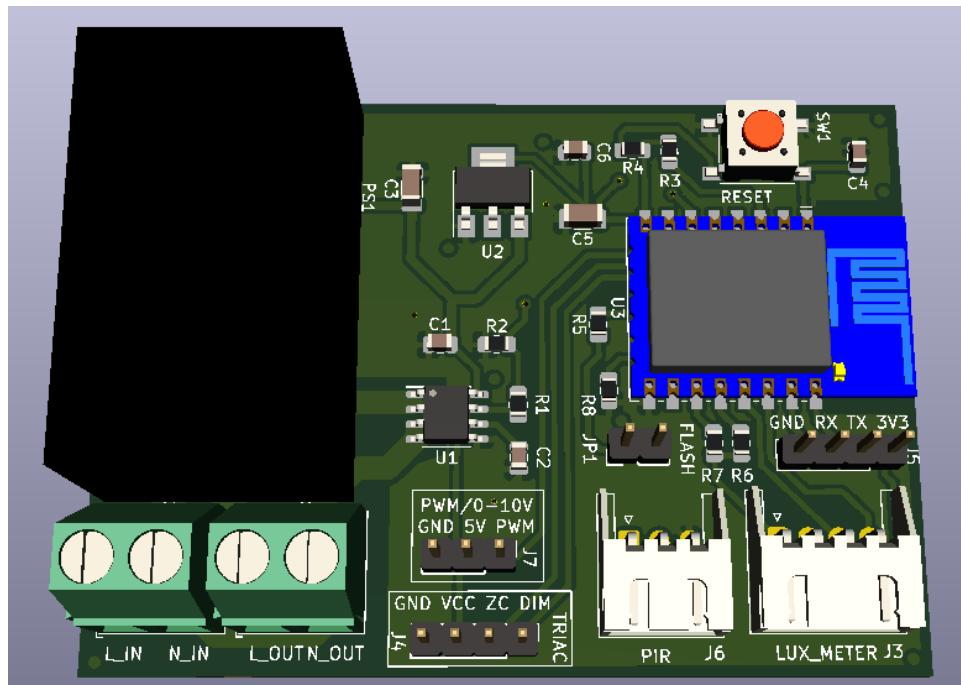


Fig. 11. Modelo en 3D de la PCB.

3.2.2 Creación del prototipo mediante grabado con CNC

Una vez desarrollado la placa PCB, el siguiente paso natural es la fabricación de la placa prototipo. Normalmente se suelen usar procesos de impresión de los Gerber en una placa de cobre fotosensible mediante insolación de rayos UV, grabado de las pistas mediante inmersión en ácido, etc., o también podríamos enviar a un fabricante la creación de la PCB, aunque sea una tirada corta de prototipos cuyo coste es mayor que los lotes grandes de producción.

Pero actualmente, gracias al auge de fresadores CNC de bajo coste, se ha desarrollado una técnica para el grabado de las pistas de circuito impreso usando estas máquinas. En este caso, se ha aprovechado el desarrollo de este proyecto para probar esta nueva técnica.

A continuación, una breve descripción del proceso.

- **Generación del programa Gcode a partir del Gerber:** Partimos de los ficheros Gerber con la definición de las pistas de cobre de las caras superior e inferior. Necesitamos generar a partir de estos ficheros el programa en código GCode para ejecutar en la fresadora CNC. Para ello vamos a utilizar el software Flatcam [2], que, con la información del diámetro y ángulo de la fresa instalada, genera el código del programa para la CNC en lenguaje Gcode. En las figuras 12 y 13 se muestran el software Flatcam con el Gerber de la cara superior del proyecto Luces.

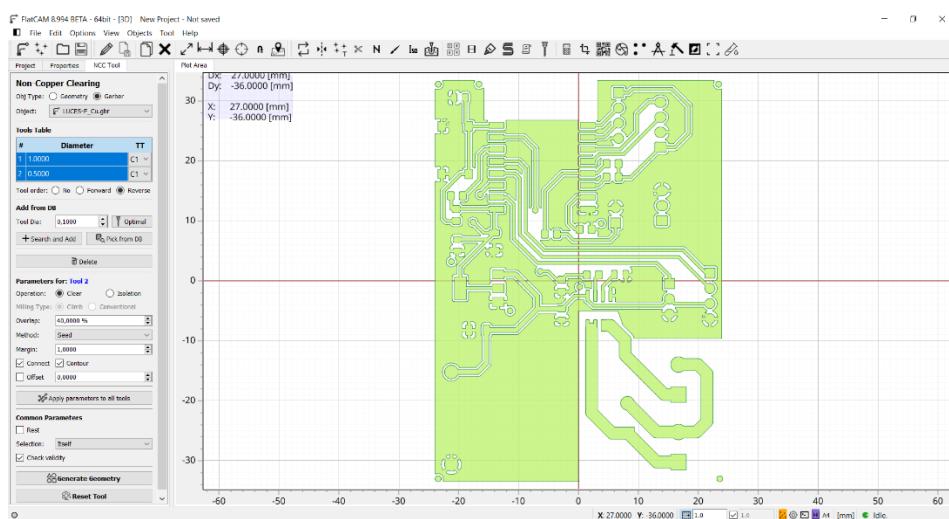


Fig. 12. Software Flatcam, para pasar el Gerber a Gcode.

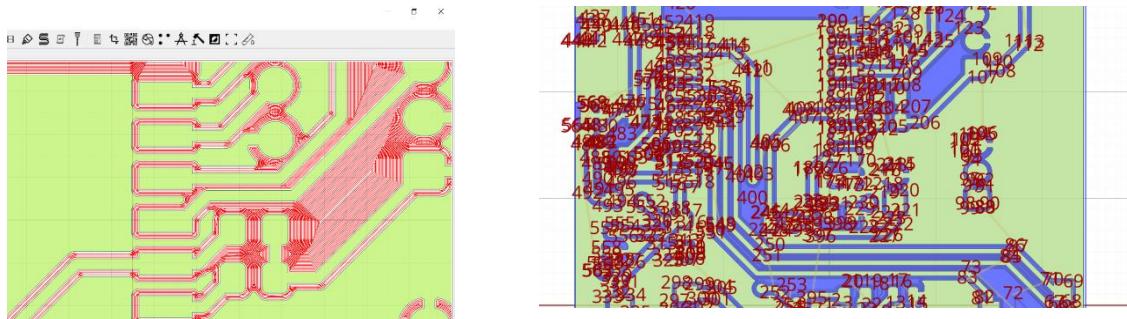


Fig. 13. Detalles del procesado del Gerber con Flatcam..

- Carga del programa Gcode en la CNC:** cargamos el programa Gcode generado, figura 14, en el software que controla la fresadora CNC, es este caso se usó *Candle*.

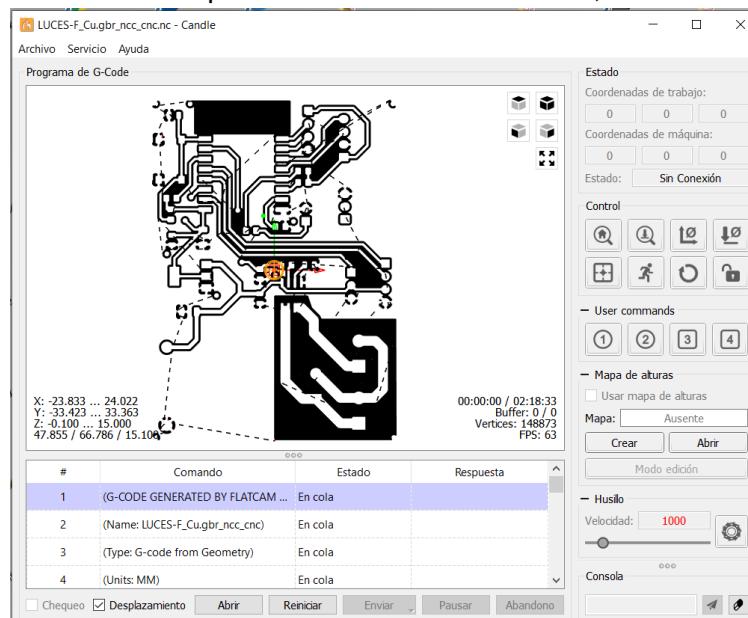


Fig. 14. Programa de control de la CNC con el Gcode cargado.

- **Fresado del cobre:** Tras diferentes configuraciones y medida de la planeidad de la placa PCB, se inicia el proceso de fresado de las zonas donde no debe haber cobre. Este proceso se muestra en la figura 15.

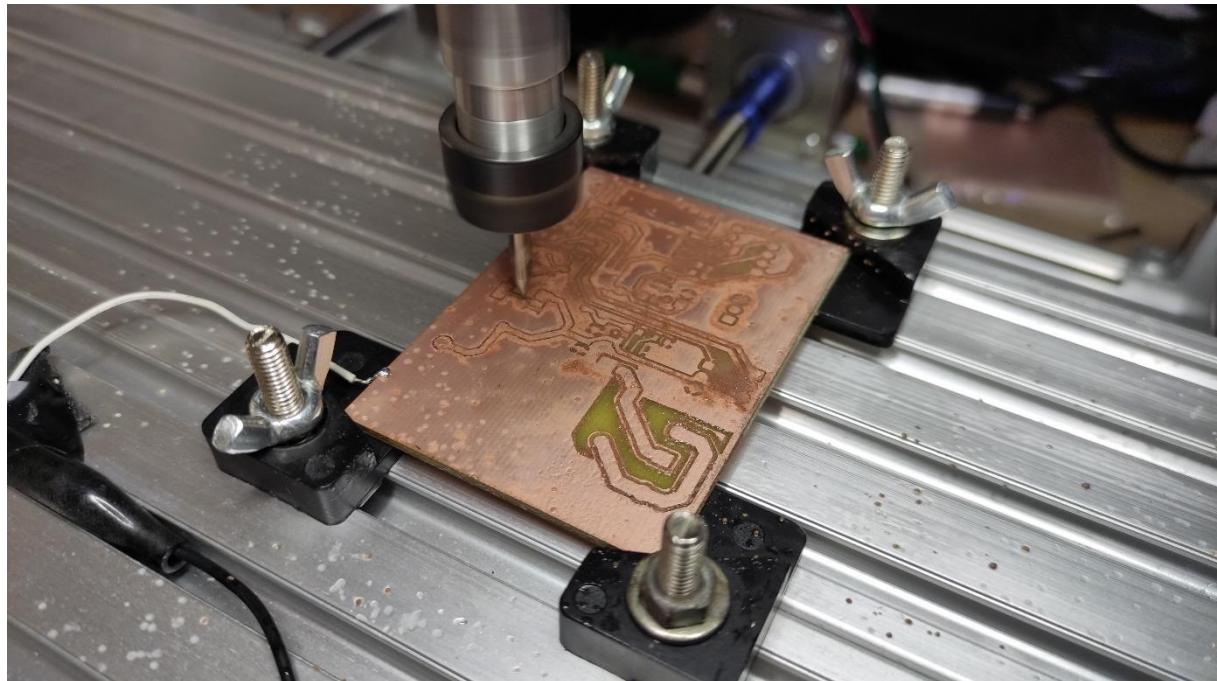


Fig. 15. Fresado del cobre no necesario en la PCB.

- **Taladrado de las vías:** Para el taladrado de las vías, en este caso se realizará a partir del fichero Excellon, la siguiente imagen muestra el fichero cargado para generar el código Gcode.

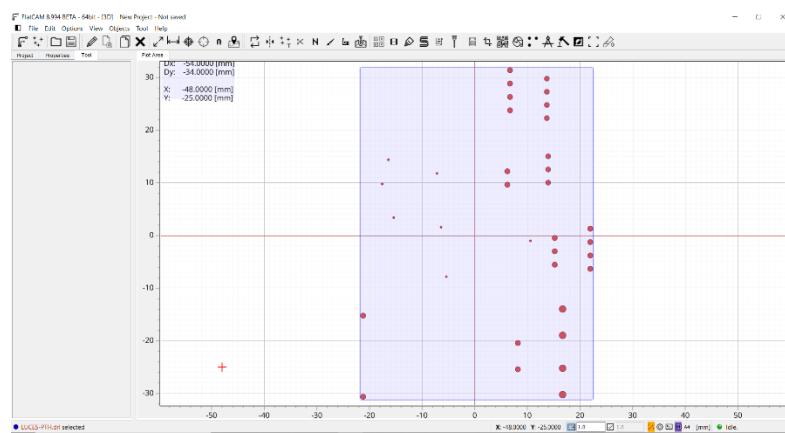


Fig. 16. Vías: Generando código Gcode a partir del Excellon.

- **Revisión del trazado de las pistas:** Una vez obtenida la placa con las vías y pistas definidas, con el multímetro en modo de conducción, podemos medir la continuidad de las pistas y posibles cortocircuitos. La PCB al final tiene este aspecto:

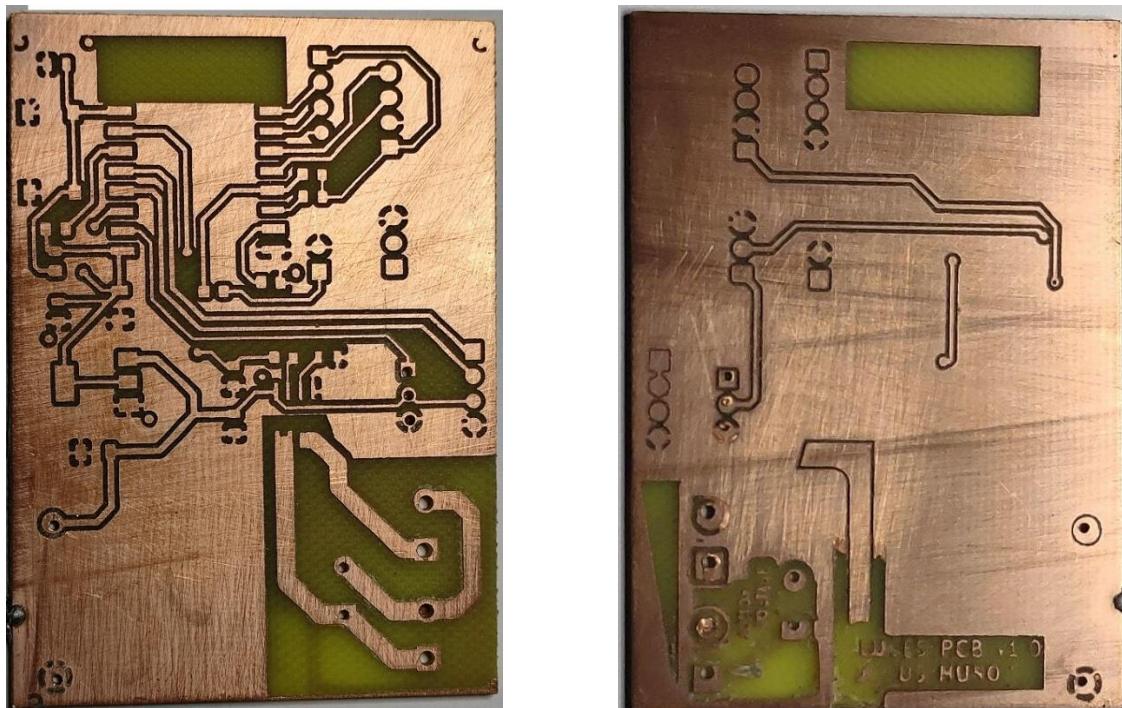


Fig. 17. Placa PCB

- **Soldadura de componentes:** Tras el chequeo eléctrico de las pistas, se sueldan los componentes y las vías, se utiliza un hilo de cobre del diámetro de las vías, 0,7 mm. Finalmente, la placa quedó como muestra en la figura 18, junto a una tarjeta microSD para tener una referencia del tamaño.

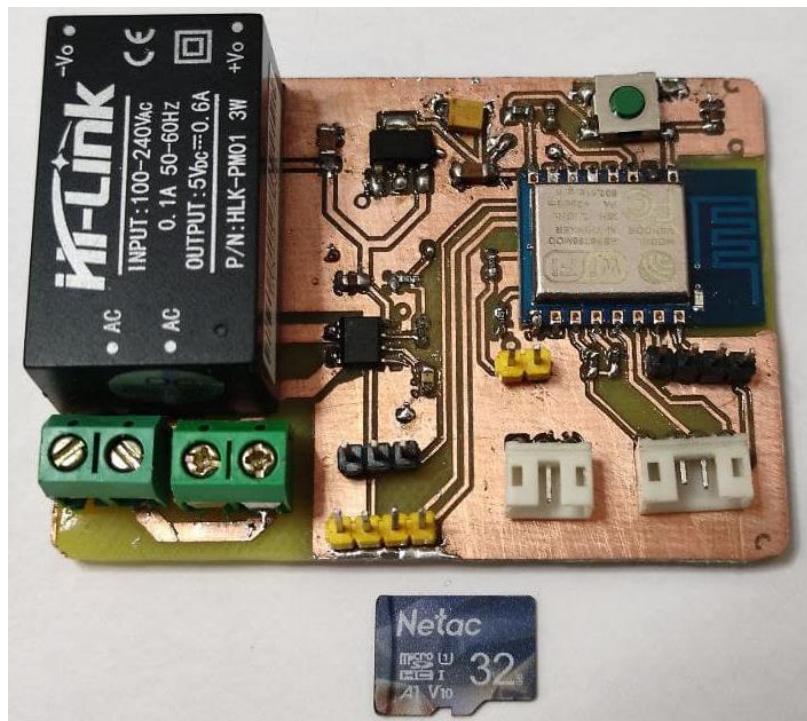


Fig. 18. PCB con los componentes soldados.

4 Implementación Software

El código fuente del software desarrollado está disponible en el siguiente enlace: <https://github.com/jesusmunozmartinez/LUCES> y descrito con más profundidad en el Anexo II.

Para la implementación del software del sistema Luces se han usado los siguientes lenguajes de programación:

- HTML+CSS y JavaScript para implementar el panel de control que corre en el servidor web que ejecuta el nodo central.
- C/C++ para programar los firmwares que corren en las motas inalámbricas y el nodo central

4.1 Firmwares del Nodo central y motas inalámbricas.

Los firmwares han sido implementados usando el IDE Visual Studio Code, editor gratuito de Microsoft de código abierto y la extensión PlatformIO IDE [3], figura, 19, orientada a la programación de firmware para sistemas empotrados y microcontroladores.

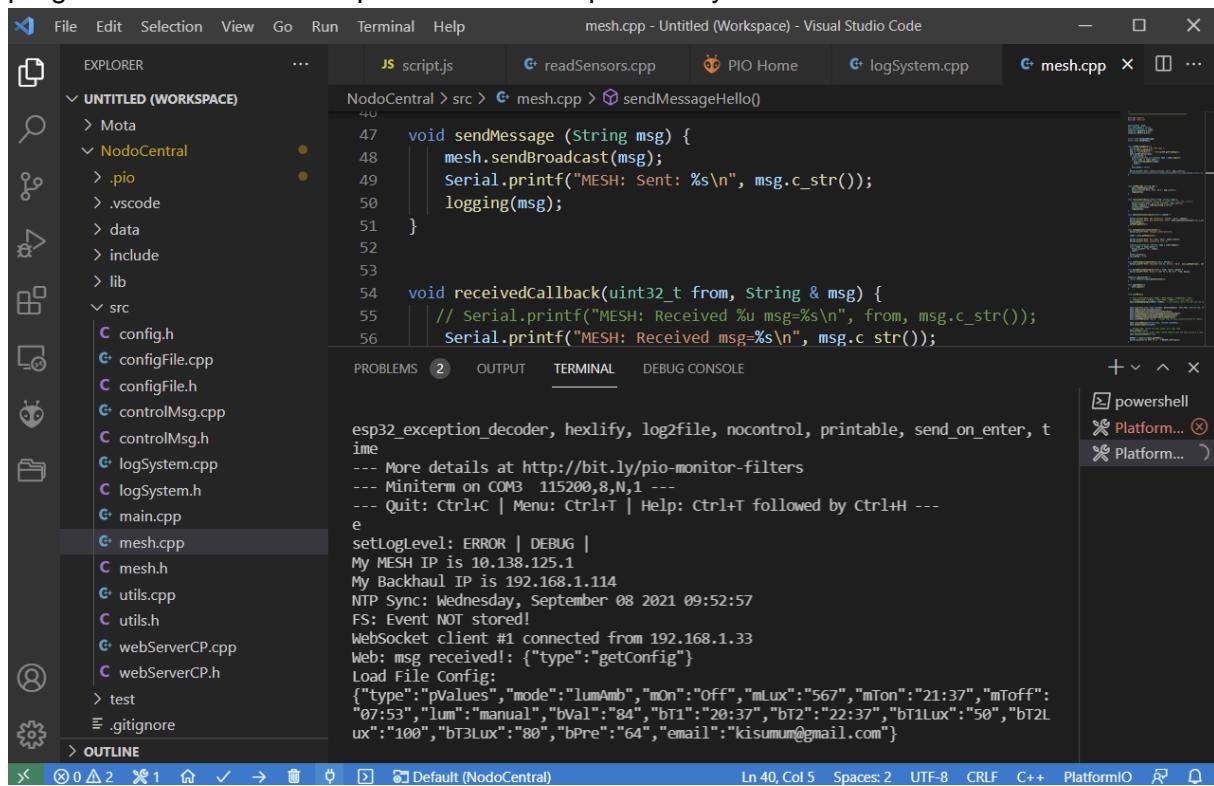


Fig. 19. Programando el firmware con Visual Studio Code + PlatformIO.

En los firmwares se han implementado las siguientes funciones que destacaríamos como más interesantes:

- **Red mallada** para las luminarias: Se ha utilizado la librería “painlessMesh” [4], para la gestión de la funcionalidad ESP-WIFI-MESH de los SoC de Expressif. De esta forma creamos una red Ad-hoc que no necesita controlador o enrutador. El nodo central hace la función de puente entre la red de luminarias y la red de área local además de implementar un servidor web para el panel de control. Esta red mallada,

no es una red IP, los nodos se identifican entre ellos por un identificador creado a partir de su MAC.

- **Multitarea cooperativa:** Se ha utilizado la librería “TaskScheduler” [5] para implementar la multitarea cooperativa de forma alternativa a como lo haría FreeRTOS
- **Sistema de archivos:** Se ha aprovechado parte de la memoria flash para crear un sistema de archivos donde guardar los archivos de configuración, registro y del servidor web. Se han usado las librerías SPIFFS y LittleFS en el ESP32 y ESP8266 respectivamente para implementar esta función.
- **Servidor web con websockets:** En el ESP32 utilizando la librería “ESPAsyncWebServer”, se han implementado un servidor web con soporte para WebSocket, que permite que cualquier cambio en los controles de configuración del servidor web se vea reflejado automáticamente en el navegador web de otros clientes conectados, y además elimina la necesidad de usar botones de enviar o similar.
- **Sincronización de la hora:** El nodo central es capaz de sincronizar su hora con servidores NTP. A su vez cuando una mota se conecta a la red mallada se sincroniza su hora con la del nodo central. Se han usado en esta ocasión las librerías “Time” y “TimeLib”.
- **Mensajes basados en JSON:** En este caso con la librería “Arduino_JSON”, se pueden encapsular los mensajes de comunicación entre las motas, nodo central y servidor web en este formato.
- **Registro cíclico del sistema.** Se ha implementado un registro del sistema que se guarda en la memoria flash del nodo central. En este caso se ha utilizado alguna funcionalidad de la librería “Logger_spiffs”.
- **Lectura de sensores:** Para el de luminosidad BH1750 se ha usado la librería “BH1740”, para el sensor de consumo de energía eléctrica se ha usado el conversor analógico digital (ADC) del ESP8266. Para el estado del sensor de presencia PIR se ha utilizado una GPIO.
- **Control de los módulos de alimentación de las luminarias:** Para las luminarias que funcionan con corriente DC y con el protocolo 0-10V, se ha utilizado una salida PWM del ESP8266. Para las luminarias que utilizan el módulo TRIAC se utiliza una salida GPIO que controla el disparo del TRIAC y una interrupción que detecta el cruce por cero de la señal alterna.
- **Envío de correo electrónico:** Para aviso en caso de fallo de la luminaria al responsable de mantenimiento. Utilizando la librería “ESP-Mail-Client” se ha implementado esta funcionalidad.
- **Modo de configuración.** En este modo, el nodo central se configura como un punto de acceso donde el usuario puede conectarse y acceder automáticamente a un portal web donde buscar las redes inalámbricas disponibles y conectarse a ella. Tras esto el nodo central se reinicia en su modo normal de operación. Se puede acceder a este modo desde el panel de control o desde un botón en el hardware del nodo central.
- **Control de intensidad lumínica suavizado.** Mediante una tarea temporizada los cambios de los valores de intensidad entre el valor normal y cuando no se detecta uso de la vía se realizan de forma progresiva para un mayor confort visual y evitar el deslumbramiento.

4.2 Comunicaciones entre subsistemas

Se ha implementado un protocolo de comunicaciones basado en mensajes JSON ya que permite una mejor visualización del funcionamiento del sistema, finalidad de un trabajo académico como éste, ya que facilita una mejor compresión al acercarse más al lenguaje natural del ser humano. Si buscásemos la seguridad u optimización de los mensajes se podría haber codificado en BASE64 u otro formato más cercano a la máquina.

Finalmente, los mensajes JSON se usan en las comunicaciones entre el servidor web y el nodo central y también entre el nodo central y las motas inalámbricas. Hay que comentar que los mensajes para el mantenimiento de la red usando la librería *painlessMesh*, están implementados también usando el formato JSON.

En el siguiente diagrama de bloques se visualiza el intercambio de los principales tipos de mensajes:

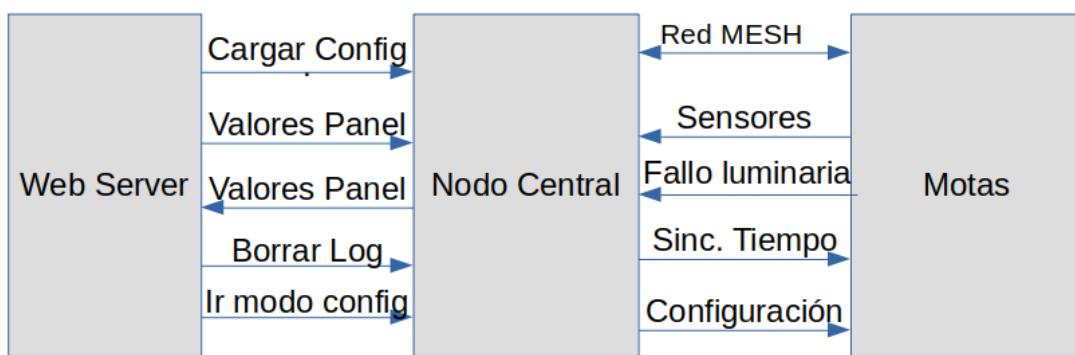


Fig. 20. Comunicaciones entre los distintos subsistemas.

Los mensajes están formados por un campo con el número de nodo que lo generó, otro con el del tipo de mensaje y el resto de los campos contienen información como valores de configuración, tiempo, etc. En la figura 21, se muestra el tratamiento de los mensajes del servidor web según los tipos a los que pertenezcan.

```

Serial.println("Web: msg received!: "+ message);
JSONVar myMsg = JSON.parse(message.c_str());
if (strcmp(myMsg["type"], "getConfig") == 0) {
    sendConfigValues();
}
if (strcmp(myMsg["type"], "pValues") == 0){
    notifyClients(message);
    saveConfig(myMsg);
    sendConfig2Nodes();
}
if (strcmp(myMsg["type"], "clearLog") == 0) {
    clearLog();
}
if (strcmp(myMsg["type"], "gotoConfig") == 0) {
    gotoConfigMode();
}

```

Fig. 21. Tratamiento de mensajes del servidor web según sus tipos.

En las motas, aparte de los mensajes de mantenimiento de la red, solo recibe mensajes de

sincronización de tiempo (tipo T) y mensajes de configuración de la mota (Tipo C). En la figura 22 se muestra su tratamiento:

```
Serial.printf("Received from %u msg=%s\n", from, msg.c_str());
JSONVar myMsg = JSON.parse(msg.c_str());
if (strcmp(myMsg["type"], "C") == 0) {
    config = JSON.parse(msg.c_str());
    saveConfig();
}
if (strcmp(myMsg["type"], "T") == 0) {
    syncTime(myMsg);
```

Fig. 22. Tratamiento de mensajes recibidos en la mota.

Para más detalles, de las funciones que ejecutan los mensajes, consultar el anexo II, donde se describe el código del programa.

4.3 Panel de control

En la figura 23, se muestra una visual del aspecto del panel de control en modo normal de operación visto en el navegador de un PC:

The screenshot shows a web browser window titled "LUCES PANEL DE CONTROL". The address bar indicates "No es seguro | 192.168.1.114". The main content area has a header "LUCES: Panel de control". Below it, there are two tabs: "Información" and "Configuracion y control".

Información tab content:

Nodo	Lum. Amb. (Lx)	Consumo(W/h)	Presencia	Estado	Brillo(%)	Fallo
2293	25	-13.48	true	ON	89	Si

Configuracion y control tab content:

- Modo de encendido:** Temporizado
- Ajuste de intensidad:** Franja Horaria
- Por franja horaria:**
 - Ajusta el valor de una franja horaria que será utilizada cada día para encender las luminarias.
 - Hora de encendido: 21:37
 - Hora de apagado: 07:53
- Ajuste por franja horaria:**
 - Selecciona el nivel de intensidad de las luminarias según la franja horaria:

21:37	-	20:37	Brillo (0-100%):	<input type="range"/>
20:37	-	22:37	Brillo (0-100%):	<input type="range"/>
22:37	-	07:53	Brillo (0-100%):	<input type="range"/>
 - Ajusta la intensidad de las luminarias en el caso de que no se detecte uso de la vía:

 Brillo sin uso de la vía: 67 %



Fig. 23. Aspecto del panel de control.

El panel de control está basado en la utilización del protocolo de comunicaciones **WebSocket**, figura 24, gracias al que cualquier cambio en la configuración del panel de control se actualiza automáticamente en todos los clientes del servidor web, en el nodo central y las motas.

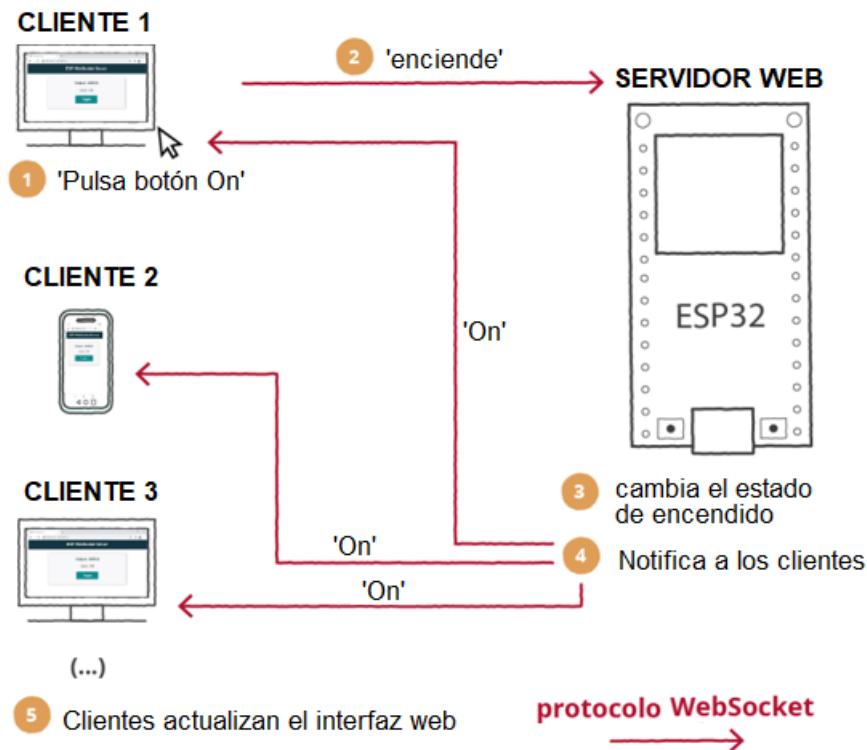


Fig. 24. Comunicación con WebSocket. Imagen traducida de la de Rui Santos [6].

El panel de control sigue el siguiente esquema de configuración de los modos de funcionamiento de las luminarias.

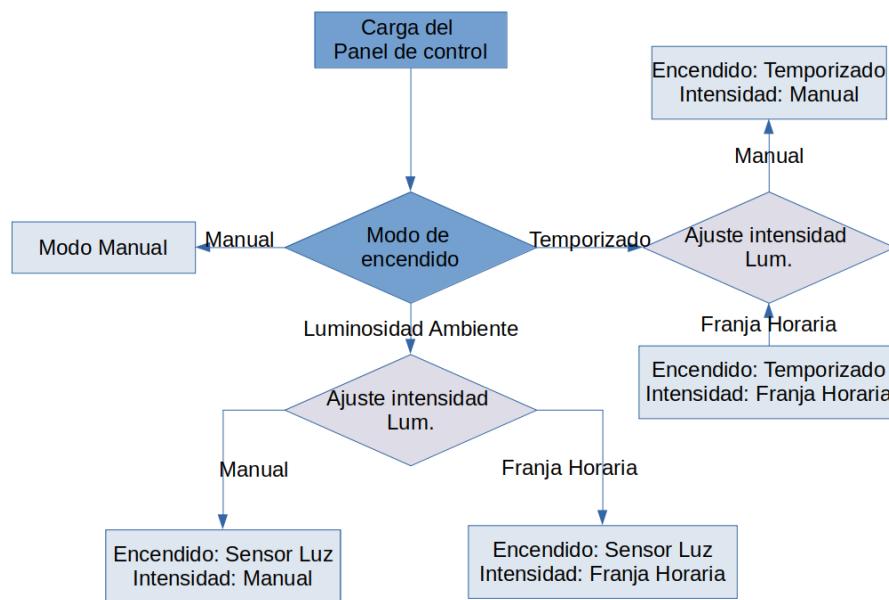


Fig. 25. Flujo de configuración de las luminarias.

En las siguientes imágenes se puede ver el aspecto y las posibilidades de configuración de las luminarias mediante el panel de control.



Modo de encendido: Por luminosidad amb.

Ajuste de intensidad: Manual

Encendido por luminosidad ambiente

Ajusta el valor de luxes medidos en el ambiente que será usado para encender y apagar las luminarias.
Como referencia la salida o puesta de sol en un día despejado son 660 Lux.

Luxes: 174

Ajuste manual de intensidad

Ajusta la intensidad de las luminarias:

Brillo: 83 %

Ajusta la intensidad de las luminarias en el caso de que **no se detecte uso de la vía**:

Brillo sin uso de la vía: 30 %

Modo de encendido: Por luminosidad amb.

Ajuste de intensidad: Franja Horaria

Encendido por luminosidad ambiente

Ajusta el valor de luxes medidos en el ambiente que será usado para encender y apagar las luminarias.
Como referencia la salida o puesta de sol en un día despejado son 660 Lux.

Luxes: 174

Ajuste por franja horaria

Selecciona el nivel de intensidad de las luminarias según la franja horaria:

21 : 37	-	23 : 37	Brillo (0-100%):	<input type="range"/>
23 : 37	-	06 : 00	Brillo (0-100%):	<input type="range"/>
06 : 00	-	07 : 53	Brillo (0-100%):	<input type="range"/>

Ajusta la intensidad de las luminarias en el caso de que **no se detecte uso de la vía**:

Brillo sin uso de la vía: 24 %

Modo de encendido: Temporizado

Ajuste de intensidad: Manual

Por franja horaria

Ajusta el valor de una franja horaria que será utilizada cada día para encender las luminarias.

Hora de encendido: 21 : 37

Hora de apagado: 07 : 53

Ajuste manual de intensidad

Ajusta la intensidad de las luminarias:

Brillo: 83 %

Ajusta la intensidad de las luminarias en el caso de que **no se detecte uso de la vía**:

Brillo sin uso de la vía: 24 %

Modo de encendido: Temporizado

Ajuste de intensidad: Franja Horaria

Por franja horaria

Ajusta el valor de una franja horaria que será utilizada cada día para encender las luminarias.

Hora de encendido: 21 : 37

Hora de apagado: 07 : 53

Ajuste por franja horaria

Selecciona el nivel de intensidad de las luminarias según la franja horaria:

21 : 37	-	23 : 37	Brillo (0-100%):	<input type="range"/>
23 : 37	-	06 : 00	Brillo (0-100%):	<input type="range"/>
06 : 00	-	07 : 53	Brillo (0-100%):	<input type="range"/>

Ajusta la intensidad de las luminarias en el caso de que **no se detecte uso de la vía**:

Brillo sin uso de la vía: 24 %

Fig. 26. Modos de configuración del panel de control.

Desde el panel de control se puede acceder al “modo de configuración”, además desde un botón hardware.

Modo de configuración del punto de acceso

En el modo de configuración podrá configurar la red WiFi a la que desea conectarse, actualizar el firmware y consultar información técnica del hardware.

[Ir a modo de configuración](#)

Fig. 27. Acceso al modo de configuración desde el panel de control.

Los detalles de funcionamiento del modo de configuración se describen en el Anexo IV Manual de usuario, aunque su principal función es la de administrar la configuración de la red WiFi que se usará de backhaul o de control. La siguiente imagen muestra un aspecto de

este.

23:48 ☀️ 🔋

X

✓

LUCES_CONFIG

Conectarse automáticamente

k@m	
k@m_invitados	

SSID

Password

Fig. 28. Modo de configuración.

5 Referencias

- [1] Espressif, «ESP32-DevKitC V4 Getting Started Guide,» [En línea]. Available: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/hw-reference/esp32/get-started-devkitc.html>. [Último acceso: Julio 2021].
- [2] J. P. Caram, «FlatCAM,» Enero 2021. [En línea]. Available: <http://flatcam.org/>.
- [3] Platformio.org, «PlatformIO IDE,» Julio 2021. [En línea]. Available: <https://platformio.org/platformio-ide>.
- [4] PainlessMesh, «PainlessMesh,» BlackEdder, [En línea]. Available: <https://gitlab.com/painlessMesh/painlessMesh/-/wikis/home>. [Último acceso: Julio 2021].
- [5] A. Arkhipenko, «TaskScheduler,» [En línea]. Available: <https://github.com/arkhipenko/TaskScheduler>. [Último acceso: Julio 2021].
- [6] R. Santos, «Random nerd tutorials,» Julio 2021. [En línea]. Available: <https://randomnerdtutorials.com/>.
- [7] Espressif, «ESP-WIFI-MESH,» Espressif, [En línea]. Available: <https://www.espressif.com/en/products/sdks/esp-wifi-mesh/overview>. [Último acceso: Agosto 2021].
- [8] F14V, «Thingiverse,» [En línea]. Available: <https://www.thingiverse.com/thing:2786523>. [Último acceso: septiembre 2021].

Pruebas

RESUMEN DE DOCUMENTO

Descripción de las pruebas de verificación del funcionamiento individual de los subsistemas que forman la arquitectura y su correcta comunicación, así como de las pruebas de validación de los requisitos del sistema completo.

TÍTULO DOCUMENTO	Pruebas
FECHA DE ENTREGA	30/09/2021
AUTOR/ES	Jesús Muñoz Martínez

COMENTARIOS	VERSIÓN	FECHA
Versión Inicial	1.0	06/08/2021

Índice

1	VERIFICACIÓN DE REQUISITOS	5
1.1	ENTORNO 1: PRUEBAS DE LABORATORIO.	6
1.1.1	Prueba. Lectura de sensores en local	6
1.1.2	Prueba: Control de luminaria en local.	6
1.1.3	Prueba: Nodo central en local	7
1.1.4	Prueba: Registro del sistema	8
1.1.5	Prueba: Establecimiento red MESH.	9
1.1.6	Prueba: Nodo central y motas en red.	10
1.1.7	Prueba: Fallo de luminaria	11
1.1.8	Prueba: Actualización de firmware	12
1.2	ENTORNO 2. VALIDACIÓN DEL PROTOTIPO	13
1.2.1	Prueba. Establecimiento de la red MESH y lectura de medidas.	13
1.2.2	Prueba: Configuración y control de luminaria.	14
1.2.3	Prueba: Fallo de luminaria.	15
1.2.4	Prueba: Revisión del registro del sistema	16
2	PLAN DE ACEPTACIÓN	18
2.1	PRUEBAS	18

Índice de figuras.

<i>Fig. 1. Evidencia de prueba: Lectura de sensores en local.....</i>	6
<i>Fig. 2. Evidencia de prueba: Control de luminaria en local</i>	7
<i>Fig. 3. Evidencia de prueba: Nodo central en local</i>	8
<i>Fig. 4. Evidencia de prueba: Registro de sistema</i>	9
<i>Fig. 5. Evidencia de prueba: Establecimiento red MESH</i>	10
<i>Fig. 6. Evidencia de prueba: Nodo central y motas en red</i>	11
<i>Fig. 7. Evidencia de prueba: Fallo de luminaria.....</i>	12
<i>Fig. 8. Evidencia de prueba: Actualización de firmware.....</i>	12
<i>Fig. 9. Evidencia de prueba: Establecimiento de la red MESH y lectura de medias</i>	14
<i>Fig. 10. Evidencia de prueba: Configuración y control de luminaria</i>	15
<i>Fig. 11. Evidencia de prueba: Fallo de luminaria.....</i>	16
<i>Fig. 12. Evidencia de prueba: Revisión del registro del sistema</i>	17

Índice de tablas.

<i>Tabla 1.Requisitos.</i>	5
----------------------------	---

Definiciones, acrónimos y abreviaturas

LED	Light Emitting Diode (Diodo emisor de luz).
MESH	Topología de red mallada.
NTP	Network Time Protocol.

*Nota: En la carpeta “**Videos**” del CD-ROM y de la nube de GitHub se encuentran videos de diferentes pruebas realizadas sobre los prototipos.*

1 Verificación de requisitos

La siguiente tabla muestra los requisitos implementados :

Id	Nombre	Prioridad	Precedencia
R1.1	Esquemáticos	F	
R1.2	PCB	F	R1.2
R1.3	ManualInstalación	D	R1.2
R1.4	ManualUsuario	D	R4.* - R13.*.
R2	Firmware	F	R4.*-R7, R8.*- R11.*.
R3.1	NodoCentral	F	
R3.2	MotaWireless	F	
Requisitos aplicables al nodo central			
R4.1	NCEncendidoManual	F	R8.*.
R4.2	NCApagadoManual	F	R8.*.
R4.3	NCEncendidoTemporizado	F	R8.* , R9.6, R10.2.
R4.4	NCEncendidoAutoLuminosidadAmbiente	D	R8.* , R9.1, R10.1.
R5.1	NCAjusteLuminosidadLuminarias	F	R8.* , R9.1, R10.1.
R5.2	NCAjusteLuminosidadFranjaHoraria	F	R8.* , R9.1, R9.5
R5.3	NCAjusteLuminosidadPresencia	O	R8.* , R9.1, R9.2.
R6.1	NCInfoLuminosidad	O	R9.1, R11.2.
R6.2	NCInfoPresencia	O	R9.2, R11.2.
R6.3	NCInfoConsumo	O	R9.5, R11.2.
R6.4	NCInfoFallo	F	R9.3, R11.1.
R7	RegistroNodoCentral	F	R11.1, R11.2.
Requisitos aplicables a las motas			
R8.1	ControlLuminariaTipo1	F	
R8.2	ControlLuminariaTipo2	D	
R8.3	ControlLuminariaTipo3	O	
R9.1	MedirLuminosidad	D	
R9.2	DetectarPresencia	O	
R9.3	DetectarFallo	F	R9.1, R9.4
R9.4	MedirConsumo	O	
R9.5	MediciónHoraria	F	
R10.1	NivelLuminosidadProgramable	F	
R10.2	TemporizaciónProgramable	F	R9.5
R11.1	AvisoFallo	F	R4.2, R9.3.
R11.2	EnvioEstado	D	R4.2, R9.*.
R12	OTAFirmware	F	
R13	AlimentacionMotaRedElectrica	F	

Tabla 1.Requisitos.

1.1 Entorno 1: Pruebas de laboratorio.

Se cuenta con el sistema de iluminación software/hardware completo ya desarrollado. Se encuentra desplegado en un entorno de laboratorio, los nodos tienen habilitados los puertos de depuración. Se dispone un PC de pruebas que puede conectarse a cualquier tipo de nodo a través del puerto de depuración y por la red inalámbrica. Este PC tiene un software de pruebas que le permite enviar y recibir mensajes y actuar como mota o nodo central.

1.1.1 Prueba. Lectura de sensores en local

Requisito/s		R3.2, R9.1, R9.2, R9.4, R9.5, R11.2, R13
Prueba		<p>Se conectará el PC de pruebas a una mota inalámbrica por su puerto de depuración y por la red inalámbrica.</p> <p>Se encenderá la mota con el modo depuración activado y se leerán los valores del sensor de luminosidad, presencia y consumo por el puerto de depuración y mediante los mensajes recibidos en la red inalámbrica.</p> <p>Se pondrá la frecuencia de lectura de sensores a 5 segundo.</p> <p>Se simularán distintas situaciones de luminosidad, presencia y consumo.</p>
Salida		<ol style="list-style-type: none"> En el PC de pruebas en la terminal 1 se muestra los datos de los sensores recibidos mediante el puerto de depuración, y en la terminal 2 los datos de los sensores recibidos por la red inalámbrica. Mensaje con las medidas de los sensores en las interfaces mencionadas en el punto a.
Errores		<ol style="list-style-type: none"> No se reciben datos de los sensores o son erróneos Los valores leídos por el puerto de depuración difieren de los recibidos por la red inalámbrica. Se pierden mensajes por la red inalámbrica que si se han recibido por el puerto de depuración.
Evidencia		<pre> CTRL: Bright to 133 Modo LumAmb, manual on CTRL: Bright to 133 Time: Mon Sep 6 21:08:01 2021 Broadcast: {"node":"2753","type":"S","lux":"201","I":"1.15","pir":false,"onOff":"ON","bright":"52","fail":"NO"} CTRL: Bright to 133 Modo LumAmb, manual on CTRL: Bright to 133 Modo LumAmb, manual on CTRL: Bright to 133 [Jmuoz@jmuoz-Latitude-5300:~ {"event":"receive","nodeTime":567042367,"time":"2021-09-06T21:07:37","nodeId":2785102753,"msg":{"node":2753,"type":"S","lux":202,"I":1.21,"pir":false,"onOff":"ON","bright":52,"fail":"NO"}} {"event":"receive","nodeTime":572577205,"time":"2021-09-06T21:07:42","nodeId":2785102753,"msg":{"node":2753,"type":"S","lux":202,"I":1.22,"pir":false,"onOff":"ON","bright":52,"fail":"NO"}} {"event":"receive","nodeTime":579060790,"time":"2021-09-06T21:07:49","nodeId":2785102753,"msg":{"node":2753,"type":"S","lux":201,"I":1.21,"pir":false,"onOff":"ON","bright":52,"fail":"NO"}} {"event":"receive","nodeTime":585187175,"time":"2021-09-06T21:07:55","nodeId":2785102753,"msg":{"node":2753,"type":"S","lux":201,"I":1.21,"pir":false,"onOff":"ON","bright":52,"fail":"NO"}} {"event":"receive","nodeTime":591491601,"time":"2021-09-06T21:08:01","nodeId":2785102753,"msg":{"node":2753,"type":"S","lux":201,"I":1.15,"pir":false,"onOff":"ON","bright":52,"fail":"NO"}} </pre>
Resultado		PASA.

1.1.2 Prueba: Control de luminaria en local.

Requisito/s	R3.2, R8. *, R9.1, R9.2, R9.4, R9.5, R10. *, R11.2, R13
Prueba	<p>Se conectará el PC de pruebas a una mota inalámbrica por su puerto de depuración y por la red inalámbrica.</p> <p>Se encenderá la mota con el modo depuración activado y se le enviaran mensajes de control con los distintos modos de funcionamiento y ajuste de intensidad posibles.</p> <p>Se hará la prueba con todos los tipos de luminarias soportados.</p>
Salida	<ul style="list-style-type: none"> a. En el PC de pruebas en la terminal 1 se muestra los datos de los mensajes recibidos mediante el puerto de depuración, y en la terminal 2 los datos de los mensajes recibidos por la red inalámbrica. b. Luminaria encendida y brillo de ésta. c. Mensaje de control y su respuesta en las interfaces mencionadas en el punto a.
Errores	<ul style="list-style-type: none"> a. La mota no responde a la orden de control enviada por algunas de las interfaces. b. La mota no devuelve mensaje de respuesta a la red inalámbrica. c. Existe un retraso considerable desde que la mota recibe la orden y se controla la luminaria. d. Algun tipo de luminaria no se puede controlar.
Evidencia	<pre> { "type": "C", "mode": "lumAmb", "mOn": "Off", "mLux": "736", "mTon": "21:37", "mToff": "07:53", "lum": "manual", "bVal": "75", } Modo LumAmb, manual on CRTL: Bright to 191 Modo LumAmb, manual on CRTL: Bright to 191 Time: Mon Sep 6 21:10:57 2021 Broadcast: {"node": "2753", "type": "S", "lux": "201", "I": "1.19", "pir": false, "onOff": "ON", "bright": "75", "fail": "NO"} n CRTL: Bright to 196 Modo LumAmb, manual on CRTL: Bright to 196 </pre> <p>Fig. 2. Evidencia de prueba: Control de luminaria en local</p>
Resultado	PASA

1.1.3 Prueba: Nodo central en local

Requisito/s	R3.1, R4. *, R5. *, R6. *
Prueba	<p>Se conectará el PC de pruebas al nodo central por su puerto de depuración, por la red inalámbrica y por su panel de control.</p> <p>Se encenderá el nodo central con el modo depuración activado y se le enviaran mensajes desde el PC de pruebas como si de una mota inalámbrica se tratase.</p>

	Desde el nodo central se enviarán todos los mensajes de control al PC que emula ser una mota inalámbrica.
Salida	<ul style="list-style-type: none"> a. En el PC de pruebas en la terminal 1 se muestra los datos de los mensajes recibidos mediante el puerto de depuración, y en la terminal 2 los datos de los mensajes recibidos por la red inalámbrica. b. En el navegador web se muestra la información del panel de control. c. Mensaje de control en las interfaces mencionadas en el punto a.
Errores	<ul style="list-style-type: none"> a. El nodo central no interpreta correctamente los mensajes recibidos de la mota emulada. b. El nodo central no visualiza correctamente en el panel de control la información recibida en los mensajes. c. Los mensajes de control que envía el nodo central están malformados. d. El nodo central no es capaz de manejar todos los mensajes y pierde información.
Evidencia	<pre> MESH: RECEIVED msg=1 node : 2293 , type : S , lux : 25 , I : -13.48 , pir : true, onOff : ON , bright : 63 , fail : 0 MESH: Received msg={"node": "2293", "type": "S", "lux": "25", "I": "-13.48", "pir": true, "onOff": "ON", "bright": "63", "fail": "0"} MESH: Received msg=1 node : 2293 , type : S , lux : 25 , I : -13.48 , pir : true, onOff : ON , bright : 63 , fail : 0 MESH: Received msg={"node": "2293", "type": "S", "lux": "25", "I": "-13.48", "pir": true, "onOff": "ON", "bright": "63", "fail": "0"} MESH: Received msg=1 node : 2293 , type : S , lux : 25 , I : -13.48 , pir : true, onOff : ON , bright : 63 , fail : 0 MESH: Received msg={"node": "2293", "type": "S", "lux": "25", "I": "-13.48", "pir": true, "onOff": "ON", "bright": "63", "fail": "0"}[{"event": "offset", "nodeTime": 403117036, "time": "2021-09-06T21:20:36", "offset": -4707}, {"event": "offset", "nodeTime": 403122353, "time": "2021-09-06T21:20:36", "offset": -2104}, {"event": "receive", "nodeTime": 406520026, "time": "2021-09-06T21:20:40", "nodeId": 2144992293, "msg": {"node": "2293", "type": "S", "lux": "25", "I": "-13.48", "pir": true, "onOff": "ON", "bright": "63", "fail": "SI"}}, {"event": "receive", "nodeTime": 411845394, "time": "2021-09-06T21:20:45", "nodeId": 2144992293, "msg": {"node": "2293", "type": "S", "lux": "25", "I": "-13.48", "pir": true, "onOff": "ON", "bright": "63", "fail": "SI"}}, {"event": "receive", "nodeTime": 421061854, "time": "2021-09-06T21:20:54", "nodeId": 2144992293, "msg": {"node": "2293", "type": "S", "lux": "25", "I": "-13.48", "pir": true, "onOff": "ON", "bright": "63", "fail": "SI"}}, {"event": "receive", "nodeTime": 430483288, "time": "2021-09-06T21:21:04", "nodeId": 2144992293, "msg": {"node": "2293", "type": "S", "lux": "25", "I": "-13.48", "pir": true, "onOff": "ON", "bright": "63", "fail": "SI"}} </pre>
Resultado	PASA

1.1.4 Prueba: Registro del sistema

Requisito/s	R3.1, R7
Prueba	<p>Se conectará el PC de pruebas al nodo central por su puerto de depuración, por la red inalámbrica y por su panel de control.</p> <p>Se encenderá el nodo central con el modo depuración activado y se le enviarán mensajes desde el PC de pruebas como si de una mota inalámbrica se tratase.</p> <p>Se configurará el tamaño del registro del sistema a un tamaño de 100k o menor para realizar esta prueba.</p> <p>Se comprobará que los mensajes son almacenados correctamente en el registro del sistema y que cuando se llena se sobrescriben en el orden correcto.</p> <p>Se ejecutarán las operaciones del panel control asociadas al registro del sistema como visualizar, descargar y borrar.</p>

Salida	<ul style="list-style-type: none"> a. En el PC de pruebas en la terminal 1 se muestra los datos de los mensajes recibidos mediante el puerto de depuración, y en la terminal 2 los datos de los mensajes recibidos por la red inalámbrica. b. En el navegador web se muestra la información del panel de control. c. Visualización del registro del sistema por el panel de control y la interfaz de depuración.
Errores	<ul style="list-style-type: none"> a. El nodo central deja de responder cuando el registro del sistema se llena. b. El nodo central no almacena algunos mensajes en el registro del sistema. c. El nodo central no sobrescribe correctamente al llenarse el registro del sistema. d. El registro del sistema no se vacía con la operación borrar.
Evidencia	<pre>MESH: Received msg={"node": "2293", "type": "S", "lux": "25", "I": "-13.48", "pir": true, "onOff": "ON", "bright": "74", "fai"} ! Web: msg received!: {"type": "clearLog"} ← FS: Log cleared!</pre> <pre>MESH: Received msg={"node": "2293", "type": "S", "lux": "25", "I": "-13.48", "pir": true, "onOff": "ON", "bright": "74", "fai"} ! MESH: Received msg={"node": "2293", "type": "S", "lux": "25", "I": "-13.48", "pir": true, "onOff": "ON", "bright": "74", "fai"} ! My Backhaul IP is 192.168.1.114 NTP Sync: Monday, September 06 2021 21:26:22 FS: Event stored! MESH: Sent: {"type": "T", "time": 1630956382, "hOffset": 2} → WebSocket client #1 connected from 192.168.1.33 Web: msg received!: {"type": "getConfig"} ← Load File Config: {"type": "pValues", "mode": "lumAmb", "mOn": "Off", "mLux": "675", "mTon": "21:37", "mToff": "07:53", "lum": "manual", "bVal": 1} MESH: Received msg={"node": "2293", "type": "S", "lux": "25", "I": "-13.48", "pir": true, "onOff": "ON", "bright": "74", "fai"} ! MESH: Received msg={"node": "2293", "type": "S", "lux": "25", "I": "-13.48", "pir": true, "onOff": "ON", "bright": "74", "fai"} ! □</pre> <p style="text-align: center;">Fig. 4. Evidencia de prueba: Registro de sistema</p>
Resultado	PASA

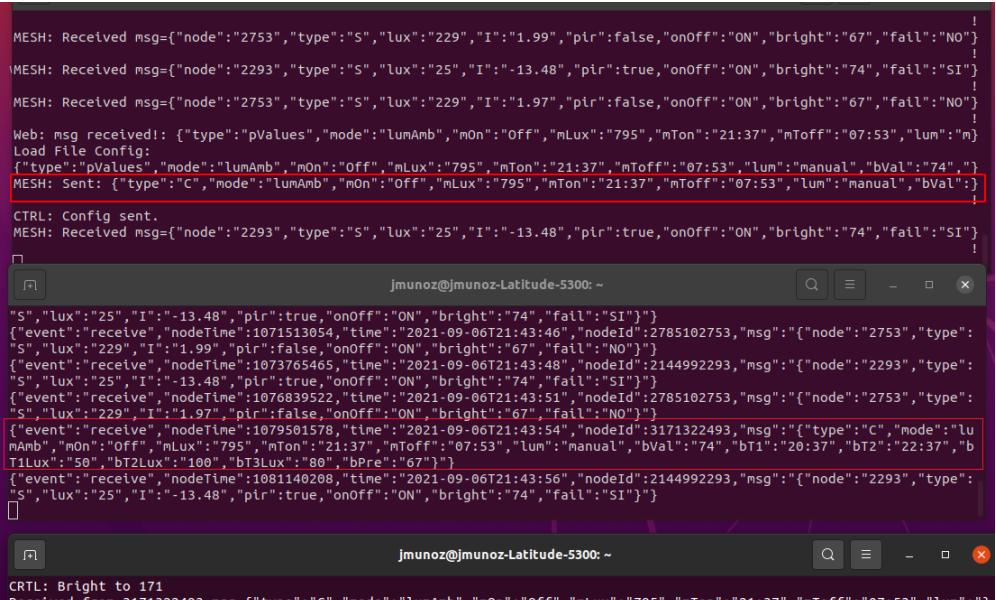
1.1.5 Prueba: Establecimiento red MESH.

Requisito/s	R3.* , R6.* , R7
Prueba	<p>Se conectará el PC de pruebas al nodo central por su puerto de depuración, por la red inalámbrica y por su panel de control.</p> <p>Se encenderá el nodo central con el modo depuración activado y se empezarán a encender las motas inalámbricas.</p> <p>Se comprobará que la red MESH se está formando por los mensajes de anuncio que llegan al nodo central.</p> <p>Se observará en el panel de control si la red está completa.</p>
Salida	<ul style="list-style-type: none"> a. En el PC de pruebas en la terminal 1 se muestra los mensajes de depuración del nodo central y en la terminal 2 los mensajes que circulan en la red inalámbrica, incluidos los mensajes de anuncio de asociación a la red recibidos. b. En el navegador web se muestra el panel de control donde se

	<p>pueden ver las motas asociadas a la red.</p> <p>c. Mensaje de anuncio y asociación red en las interfaces mencionadas en el punto a.</p>
Errores	<p>a. El nodo central no es capaz de iniciar la red MESH.</p> <p>b. El nodo central no visualiza correctamente las motas en el panel de control.</p> <p>c. Motas inalámbricas no son capaces de asociarse a la red de malla.</p>
Evidencia	<pre> MESH: Received msg={"node": "2293", "type": "S", "lux": "25", "I": "-13.48", "pir": true, "onOff": "ON", "bright": "63", "fail": "NO"} MESH: Received msg={"node": "2293", "type": "S", "lux": "25", "I": "-13.48", "pir": true, "onOff": "ON", "bright": "63", "fail": "NO"} MESH: Changed connections MESH: Num nodes: 2 MESH: New Connection, nodeId = 2083651461 MESH: Connection list: 2144992293 2083651461 MESH: New Connection, { "nodeId": 3171322493, "root": true, "subs": [{ "nodeId": 2144992293 }, { "nodeId": 2083651461 }] } MESH: Sent: {"type": "T", "time": 1630955786, "hOffset": 2} FS: Event stored! </pre> <pre> setLogLevel: ERROR {"event": "connect", "nodeTime": 1238351654, "time": "2021-09-06T21:02:19", "nodeId": 3171322493, "layout": {"nodeId": 4059128081, "subs": [{"nodeId": 3171322493, "root": true, "subs": [{"nodeId": 2785102753}]}]}} {"event": "change", "nodeTime": 1238351912, "time": "2021-09-06T21:02:19", "layout": {"nodeId": 4059128081, "subs": [{"nodeId": 3171322493, "root": true, "subs": [{"nodeId": 2785102753}]}]}} {"event": "receive", "nodeTime": 1238369917, "time": "2021-09-06T21:02:19", "nodeId": 3171322493, "msg": {"type": "T", "time": 1630954939, "hOffset": 2}} {"event": "receive", "nodeTime": 1241362206, "time": "2021-09-06T21:02:22", "nodeId": 2785102753, "msg": {"node": "2753", "type": "S", "lux": "200", "I": "1.19", "pir": false, "onOff": "ON", "bright": "52", "fail": "NO"}} {"event": "receive", "nodeTime": 1246265793, "time": "2021-09-06T21:02:27", "nodeId": 2785102753, "msg": {"node": "2753", "type": "S", "lux": "200", "I": "1.21", "pir": false, "onOff": "ON", "bright": "52", "fail": "NO"}} {"event": "offset", "nodeTime": 259123664, "time": "2021-09-06T21:02:29", "offset": -989246788} {"event": "offset", "nodeTime": 4059480063, "time": "2021-09-06T21:02:29", "offset": -494620072} {"event": "offset", "nodeTime": 259345425, "time": "2021-09-06T21:02:29", "offset": 494620377} {"event": "offset", "nodeTime": 259562166, "time": "2021-09-06T21:02:29", "offset": -2556} </pre>
Resultado	PASA

1.1.6 Prueba: Nodo central y motas en red.

Requisito/s	R3.* , R4.* , R5.* , R6.* , R8.* , R9.* , R10.* , R11.2
Prueba	<p>Se conectará el PC de pruebas al nodo central por su puerto de depuración, por la red inalámbrica y por su panel de control.</p> <p>Se iniciará la red MESH encendiendo todos los nodos.</p> <p>Desde el nodo central se enviarán todos los mensajes de control a las motas.</p>
Salida	<p>a. En el PC de pruebas en la terminal 1 se muestra los datos de los mensajes recibidos mediante el puerto de depuración, y en la terminal 2 los datos de los mensajes recibidos por la red inalámbrica.</p> <p>b. En el navegador web se muestra la información del panel de control.</p> <p>c. Mensaje del sistema de iluminación en las interfaces mencionadas en el punto a.</p>

Errores	<ul style="list-style-type: none"> a. El nodo central mezcla mensajes de distintas motas. b. Motas envían mensajes de estado a todos los nodos (broadcast) y no solo al nodo central. c. El nodo central no es capaz de manejar todos los mensajes y pierde información.
Evidencias	 <p>Fig. 6. Evidencia de prueba: Nodo central y motas en red</p>
Resultado	PASA

1.1.7 Prueba: Fallo de luminaria

Requisito/s	R3.* , R6.4, R7, R9.3, R11.1.
Prueba	<p>Se conectará el PC de pruebas al nodo central por su puerto de depuración, por la red inalámbrica y por su panel de control.</p> <p>Se iniciará la red MESH encendiendo todos los nodos.</p> <p>Se desconectará una luminaria de una mota provocando así el evento de fallo.</p>
Salida	<ul style="list-style-type: none"> a. En el PC de pruebas en la terminal 1 se muestra los datos de los mensajes recibidos mediante el puerto de depuración, y en la terminal 2 los datos de los mensajes recibidos por la red inalámbrica. b. En el navegador web se muestra la información del panel de control. c. Mensaje de fallo en las interfaces mencionadas en el punto a. d. Correo electrónico enviado a la cuenta de mantenimiento.
Errores	<ul style="list-style-type: none"> a. El nodo central no recibe ningún mensaje de fallo. b. El nodo central no manda email de aviso por fallo en la luminaria. c. La mota no envía el mensaje de fallo. d. El mensaje de fallo no se almacena en el registro del sistema.

Evidencia	<pre>MESH: Received msg={"node": "2293", "type": "S", "lux": "25", "I": "-13.48", "pir": true, "onOff": "ON", "bright": "74", "fail": "SI"}! MESH: Received msg={"node": "2293", "type": "S", "lux": "25", "I": "-13.48", "pir": true, "onOff": "ON", "bright": "74", "fail": "SI"}!</pre> <p style="text-align: center;"><i>Fig. 7. Evidencia de prueba: Fallo de luminaria</i></p>
Resultado	PASA

1.1.8 Prueba: Actualización de firmware

	R3.* , R7, R8, R12.
Prueba	Se subirán a la red de motas inalámbricas mediante el fichero con la el firmware correcta, y otra con firmware inválido.
Salida	<ul style="list-style-type: none"> a. Se comprueba que el archivo de firmware es válido. b. En el registro del sistema estarán los mensajes del resultado de la actualización.
Errores	<ul style="list-style-type: none"> a. Los nodos no se actualizan correctamente con sus correspondientes firmwares. b. Un nodo se actualiza con el firmware inválido. c. En el registro del sistema no quedan registrados los eventos de actualización.
Evidencia	<pre>Connection list: 539276749 Time: Wed Sep 8 19:02:07 2021 Broadcast: {"node": "2753", "type": "S", "lux": "47", "I": "0.34", "pir": false, "onOff": "ON", "bright": "24", "fail": "ST"} Received from 539276749 msg:{"plugin": "ota", "type": "version", "md5": "afcdcd9d9a84ac38f78ef215e7e7cae66", "hardware": "ESP8266", "nodeType": "Mota", "noPart": 389, "forced": true} Changed connections Num nodes: 0 Connection list: Time: Wed Sep 8 19:02:17 2021 Broadcast: {"node": "2753", "type": "S", "lux": "47", "I": "-0.38", "pir": false, "onOff": "ON", "bright": "24", "fail": "SI"} Firmware version: Luces ESP32 root 080921.bin *wm:[2] Starting Config Portal *wm:[2] Disabling STA *wm:[2] Enabling AP</pre> <p style="text-align: center;"><i>Fig. 8. Evidencia de prueba: Actualización de firmware</i></p>
Resultado	PASA

1.2 Entorno 2. Validación del prototipo

Se cuenta con el sistema desplegado en su totalidad con los nodos en sus versiones de producción, ya instalados y alimentados. PC de pruebas con dos interfaces de red inalámbricas para poder acceder a la red MESH y a la red de *backhaul* y software para esnifar los mensajes que circulan por ambas redes.

1.2.1 Prueba. Establecimiento de la red MESH y lectura de medidas.

Requisito/s	R3.*, R6.* , R7, R9.* , R11.2, R13
Prueba	<p>Se iniciará la red MESH encendiendo todos los nodos.</p> <p>Se conectará el PC de pruebas a la red MESH para visualizar los mensajes que circulan por ella y al panel de control del nodo central.</p> <p>Se accede al panel de control del nodo central.</p> <p>Se comprobará si la red MESH está completada comprobando los mensajes de anuncio y asociación que circulan por la red MESH.</p> <p>Se observará en el panel de control si la red está completa y los valores de las medidas de los sensores de las distintas motas.</p>
Salida	<ul style="list-style-type: none"> a. Mediante el panel de control del nodo central se pueden visualizar las motas conectadas a la red y las medidas de sus sensores. b. Consultando los mensajes almacenados en el registro del sistema del nodo central. c. Mensajes de anuncio y asociación red circulando por la red MESH.
Errores	<ul style="list-style-type: none"> a. El nodo central no es capaz de iniciar la red MESH. b. El nodo central no visualiza correctamente las motas en el panel de control. c. Motas inalámbricas no son capaces de asociarse a la red de malla. d. El nodo central no recibe mensajes con las medidas de los sensores.
Evidencia	<pre> MESH: Num nodes: 3 MESH: Connection list: 2144992293 2785102753 1233185187 MESH: New Connection, nodeId = 1233185187 MESH: New Connection, { "nodeId": 3171322493, "root": true, "subs": [{ "nodeId": 2144992293 }, { "nodeId": 2785102753 }, { "nodeId": 1233185187 }] } MESH: Sent: {"type":"T","time":1630956974,"hOffset":2} FS: Event stored! </pre>

Información							
Nodo	Lum. Amb. (Lx)	Consumo(W/h)	Presencia	Estado	Brillo(%)	Fallo	
2753	181	1.48	false	ON	84	NO	

Fig. 9. Evidencia de prueba: Establecimiento de la red MESH y lectura de medias

1.2.2 Prueba: Configuración y control de luminaria.

Requisito/s	R3.* , R4.* , R5.* , R6.* , R8.* , R9.* , R10.* , R11.2
Prueba	<p>Se iniciará la red MESH encendiendo todos los nodos.</p> <p>Se conectará el PC de pruebas a la red MESH para visualizar los mensajes que circulan por ella y al panel de control del nodo central.</p> <p>Se accede al panel de control del nodo central y se enviarán todos los mensajes de control y configuración a las motas.</p>
Salida	<ol style="list-style-type: none"> En el PC de pruebas en la terminal se muestra todos los mensajes que circulan por red. En el explorador web del PC de pruebas se muestra el panel de control y el registro del sistema. Mensaje de la configuración y sus respuestas en el PC de pruebas y registro del sistema.
Errores	<ol style="list-style-type: none"> El nodo central mezcla mensajes de distintas motas. Motas envían mensajes de estado a todos los nodos (broadcast) y no solo al nodo central. El nodo central no es capaz de manejar todos los mensajes y pierde información.
Evidencia	

Resultado	PASA

1.2.3 Prueba: Fallo de luminaria.

Requisito/s	R3.* , R6.4, R7, R9.3, R11.1.
Prueba	<p>Se iniciará la red MESH encendiendo todos los nodos. Se conectará el PC de pruebas a la red MESH para visualizar los mensajes que circulan por ella y al panel de control del nodo central. Se desconectará una luminaria de una mota provocando así el evento de fallo.</p>
Salida	<ol style="list-style-type: none"> En el PC de pruebas en la terminal se muestra todos los mensajes que circulan por red. En el explorador web del PC de pruebas se muestra el panel de control y el registro del sistema. Mensaje de fallo en la terminal del PC de pruebas y registro del sistema. Correo electrónico enviado a la cuenta de mantenimiento.
Errores	<ol style="list-style-type: none"> El nodo central no recibe ningún mensaje de fallo. El nodo central no manda email de aviso por fallo en la luminaria. La mota no envía el mensaje de fallo. El mensaje de fallo no se almacena en el registro del sistema.

Evidencia	<table border="1"> <thead> <tr> <th>Nodo</th><th>Lum. Amb. (Lx)</th><th>Consumo(W/h)</th><th>Presencia</th><th>Estado</th><th>Brillo(%)</th><th>Fallo</th></tr> </thead> <tbody> <tr> <td>2753</td><td>54</td><td>-13.44</td><td>false</td><td>ON</td><td>75</td><td>SI</td></tr> </tbody> </table> <p>[Descargar] [Borrar Registro] [Recargar registro]</p> <pre> Sun Sep 12 14:09:00 2021: {"node": "2753", "type": "S", "lux": "57", "I": "-13.44", "pir": false, "onOff": "ON", "bright": "75", "fail": "SI"} Sun Sep 12 14:09:08 2021: {"node": "2753", "type": "S", "lux": "57", "I": "-13.44", "pir": false, "onOff": "ON", "bright": "75", "fail": "SI"} Sun Sep 12 14:09:14 2021: {"node": "2753", "type": "S", "lux": "57", "I": "-13.44", "pir": false, "onOff": "ON", "bright": "75", "fail": "SI"} Sun Sep 12 14:09:23 2021: {"node": "2753", "type": "S", "lux": "55", "I": "-13.44", "pir": false, "onOff": "ON", "bright": "75", "fail": "SI"} Sun Sep 12 14:09:31 2021: {"node": "2753", "type": "S", "lux": "55", "I": "-13.44", "pir": false, "onOff": "ON", "bright": "75", "fail": "SI"} Sun Sep 12 14:09:39 2021: {"node": "2753", "type": "S", "lux": "55", "I": "-13.44", "pir": false, "onOff": "ON", "bright": "75", "fail": "SI"} Sun Sep 12 14:09:49 2021: {"node": "2753", "type": "S", "lux": "56", "I": "-13.44", "pir": false, "onOff": "ON", "bright": "75", "fail": "SI"} Sun Sep 12 14:09:57 2021: {"node": "2753", "type": "S", "lux": "56", "I": "-13.44", "pir": false, "onOff": "ON", "bright": "75", "fail": "SI"} Sun Sep 12 14:10:02 2021: {"node": "2753", "type": "S", "lux": "56", "I": "-13.44", "pir": false, "onOff": "ON", "bright": "75", "fail": "SI"} Sun Sep 12 14:10:08 2021: {"node": "2753", "type": "S", "lux": "56", "I": "-13.44", "pir": false, "onOff": "ON", "bright": "75", "fail": "SI"} Sun Sep 12 14:10:21 2021: {"node": "2785102753", "type": "F"} ← Sun Sep 12 14:10:22 2021: {"node": "2753", "type": "S", "lux": "56", "I": "-13.44", "pir": false, "onOff": "ON", "bright": "75", "fail": "SI"} → </pre> <p>Fig. 11. Evidencia de prueba: Fallo de luminaria</p>	Nodo	Lum. Amb. (Lx)	Consumo(W/h)	Presencia	Estado	Brillo(%)	Fallo	2753	54	-13.44	false	ON	75	SI
Nodo	Lum. Amb. (Lx)	Consumo(W/h)	Presencia	Estado	Brillo(%)	Fallo									
2753	54	-13.44	false	ON	75	SI									
Resultado	PASA														

1.2.4 Prueba: Revisión del registro del sistema

Requisito/s	R3.1, R7
Prueba	<p>Acceder al panel de control y borrar el registro del sistema.</p> <p>Inicializar la red y todos sus nodos.</p> <p>Visualizar el registro del sistema y comprobar si hay mensajes de fallo y su origen.</p>
Salida	a. Visualización del registro del sistema por el panel de control y/o descarga del mismo.
Errores	a. Hay registrados mensajes de error no esperados en el registro del sistema.
Evidencia	<p>[Descargar] [Borrar Registro] [Recargar registro]</p> <pre> { "type": "C", "mode": "lumAmb", "mUn": "Off", "mLux": "443", "mOn": "21:37", "mOff": "07:53", "lum": "manual", "bVal": "74", "b1": "20:37", "b2": "22:37", "bt1Lux": "50", "bt2Lux": "100", "bt3Lux": "80", "bPre": "67" } Mon Sep 6 21:28:35 2021: {"type": "C", "mode": "lumAmb", "mOn": "Off", "mLux": "444", "mOn": "21:37", "mOff": "07:53", "lum": "manual", "bVal": "74", "bt1": "20:37", "bt2": "22:37", "bt1Lux": "50", "bt2Lux": "100", "bt3Lux": "80", "bPre": "67" } Mon Sep 6 21:28:36 2021: {"type": "C", "mode": "lumAmb", "mOn": "Off", "mLux": "445", "mOn": "21:37", "mOff": "07:53", "lum": "manual", "bVal": "74", "bt1": "20:37", "bt2": "22:37", "bt1Lux": "50", "bt2Lux": "100", "bt3Lux": "80", "bPre": "67" } Mon Sep 6 21:28:37 2021: {"node": "2293", "type": "S", "lux": "25", "I": "-13.48", "pir": true, "onOff": "ON", "bright": "74", "fail": "SI" } Mon Sep 6 21:28:38 2021: </pre>

<p align="center"><i>Fig. 12. Evidencia de prueba: Revisión del registro del sistema</i></p>	

Resultado	PASA
------------------	------

2 Plan de aceptación

Nombre del entregable	Pruebas asociadas para su validación	Responsable
Sistema de control de iluminación	Sci1, Sci2, Sci3 y Sci4	Director de la empresa
Firmware	Fir1	Director de la empresa
Manuales del producto.	Mnl1	Director de la empresa
Esquemáticos y planos PCB	Epp1	Director de la empresa

2.1 Pruebas

Establecimiento de la red y lectura de medidas	Sci1
Descripción: Se procederá a iniciar el sistema de iluminación y comprobar que se conectan todos los nodos y envían correctamente las medidas de sus sensores.	
Prerrequisitos Sistema de iluminación instalado en el cliente final, pero con la alimentación desconectada.	
Pasos: <ol style="list-style-type: none"> El usuario conecta la alimentación de todos los nodos instalados. El usuario accede al panel de control del nodo central. El usuario comprueba que en la red aparecen todos los nodos instalados. El usuario comprueba que todos los nodos envían medidas de sus sensores. 	
Resultado esperado: Todos los nodos de la red aparecen listados en el panel de control y las medidas de los sensores de los nodos son correctas.	
Resultado obtenido: Dirección Hotel Atlántico Resultado: PASA	Javier Vizcaya

Configuración y control de luminaria	Sci2
Descripción: Se va a comprobar todos los modos de funcionamiento y control de las luminarias	
Prerrequisitos Sistema de iluminación instalado y funcionando en el cliente final.	
Pasos: <ol style="list-style-type: none"> El usuario accede al panel de control del nodo central. El usuario enciende las luminarias. El usuario ajusta el nivel de intensidad de las luminarias. El usuario activa la opción de detección de presencia, comprueba su funcionamiento y la desactiva. El usuario apaga la luminaria. El usuario programa la hora del encendido y apagado de las luminarias. 	

7. El usuario programa los niveles de iluminación por franjas horarias.
8. El usuario comprueba que las luminarias se encienden en el tiempo programado en el paso 6 y los niveles coinciden con los configurados en el paso 7.
9. El usuario programa el modo de encendido/ apagado por nivel de luminosidad.
10. El usuario comprueba que las luminarias se encienden y apagan con ese umbral de luminosidad.

Resultado esperado:

El sistema funciona correctamente en todos los modos según las especificaciones descritas en el manual de usuario.

Resultado obtenido:

Dirección Hotel Atlántico

Resultado: PASA

Javier Vizcaya

Generar fallo de luminaria	Sci3
Descripción:	
Se va a provocar un fallo en una luminaria y probar los mensajes de aviso en el panel de control y por correo electrónico.	
Prerrequisitos	
Sistema de iluminación instalado y funcionando en el cliente final.	
Pasos:	
<ol style="list-style-type: none"> 1. El usuario accede al panel de control en el nodo central. 2. El usuario configura la opción de la cuenta de correo de mantenimiento. 3. El usuario provoca un fallo desmontando una bombilla de una luminaria. 4. El usuario comprueba el mensaje de aviso en el panel de control y la recepción de un correo electrónico informando del fallo en la cuenta de mantenimiento. 	
Resultado esperado:	
El sistema de iluminación muestra mensajes de fallo de luminaria en el panel de control y envía un correo electrónico a la cuenta de mantenimiento.	
Resultado obtenido:	
<p>Dirección Hotel Atlántico</p> <p>Resultado: PASA</p> <p><i>Javier Vizcaya</i></p>	

Revisión del registro del sistema	Sci4
Descripción:	
Se va a revisar el registro del sistema y comprobar que su contenido coincide con los eventos generados en las pruebas.	
Prerrequisitos	
Sistema de iluminación instalado y funcionando en el cliente final.	
Pasos:	
<ol style="list-style-type: none"> 1. El usuario accede al panel de control en el nodo central. 2. El usuario accede a la opción de visualizar el registro del sistema. 3. El usuario comprueba que los eventos registrados coinciden con las operaciones 	

realizadas.

4. El usuario usando las opciones de filtrado comprueba que no hay mensajes de error no esperados.

Resultado esperado:

El registro del sistema muestra registrado correctamente todos los eventos de configuración de las operaciones realizadas sobre el sistema y los eventos de encendido y apagado. El registro del sistema muestra el mensaje de fallo provocado en la prueba Sci3. No deben aparecer otros mensajes de error no esperados.

Resultado obtenido:

Dirección Hotel Atlántico

Resultado: PASA

Javier Vizcaya

Actualización de firmware y verificación	Fir1
Descripción: Se va a probar que los firmwares entregados son correctos realizando una actualización de firmware de todos los nodos del sistema.	
Prerrequisitos Sistema de iluminación instalado y funcionando en el cliente final.	
Pasos: <ol style="list-style-type: none"> 1. El usuario accede al panel de control del nodo central. 2. El usuario revisa las versiones de firmware instaladas en los nodos y sus hashes. 3. El usuario realiza una actualización de firmware con los firmwares entregados. 4. El usuario comprueba que coinciden las versiones de los firmwares y sus hashes. 	
Resultado esperado: Las versiones de firmware entregadas y sus hashes coinciden con las utilizadas en las pruebas.	
Resultado obtenido:	
Dirección Hotel Atlántico	
Resultado: PASA	

1. El usuario accede al panel de control del nodo central.
2. El usuario revisa las versiones de firmware instaladas en los nodos y sus hashes.
3. El usuario realiza una actualización de firmware con los firmwares entregados.
4. El usuario comprueba que coinciden las versiones de los firmwares y sus hashes.

Javier Vizcaya

Manuales del producto	Mnl1
Descripción: Se va a comprobar que el manual de usuario y el de instalación son correctos.	
Prerrequisitos Visor de PDF en el ordenador del cliente.	
Pasos: <ol style="list-style-type: none"> 1. El usuario revisa que el manual de usuario es correcto. 2. El usuario revisa que el manual de instalación es correcto. 	
Resultado esperado: El usuario está conforme con las descripciones facilitadas en el manual de usuario y en el manual de instalación.	

1. El usuario revisa que el manual de usuario es correcto.
2. El usuario revisa que el manual de instalación es correcto.

Resultado obtenido:

Dirección Hotel Atlántico

Resultado: PASA

Javier Vizcaya

Esquematicos y planos PCB	Epp1
----------------------------------	-------------

Descripción:

Se va a comprobar que el manual de usuario y el de instalación son correctos.

Prerrequisitos

Herramienta de CAD electrónico.

Pasos:

1. El usuario revisa que los esquemáticos.
2. El usuario revisa que los planos PCB.

Resultado esperado:

El usuario comprueba que los esquemáticos y planos PCB coinciden con el prototipo usado en las pruebas y que son aptos para su producción en masa.

Resultado obtenido:

Dirección Hotel Atlántico

Resultado: PASA

Javier Vizcaya

Conclusiones y cierre

RESUMEN DE DOCUMENTO

Documentación relacionada con el cierre del proyecto, incluyendo un estudio de las conclusiones obtenidas y manual de usuario de instalación si procede.

TÍTULO DOCUMENTO	Conclusiones y cierre.
FECHA DE ENTREGA	30/09/2021
AUTOR/ES	Jesús Muñoz Martínez

COMENTARIOS	VERSIÓN	FECHA
Versión Inicial	1.0	06/08/2021

Índice

1.	CONCLUSIONES GENERALES	4
2.	MEJORAS Y LÍNEAS FUTURAS	5

Definiciones, acrónimos y abreviaturas

SoC	System on Chip (Microcontrolador y periféricos en un solo chip)
WPA2	Un cifrado de seguridad para comunicaciones inalámbricas.

1. Conclusiones generales

La meta principal de este proyecto era conseguir un sistema inteligente de control de luminarias, el cual se ha conseguido implementar satisfactoriamente con todos los requisitos planteados inicialmente, así como con nuevos requisitos que se han ido añadiendo durante el desarrollo al ver la viabilidad técnica de su implementación y las mejoras que aportaban al diseño.

El factor más importante para hacer que el sistema funcione es la implementación de una **red mallada inalámbrica** con tecnología Wifi, basada en los SoC de bajo coste del fabricante Espressif. En las pruebas iniciales se estudió el diseño de esta red con la tecnología ZigBee y los SoC CC2531 de Texas Instruments, pero, aunque viable, la menor potencia del SoC, cobertura y un precio 3 veces mayor del SoC, inclinaron la balanza hacia los SoC de Espressif con tecnología WiFi.

El **control de iluminación** implementado sobre estos integrados proporciona unos resultados confiables, ya que el hardware utilizado cumple suficientemente con la potencia exigida y el software implementado para el mantenimiento de la red mallada basado en **multitarea cooperativa**, permite que haya CPU disponible para que la red no se caiga. Además, en caso de que se desconectase el nodo central, las motas siguen funcionando independientemente con la programación establecida, por lo que pueden funcionar de forma autónoma hasta que la red se restablezca e incluso un corte de energía, ya que almacena su programación en memoria flash.

El desarrollo de un **panel de control** en el nodo central permite que el sistema se pueda configurar desde cualquier navegador web, por lo que hace el sistema independiente de la plataforma que utilice el cliente, ya sea Linux, Windows, IOS, Android, etc.

La implementación de las **comunicaciones** sobre el estándar **WiFi** con cifrado WPA2 proporciona la seguridad necesaria, así como un alcance óptimo entre luminarias.

La aplicación de los conocimientos adquiridos en el Máster de Sistemas para Entornos Inteligentes, destacando las asignaturas de Redes de Sensores, MicroKernels, Diseño de placas de Circuito Impreso, ha sido una de las bases fundamentales que ha permitido la implementación con éxito de este Trabajo Fin de Máster. Multitud de pruebas y verificaciones realizadas durante el desarrollo del sistema han servido para ir depurando su funcionamiento.

En el desarrollo de este proyecto se ha utilizado hardware y software libre y accesible, por su coste y disponibilidad. Por tanto, el resultado está también a libre disposición para la comunidad de desarrolladores, estudiantes y cualquier entusiasta de la ingeniería bajo la licencia GPLv3.

Finalmente hay que destacar la satisfacción personal de ver el resultado obtenido al aplicar los conocimientos adquiridos en este Máster y cosechar el fruto dedicado al mismo.

2. Mejoras y líneas futuras

El sistema inalámbrico de control de luminarias ha quedado operativo y funcional cumpliendo los requisitos iniciales, pero durante el desarrollo del proyecto se ha comprobado el gran potencial de este sistema. Aunque a lo largo del mismo se han ido implementado nuevas mejoras, debido a las grandes posibilidades comentadas anteriormente, se proponen las siguientes mejoras y posibles líneas de desarrollo futuras.

- Ampliar las funcionalidades del panel de control, por ejemplo:
 - Posibilidad de crear subconjuntos de luminarias.
 - Modo de consola para depuración.
 - Implementación del acceso por contraseña.
 - ...

Para las motas y el nodo central proponemos las siguientes mejoras orientadas a la integración de este sistema en la Smart City.

- En el nodo central implementar la capacidad de subir los datos recopilados a un servidor central.
- En las motas, implementar sensores para medir la calidad del aire, condiciones meteorológicas, contador de vehículos, etc., ya que las luminarias se suelen instalar en calles y parques.
- El SoC del nodo central dispone de interfaz bluetooth, por lo que se podría también utilizar para la configuración inicial o la comunicación con nodos o periféricos que usen esta tecnología.
- La ofuscación de las comunicaciones, (en este trabajo académico no se ha realizado por su aspecto pedagógico), sería recomendable en el caso de utilizar el sistema en un entorno real, aunque actualmente hay muchas soluciones comerciales que siguen sin hacerlo. Lo anterior no implica dejar de usar el cifrado WPA2 como se hace actualmente.

Muchas ideas son aplicables a este tipo de sistemas basados en redes de sensores, sobre todo alentadas por las nuevas tecnologías que se están implantando orientadas al Internet de las Cosas, como son las redes 5G.

En un futuro cada vez más cercano, veremos más sistemas como el desarrollado en este trabajo, implantados en entornos o ciudades inteligentes, que permitan la optimización de los recursos energéticos y añadan confort y bienestar a la sociedad.

Anexo I. Planos de diseño

RESUMEN DE DOCUMENTO

Anexo que incluye la identificación planos de diseño de la placa de circuito impreso.

TÍTULO DOCUMENTO	Anexo I. Planos de diseño.
FECHA DE ENTREGA	30/09/2021
AUTOR/ES	Jesús Muñoz Martínez

COMENTARIOS	VERSIÓN	FECHA
Versión Inicial	1.0	30/08/2021

Índice

1	ESQUEMÁTICO PCB.	4
2	CAPAS DE DISEÑO DE LA PCB	6

Índice de figuras.

Fig. 1. Visión general de los componentes en la PCB	6
Fig. 2. Modelado tridimensional de la PCB	6
Fig. 3. LUCES-F_Cu.gbr	7
Fig. 4. LUCES-B_Cu.gbr	7
Fig. 5. LUCES-F_Paste.gbr:	7
Fig. 6. LUCES-F_Mask.gbr:	7
Fig. 7. LUCES-B_Mask.gbr:	7
Fig. 8. LUCES-F_SilkS.gbr:	7
Fig. 9. LUCES-B_SilkS.gbr:	8
Fig. 10. F.Fab:	8
Fig. 11. LUCES-Edge_Cuts.gbr:	8
Fig. 12. LUCES-PTH.drl. Fichero con los taladros a realizar en la PCB.	8

Definiciones, acrónimos y abreviaturas

PCB Placa de circuito impreso

1 Esquemático PCB.

Se muestra en la página siguiente a tamaño completo.

A

A

B

B

C

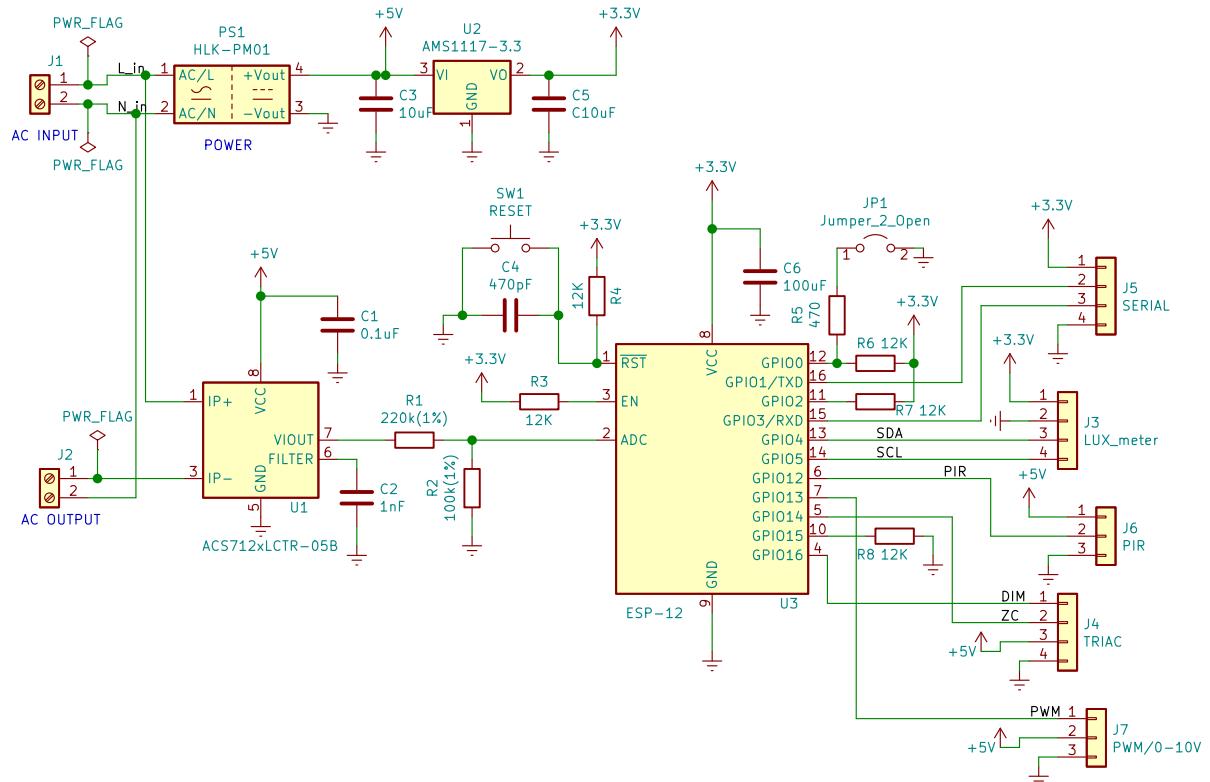
C

D

D

E

E



Jesús Muñoz Martínez

Sheet: /
File: LUCES.sch**Title: TFM Luces: Mota**Size: A4 | Date: 2021-08-01
KiCad E.D.A. kicad (5.1.10)-1Rev: 1.0
Id: 1/1

2 Capas de diseño de la PCB

En esta sección se van a identificar las capas de diseño de la PCB definidas en los archivos Gerber y Excellon adjuntos a este proyecto. Primero se muestra una visión general de los componentes utilizados y colocados en la PCB, mostrando varias capaz en un mismo dibujo, y también, su representación en modelado tridimensional.

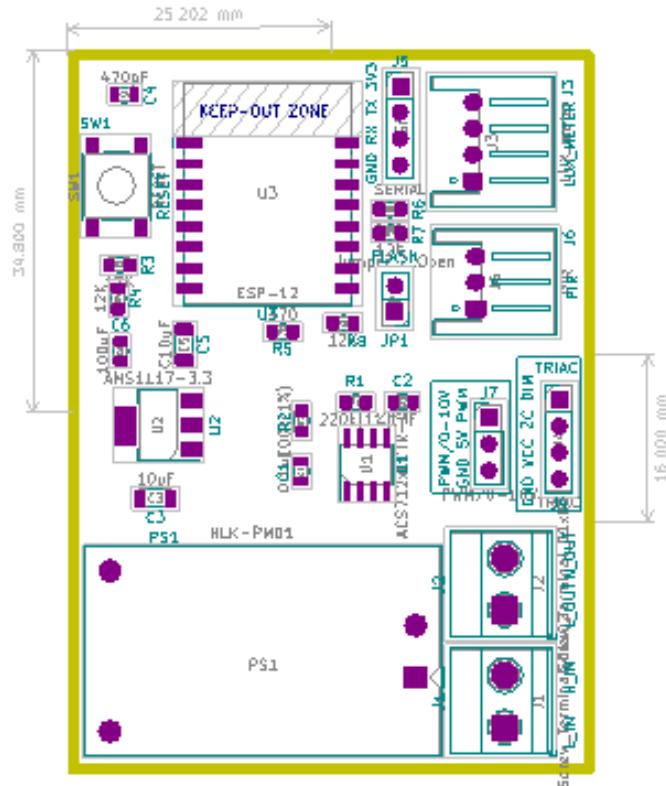


Fig. 1. Visión general de los componentes en la PCB

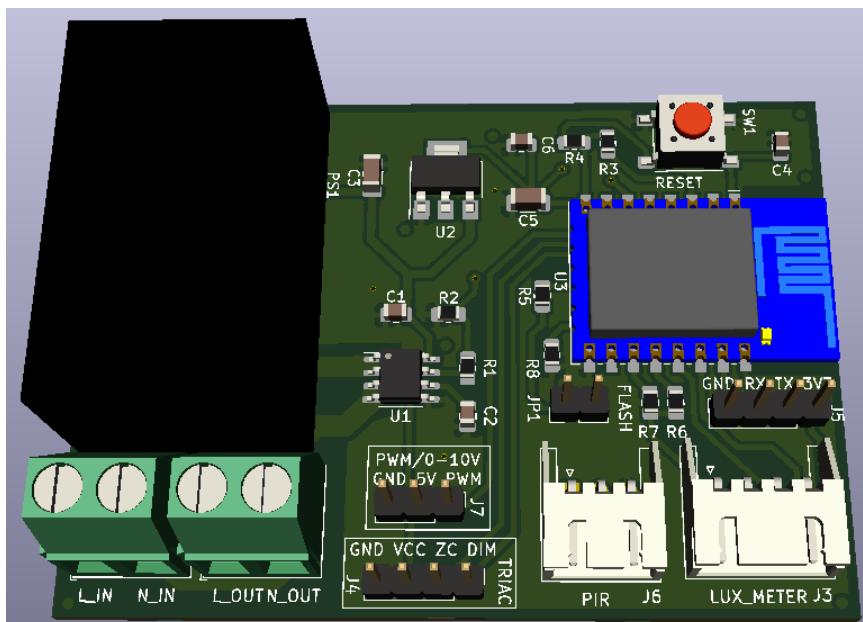


Fig. 2. Modelado tridimensional de la PCB

La identificación de los archivos de fabricación con las información de la capa que contiene la siguiente:

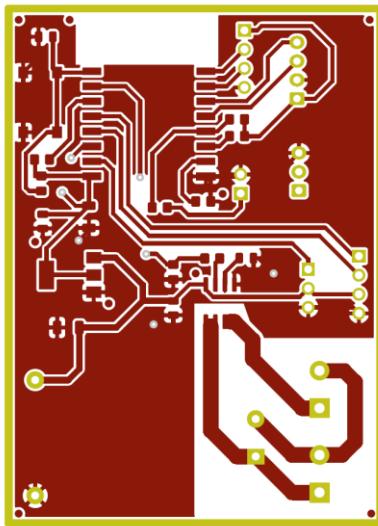


Fig. 3. LUCES-F_Cu.gbr.
Cobre de la capa superior

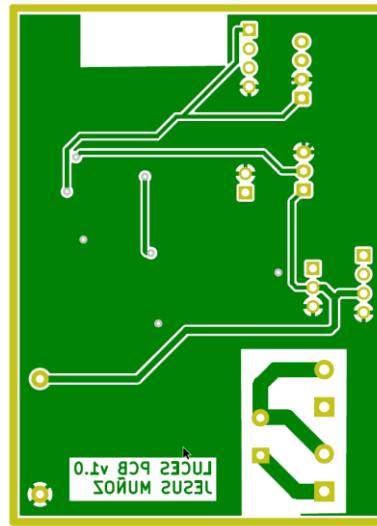


Fig. 4. LUCES-B_Cu.gbr
Cobre de la capa inferior

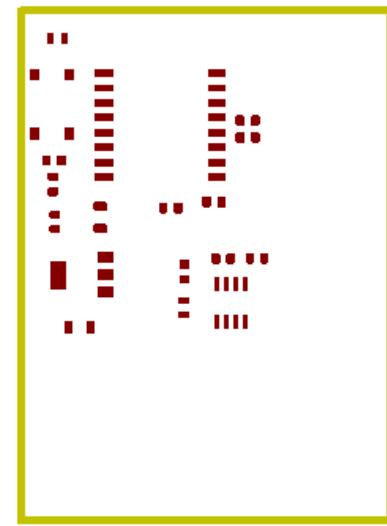


Fig. 5. LUCES-F_Paste.gbr:
Pasta de soldadura en la cara
superior

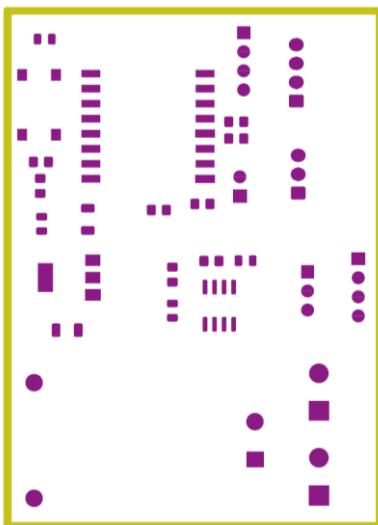


Fig. 6. LUCES-F_Mask.gbr:
Máscara de soldadura superior

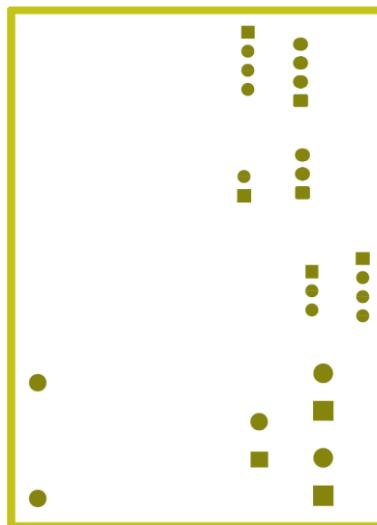


Fig. 7. LUCES-B_Mask.gbr:
Máscara de soldadura inferior

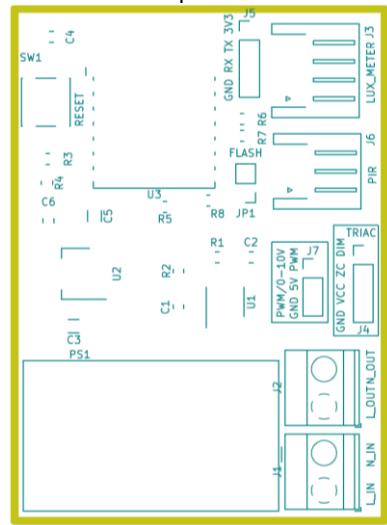


Fig. 8. LUCES-F_SilkS.gbr:
Serigrafía de la cara superior

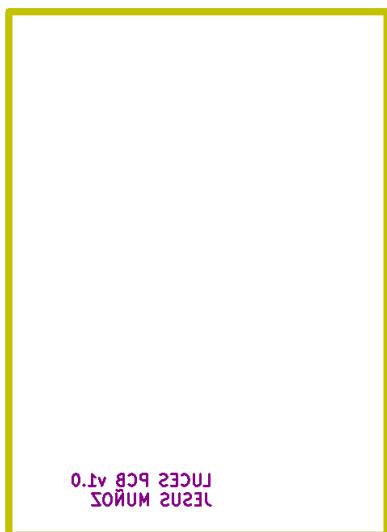


Fig. 9. LUCES-B_SilkS.gbr:
Serigrafía de la cara superior

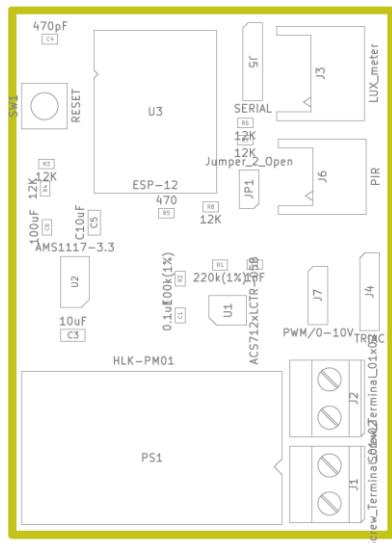
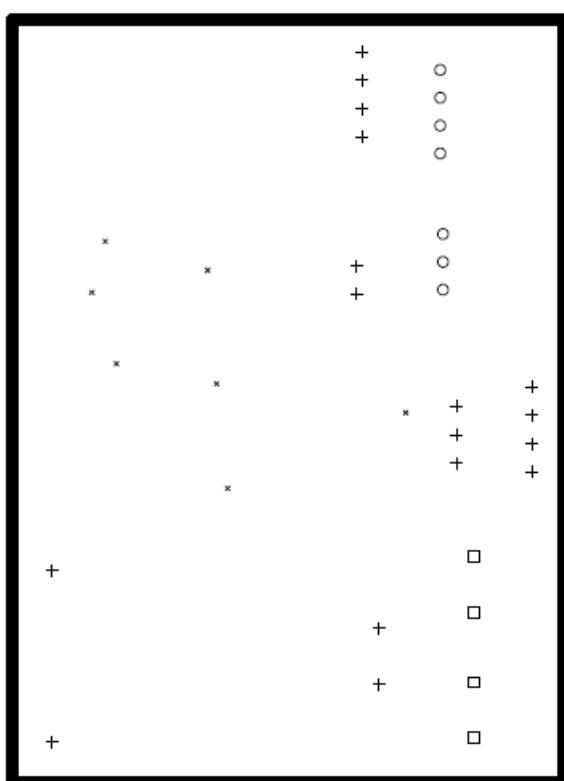


Fig. 10. F.Fab:
Capa de documentación para
fabricación



Fig. 11. LUCES-Edge_Cuts.gbr:
Capa de corte de borde de la PCB



Drill Map:

- * 0.400mm / 0.0157" (7 holes)
- o 0.950mm / 0.0374" (7 holes)
- + 1.000mm / 0.0394" (17 holes)
- 1.300mm / 0.0512" (4 holes)

Fig. 12. LUCES-PTH.drl. Fichero con los taladros a realizar en la PCB.

Anexo II. Código de programación

RESUMEN DE DOCUMENTO

Anexo con la descripción del código de programa desarrollado para la creación de los firmwares y el panel de control.

TÍTULO DOCUMENTO	Diseño e implementación del proyecto: LUCES
FECHA DE ENTREGA	30/09/2021
AUTOR/ES	Jesús Muñoz Martínez

COMENTARIOS	VERSIÓN	FECHA
Versión Inicial	1.0	30/08/2021

Índice

1. INTRODUCCIÓN	5
2. FIRMWARE DE LA MOTA INALÁMBRICA	6
2.1 PROGRAMA PRINCIPAL Y ARCHIVO DE CONFIGURACIÓN	6
2.2 GESTIÓN DE LA RED MALLADA	8
2.3 LECTURA DE LOS SENSORES Y ENVÍO DE MENSAJES	9
2.4 CONTROL DEL ENCENDIDO Y BRILLO DE LA LUMINARIA	10
2.5 ARCHIVO DE CONFIGURACIÓN Y SINCRONIZACIÓN DE LA HORA	10
3. FIRMWARE DEL NODO CENTRAL	12
3.1 PROGRAMA PRINCIPAL Y ARCHIVO DE CONFIGURACIÓN	12
3.2 GESTIÓN DE LA RED MALLADA	13
3.3 MODO DE CONFIGURACIÓN	14
3.4 SERVIDOR WEB Y CONTROL DE MENSAJES	14
3.5 ARCHIVOS DE CONFIGURACIÓN Y SINCRONIZACIÓN DE LA HORA	15
3.6 REGISTRO DEL SISTEMA	15
4. REFERENCIAS	17

Índice de figuras.

<i>Fig. 1. Arquitectura Lógica</i>	5
<i>Fig. 2. Listado de archivos del código fuente del firmware de la mota inalámbrica</i>	6
<i>Fig. 3. Archivo de configuración del firmware</i>	6
<i>Fig. 4. Variables para los subsistemas principales</i>	7
<i>Fig. 5. Tareas funcionando en el sistema</i>	7
<i>Fig. 6. Bucle Loop del programa principal</i>	7
<i>Fig. 7. Funciones de inicialización</i>	8
<i>Fig. 8. Funciones para la gestión de la red mesh</i>	8
<i>Fig. 9. Lectura de los sensores</i>	9
<i>Fig. 10. Lectura y cálculo de la corriente</i>	9
<i>Fig. 11. Función que comprueba el fallo de la luminaria</i>	9
<i>Fig. 12. Parte del código de la función PowerLum</i>	10
<i>Fig. 13. Función que carga el archivo de configuración</i>	11
<i>Fig. 14. Función de sincronización de la hora</i>	11
<i>Fig. 15. Listado de archivos del código fuente del firmware de la mota inalámbrica</i>	12
<i>Fig. 16. Algunas líneas del archivo de configuración del firmware</i>	12
<i>Fig. 17. Tareas funcionando en el sistema</i>	12
<i>Fig. 18. ISR para acceder al panel de control</i>	13
<i>Fig. 19. Bucle Loop del programa principal</i>	13
<i>Fig. 20. Sentencia If para inicializar el modo de configuración</i>	13
<i>Fig. 21. receivedCallback en el nodo central</i>	14
<i>Fig. 22. Modo de configuración</i>	14
<i>Fig. 23. Gestión de la recepción de mensajes</i>	15
<i>Fig. 24. Función de sincronización de la hora con las motas</i>	15
<i>Fig. 25. Sentencia para controlar que el registro del sistema sea circular</i>	16

Definiciones, acrónimos y abreviaturas

ADC	Conversor analógico digital
LED	Light Emitting Diode (Diodo emisor de luz).
MESH	Topología de red mallada.
NTP	Network Time Protocol..
PWM	Pulse Wide Modulation (Modulación por ancho de pulso).

1. Introducción

En este anexo se va a describir el código de programa utilizado para la programación de los firmwares de los componentes del sistema, mota inalámbrica, nodo central y panel de control (integrado en el nodo central). La arquitectura lógica descrita en la memoria, figura 1, describe las funcionalidades que se ha implementado en ambos firmwares.

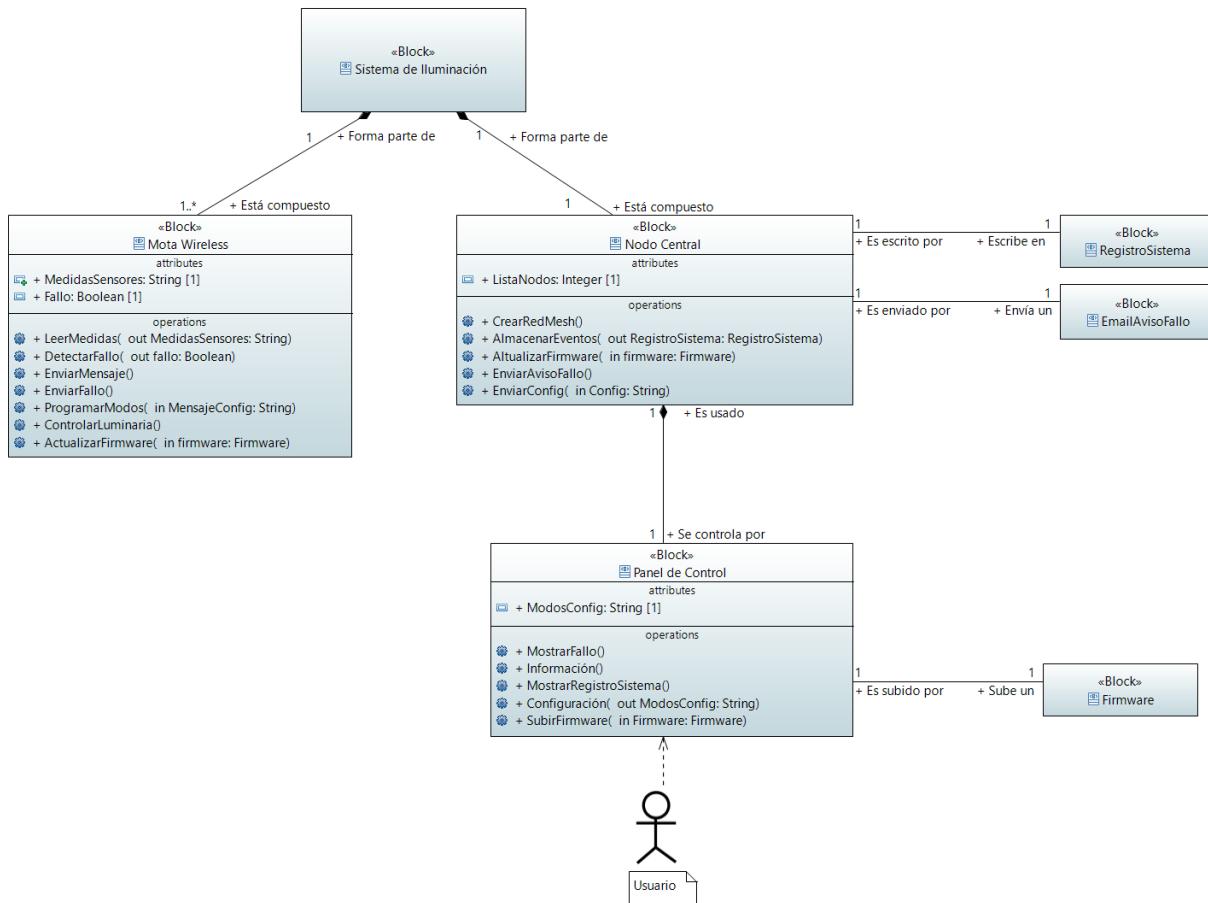


Fig. 1. Arquitectura Lógica

2. Firmware de la mota inalámbrica

El código fuente del firmware de la mota inalámbrica está programado en C/C++ y organizado según en los archivos que se pueden ver en la figura 2. Cada archivo agrupa funcionalidades asociadas al nombre de este.

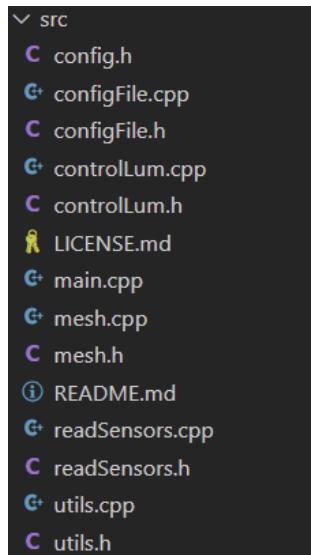


Fig. 2. Listado de archivos del código fuente del firmware de la mota inalámbrica

Los archivos README.md y LICENSE.md contienen información sobre el firmware y la licencia, GPLv2, a la que esta acogido el software y no forman parte del código de programación propiamente dicho.

2.1 Programa principal y archivo de configuración

El archivo **config.h** contiene la configuración de los parámetros del firmware como se puede ver en la figura 3:

```
const String verFw = "Luces_ESP8266_mota_080921.bin";

// MESH configuration
#define MESH_SSID      "LucesMesh"
#define MESH_PASSWORD  "████████"
#define MESH_PORT       5555

#define LED            2      // GPIO number of connected LED, ON ESP-12 IS GPIO2
#define BLINK_PERIOD   3000  // milliseconds until cycle repeat
#define BLINK_DURATION 100   // milliseconds LED is on for
#define PIRpin          12    // GPIO number of connected PIR SENSOR: GPIO4

#define PWM             13    // GPIO number of connected PWM, 1.-10v CONTROL
#define TRIAC           15    // GPIO for TRIAC control
#define ZC              D5    // GPIO for Zero cross interruption
```

Fig. 3. Archivo de configuración del firmware

El archivo **main.cpp** contiene el programa principal donde están definidas las variables globales, la configuración e inicialización del sistema y las tareas de la multitarea cooperativa.

Del as variables, figura 4, hay que destacar “*config*”, que almacena la configuración que se está ejecutando en ese momento, además esta variable se almacena en el archivo config.txt del sistema de ficheros, lo que permite que no se pierda su configuración en caso de fallo de alimentación. Las variables “*userScheduler*” y “*mesh*” son las asociadas a la multitarea y a la red mallada del sistema de iluminación.

```
JSONVar config;
scheduler userscheduler; // to control task
painlessMesh mesh;
```

Fig. 4. Variables para los subsistemas principales

Las tareas utilizadas son las mostradas en la figura 5.

```
Task taskSendStatus( TASK_SECOND * 1, TASK_FOREVER, &sendMessageStatus );
Task taskControlLum( TASK_SECOND * 1, TASK_FOREVER, &controlLum ); //start
Task taskBlinkNoNodes;
Task taskFadeLum( TASK_SECOND * 0.02, TASK_FOREVER, &fadeLum );
```

Fig. 5. Tareas funcionando en el sistema

- La tarea “taskSendStatus” se encarga de la lectura de los sensores del sistema, verificar si hay fallo en la luminaria y enviar el mensaje al nodo central del estado de la luminaria y medida de los sensores.
- La tarea “taskControlLum” se encarga de leer la configuración de “config” y aplicarla al módulo de potencia y así ajustar la luminaria a los valores deseados.
- La tarea “taskBlinkNodes” es una tarea que hace parpadear el Led de la mota con el número de nodos conectado a la red mallada. Su función es meramente informativa para el instalador.
- La tarea “taskFadeLum” se encarga de realizar los cambios de intensidad lumínica de forma suave y progresiva entre los valores con y sin presencia de usuarios de la vía. Esta tarea solo se ejecuta cuando es necesario realizar un cambio de intensidad.

Aunque no están definidas como tareas, ya que su ejecución es asíncrona, las funciones relacionadas con la red mallada son otra parte muy importante de la gestión de la mota inalámbrica, por eso en el loop del programa, figura 6, se ejecuta continuamente la función “updateMesh()”, encargada del control de los eventos que ocurren en la red Mesh.

```
// ----- Loop -----
void loop() {
    updateMesh();
    digitalWrite(LED, !onFlag);
}
```

Fig. 6. Bucle Loop del programa principal

Como comentamos, la inicialización de los subsistemas, figura 7, se realiza en el programa principal.

```
void setup() {
    pinMode(LED, OUTPUT);
    pinMode(PIRpin, INPUT);
    pinMode(A0, INPUT);
    pinMode(PWM, OUTPUT);
    pinMode(ZC, INPUT);
    pinMode(TRIAC, OUTPUT);

    Serial.begin(115200);
    printFwVersion();
    initFS();
    Wire.begin(); //I2C BUS

    initBH1750();
    loadConfig();
    initMesh();
```

```

userScheduler.addTask(taskBlinkNoNodes);
taskBlinkNoNodes.enable();

// Task Send Status
userScheduler.addTask(taskSendStatus);
taskSendStatus.enable();

// Task Control luminosity
userScheduler.addTask(taskControlLum);
taskControlLum.enable();

```

Fig. 7. Funciones de inicialización

2.2 Gestión de la red mallada

La gestión de la red mallada se realiza mediante la librería “*painlessMesh*”. Los archivos **mesh.h** y **mesh.cpp** implementan las funciones que llaman a los métodos de esta librería. A continuación, se describen estas funciones.

```

extern painlessMesh mesh;
extern Scheduler userScheduler;

String meshNodeId();
void sendMessageStatus();
void receivedCallback(uint32_t from, String & msg);
void newConnectionCallback(uint32_t nodeId);
void changedConnectionCallback();
void nodeTimeAdjustedCallback(int32_t offset);
void delayReceivedCallback(uint32_t from, int32_t delay);
void updateMesh();
void initMesh();

```

Fig. 8. Funciones para la gestión de la red mesh

- **meshNodeId**: devuelve el número de identificación del nodo de la red mallada. Está basado en la dirección MAC del dispositivo.
- **sendMessageStatus()**: llama a la función “*getReadings()*” para obtener las medidas de los sensores y el estado de la luminaria. Luego envía un mensaje con dicha información al nodo central.
- **receivedCallback**: esta función se ejecuta cuando la mota recibe un mensaje que circule por la malla. Lee el tipo de mensaje, si se trata de sincronización de la hora actual ejecuta “*syncTime()*” y si se trata de una nueva configuración de la mota ejecuta “*saveConfig()*” para guardar dicha configuración en la memoria flash y cargarla en el sistema.
- **newConnectionCallback**: Se ejecuta cuando la mota se conecta a la red MESH. Informa de este evento a la red y por el puerto serie mensaje.
- **changedConnectionCallback**: Se ejecuta al existir cambios en la malla, ya sea por un nuevo nodo o por que se haya caído alguno. Se actualiza la tarea de parpadeo del led con el número de nodos de la red.
- **nodeTimeAdjustedCallback**: Ejecuta tareas de sincronización temporal entre los nodos de la red mallada. Esta sincronización es diferente a la sincronización de la hora, ya que el tiempo es referido desde el inicio de la red mallada.
- **delayReceivedCallback**: Para la sincronización temporal entre los nodos, la función retrasa o adelanta la temporización interna para ajustarse a la de la red mallada.
- **updateMesh**: Tarea que ejecuta varias tareas de mantenimiento de la red mallada. Es necesaria para su funcionamiento.
- **initMesh**: Inicializa la conexión a la red mallada, para ello:
 - Activa el interfaz WiFi.

- Empieza a buscar otras redes wifi que formen parte de una malla
- Inicia sesión en el mejor nodo de red de malla que encuentra, si no encuentra nada, comienza una nueva búsqueda en 5 segundos.

2.3 Lectura de los sensores y envío de mensajes

En los archivos **readSensor.h** y **readSensor.cpp**, están escritas las funciones para las lecturas de los sensores, como por ejemplo la encargada de inicializar el sensor de iluminación, función “*initBH1750()*”. La función más interesante es *getReadings()* que es llamada por tarea de envío de mensaje de estado a través de la función “*sendMessageStatus()*”, por lo tanto, su ejecución es cíclica. Desde “*getReadings()*”, se llama al resto de funciones implicadas con los sensores, figura 9.

```
printLocalTime();
JsonReadings["node"] = node;
JsonReadings["type"] = "S"; //S: info from sensors
JsonReadings["lux"] = luxesStr;
JsonReadings["I"] = currentStr;
onPresence = digitalRead(PIRpin);
JsonReadings["pir"] = onPresence;
JsonReadings["onOff"] = onOff;
JsonReadings["bright"] = String(bright);
JsonReadings["fail"] = checkFail(current);
String readings = JSON.stringify(JsonReadings);
return readings;
```

Fig. 9. Lectura de los sensores

La función “*readCurrent()*”, figura 10, calcula el valor de la corriente leyendo el valor que hay en el conversor ADC conectado al sensor de medición de corriente.

```
float readCurrent() {
    float voltageSensor, current;
    int nMuestras = 10;
    for (int i=0; i<nMuestras;i++){
        voltageSensor= analogRead(A0)*(3.07 / 1024.0);
        current=current + (voltageSensor-2.50)/CurrentSensity;
    }
    current = current/nMuestras;
    return current;
```

Fig. 10. Lectura y cálculo de la corriente

Con “*checkFail()*”, si no la luminaria está encendida y no se detecta consumo de corriente se envía un mensaje de fallo, figura 11.

```
String checkFail(float I){
    String fail = "NO";

    if((onStatus)&&(I<0.1)){
        fail = "SI";
        failCount++;
    }else{
        fail = "NO";
    }
    if (failCount == 20) {
        JSONVar JsonFail;
        String node = meshNodeId();
        JsonFail["node"] = node;
        JsonFail["type"] = "F"; //F: fail message
        String msgFail = JSON.stringify(JsonFail);
        mesh.sendBroadcast(msgFail);
    }
    return fail;
}
```

Fig. 11. Función que comprueba el fallo de la luminaria

2.4 Control del encendido y brillo de la luminaria

Estas funcionalidades están implementadas en los ficheros **controlLum.h** y **controlLum.cpp**. En ellos están descritas funciones encargadas de conversión de datos del JSON que contiene la configuración, “*parseTime()*”, y comprobación de las temporizaciones programadas y la hora actual, “*timeOnRange()*” y “*checkTimeRange()*”.

La función “*controlLum()*”, es la que comprueba en qué modo está actualmente configurada la mota y finalmente con “*powerLum()*” y “*fadeLum()*”, se genera la señal de control para los módulos de potencia de una forma suave y progresiva, figura 12.

```

void fadeLum(){
    if (bright > brightNow){
        brightNow++;
        //int value = map(brightNow, 0, 100, 0, 255);
        analogWrite(PWM, brightNow);
    }else if (bright < brightNow){
        brightNow--;
        //int value = map(brightNow, 0, 100, 0, 255);
        analogWrite(PWM, brightNow);
    }
}

void powerLum(){
    int value;
    if (!onPresence){
        value = atoi(config["bPre"]);
        bright = value;
    }else{
        if((strcmp(config["lum"],"time") == 0)){
            value = bright;
        }else{
            value = atoi(config["bVal"]);
            bright = value;
        }
    }
    if (bright == brightNow){
        taskFadeLum.disable();
    }else{
        taskFadeLum.enable();
    }
}

```

Fig. 12. Código de la función *fadeLum* y *powerLum*.

2.5 Archivo de configuración y sincronización de la hora

En los archivos **configFile.h** y **configFile.cpp** están descritas las funciones para leer y escribir el fichero de configuración *config.txt*, que se almacena en el sistema de ficheros gestionado por la librería “*LittleFS*”. En la figura 13 se muestra la función de carga del fichero de configuración a modo de ejemplo.

```

void loadConfig(){

    File file = LittleFS.open("/config.txt", "r");
    String configStr;
    if (!file) {
        Serial.println("Failed to open file for reading");
    }
    while (file.available()) {
        configStr += char(file.read());
    }
    file.close();
    Serial.println("Load File Config:"+configStr);
    config = JSON.parse(configStr.c_str());
    if (strcmp(config["mOn"], "on") == 0) {
        onStatus = true;
    }else{
        onStatus = false;
    }
    bright = atoi(config["bVal"]);
}

```

Fig. 13. Función que carga el archivo de configuración

Por último, los ficheros **utils.h** y **utils.cpp** implementan funciones como las encargadas de la sincronización de la hora, “*printLocalTime()*” y “*syncTime()*”, figura 14. También contiene la función que imprime la versión del firmware “*printFwVersion()*”.

```

void syncTime(JSONVar jsonTime){

    JSONVar value = jsonTime["time"];
    String jsonString = JSON.stringify(value);
    time_t myTime = jsonString.toInt();
    setTime(myTime);
    value = jsonTime["hOffset"];
    jsonString = JSON.stringify(value);
    int h = jsonString.toInt();
    adjustTime(h*3600);
}

```

Fig. 14. Función de sincronización de la hora

3. Firmware del nodo central

Al igual que el código fuente del firmware de la mota inalámbrica, el del nodo central está programado en C/C++ y organizado según en los archivos que se pueden ver en la figura 15. Al existir partes de códigos iguales o muy similares al firmware de la mota inalámbricas, ese contenido será referenciado a la sección del apartado anterior.

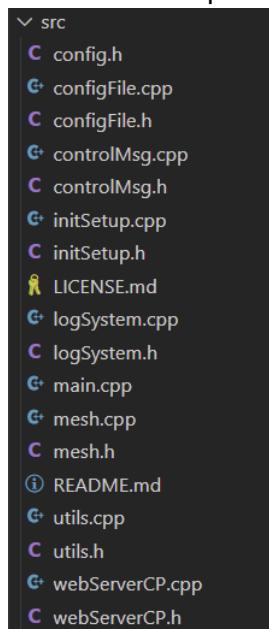


Fig. 15. Listado de archivos del código fuente del firmware de la mota inalámbrica

Los archivos README.md y LICENSE.md contienen información sobre el firmware y la licencia, GPLv2, a la que esta acogido el software y no forman parte del código de programación propiamente dicho.

3.1 Programa principal y archivo de configuración

El archivo **config.h** contiene la configuración de los parámetros del firmware, a diferencia de las motas aquí se añade la configuración de la red backhaul, servidor de correo, y servidor NTP, figura 16.

```

// NTP client
#define NTPSERVER      "pool.ntp.org"
#define GMTOFFSET_SEC  3600 //offset from GMT. 3600 from Spain
#define DAYLIGHTOFFSET 3600

// mail client
#define SMTP_HOST      "smtp.gmail.com"
#define SMTP_PORT      465
//#define AUTHOR_EMAIL  "lucestfm@gmail.com"
#define AUTHOR_PASSWORD "marypink"
  
```

Fig. 16. Algunas líneas del archivo de configuración del firmware

El archivo **main.cpp** es similar al del firmware de la mota inalámbrica. En este caso, las tareas son diferentes como se ven en la figura 17.

```

Task taskNTPSync( TASK_SECOND * 3, TASK_FOREVER, &nntpSync ); // utils.cpp
Task taskWsCleanup( TASK_SECOND * 1, TASK_FOREVER, &wsCleanupClients );
Task taskCheckBHTTP(TASK_SECOND * 1, TASK_FOREVER, &checkBHTTP);
Task taskGotoConfig(TASK_SECOND * 1, TASK_FOREVER, &gotoConfigMode);
  
```

Fig. 17. Tareas funcionando en el sistema

- La tarea “taskNTPSync” se encarga de la sincronización horaria con el servidorNTP

de internet. Se realiza en un intervalo de horas.

- La tarea “*taskWsCleanup*” se encarga de gestionar los websocket del servidor web que utiliza el panel de control. Cierra las conexiones de estos cuando ya no está conectado el cliente.
- La tarea “*taskCheckBHIP*” comprueba si hay cambios en la red de *backhaul* y comprueba la IP asignada ha cambiado.
- La tarea “*taskGotoConfig*” inicialmente no se ejecuta, se pone en marcha cuando se pulsa el botón “boot” en el nodo central, que activa la rutina de interrupción de la figura 18, o desde una llamada del panel de control.

```
void IRAM_ATTR isr() {
    serial.println("ISR");
    taskGotoConfig.enable();
}
```

Fig. 18. ISR para acceder al panel de control

Como en el caso de la mota, en el loop del programa, figura 19, se ejecuta continuamente la función “*updateMesh()*”, encargada del control de los eventos que ocurren en la red Mesh.

```
// ----- Loop -----
void loop() {
    if (!configMode){
        updateMesh();
    }
}
```

Fig. 19. Bucle Loop del programa principal

Hay que destacar que, en la inicialización, es donde se gestiona si el nodo central arranca en modo de configuración o modo normal de funcionamiento, figura 20.

```
// ----- SETUP -----
void setup() {
    Serial.begin(115200);
    printFWVersion();
    initFS();
    configMode = isConfigMode();
    if (configMode){
        configModeStart();
    }else{
        pinMode(0, INPUT_PULLUP);
        attachInterrupt(0, isr, FALLING);
        initMesh();
        initWebServer();
        initLog();

        //--- Task NTP synchronize --
        userScheduler.addTask( taskNTPSync );
    }
}
```

Fig. 20. Sentencia If para inicializar el modo de configuración

3.2 Gestión de la red mallada

La gestión de la red mallada es similar a como se realiza en la mota inalámbrica, ver 2.2, las diferencias son las siguientes:

- **initMesh:** Inicializa la conexión a la red mallada, para ello:
 - Activa el interfaz WiFi.
 - Crea una red mallada y busca la red *backhaul* para conectarse
- **sendMessage();** función que envía todos los tipos de mensajes a la red mallada. No existe la función “*sendMessageStatus()*.”
- **receivedCallback:** cuando recibe un mensaje los renvía al panel de control y los

registra en el registro del sistema. Si es un mensaje de fallo llama la función de enviar un correo de aviso en caso de fallo, “*sendEmail()*”, figura 21.

```
void receivedCallback(uint32_t from, String &msg) {
    Serial.printf("MESH: Received msg=%s\n", msg.c_str());
    JSONVar myObject = JSON.parse(msg.c_str());
    if (strcmp(myObject["type"], "F") == 0) { //fail message received
        int node = atoi(myObject["node"]);
        sendEmail(node);
    }
    notifyClients(msg);
    logging(msg);
}
```

Fig. 21. *receivedCallback* en el nodo central

3.3 Modo de configuración

En los archivos **initSetup.h** y **initSetup.cpp**, están escritas las funciones relacionadas con el modo de configuración, “*configModeStart()*”, inicia el modo de configuración creando un punto de acceso basado en la librería “*WiFiManager()*”, figura 22.

```
void configModeStart() {
    wifiManager.setAPCallback(configModeCallback);
    wifiManager.setSaveConfigCallback(saveConfigCallback);
    if(!wifiManager.startConfigPortal("LUCES_CONFIG")){
        Serial.println("No se pudo conectar");
    }
    Serial.println("Rebooting...");
    ESP.restart();
}
```

Fig. 22. Modo de configuración

En este modo el usuario busca una red WiFi a la que se conecta, La función “*saveConfigCallback()*”, guarda las SSID y las credenciales en el archivo *configMode*, que será usadas en el modo de normal de operación para conectarse a la red *backhaul*.

3.4 Servidor Web y control de mensajes

Estas funcionalidades están implementadas en los ficheros **webserverCP.h** y **webServerCP.cpp**. Con las funciones “*initWebServer()*” e “*initWebSocket()*” se inicializa el servidor web y las comunicaciones con el protocolo websocket. La función “*notifyClients()*” junto con “*onEvent()*” se encarga de actualizar los cambios en los navegadores web de los clientes conectados cuando haya alguna modificación de los controles de configuración, y la función “*wsCleanClients()*” se encargará de cerrar las conexiones websockets de los clientes que ya no estén conectados.

La función “*handleWebSocketMessage()*” gestiona la recepción de los mensajes JSON recibidos desde el servidor web, analiza el tipo de mensaje y ejecuta las funciones correspondientes asociadas al tipo, ver figura 23.

```

void handleWebSocketMessage(void *arg, uint8_t *data, size_t len) {
    AwsFrameInfo *info = (AwsFrameInfo*)arg;
    if (info->final && info->index == 0 && info->len == len && info->opcode == WS_TEXT) {
        data[len] = 0;
        message = (char*)data;
        Serial.println("Web: msg received!: " + message);
        JSONVar myMsg = JSON.parse(message.c_str());
        if (strcmp(myMsg["type"], "getConfig") == 0) {
            sendConfigValues();
        }
        if (strcmp(myMsg["type"], "pValues") == 0){
            notifyClients(message);
            saveConfig(myMsg);
            sendConfig2Nodes();
        }
        if (strcmp(myMsg["type"], "clearLog") == 0) {
            clearLog();
        }
        if (strcmp(myMsg["type"], "gotoConfig") == 0) {
            gotoConfigMode();
        }
    }
}

```

Fig. 23. Gestión de la recepción de mensajes

Los ficheros **controlMsg.h** y **controlMsg.cpp** contiene la función para enviar un correo electrónico de aviso a la cuenta de mantenimiento configurada. Para ello hace uso de la función “*sendEmail()*”. También contiene la función “*sendConfig2Nodes()*” para modificar el formato del JSON del servidor web y adaptarlo al de la red amallada, añadiendo y eliminando algunas claves del JSON

3.5 Archivos de configuración y sincronización de la hora

El funcionamiento del sistema de archivos es similar a como se realiza en la mota inalámbrica, implementados en los archivos **configFile.h** y **configFile().cpp**. En este caso, el sistema de archivos utilizado en el ESP32 se llama SPIFFS, pero con funcionamiento similar al visto en el apartado anterior. Hay que añadir que en este caso también se incluye funciones para el manejo del archivo *configMode*.

Los ficheros **utils.h** y **utils.cpp** implementan funciones encargadas de la sincronización de la hora con internet y las motas, “*ntpSync()*” y “*syncTimeNode()*” respectivamente, figura 24. También contiene la función que imprime la versión del firmware “*printFwVersion()*” y funciones auxiliares del modo de configuración como “*getSSID()*”, “*getPasswd()*” e “*isConfigMode()*”.

```

void syncTimeNode(){

    JSONVar jsonTime;
    time_t now = time(nullptr);
    jsonTime["type"] = "T"; // T: Time control messages
    jsonTime["time"] = now; //epoch time
    jsonTime["hoffset"] = 2; //for local time adjust
    String timeMsg = JSON.stringify(jsonTime);
    sendMessage(timeMsg);
}

```

Fig. 24. Función de sincronización de la hora con las motas

3.6 Registro del sistema

Por último, los ficheros **logSystem.h** y **logSystem.cpp** implementan funciones relacionadas con el registro del sistema apolladas en la librería “*LoggerSPIFFS*”.

- **initLog()**: Inicializa el archivo de registro del sistema con un tamaño especificado

- **logging()**: función que almacena un evento, que se le pasa como un string, en el registro del sistema de forma circular, figura 25.
- **clearLog()**: función que vacía el registro del sistema.

```
if(myLog.isFull()){
    myLog.flush();
    bool success = myLog.append(msg.c_str(), false);
    if(success){
        Serial.println("FS: Event stored!");
    }else {
        Serial.println("FS: Event NOT stored, file is full!");
    }
}else{
    bool success = myLog.append(msg.c_str(), false);
```

Fig. 25. Sentencia para controlar que el registro del sistema sea circular

4. Referencias

- [1] Espressif, «ESP32-DevKitC V4 Getting Started Guide,» [En línea]. Available: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/hw-reference/esp32/get-started-devkitc.html>. [Último acceso: Julio 2021].
- [2] J. P. Caram, «FlatCAM,» Enero 2021. [En línea]. Available: <http://flatcam.org/>.
- [3] Platformio.org, «PlatformIO IDE,» Julio 2021. [En línea]. Available: <https://platformio.org/platformio-ide>.
- [4] PainlessMesh, «PainlessMesh,» BlackEdder, [En línea]. Available: <https://gitlab.com/painlessMesh/painlessMesh/-/wikis/home>. [Último acceso: Julio 2021].
- [5] A. Arkhipenko, «TaskScheduler,» [En línea]. Available: <https://github.com/arkhipenko/TaskScheduler>. [Último acceso: Julio 2021].
- [6] R. Santos, «Random nerd tutorials,» Julio 2021. [En línea]. Available: <https://randomnerdtutorials.com/>.
- [7] Espressif, «ESP-WIFI-MESH,» Espressif, [En línea]. Available: <https://www.espressif.com/en/products/sdks/esp-wifi-mesh/overview>. [Último acceso: Agosto 2021].
- [8] F14V, «Thingiverse,» [En línea]. Available: <https://www.thingiverse.com/thing:2786523>. [Último acceso: septiembre 2021].

Anexo III. Manual de instalación

RESUMEN DE DOCUMENTO

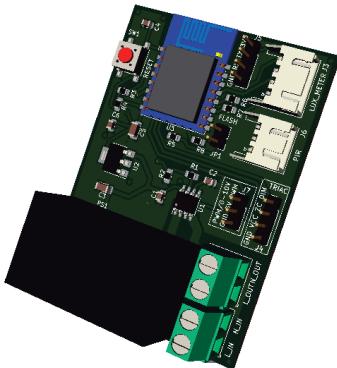
Este Anexo contiene el manual de instalación del sistema de iluminación LUCES.

TÍTULO DOCUMENTO	Anexo III. Manual de instalación.
FECHA DE ENTREGA	30/09/2021
AUTOR/ES	Jesús Muñoz Martínez

COMENTARIOS	VERSIÓN	FECHA
Versión Inicial	1.0	06/09/2021



Manual de instalación V1.0



Sistema inalámbrico de control de luminaria

Especificaciones	2
Características	2
Conexión de los sensores	3
Conexión de los módulos de potencia	5
Conexión de la alimentación AC.....	7
Puesta en marcha	7
Problemas frecuentes	9

1

Para una descripción del funcionamiento detallado del nodo central, consulte el manual del usuario.

El dispositivo es un controlador de luminarias inteligente mediante el protocolo Wi-Fi. Las motas inalámbricas integradas en las luminarias permiten un control de intensidad lumínica, detectar la luminosidad solar, consumo, presencia de personas, etc., de tal forma que permite apagar o encender las luminarias de manera remota, programar el encendido o apagado en función de la luminosidad, solar o temporizado.



Control remoto
desde cualquier
plataforma



Programación
temporizada



Control
compartido



Monitoreo de
Alimentación



Estado de
sincronización

Conexión de los sensores

Sensor de luminosidad.

El sensor de luminosidad se presenta con el siguiente aspecto, el color del soporte puede diferir.



Elija una buena ubicación del sensor de luminosidad donde su medida no se vea afectada por la luz que emite la luminaria. Busque una zona de sombra de la luz de la luminaria como puede ser la parte superior o detrás del poste. La siguiente imagen muestra posibles ubicaciones recomendables.

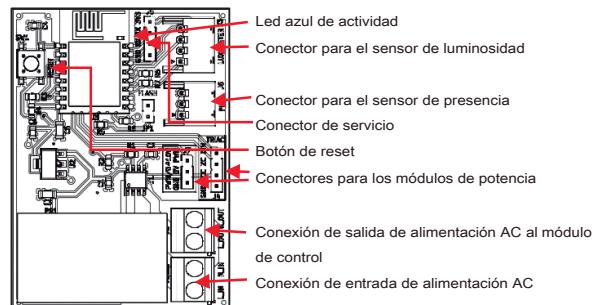


3

Especificaciones

Modelo	LUCES
Entrada	100-240V AC 50/60Hz
Salida	100-240V AC 50/60Hz 5A máxima
Carga máxima	1200W/5A
Wi-Fi	IEEE 802.11 b/g/n 2.4GHz
Sistemas operativos	Android/iOS/Windows/Linux/MacOS
Módulos de control de luminaria	3 (uno por cada mota)
máximo número de motas	>50
Temperatura de funcionamiento	-10°C~40°C
Dimensiones	70x50x34mm

Introducción al producto



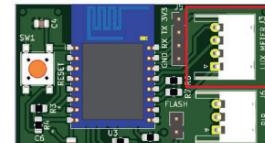
⚠️ ¡Atención! Tenga cuidado de asegurarse que ha desconectado los cables de alimentación AC antes de conectarlos.

Características

Este sistema de control de luminaria está basado en un control integrado en un Nodo Central, accesible al usuario, y en un conjunto de motas inalámbricas instaladas por un profesional cualificado. Es por ello, que este manual, orientado al instalador, se centra en las motas inalámbricas.

2

Una vez ubicado el sensor, conéctelo en el socket etiquetado como LUXMETER:



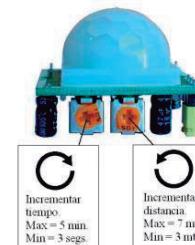
⚠️ El sensor de luminosidad no necesita ningún tipo de ajuste ni calibración.
Sensor de presencia de uso de la vía

El sensor de presencia de uso de la vía (PIR) se presenta con el siguiente aspecto, el color del soporte puede diferir.



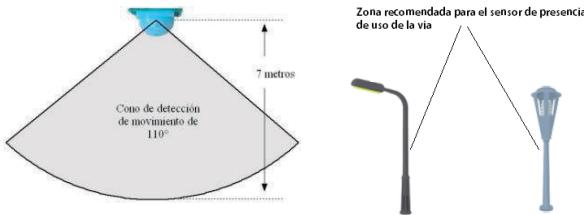
Antes de instalar él módulo puede ajustar los siguientes parámetros usando los potenciómetros que posee el módulo:

- Tiempo que permanece activo la señal de detección, desde 3 segundos hasta 5 minutos
- Distancia máxima de detección, desde 3 metros hasta 7 metros.

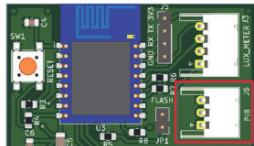


Elija una buena ubicación del sensor de presencia basándose que su rango de detección tiene un alcance de hasta 7 metros y ángulo de 110°. La siguiente imagen muestra posibles ubicaciones recomendables.

4



Una vez ubicado el sensor, conéctelo en el socket etiquetado como PIR:

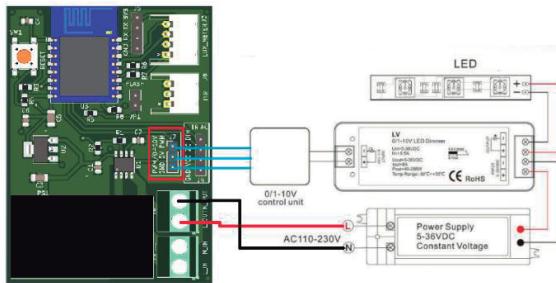


Conexión de los módulos de potencia

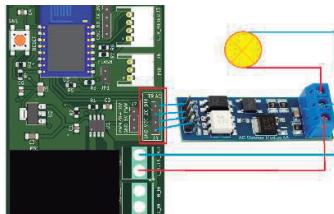
La mota inalámbrica dispone de los siguientes módulos de control/ potencia a los que se puede conectar según la luminaria a controlar:

1. Módulo de potencia para protocolo 0-10V.

Utilice este módulo para controlar luminarias con interfaz 0-10V. Conecte el módulo de potencia al conector J7 que se muestra en la siguiente imagen:

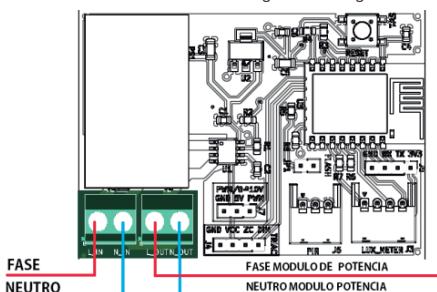


5



Conexión de la alimentación AC

Realice la conexión de la alimentación AC siguiendo el siguiente esquema:



Puesta en marcha

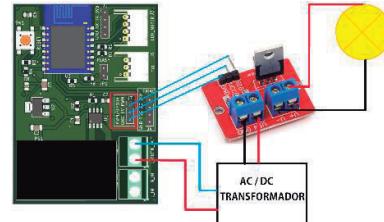
Revise que ha realizado todas las conexiones que a continuación se detallan:

1. Sensor de luminosidad ambiente y de presencia
2. Módulo de potencia, cableado de control y de alimentación
3. Alimentación AC de la mota inalámbrica.

Si todo está correctamente conectado puede alimentar la Mota inalámbrica, el led azul mostrará actividad y se conectará al nodo central.

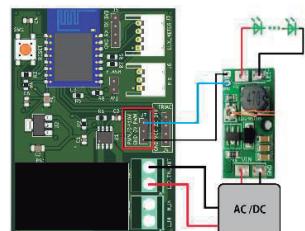
2. Módulo de potencia para tensiones DC.

Utilice este módulo para controlar luminarias basadas en regulación de intensidad por ajuste de la tensión DC. Conecte el módulo de potencia al conector J7 que se muestra en la siguiente imagen:



3. Módulo de potencia para corriente DC.

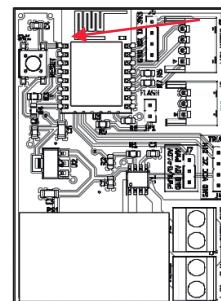
Utilice este módulo para controlar luminarias basadas en regulación de la intensidad por limitación de la corriente DC. Conecte el módulo de potencia al conector J7 que se muestra en la siguiente imagen:



4. Módulo de potencia para tensiones AC.

Utilice este módulo para controlar luminarias que funcionen con corriente AC (100-240V) y tengan capacidad de regulación de luminosidad como las bombillas incandescentes. Las bombillas LED deben ser regulables, en caso contrario apreciará un parpadeo en las mismas. Conecte el módulo de potencia al conector J4.

6



LED azul de actividad

Si necesitas configurar el nodo central, presiona sin soltar el botón de acceso al modo configuración, luego el dispositivo entra en el modo de configuración donde podrás configurarlo conectándose al punto de acceso creado.



Botón de acceso al Modo de configuración

Problemas frecuentes

P: ¿Por qué el dispositivo no se conecta a la red Wi-Fi MESH de mi instalación?

R: El dispositivo recién añadido necesita hasta 1 a 2 minutos para conectarse a la red Wi-Fi MESH. Si permanece sin conexión durante mucho tiempo, pulse el botón de reset. Si no ha solucionado el problema verifique los siguientes puntos

- ① Es posible que el Nodo central esté apagado. Verifique su funcionamiento
- ② Es posible que el dispositivo no tenga alimentación AC. Verifique si hay actividad en el led azul.
- ③ Revise las conexiones con el módulo de potencia instalado.

P: ¿Por qué el dispositivo no detecta la presencia de usuarios en la vía correctamente?

R: El dispositivo de detección de personas (PIR) debe calibrarse correctamente según la zona a controlar, revise la sección "conexión de los sensores" para más información. Si no ha solucionado el problema verifique los siguientes puntos

- ① Es posible que el dispositivo de detección PIR no tenga alimentación. Verifique si hay actividad en el del mismo.
- ② Revise las conexiones y el cableado con el módulo PIR.

P: ¿Por qué el dispositivo no detecta correctamente la luminosidad ambiente?

R: El dispositivo de detección de personas (PIR) debe calibrarse correctamente según la zona a controlar, revise la sección "conexión de los sensores" para más información. Si no ha solucionado el problema verifique los siguientes puntos

- ① Es posible que el dispositivo de detección PIR no tenga alimentación. Verifique si hay actividad en el del mismo.
- ② Revise las conexiones y el cableado con el módulo PIR.

FCC Warning

Changes or modifications not expressly approved by the party responsible for compliance could avoid the user's authority to operate the equipment.

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

FCC Radiation Exposure Statement:

This equipment complies with FCC radiation exposure limits set forth for an uncontrolled environment. This equipment should be installed and operated with minimum distance 20cm between the radiator & your body. This transmitter must not be co-located or operating in conjunction with any other antenna or transmitter.

Note:

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

For FCC:

Frequency range

Wi-Fi: 2412-2462MHz BT: 2402-2480MHz

Maximum RF output power of the product

Wi-Fi: 17.85dBm BT: -1.90dBm

For CE RED:

Frequency range

Wi-Fi: 2412-2472MHz BT: 2402-2480MHz

Maximum RF output power of the product

Wi-Fi: 18.36dBm BT: 3.93dBm (Inclusion antenna gain)

RF exposure

RF exposure information: The Maximum Permissible Exposure (MPE) level is calculated based on a distance of d=20 cm between the device and the human body. To maintain compliance with the RF exposure requirement, a separation distance of 20 cm between the device and the human should be maintained.



Anexo IV. Manual de usuario

RESUMEN DE DOCUMENTO

Este Anexo contiene el manual de usuario para la utilización y configuración del sistema de iluminación LUCES una vez instalado.

TÍTULO DOCUMENTO	Anexo IV. Manual de usuario
FECHA DE ENTREGA	30/09/2021
AUTOR/ES	Jesús Muñoz Martínez

COMENTARIOS	VERSIÓN	FECHA
Versión Inicial	1.0	06/09/2021



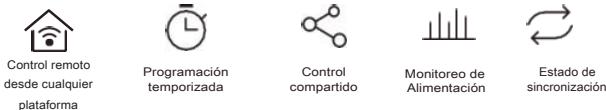
Manual de usuario V1.0



Sistema inalámbrico de control de luminaria

Especificaciones.....	2
Características	2
Primer encendido	3
Modo de configuración	3
Panel de control	4
Cambiar Red	7
Problemas frecuentes	8

1



Primer encendido

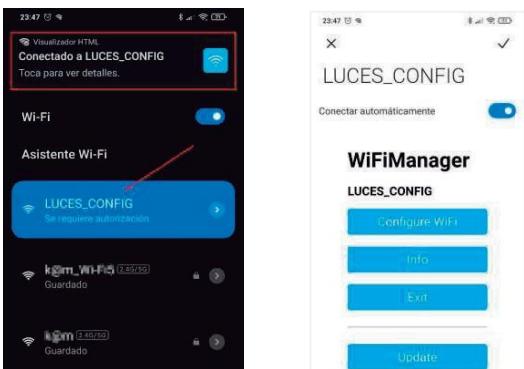
Después de encender el nodo central por primera vez, el dispositivo se pondrá en modo de configuración durante el primer uso.

ⓘ Si el dispositivo no entra en modo de configuración, podrá acceder a dicho modo mediante el botón de acceso en el panel de control o mediante la pulsación del botón Modo de configuración durante un segundo. A continuación, suelte el botón.

Modo de configuración

Después de encender el nodo central por primera vez, el dispositivo se pondrá en modo de configuración durante el primer uso.

Para acceder al dispositivo, conéctese a la red WiFi con el nombre LUCES_CONFIG.



Una vez conectado a la red LUCES_CONFIG, podrá acceder directamente al panel de configuración abriendo su navegador o pulsando en su teléfono en "Toca para ver detalles".

En caso de que esto no ocurra puede acceder manualmente en la dirección <http://192.168.4.1>.

3

Especificaciones

Modelo	LUCES
Entrada	100-240V AC 50/60Hz
Salida	100-240V AC 50/60Hz 5A máxima
Carga máxima	1200W/5A
Wi-Fi	IEEE 802.11 b/g/n 2.4GHz
Sistemas operativos	Android/iOS/Windows/Linux/MacOS
Módulos de control de luminaria	3 (uno por cada mota)
máximo número de motas	>50
Temperatura de funcionamiento	-10C~40C
Dimensiones	70x50x34mm

Introducción al producto



Nodo central

ⓘ Para el nodo central se recomienda usar el adaptador de corriente de 1A suministrado.

Características

Este sistema de control de luminaria está basado en un control integrado en un Nodo Central, accesible al usuario, y en un conjunto de motas inalámbricas instaladas por un profesional cualificado. Es por ello, que este manual, orientado al usuario, se centra en el uso del Nodo Central.

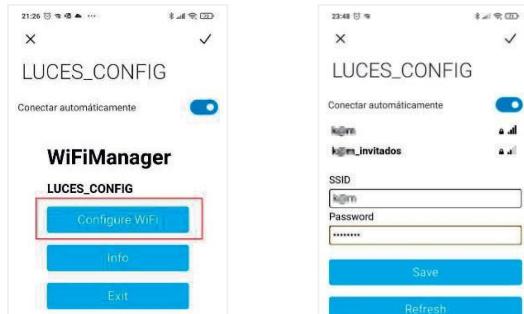
Para descripción del funcionamiento detallado de las motas inalámbricas, consulte el manual de instalación.

El dispositivo es un controlador de luminarias inteligente mediante el protocolo Wi-Fi. Las motas inalámbricas integradas en las luminarias permiten un control de intensidad lumínica, detectar la luminosidad solar, consumo, presencia de personas, etc., de tal forma que permite apagar o encender las luminarias de manera remota, programar el encendido o apagado en función de la luminosidad, solar o temporizado.

2

Añadir red local

Una vez dentro del panel de configuración, pulse sobre el botón "Configure WiFi" para iniciar el escaneo de redes inalámbricas . Tras unos instantes aparecerá el listado con las redes encontradas.



Seleccione la red, introduzca la contraseña de acceso y el Nodo Central se conectará a la red indicada con asignación dinámica de dirección IP. Realice una búsqueda en su red con el hostname "LUCES" o verifique en la tabla DHCP de su router la dirección IP asignada.

ⓘ Recomendamos configurar una IP fija en su router para mayor comodidad. Consulte a su administrador de red para más información.

Panel de control

Una vez conectado el Nodo Central a su red local con salida a internet, se sincronizará automáticamente la hora del sistema LUCES. Acceda al panel de control donde podrá realizar las siguientes funciones:

1. Información de la red de motas inalámbricas.

En el panel de control se muestran en tiempo real la información de las motas conectadas, su estado, las medidas de sus sensores y si se ha detectado fallo en la luminaria.

Información						
Nodo	Lum. Amb. (Lx)	Consumo(W/h)	Presencia	Estado	Brillo(%)	Fallo
2753	30	1.54	false	ON	71	NO
2293	25	-13.48	true	OFF	0	NO

4

2. Configuración y control

Aquí podrá configurar el sistema de iluminación entre sus tres modos de encendido y sus tres modos de ajuste de intensidad lumínica. El ajuste de intensidad lumínica por detección de presencia está activado por defecto en todas las configuraciones, para intentar maximizar el ahorro de energía.



Modo Manual:

Este modo permite encender y apagar las luminarias en cualquier momento. Permite ajustar la intensidad lumínica y también la intensidad cuando no se detecten personas en las zonas donde se han instalado las luminarias.



Modo de encendido por luminosidad ambiente:

Este modo permite encender y apagar las luminarias según el nivel lumínico configurado. Las luminarias se encenderán cuando el nivel de luminosidad medida sea inferior al valor ajustado. Las luminarias se apagaran cuando el nivel lumínico sea superior al ajustado.

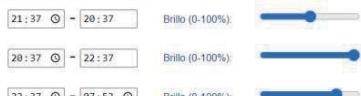


5

Luego, este modo dispone permite configurar manualmente la intensidad lumínica o mediante una selección por franja horaria, pensado para poner diferentes ajustes de luminosidad para el anochecer, noche y amanecer.

Ajuste por franja horaria

Selecciona el nivel de intensidad de las luminarias según la franja horaria:



Ajusta la intensidad de las luminarias en el caso de que **no se detecte uso de la vía**:

Brillo sin uso de la vía: 52 %

Modo de encendido temporizado:

En este modo el encendido y el apagado se realiza según las horas programadas. Los modos de ajuste de intensidad lumínica disponibles en este modo son todos los disponibles, así que podremos configurarlo tanto manualmente como por franja horaria y siempre con el modo de detección de uso de la vía funcionando.

Por franja horaria

Ajusta el valor de una franja horaria que será utilizada cada día para encender las luminarias.

Hora de encendido: 21:37
Hora de apagado: 07:53

3. Añadir cuenta de correo de mantenimiento

El sistema permite configurar una cuenta de correo donde se avisará en caso de fallo de luminaria.

Cuenta de correo de mantenimiento

Recibe alertas en caso de fallo y otros avisos en la cuenta indicada en el siguiente campo:

tester@gmail.com Guardar

4. Registro del sistema

En esta sección podrá visualizar los eventos y mensajes registrados en el sistema de control Luces. Podrá descargar el fichero así como borrarlo y/o recargar el visualizador para ver los últimos eventos registrados.

6

Problemas frecuentes

P: ¿Por qué el dispositivo no se conecta a la red Wi-Fi de mi instalación?

R: El dispositivo recién añadido necesita de 1 a 2 minutos para conectarse a Wi-Fi y a la red. Si permanece sin conexión durante mucho tiempo, evalúe la situación analizando los siguientes puntos.

- a) Es posible que haya introducido una contraseña de Wi-Fi incorrecta.
- a) Es posible que la distancia entre el interruptor y el router sea demasiado grande o que el entorno provoque interferencias. Piense en acercarlos entre sí. Si falla, vuelva a añadirlo.
- c) La red 5G Wi-Fi no es compatible y solo admite la red inalámbrica de 2,4 GHz.
- d) Es posible que el filtrado de direcciones MAC esté activado. Desactívelo.

Si no ha sido posible solucionar el problema mediante ninguno de los métodos anteriores, puede abrir la red de datos móviles del teléfono para crear un punto de acceso Wi-Fi y, a continuación, volver a añadir el dispositivo.

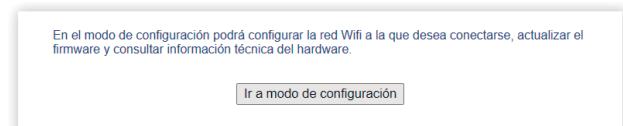
Registro del sistema

Descargar Borrar Registro Recargar registro

```
Sat Sep 11 12:17:07 2021:  
{"node": "2295", "lux": "25", "z": "-13.48", "pir": "true", "onOff": "OFF", "bright": "0", "fail": "NO"}  
Sat Sep 11 12:17:19 2021:  
{"node": "2793", "lux": "91", "z": "1.15", "pir": "false", "onOff": "ON", "bright": "96", "fail": "NO"}  
Sat Sep 11 12:18:38 2021: {"type": "T", "time": 163135518, "hoffst": 2}  
Sat Sep 11 12:18:41 2021:  
{"node": "2295", "lux": "25", "z": "-13.48", "pir": "true", "onOff": "OFF", "bright": "0", "fail": "NO"}  
Sat Sep 11 12:18:54 2021:
```

5. Acceder al modo de configuración

Puede acceder al modo de configuración pulsando el botón que se muestra en la imagen.



Cambiar Red

Si necesitas cambiar la red, presiona sin soltar el botón de acceso al modo configuración, luego el dispositivo entra en el modo de configuración donde podrás configurarlo a una nueva red de nuevo.



Botón de acceso al Modo de configuración

7

8

FCC Warning

Changes or modifications not expressly approved by the party responsible for compliance could avoid the user's authority to operate the equipment.

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

FCC Radiation Exposure Statement:

This equipment complies with FCC radiation exposure limits set forth for an uncontrolled environment. This equipment should be installed and operated with minimum distance 20cm between the radiator & your body. This transmitter must not be co-located or operating in conjunction with any other antenna or transmitter.

Note:

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

9

For FCC:

Frequency range

Wi-Fi: 2412-2462MHz BT: 2402-2480MHz

Maximum RF output power of the product

Wi-Fi: 17.85dBm BT: -1.90dBm

For CE RED:

Frequency range

Wi-Fi: 2412-2472MHz BT: 2402-2480MHz

Maximum RF output power of the product

Wi-Fi: 18.36dBm BT: 3.93dBm (Inclusion antenna gain)

RF exposure

RF exposure information: The Maximum Permissible Exposure (MPE) level is calculated based on a distance of d=20 cm between the device and the human body. To maintain compliance with the RF exposure requirement, a separation distance of 20 cm between the device and the human should be maintained.



10

