



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Aeroespacial
y Diseño Industrial

Desarrollo de un sistema embebido para pantalla LED
modular

Trabajo Fin de Grado

Grado en Ingeniería Electrónica Industrial y Automática

AUTOR/A: Sanz Peña, José David

Tutor/a: Rodríguez Ballester, Francisco

CURSO ACADÉMICO: 2022/2023



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



TRABAJO FIN DE GRADO

Grado en Ingeniería Electrónica
Industrial y Automática

DESARROLLO DE UN SISTEMA
EMBEBIDO PARA PANTALLA LED
MODULAR

Desarrollado por: D. David Sanz Peña

Dirigido por: D. Francisco Rodríguez Ballester

Curso 2022-2023

DESARROLLO DE UN SISTEMA EMBEBIDO PARA PANTALLA LED MODULAR

MEMORIA.....	3
PRESUPUESTO	63
PLIEGO DE CONDICIONES.....	72
DOCUMENTACIÓN.....	Adjunta



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Grado en Ingeniería Electrónica Industrial y Automática

DESARROLLO DE UN SISTEMA EMBEBIDO PARA PANTALLA LED MODULAR

DOCUMENTO: MEMORIA

Desarrollado por: D. David Sanz Peña

Dirigido por: D. Francisco Rodríguez Ballester

Curso 2022-2023

Contenido

RESUMEN	5
RESUM	6
ABSTRACT	7
OBJETO	8
ANTECEDENTES	9
Antecedentes de los leds	9
Antecedentes de la comunicación inalámbrica	11
Antecedentes de las PCBs	11
NECESIDADES	13
ALTERNATIVAS	15
SOLUCIÓN ADOPTADA	16
Solución al problema del consumo elevado y justificación de la <i>modularización</i> de la pantalla.....	16
Composición de los módulos	16
Selección de los leds.....	18
Requerimientos temporales de actualización de la imagen	19
Justificación solución modular	19
Funcionamiento de los módulos	20
Versatilidad de los módulos	21
Elección del método de comunicación	34
Elección del microcontrolador	34
Elección del método de control	35
Explicación funcionamiento de manejo de LED	35
Explicación protocolo de comunicación de datos.....	36
FW del dispositivo	42
COMPONENTES DEL PROYECTO.....	50
SELECCIÓN DE COMPONENTES Y ESQUEMAS ELECTRICOS.	50
OBJETIVO DESARROLLO SOSTENIBLE (ODS).....	61

RESUMEN

El trabajo consiste en el desarrollo de un sistema embebido modular encargado de representar mediante leds imágenes en movimiento recibidas inalámbricamente por una aplicación externa formando pantallas que puedan tener diferentes configuraciones.

Las principales limitaciones del proyecto son la variabilidad en la composición de pantallas, la transmisión de datos inalámbricamente, la representación de imagen en movimiento a una velocidad de actualización indetectable por el ojo, la comunicación de los microcontroladores con la app y el desarrollo del circuito embebido.

El uso al que va enfocado el proyecto es a la generación de pantallas de grandes dimensiones y formas variables compuestas por módulos para formar entre otros, muros led, pantallas gigantes para conciertos, paneles publicitarios led, etc.

RESUM

El treball consisteix en el desenvolupament d'un sistema embegut modular encarregat de representar mitjançant imatges leds en moviment rebudes sense fils per una aplicació externa formant pantalles que puguin tenir diferents configuracions.

Les limitacions principals del projecte són la variabilitat en la composició de pantalles, la transmissió de dades sense fils, la representació d'imatge en moviment a una velocitat d'actualització indetectable per l'ull, la comunicació dels microcontroladors amb l'app i el desenvolupament del circuit embegut .

L'ús a què va enfocat el projecte és a la generació de pantalles de grans dimensions i formes variables compostes per mòduls per formar entre d'altres, murs led, pantalles gegants per a concerts, panells publicitaris led, etc.

ABSTRACT

The work consists of developing a modular embedded system responsible for displaying images in motion using LEDs, which are wirelessly received from an external application, forming screens with different configurations. The main limitations of the project include the variability in screen compositions, wireless data transmission, displaying moving images at a refresh rate undetectable by the human eye, microcontroller communication with the app, and the development of the embedded circuit.

The project is aimed at generating large-sized and variably shaped screens composed of modules to create, among others, LED walls, giant screens for concerts, LED advertising panels, etc.

OBJETO

Se llevará a cabo el desarrollo de un panel de leds inteligente a bajo coste que puede ser utilizado en distintos entornos, desde la creación de ambientes en un pub, a pantallas gigantes (compuestas por distintos módulos), en un concierto de gran envergadura; a pantallas-pared para casas de lujo o simplemente para pantallas publicitarias, como en estadios de futbol. Estas pantallas estarán conformadas de módulos, haciendo versátil la configuración de estas. Además, las mismas emitirán imágenes en movimiento recibidas mediante comunicación inalámbrica.

ANTECEDENTES

Antecedentes de los leds

Desde la prehistoria los humanos hemos utilizado señales luminosas para transmitir información, para el uso cotidiano, o para entretenernos. En la prehistoria y los tiempos antiguos utilizábamos el fuego para todo, como iluminación cotidiana, para transmitir señales o para hacer ritos antiguos con los sabios de las tribus.

Con la llegada de la modernidad y el descubrimiento de la corriente eléctrica, se abrió paso todo un mundo de nuevas tecnologías e innovación. Uno de los momentos más importantes de esta revolución tecnológica fue el descubrimiento de la bombilla por parte de Thomas Edison el 21 de octubre de 1879. A partir de este momento fuimos capaces de crear luz al hacer pasar una corriente eléctrica por una resistencia que se volvía incandescente.



Imagen 1: Imagen de una bombilla obtenida de Google imágenes.

Desde este momento los humanos hemos utilizado nuestra capacidad e ingenio, para crear iluminaciones de todo tipo incluyendo desde la iluminación cotidiana que utilizamos día a día hasta espectáculos luminosos avanzados.

Con el paso del tiempo los humanos hemos buscado reducir el consumo energético en nuestras necesidades lumínicas, de este modo en 1927 la compañía estadounidense General Electric patentó la bombilla de bajo consumo, que no se comercializó de todos modos hasta 1938 cuando ya estuvo perfeccionada.

Tras la segunda guerra mundial y una vez las economías mundiales se hubieron recuperado, volvió a haber interés en la mejora del estado del arte en cuanto a iluminación se refiere, y fue la misma compañía estadounidense, GE, General Electric, la que, a través del ingeniero Nick Holonyak en 1962, inventó la tecnología led con el

uso de diodos y ya anticipo que en un futuro las lámparas incandescentes de Edison serían sustituidas por lámparas led.



Imagen 2: tira de leds obtenida de Google imágenes

La predicción se ha cumplido, y actualmente la mayoría de la iluminación moderna cuenta con tecnología led. El problema es que un led por sí solo no es muy luminiscente, y necesita colocarse en grupos, llamados paneles, para cumplir con finalidades tan diversas como son un semáforo, indicar una salida de emergencia, los faros de un coche, o las comentadas en el objeto de esta presentación, como son el uso en grandes paneles con una infraestructura compleja para formar pantallas gigantes para distintos fines.



Imagen 3: pantalla gigante empleada en un concierto de música.

En los siguientes apartados veremos que la agrupación de los leds tiene diversos problemas que explicaremos y resolveremos más adelante.

Antecedentes de la comunicación inalámbrica

Además de la tecnología led, en el desarrollo de este proyecto se emplearán las tecnologías de comunicación inalámbrica. Estas tecnologías emplean las ondas electromagnéticas para transmitir información de unos dispositivos a otros y tienen diversas utilidades en la actualidad; desde la tecnología wifi, hasta la radio, pasando por el bluetooth y llegando incluso a alcanzar el GPS que también emplea ondas electromagnéticas para transmitir los datos desde los satélites hasta los dispositivos.

La historia de esta tecnología comienza en 1888 cuando el físico alemán Rudolf Hertz, cuyo apellido da nombre a la unidad de medida de la frecuencia en reconocimiento, Hertz; consiguió realizar la primera transmisión de información con ondas electromagnéticas empleando un oscilador. Sin embargo, no fue hasta 1896, cuando el ruso, Alexander Stepanovich Popov logró llevar a cabo una comunicación a 250m de distancia gracias al desarrollo de la antena, que el mismo llevo a cabo. En el año 1902 se llevó a cabo la primera comunicación trasatlántica, entre Canadá y Reino Unido y durante la primera guerra mundial se llevaron a cabo numerosos desarrollos en el ámbito de la guerra que empleaban la tecnología recientemente descubierta.

Desde entonces, la comunicación inalámbrica ha evolucionado significativamente, con la invención de tecnologías como la radio y la televisión, que permitieron la transmisión de información a larga distancia. En los años 80 y 90 surgieron tecnologías de comunicación inalámbrica como el teléfono móvil, que permitían la comunicación interpersonal sin necesidad de cables.

En la actualidad, existen diversas tecnologías de comunicación inalámbrica que se utilizan en diferentes aplicaciones, como el wifi para la conexión a internet en hogares y empresas, el bluetooth para la comunicación entre dispositivos cercanos, el GPS para la navegación y localización, y la tecnología celular para la comunicación móvil en todo el mundo.

En el futuro, se espera que esta tecnología siga evolucionando con la implementación de tecnologías como la 5G y el *Internet of Things*, conjunto de tecnologías en el cual se puede incluir este proyecto.

Antecedentes de las PCBs

Como más tarde explicaremos, en el proyecto se emplean dos PCBs. Los antecedentes de esta tecnología comienzan en 1903 cuando el alemán Albert Hanson patentó un dispositivo que combinaba un elemento conductor con un aislante dispuesto en capas y que se uniría mediante perforaciones dando origen a la idea de las PCBs, con el paso del tiempo otros científicos como Paul Eisler perfeccionaron el concepto, sin embargo, no fue hasta la segunda guerra mundial cuando se adoptó esta tecnología por parte

del gobierno de los estados unidos. Hacia mediados de la década de 1950 se extendió su uso a uno comercial con la producción en masa de las PCBs, lo que permitió la creación de dispositivos electrónicos más pequeños y eficientes.

En la década de 1960, la aparición de la tecnología de montaje superficial (SMT) permitió una mayor densidad de componentes en las PCBs, lo que permitió el desarrollo de dispositivos aún más pequeños y complejos. Desde entonces, las PCBs se han convertido en una parte esencial de la mayoría de los dispositivos electrónicos, desde teléfonos móviles y ordenadores hasta equipos médicos y sistemas de comunicaciones. En la actualidad, las PCBs se fabrican utilizando una variedad de materiales, incluyendo fibra de vidrio, resinas epoxi, poliimida y materiales compuestos.

Las PCBs han sido fundamentales para el desarrollo y miniaturización de dispositivos electrónicos en la historia reciente, y se espera que sigan siendo una tecnología vital en el futuro de la electrónica y la tecnología en general.

NECESIDADES

A continuación, se van a desarrollar las necesidades que son necesarias cumplir para lograr el dispositivo que se desea implementar.

Para poder tener un gran panel con muchos leds y formar imágenes en movimiento con complejidad nos encontramos con diversos problemas que a continuación explicaremos.

En primer lugar, es comúnmente conocido que un led por sí solo consume una cantidad insignificante de potencia, del orden de 40 mW, sin embargo, esta aseveración no es correcta al cien por cien, porque el principal problema de un led es que necesitas que funcione con corriente continua, ya que, por definición, cualquier diodo, incluyendo los leds, solo funciona en un sentido de la corriente. Esto implica que, aunque un led puede consumir únicamente 20 mA alimentado a una tensión de 5V, en el momento en el que usas 100 leds necesitas una alimentación de corriente continua que suministre 2 A de corriente, y si necesitas una pantalla gigante de 100 000 leds, necesitas que el suministro de la corriente sea de 2000 A, lo cual es una cifra de dimensiones verdaderamente elevadas para fuentes de corriente continua.

A este problema debemos añadir que para poder encender un led de un color determinado necesitamos que este led sea RGB, es decir, que cada led cuente con 3 leds, uno rojo, otro verde y otro azul, respectivamente, y con la combinación de intensidad de estos 3 colores obtener un total de 16 millones de tonos distintos de luz, 16 777 216 colores en concreto. Esta versatilidad es ideal para el proyecto, sin embargo, implica triplicar el número de leds necesarios para cualquier finalidad triplicando del mismo modo el consumo de energía requerido. En el ejemplo anterior de una pantalla gigante que necesita 100 000 leds, necesitaríamos una fuente que proporcione 6000 A, algo fuera del alcance de cualquier dispositivo.

Por otro lado, tenemos el problema de que a cada led se le debe indicar de qué color debe iluminar, y, por lo tanto, con qué intensidad debe iluminar cada led RGB.

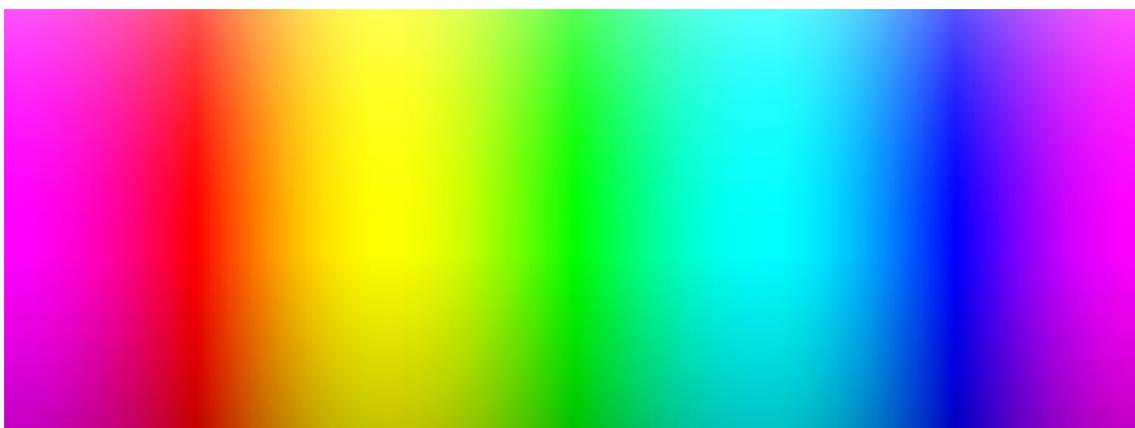


Imagen 4: banda de colores. Obtenida de Google Imágenes.

Por último, debemos encontrar la manera de poder transmitir la imagen que deseamos mostrar al dispositivo de un modo inalámbrico, ya sea a través de wifi,

bluetooth, LoRa, u otros tipos de comunicación inalámbrica, y de este modo conseguir no solamente imágenes fijas sino también imágenes en movimiento que puedan mostrar, por ejemplo, un anuncio en las vallas publicitarias del estadio Santiago Bernabéu.

En este sentido nos enfrentamos fundamentalmente a 4 problemas: como aportar suficiente corriente a todos los leds; como indicar el grado de luz al que debe iluminar cada *intraled* que compone el led RGB; como transmitir la información del color que cada led debe mostrar a suficiente velocidad como para representar imágenes en movimiento, y, por último, como distinguir los distintos módulos de tal modo que cada uno muestre la imagen que debe mostrar para que entre todos conformen una imagen completa.

ALTERNATIVAS

Como alternativa al uso de leds para pantallas gigantes existen básicamente dos soluciones que cumplen solo en cierta medida su cometido.

En primer lugar, el uso de imágenes fijas en superficies de metal propias de los paneles o vallas publicitarias. Aunque es una posible solución, se pierde la capacidad de variar el contenido al instante de un modo sencillo además de perder el factor de que la pantalla sea luminosa y no una simple imagen fija. Por ejemplo, en los paneles luminosos de Times Square, de la ciudad de Nueva York, o en Gran Vía, en Madrid, tendrían que estar continuamente cambiando las imágenes de publicidad en lugar de simplemente cambiar la imagen transmitida por los leds, perdiendo además el factor de iluminación que aportan las luces leds. Por supuesto para los usos de las pantallas gigantes led en conciertos o muros con imágenes variables como en la calle Gran Vía de Madrid esta solución no cubriría las necesidades.



Imagen 5: foto de la calle Gran Vía de la ciudad de Madrid. Obtenida de Google Imágenes.

En segundo lugar, se pueden utilizar bombillas de corriente alterna ya sean incandescentes o de bajo consumo de distintos colores y mostrar los paneles o pantallas de un modo semejante a un panel led, pero con un consumo desmesurado que hace completamente inviable este sistema para todo aquello que no sea un pequeño conjunto de pocas luces. Además, esta solución dificulta en gran medida la versatilidad de color que aportan los leds.

SOLUCIÓN ADOPTADA

A continuación, se explican las diferentes soluciones que se han aportado al producto definitivo diseñado:

Solución al problema del consumo elevado y justificación de la *modularización* de la pantalla

Para solucionar el problema del sobreconsumo generado por los leds para una fuente de corriente continua la solución es dividir nuestras pantallas de gran tamaño en módulos con unos pocos leds que requieran de consumos asumibles por una fuente de alimentación estándar del mercado y formar pantallas gigantes a partir de la combinación de diversos módulos independientes.

Con esta solución obtenemos asociada una ventaja añadida, que es la versatilidad que nos permite, ya que con el mismo producto se podrán formar desde pantallas pequeñas para decorar una habitación, a paneles de grandes dimensiones que requieran verse desde la distancia, incluyendo también otros usos como los paneles luminosos de farmacias (Imagen 6).



Imagen 6: módulo de farmacia. Obtenido de Google Imágenes.

Composición de los módulos

Estos módulos, cada uno de los cuales contará con una fuente de alimentación propia, estarán constituidos a su vez por 9 submódulos que consistirán cada uno en un cuadrado de 8x8 leds y unas dimensiones de 10x10 cm como se puede apreciar en la Imagen 7. Cada submódulo estará compuesto por una PCB de control y una PCB de leds que será reemplazable de un modo sencillo de modo que la estructura de control de la imagen sea independiente de la placa que posee los leds, facilitando a su vez el mantenimiento. Además, el conjunto PCBs de leds, PCBs de control y fuente de alimentación deberá estar asociado a un housing, el cual variará dependiendo de la finalidad del dispositivo. En el caso de las pantallas gigantes el housing tendrá la forma

estándar de un cuadrado de 30x30 cm conformando un módulo completo. En el caso de otras utilidades como para pequeños paneles luminosos o luminarias de farmacias la forma podrá variar a la específica formando la característica cruz de las farmacias o distintas formas como “L” o “-“, dependiendo de las necesidades del producto utilizando la cantidad de submódulos que sea necesaria.

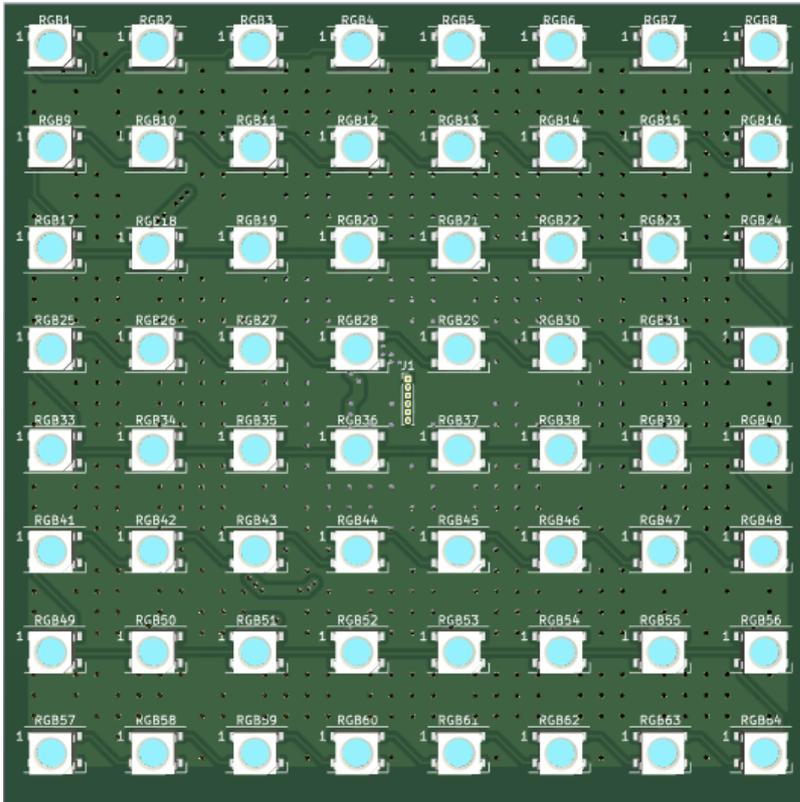


Imagen 7: cara de componentes de PCB submódulo_leds. (Creación propia)

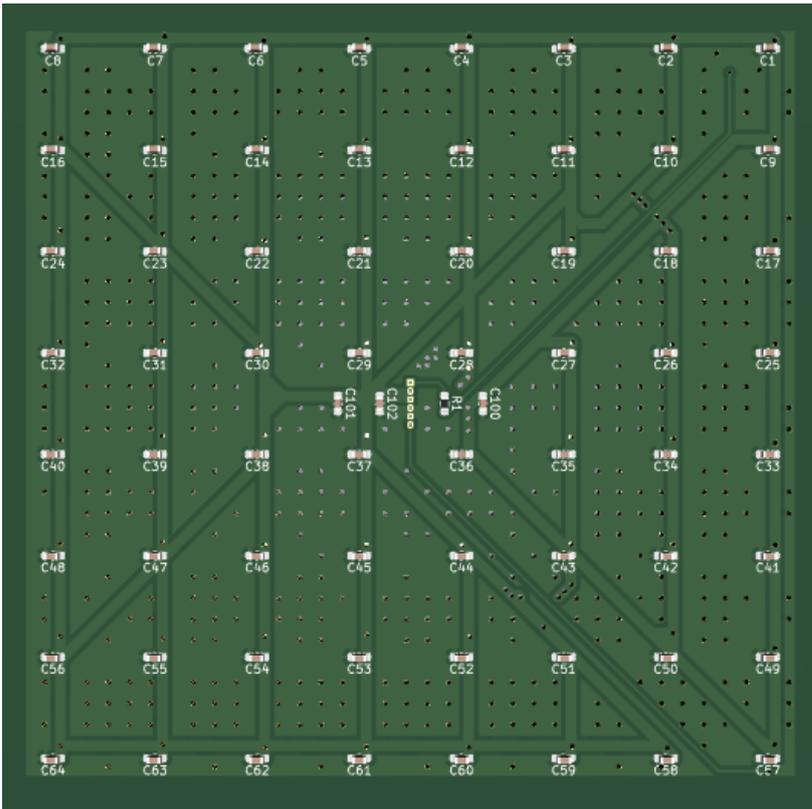


Imagen 8: cara de soldadura de PCB submódulo_leds. (Creación propia)

De tal modo, cada módulo, teniendo en cuenta que se trata un módulo completo con 9 submódulos, tendrá unas dimensiones finales de 30x30cm y 576 leds RGB controlados mediante un mismo microcontrolador y alimentados con una fuente de alimentación de 100W a 5V como más tarde explicaremos.

Selección de los leds

Con respecto al segundo desafío, el problema de informar a cada led en cada momento de cómo se debe iluminar nuestra estrategia es el uso de leds RGB inteligentes (pixel leds), los WS2812B en concreto. Estos leds están compuestos como ya hemos mencionado por otros 3 leds internos, uno rojo, uno verde y uno azul, para con la combinación de intensidades de todos formar un espectro de hasta 16 millones de tonalidades (16 777 216). Sin embargo, la característica principal de estos leds es que cuentan con un controlador interno que es capaz de interpretar las señales enviadas por otro microcontrolador externo que actúe como controlador principal del módulo enviando pulsos de distintas longitudes a nivel alto y nivel bajo, como más tarde se detallará, indicando así la intensidad a la que el led debe mostrar cada color. El uso de este modo de control de los leds es idóneo, ya que nos permite controlar tanto el color de los leds como el tiempo en el que los leds deben mostrar el color indicado, permitiéndonos formar imágenes que se perciban en movimiento por el ojo humano.

Requerimientos temporales de actualización de la imagen

Sin embargo, esto aporta una nueva limitación al problema, y esta es que, dado que los leds se encienden mediante ciclos del microcontrolador a nivel alto o bajo, al final el recurso que estamos utilizando para definir el color de una serie de leds es el tiempo. Y, aunque estos ciclos son bajos, del orden de 25 ns, si utilizamos un número muy alto de leds controlados por el mismo microcontrolador, es posible que, al llegar la señal de los últimos leds transmitida por el micro, haya pasado tanto tiempo que se pueda apreciar cierta marea en el panel al ojo humano, ya que para que el ojo humano perciba que una imagen está en movimiento es ideal que cuente con una actualización de la imagen de al menos 60 fps. Esto quiere decir que la imagen debe actualizarse completamente 60 veces por segundo, o lo que es lo mismo, actualizar la imagen cada 16,7 ms. Es decir, cada 16,7 ms la MCU debe haber recibido la imagen a mostrar, debe haber procesado los datos y debe haber enviado todos estos datos a cada led. Por este motivo, se ha optado por la opción de que cada GPIO del microcontrolador se encargue de enviar los datos a un submódulo, compuesto cada uno por 64 leds (8x8) utilizando un total de 9 señales para controlar los 9 submódulos. Este sistema, aunque efectivo, trae nuevos desafíos: ¿Cómo se logra que los módulos sean versátiles si cada submódulo es alimentado por una señal diferente? ¿Cómo llevamos a cabo las conexiones para que a partir de un solo MCU lleguen las señales a los 9 submódulos? ¿Cómo podemos hacer que todas las PCBs de los diferentes submódulos sean iguales y no tener que hacer una PCB diferente para cada posición del módulo?

Justificación solución modular

Tanto el refresco de la imagen como el elevado consumo son las principales limitaciones que hace que los módulos de led sean de tamaño reducido y que debamos dividir la imagen en partes encargándole la ejecución a numerosos MCUs indicándoles que parte de una imagen mayor deben mostrar.

Para solventar este desafío, dado que, como hemos visto, ni un solo micro puede controlar toda una pantalla gigante ni una sola fuente de alimentación alimentarla, se ha optado por tartar a cada módulo de manera independiente haciéndole llegar los datos que deben procesar de modo inalámbrico, de modo que cada módulo solo deba encargarse de procesar los datos recibidos y enviarlos a cada led en concreto, sin ser consciente de la sección de la imagen completa que representa. En este sentido cada módulo contará con una dirección específica del mismo la cual, al ser reconocida en un mensaje transmitido por la aplicación de control de la pantalla indicará al módulo que los parámetros recibidos son los de su fragmento de imagen.

A continuación, se procederá a la explicación del funcionamiento de los módulos donde también se dará respuesta a las cuestiones anteriormente planteadas acerca de la versatilidad y el modo de conexión de los módulos.

Funcionamiento de los módulos

Como se ha mencionado, cada módulo estará compuesto a su vez de 9 submódulos, formando un conjunto cuadrilátero de 30x30 cm y 576 leds (24x24) (ver Imagen 9). Estos 9 submódulos estarán controlados por un solo microcontrolador en uno de ellos que será el submódulo central o principal. La disposición de los submódulos será semejante a la de los números del 1 al 9 en la calculadora (Imagen 10), siendo el 5 el dispositivo central que cuente con MCU y con la entrada de la corriente de la fuente y que suministre tanto la corriente como la señal a cada led del módulo, incluyendo los de los submódulos asociados y los de el mismo submódulo 5.

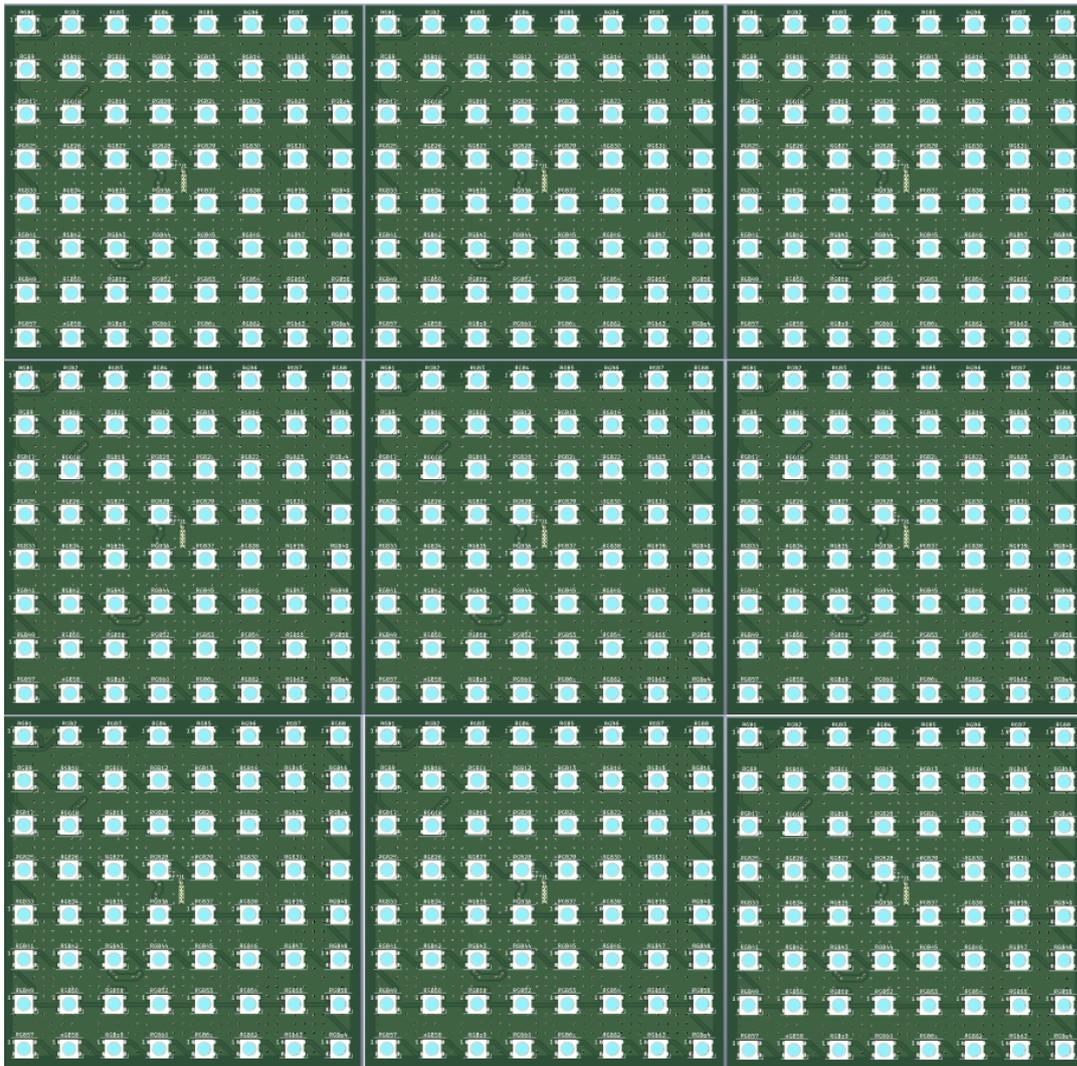


Imagen 9: vista cenital de módulo compuesto de 9 submódulos. (creación propia)

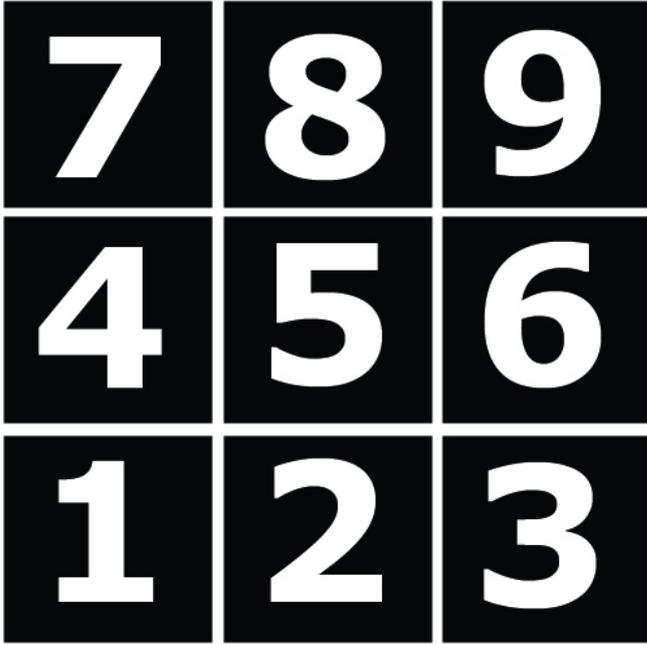


Imagen 10: posición números de la calculadora. (creación propia)

Versatilidad de los módulos

Por tanto, y respondiendo a la pregunta de cómo hacer que todas las PCBs de control de los submódulos sean iguales aportando versatilidad, a pesar de que las PCBs de control centrales (5) lleven un MCU mientras las otras solamente se encargan de transmitir la línea de datos a los submódulos y alimentarlas, la solución adoptada ha sido emplear una PCB de control versátil, complementaria a la una PCB común portadora de los leds (Imagen 7 e Imagen 8) la cual será intercambiable, y reemplazable en caso de que algún led se dañe; que en caso de ser la central lleve incorporado el microcontrolador seleccionado, mientras que en los otros submódulos, 1, 2, 3, 4, 6, 7, 8 y 9; simplemente transmitan las señales entregadas por la PCB de control central a los submódulos asociados o a su PCB de leds.

De tal modo, todas las PCBs de control estarán unidas mediante pines macho hembra como los de la Imagen 11 y la Imagen 12 formando una red de transmisión de datos por debajo de las PCBs portadoras de leds haciendo que estas últimas simplemente se unan a la estructura de soporte con sus pines de conexión sin importar en absoluto la posición del módulo que ocupen.

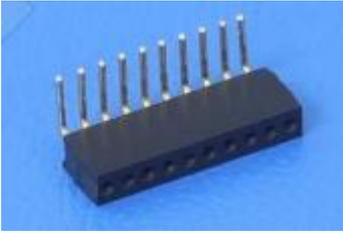


Imagen 11: pines hembra de unión de las PCBs de control. Obtenida de la web Conexcon.com

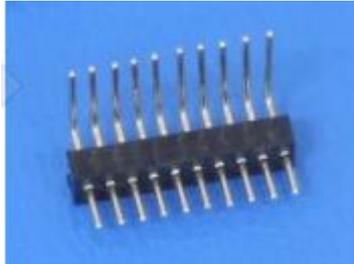
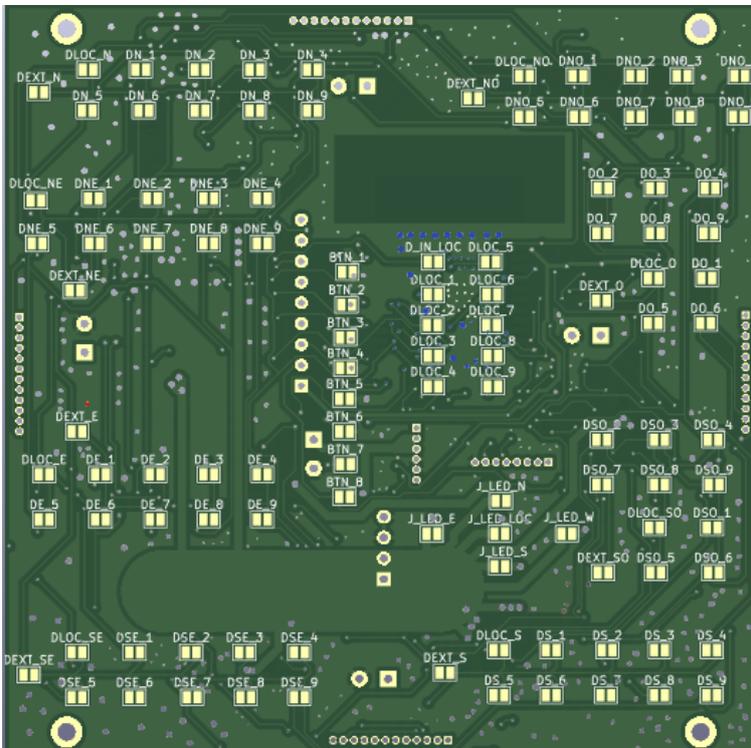


Imagen 12: pines macho de unión de las PCBs de control. Obtenida de la web Conexcon.com

A través de los pines macho-hembra, se transmitirá la corriente y los datos de los leds de cada submodulo al siguiente, partiendo siempre los datos del submodulo central (5) a los de la periferia. Para ello se seguirá el siguiente mecanismo de selección de canales de transmisión mediante switches. Destacar que en el ejemplo desarrollado en este proyecto se ha empleado switches de soldadura (ver Imagen 13), que también pueden ser empleados como huellas de resistencias jumper SMD de 0 ohmios y tamaño 0805.



1. Solo la PCB de control central (5) contará con un microcontrolador. A partir de esta PCB salen todos los datos para todos los leds y se controla la alimentación de todas las placas. El resto de las PCBs no tendrán el microcontrolador montado.
2. El microcontrolador transmite, como podemos ver en la Imagen 14, a través de los pines DATAx la trama de encendido de los leds y a través de los pines Alim_LEDS_X activa la alimentación de los distintos submódulos.

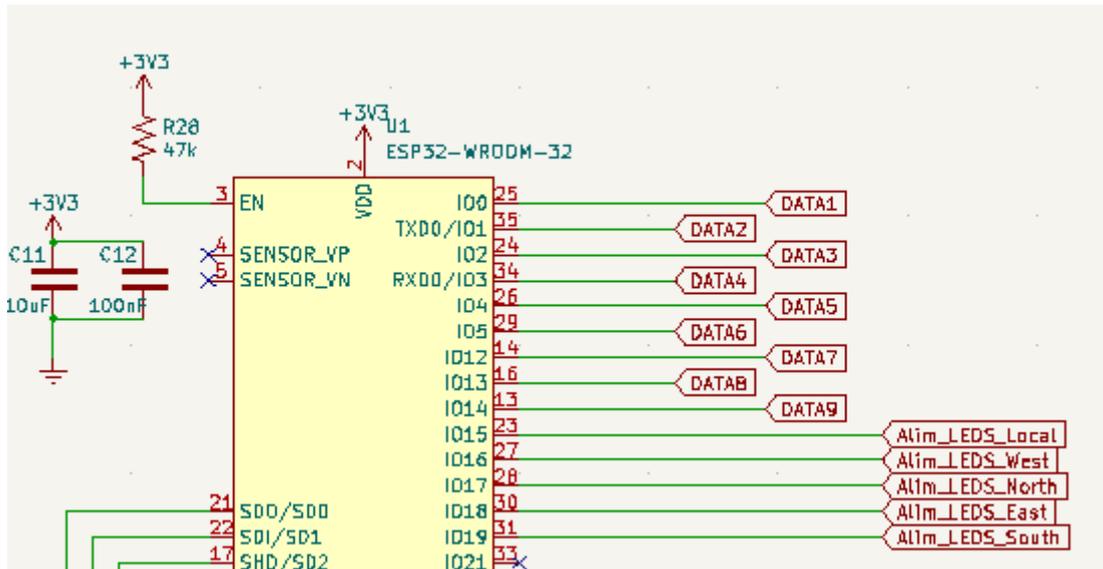


Imagen 14: conexiones microcontrolador de PCB de control.

3. En primer lugar, se va a explicar el proceso de control de la alimentación. En la Tabla 1: definición de las señales de alimentación. ,se explica para que sirve cada señal de alimentación.

Tabla 1: definición de las señales de alimentación.

Alim_LEDS_Local	Sirve para activar la alimentación en la PCB de control 5. Esta PCB es la local, la que posee el microcontrolador.
Alim_LEDS_West	Sirve para activar la alimentación en la PCB de control 4. Se considera la PCB West por analogía a los puntos cardinales.
Alim_LEDS_North	Sirve para activar la alimentación en la PCB de control 8. Se considera la PCB North por analogía a los puntos cardinales.
Alim_LEDS_East	Sirve para activar la alimentación en la PCB de control 6. Se considera la PCB East por analogía a los puntos cardinales.
Alim_LEDS_South	Sirve para activar la alimentación en la PCB de control 2. Se considera la PCB South por analogía a los puntos cardinales.

A partir de las PCBs de control 2,4,6 y 8 se alimentarán a las PCBs 3,1 9 y 7 respectivamente, como a continuación se desarrollará.

4. En la Imagen 15, se puede apreciar el circuito de control de la alimentación. Al llevar a nivel alto la señal, en este caso la señal Alim_LEDS_North, se activa el

- La alimentación se transmite desde la PCB de control 5 hasta la PCB de control 8 a través de la señal V_LED_North, pero, como es lógico, entra a la PCB de control 8 a través de la señal V_LED_South. Y de ahí a través del jumper J_LED_S1, se une a la alimentación global de esta PCB de control (+5V).

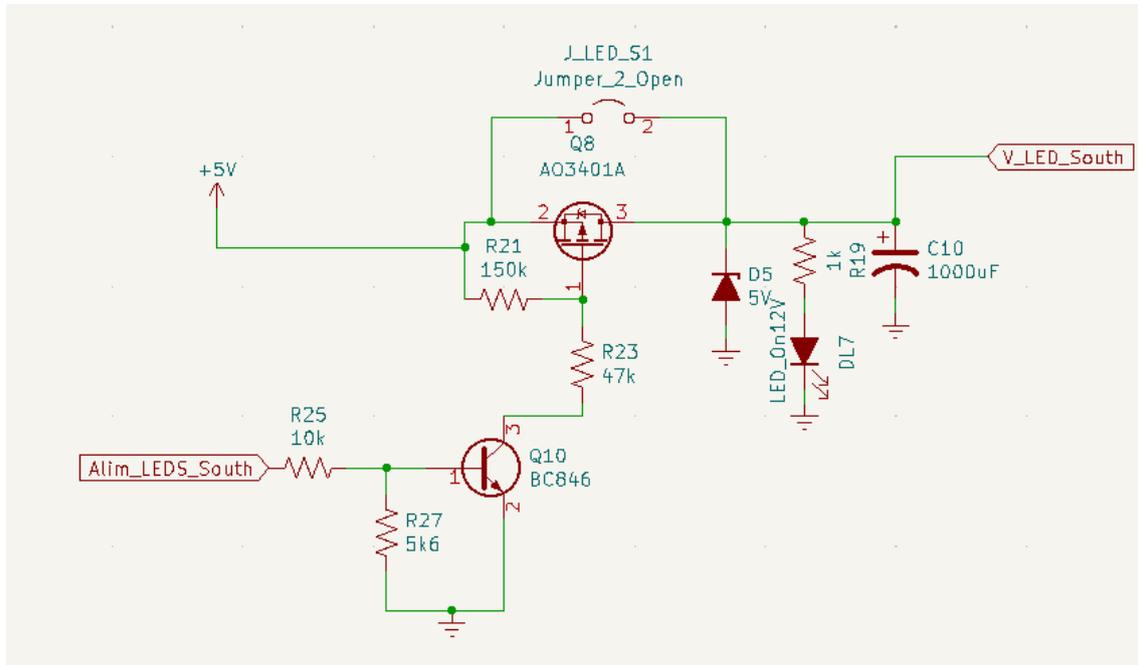


Imagen 16: circuito de control de la alimentación de los leds South.

- Como es lógico la PCB central, 5, solo tiene acceso directo a las PCBs 4,6,8 y 2. Por lo tanto, a través de las PCBs 4, 6, 8 y 2 deben alimentarse las PCBs restantes, 1,3,7 y 9. En este sentido, y, aunque es versátil por si acaso las necesidades de la forma del módulo lo requiere, por ejemplo en un módulo con forma de H; en módulos comunes con 9 submódulos, lo habitual será que a través del módulo 8 se alimente el 7; a través del 4 el 1; a través del 2 el submódulo 3, y a través del 6 el submódulo 9. Continuando con el ejemplo, donde ya había corriente en el submódulo 8, para pasar esta al submódulo 7 se utilizará el circuito de alimentación West (Imagen 17). En él, a través del jumper J_LED_W1 se transmitirá la alimentación de +5V hacia V_LED_West que se unirá mediante pines macho-hembra a V_LED_East de la PCB de control 7.

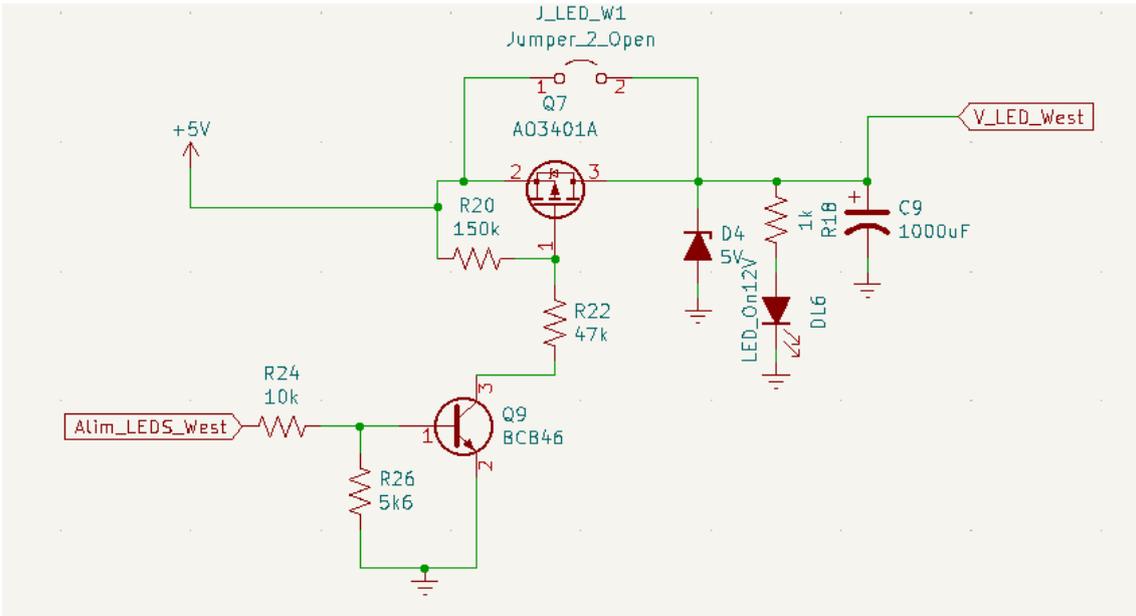


Imagen 17: circuito de control de la alimentación de los leds West.

7. A través del circuito de alimentación East de la PCB de control 7 , cerrando el jumper J_LED_E1 se transmite la alimentación a la red +5V de la PCB 7.

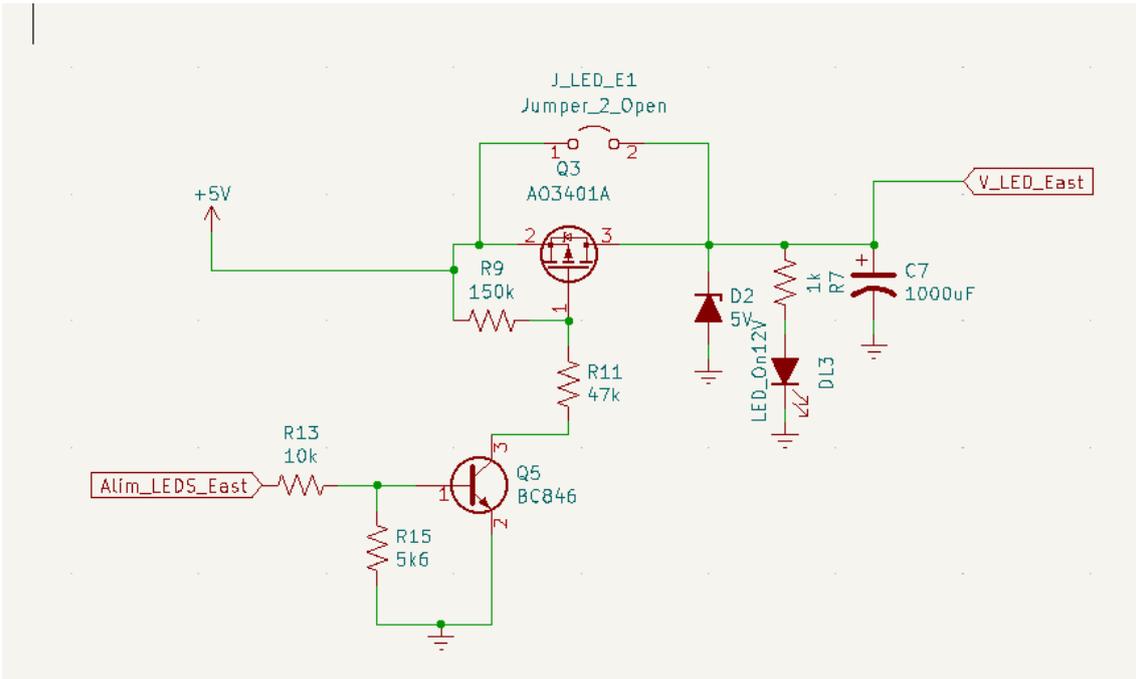


Imagen 18: circuito de control de la alimentación de los leds East

8. Una vez ya hay corriente en la red +5V está se transmite a la alimentación de la placa de LEDs a través de la señal V_LED_Local. Que deberá ser unida a la red +5V a través del jumper J_LED_LOC1. (Ver Imagen 19)

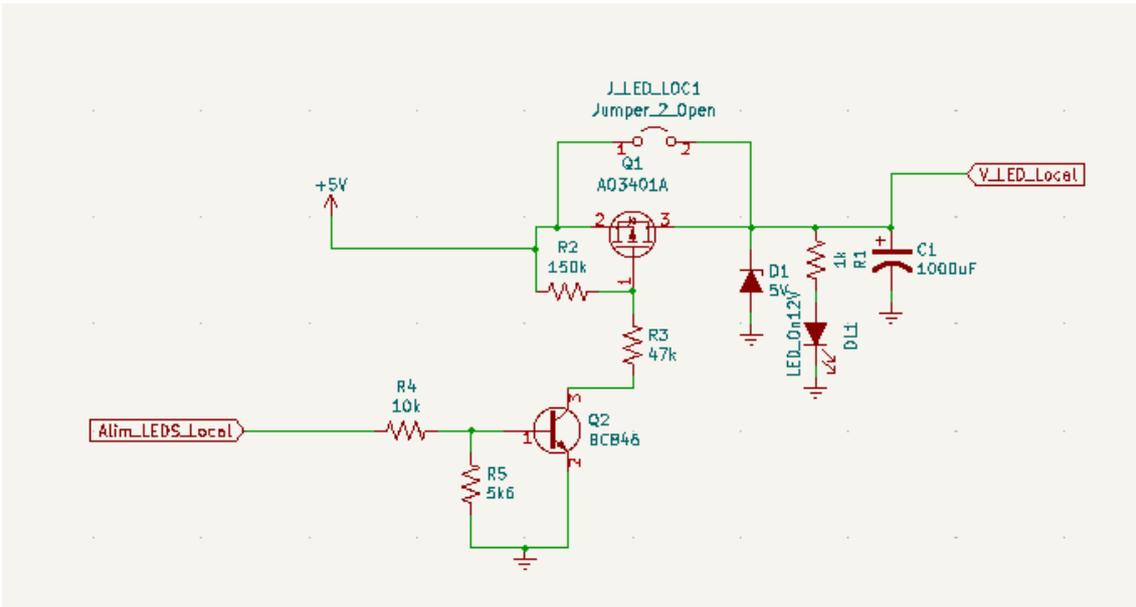


Imagen 19: circuito de control de la alimentación de los leds Local

- La placa de leds será alimentada a través del conector de la Imagen 20 independientemente del submódulo del que se trate.

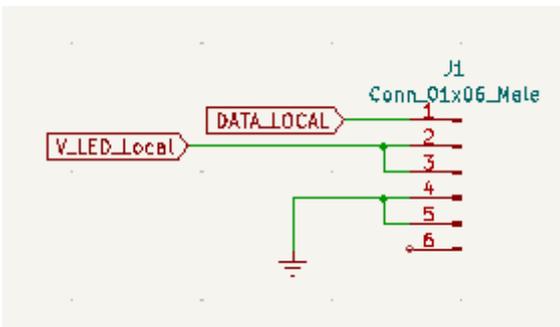


Imagen 20: conector alimentación y datos leds.

De este modo, siguiendo la misma filosofía con el resto de PCBs de control, con una sola placa central se controlará la alimentación de los 9 submódulos manteniendo un único diseño de PCB de control. En la siguiente imagen se detalla el recorrido de la alimentación para clarificar el proceso.

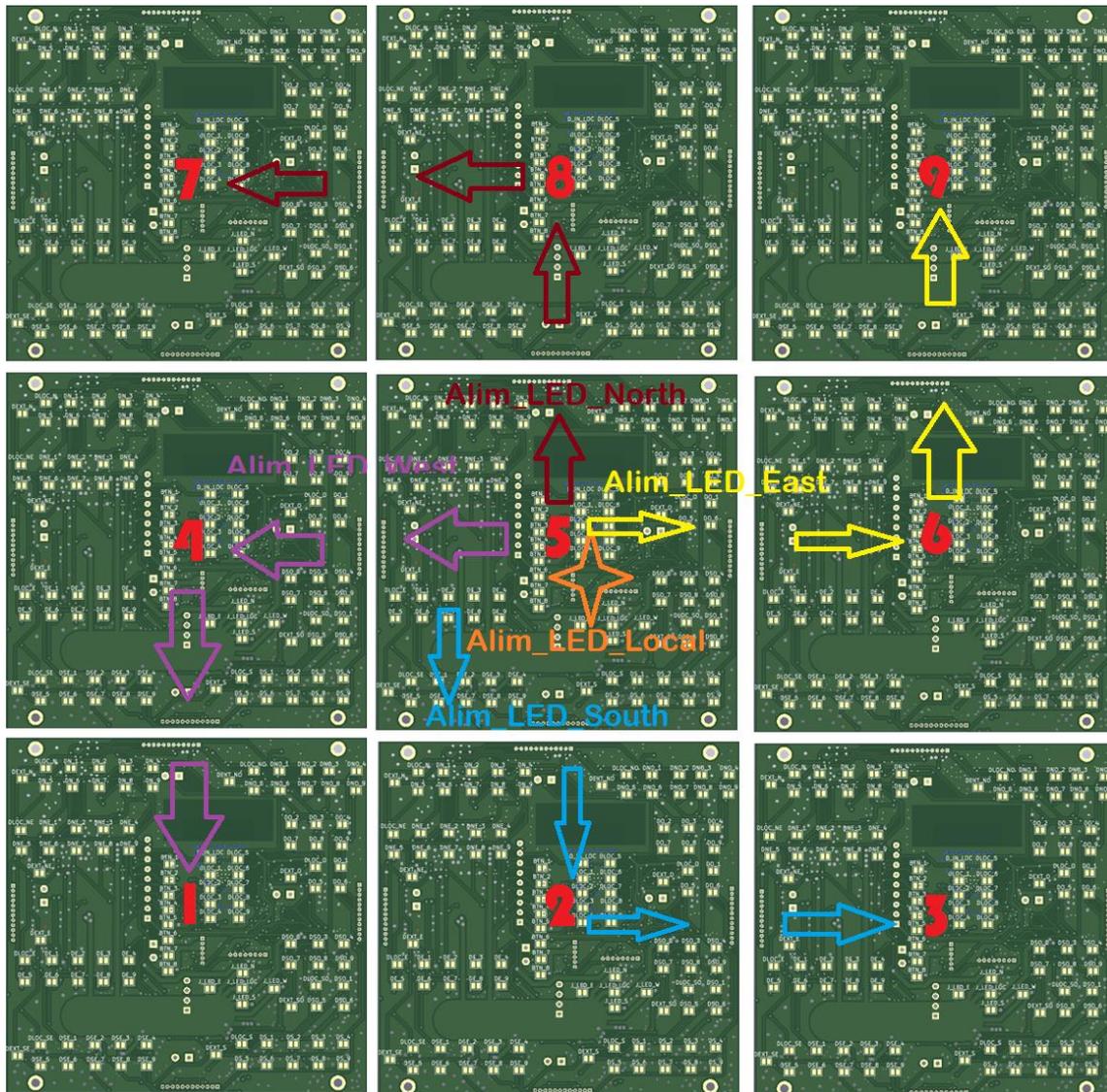


Imagen 21:esquema recorrido señal de alimentación. (Creación propia)

- Canal Naranja: a través del circuito Alim_LED_Local se transmite la corriente a los leds desde la fuente de alimentación que se conecta en la misma PCB de control 5.
- Canal Azul: a través del circuito Alim_LED_South de la placa 5 se transmite la corriente a la placa 2 a través del circuito norte de la placa 2. Esta a su vez transmite la señal a la placa 3 a través de su circuito de alimentación este y, tanto la placa 2 como la 3 transmiten a los leds la alimentación mediante sus circuitos de alimentación local.
- Canal Lila: a través del circuito Alim_LED_West de la placa 5 se transmite la corriente a la placa 4 a través del circuito East de la placa 4. Esta a su vez transmite la señal a la placa 1 a través de su circuito de alimentación South y, tanto la placa 2 como la 3 transmiten a los leds la alimentación mediante sus circuitos de alimentación local.
- Canal Amarillo: a través del circuito Alim_LED_East de la placa 5 se transmite la corriente a la placa 6 a través del circuito West de la placa 6. Esta a su vez

transmite la señal a la placa 9 a través de su circuito de alimentación North y, tanto la placa 6 como la 9 transmiten a los leds la alimentación mediante sus circuitos de alimentación local.

- Canal Granate: a través del circuito Alim_LED_North de la placa 5 se transmite la corriente a la placa 8 a través del circuito South de la placa 8. Esta a su vez transmite la señal a la placa 7 a través de su circuito de alimentación West y, tanto la placa 8 como la 7 transmiten a los leds la alimentación mediante sus circuitos de alimentación local.

De un modo semejante a la alimentación se gestionará la transmisión de los datos de iluminación a los leds siguiendo los mismos canales antes descritos.

Existen 9 señales de datos que salen del microcontrolador (Imagen 22), cada una de ellas deberá llegar hasta su respectiva PCB de leds, pero mediante 9 PCBs de control que sean iguales. Esto se logra uniendo los diferentes jumpers, ya sea mediante soldadura o mediante resistencias jumper de 0 Ohm (0805) (Imagen 23).

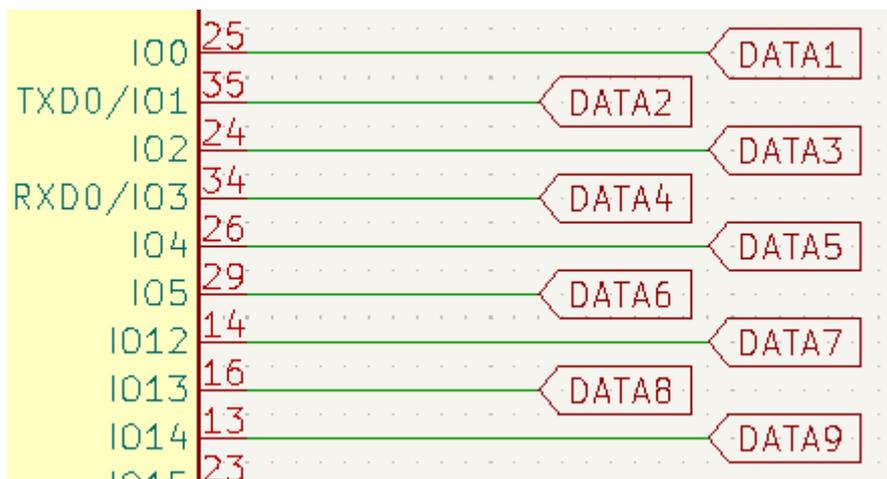


Imagen 22: conexiones de datos PCB de control. Desarrollo propio

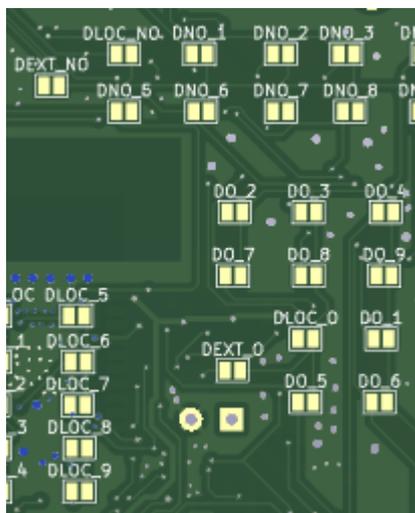


Imagen 23: jumpers PCB de control. Creación propia

A continuación, se explicará la conexión más común de las distintas placas, la que ha sido tomada en cuenta en el FW desarrollado y que se asocia a la configuración 0xFF, como más adelante se desarrollará. En caso de cambiar las conexiones el dispositivo encendería los submódulos de un modo desordenado. El criterio consiste en transmitir cada DATA con su PCB asociada. Ej.: DATA4 a la PCB de control 4.

En cada placa de control, para alimentar a sus leds en su pcb de leds, se transmite la señal a través del siguiente conector.

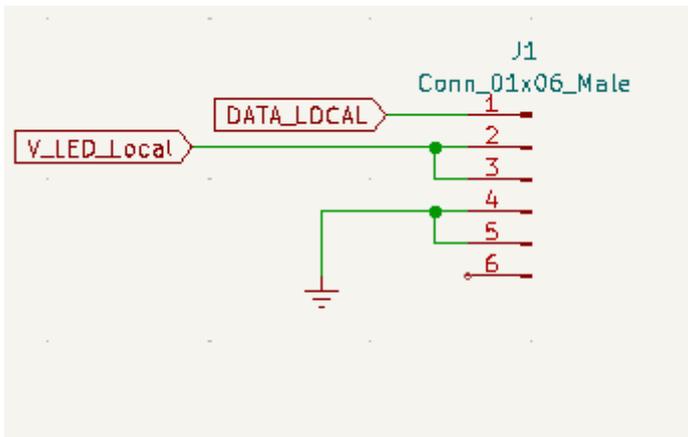


Imagen 24: esquema del conector de la PCB de control. Creación propia.

Esto es sencillo en la PCB de control 5 ya que simplemente se une el jumper de DATA5 con DATA_LOCAL.

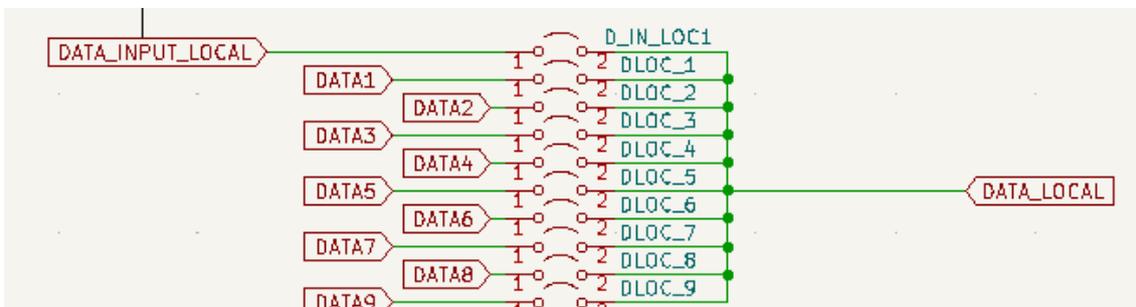


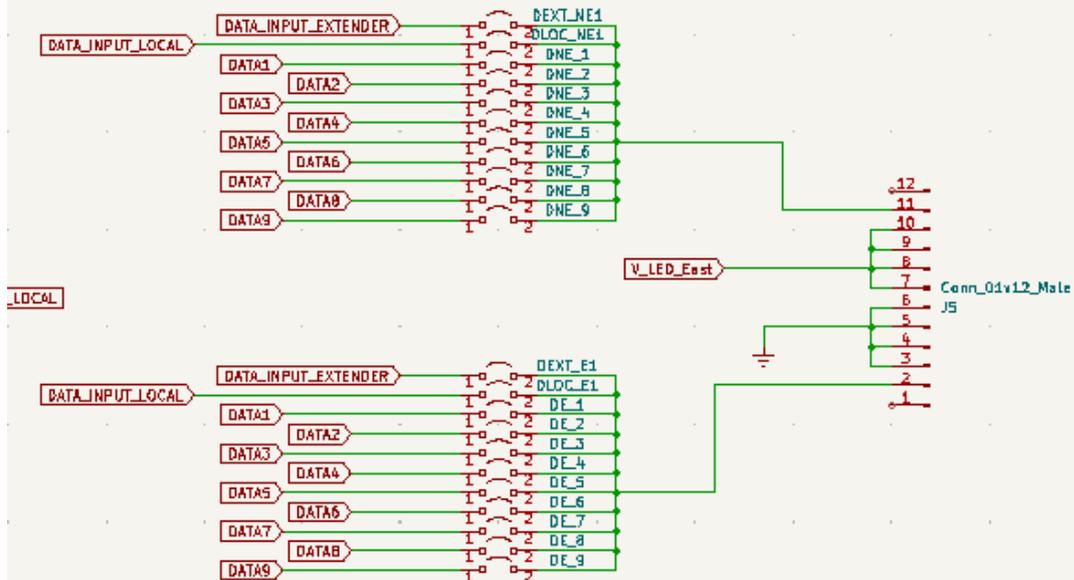
Imagen 25: jumpers de conexión de PCB central.

En cambio, para las demás PCBs es necesario crear un sistema. Ya que, a través de la PCB 5 se debe transmitir los datos a todas las demás.

Se utilizará de ejemplo para ilustrar las PCBs 6 y 9.

A través, del bloque de jumpers este, se unirá el jumper DATA6 con el conector de salida. Además, se deberá transmitir la señal de DATA9 a través del mismo conector uniendo el jumper asociado en el bloque de jumpers noreste. Imagen 26

DATA NORESTE (9)



DATA ESTE (6)

Imagen 26: conexiones lado este y noreste PCB.

Las señales llegarán a la PCB 6 a través, como es lógico del conector del lado oeste. Imagen 27.

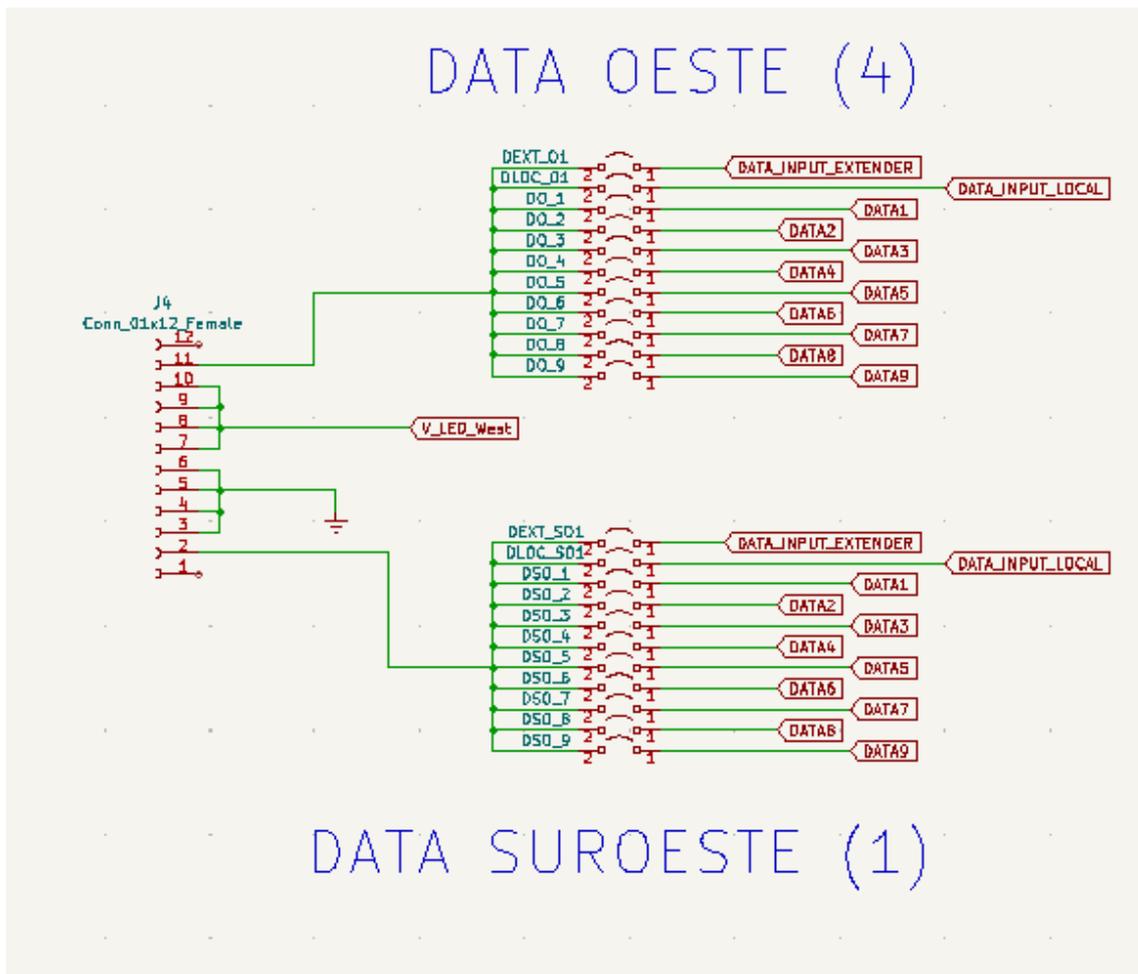


Imagen 27.conexiones lado oeste y suroeste PCB de control.

A través del pin de datos de abajo se recibe la señal DATA6 la cual en el bloque de jumpers suroeste de la PCB 6 deberá unirse con la red DATA_IMPUL_LOCAL. A continuación, se une a través del bloque de jumpers central (Imagen 25), la señal DATA_IMPUL_LOCAL con la señal DATA_LOCAL con el jumper dispuesto para tal efecto. Una vez la señal llega a la red DATA_LOCAL esta se une con la PCB de los leds asociada a través del conector de la Imagen 24 del mismo modo que la PCB número 5.

Para la PCB número 9 el recorrido es semejante, pero enviando la señal recibida hasta la siguiente PCB.

Comienza el recorrido como se ha comentado uniendo el jumper de DATA9 del bloque de jumpers noreste de la PCB 5. Esta señal se recibe en la PCB 6 a través del conector oeste y se une con la red DATA_INPUT_EXTENDER a través del bloque de jumpers oeste.

A continuación, se transmite la señal a través del bloque conector norte uniendo el jumper DATA_INPUT_EXTENDER del bloque de jumpers NORTE de la placa 6. Imagen 28

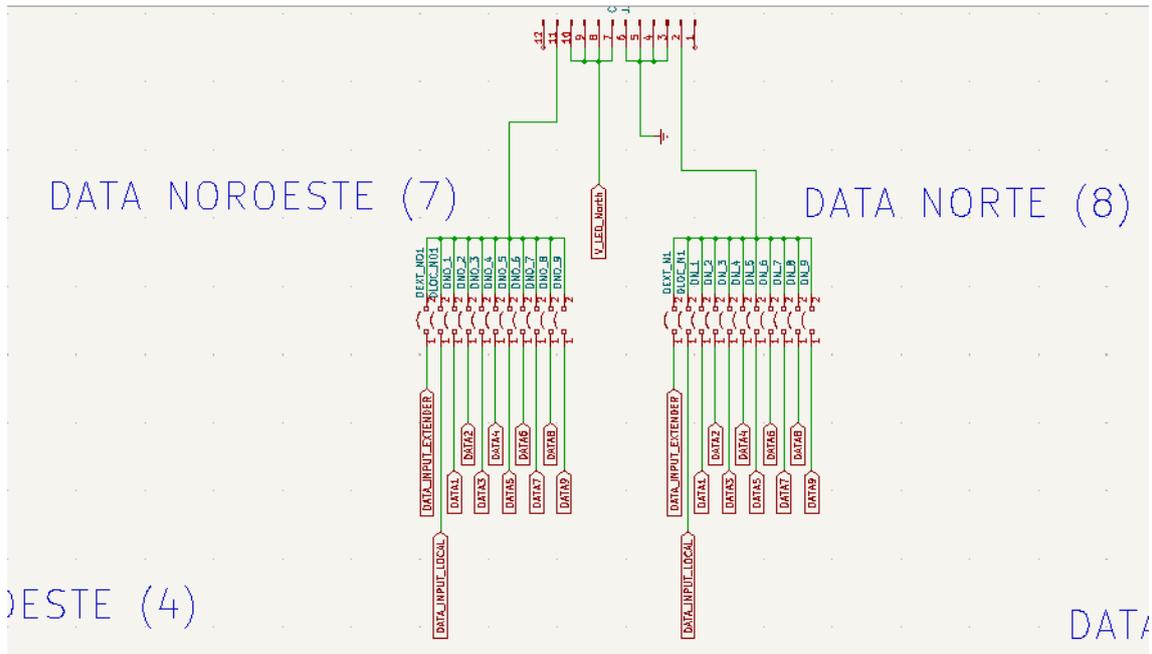


Imagen 28:conector norte PCB de control

La señal se recibe ya en la placa 9 a través del conector sur de la placa 9 uniendo el jumper DATA_INPUT_LOCAL del bloque de jumpers sureste, (Imagen 29). A continuación, se une en el jumper de DATA_INPUT_LOCAL con el conector central de la PCB (Imagen 25) llegando así la señal a la red DATA_LOCAL que a su vez está unida con la PCB de leds por el conector central mostrado anteriormente en la Imagen 24

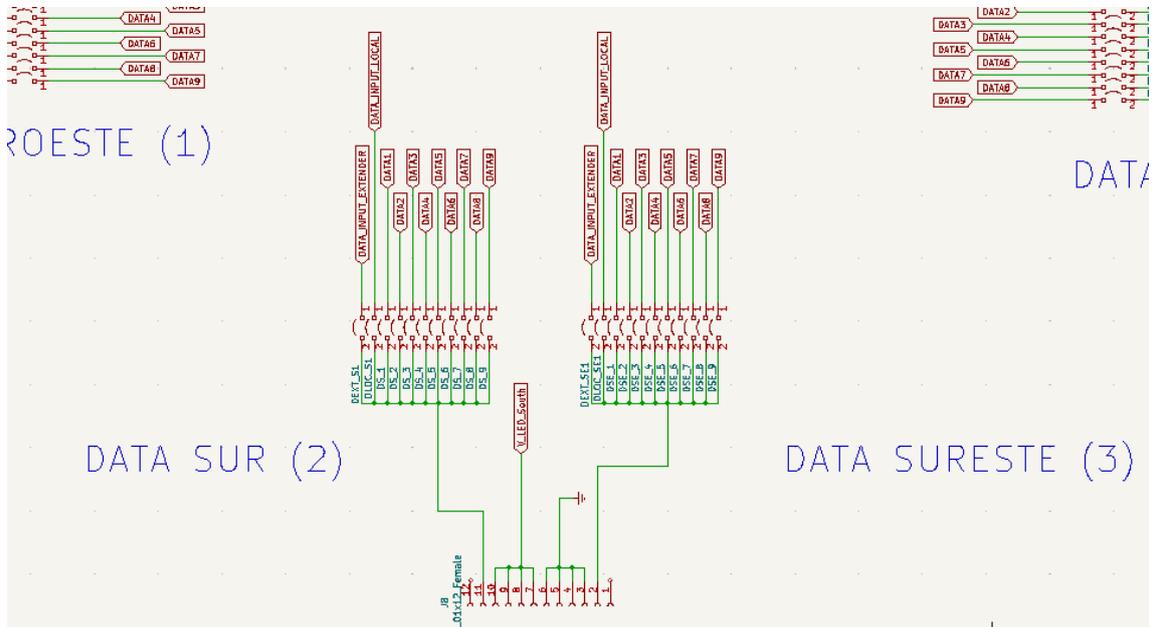


Imagen 29:conector sur PCB de control

De un modo análogo, se transmiten las demás señales de datos de las diferentes placas. La 8, la 4 y la 2 directamente a través de la placa 5 y la 7, la 1 y la 3 a través de las PCBs de control 8, 4 y 2 respectivamente.

Con este funcionamiento de los módulos es posible estandarizar la fabricación de módulos, generando únicamente 2 PCBs con sus respectivos gerbes y estandarizando el montaje de las distintas resistencias jumper y del microcontrolador para cada número de submódulo dependiendo de los requerimientos de la forma del módulo abaratando costes de fabricación de PCBs y facilitando el montaje.

Elección del método de comunicación

A la hora de tomar la decisión acerca del método utilizado para transmitir la información debemos tener en cuenta 3 factores:

En primer lugar, debemos valorar la facilidad de uso del mismo, ya que, por ejemplo, la comunicación lora necesita que exista alrededor una infraestructura de LoRa limitando la utilidad del producto o las comunicaciones LTE o NbloT requieren de tarjetas sim, aumentando costes. En esta línea, los métodos de comunicación más versátiles son el bluetooth y el wifi.

Los otros dos factores a tener en cuenta son la cantidad y velocidad de datos a transmitir y la limitación espacial de la tecnología utilizada.

En ambos casos la tecnología con mejores características es el wifi, ya que, por un lado, permite la transmisión de grandes cantidades de datos a grandes velocidades, cumpliendo el requisito del refresco de la imagen a 60 fps y por el otro no requiere estar cerca de la pantalla para interactuar con ella. De este modo podrías controlar la pantalla desde grandes distancias frente a los pocos metros de margen que el bluetooth permitiría.

Elección del microcontrolador

A continuación, trataremos la elección del microcontrolador:

Para elegir el microcontrolador correcto debíamos tener en cuenta 3 factores: las características profesionales del mismo, en especial en términos de características técnicas; la velocidad de los ciclos de reloj, para que nos permita refrescar la imagen cumpliendo los tiempos establecidos, y la facilidad para la transmisión de datos de modo inalámbrico.

Dado que nuestro método de comunicación elegido ha sido la comunicación wifi y que es un microcontrolador de características profesionales que nos permita tener unos ciclos de reloj que se ajustan a las necesidades, y que es fiable y cuenta con amplia experiencia de uso colectiva, lo que aporta facilidad a la hora de programar utilizando librerías existentes o teniendo acceso a como otros usuarios han resuelto sus problemas, se ha seleccionado un microcontrolador ESP32 de la marca Espressif. En concreto hemos seleccionado el ESP32-Wroom-32 ya que reúne las características

necesarias, posee un precio competitivo y hay stock del mismo en las páginas de compra de componentes electrónicos más comunes (Mouser y Digikey).

Elección del método de control

Para el control de la pantalla completa compuesta de distintos módulos, y a su vez de submódulos, utilizaremos un dispositivo móvil, como puede ser el móvil personal de cada uno, con una aplicación para Android desarrollada en Kotlin, variante de Java, la cual, no será el objeto de desarrollo de este proyecto. Esta aplicación transmitirá la imagen que debe mostrar, elegida por el usuario, dividida en los distintos módulos y en sus respectivos submódulos, enviando a través de wifi a cada microcontrolador los datos que este se encargará de transmitir a cada led con el protocolo que a continuación explicaremos.

Es decir, la aplicación procesará la imagen a mostrar y transmitirá, según el protocolo que a continuación detallaremos a los distintos módulos, que simplemente representarán la subimagen que les ha llegado, sin tener conocimiento global de la sección de imagen completa que componen. Esta comunicación deberá producirse a una velocidad de 60 veces por segundo para alcanzar los parámetros de calidad que requiere el proyecto obteniendo así videos de alta calidad, o lo que es lo mismo, cada 16.7ms.

Explicación funcionamiento de manejo de LED

Como podemos ver en la siguiente imagen (Imagen 30) extraída del datasheet de los leds seleccionados, WS2812B, los leds reciben 3 bytes de datos para definir el color que deben alumbrar.

Composition of 24bit data:

G7	G6	G5	G4	G3	G2	G1	G0	R7	R6	R5	R4	R3	R2	R1	R0	B7	B6	B5	B4	B3	B2	B1	B0
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Note: Follow the order of GRB to sent data and the high bit sent at first.

Imagen 30: tabla de funcionamiento leds WS2812B. Datasheet de leds WS2812B.

El primer byte indica el grado de iluminación en verde (green), el segundo del rojo (red) y el tercero del azul (blue). De tal modo el led será capaz de iluminar cada color en un ratio del 0 al 255 siendo 0 la iluminación nula y 255 la iluminación más alta.

Por ejemplo, si el led recibiera los siguientes bytes: 0x000FFF (Desplegado en binario: 0b00000000 0b00001111 0b11111111), el led iluminaría plenamente el color azul, a la mitad el rojo y mantendría a 0 el led verde.

Estos datos son recibidos, como se puede ver en la Imagen 31 a través del pin 4 (DIN). Sin embargo, si se quiere transmitir la información a más de un led, esto se puede hacer mediante el uso del pin 2 (DOUT), ya que estos leds tienen la capacidad de recibir una serie de datos de más de 3 bytes filtrando los 3 primeros para sí mismo y enviando la serie restante por su pin de salida DOUT.

NO.	Symbol	Function description
1	VDD	Power supply LED
2	DOUT	Control data signal output
3	VSS	Ground
4	DIN	Control data signal input

Imagen 31: pines de los leds WS2812B. Datasheet de los leds WS2812B.

Como se ha comentado anteriormente, los datos transmitidos a los leds requieren el empleo de ciclos de reloj de un microcontrolador principal. Esto es debido a como interpretan los leds los unos y ceros que componen la trama recibida. Como se puede ver en la Imagen 32 un periodo de tiempo corto a nivel alto seguido de uno largo a nivel bajo es interpretado como un 0 mientras que lo contrario es interpretado como un 1.

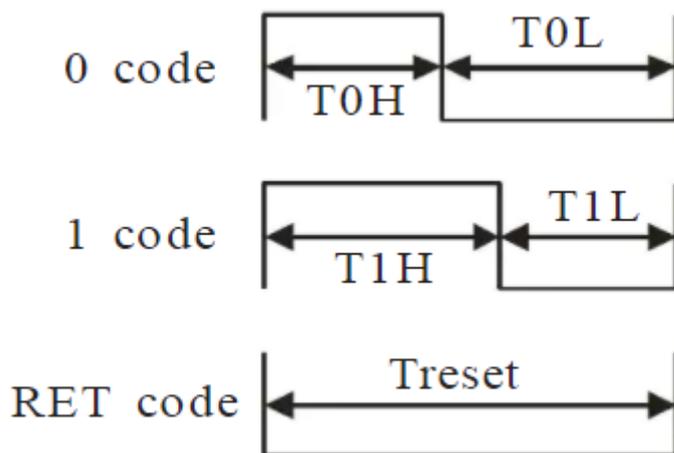


Imagen 32: procesamiento de datos de los leds WS2812B. Datasheet de los leds WS2812B.

Explicación del protocolo de comunicación de datos

Para la transmisión de los mensajes emplearemos un formato estandarizado, compuesto de diferentes partes, para todos los módulos simplificando de tal modo el proceso. Se utilizarán caracteres en hexadecimal, de tal modo si el mensaje está transmitiendo un valor en binario de, por ejemplo, 0110 1110, la aplicación transmitirá el carácter '6E' simplificando de tal modo la lectura y entendimiento del mensaje.

La trama enviada seguirá un orden concreto que siempre será constante, independientemente del número de submódulos o módulos de los que la pantalla se componga.

Se parte de la base de que hay 9 submódulos de 64 leds cada uno y cada cual se activa con 24 bits como se ha explicado en el punto anterior. (Imagen 30)

Composition of 24bit data:

G7	G6	G5	G4	G3	G2	G1	G0	R7	R6	R5	R4	R3	R2	R1	R0	B7	B6	B5	B4	B3	B2	B1	B0
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Note: Follow the order of GRB to sent data and the high bit sent at first.

Imagen 33: tabla de funcionamiento leds WS2812B. Datasheet de leds WS2812B.

A continuación, se detalla la trama de comunicación:

1. Dirección: un byte, hasta 255 valores. Este campo sirve para que los distintos módulos distingan las tramas y escuchen solo las que tienen su dirección. El valor de la dirección con el que se comparará estará indicado por HW dependiendo de la colocación de los pines específicos del microcontrolador como se muestra en la Imagen 34. Mediante jumpers de soldadura o resistencias jumper de 0 Ohms (0805) se podrá indicar la dirección formando un byte con un 1 en la posición de cada jumper abierto y un 0 en cada posición cerrada.

Este hecho es clave porque implica que el máximo número de módulos colocables por pantalla es 255 módulos.

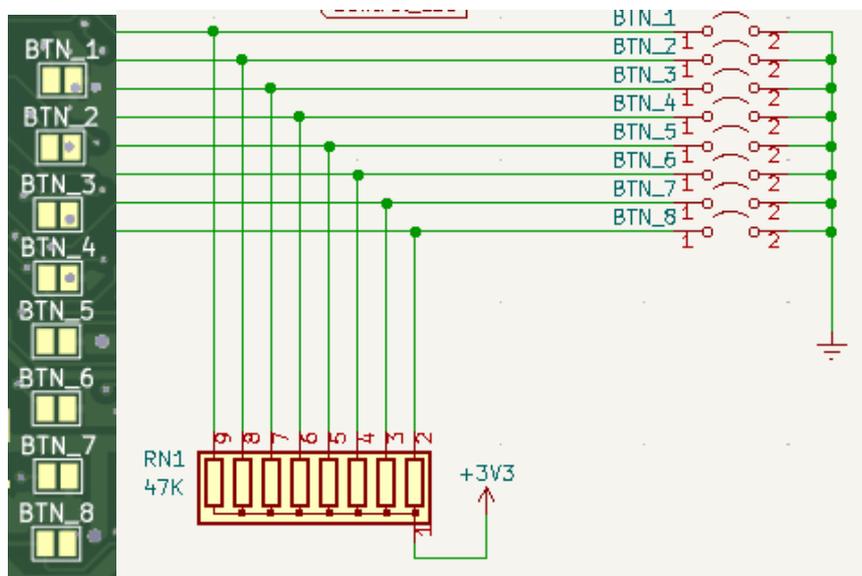


Imagen 34: sección de la cara de soldadura de la PCB de control y esquema adjunto. Desarrollo propio.

2. Serial number del dispositivo, 8 bytes. para distinguir a los distintos equipos. Es grabado por FW. Es decir, cada pantalla tendrá un número de serie definido por FW específico de cada pantalla y común a todos los módulos que la componen. Ej: 0xFF2345ABFFFFFFFF

3. Valores Indicativos: este apartado se compone de 2 bytes, hasta 65 535 valores diferentes. Sirve para tener espacio para indicar distintas opciones a la pantalla, por ejemplo, colocar el logo de la empresa, ir a negro, repetir bucles concretos para mostrar una figura previamente definida, mostrar patrones. Es útil por ejemplo para dejar el módulo iluminando durante un periodo de tiempo en el que no va a haber comunicación. Se plantea como un modo de funcionamiento automático predefinido por FW al que se accede mediante estos bytes. En caso de funcionamiento estándar mantener en 0x00.

4. Configuración: 1 bytes de configuración en la que se indicará si la trama corresponde a un submódulo compuesto de 9 submódulos, 3, en forma de H, de farmacia, etc. Se indicará con 0xFF, en el caso de ser un módulo estándar con las conexiones comunes explicadas en el apartado anterior. Se parte de la premisa de que el submódulo principal 5 siempre va a estar activo. Por tanto, cada bit del Byte de configuración define si están activos o no los submódulos 1-4 y 6-9. Por ejemplo, un módulo de farmacia tiene activos el 5 el 2, el 4, el 6 y el 8, por tanto, se enviaría una configuración tal como esta: 0b01011010. O lo que es lo mismo 0x5A.

5. Data: 1728 bytes (3 colores x 9 submódulos x 64 leds) que indican la intensidad con la que se debe iluminar cada led de cada color en el orden green, red, blue. De tal modo la trama indicará la luminosidad del led green del RGB 1 del submódulo 1, seguido por el led red y blue del RGB 1 del submódulo 1, seguido por el led green del RGB 2 del submódulo 1 y así sucesivamente pasando por los demás submódulos en orden ascendente hasta llegar al led blue del RGB 64 del submódulo 9.

En caso de que el módulo no sea estándar (9 submódulos), sino que tenga formas alternativas se completarán las posiciones que no estén conectadas con ceros. Para estandarizar el mensaje.

Estos datos serán procesados por el microcontrolador aplicando el funcionamiento indicado dependiendo de los datos recibidos.

Un ejemplo de trama sería la siguiente: (explicación a continuación)

```

FFFF2345ABFFFFFFFFF0FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0000FF0000FF0000FF0000FF0000FF
0000FF0000FF0000FF0000FF0000FF0000FF0000FF0000FF0000FF0000FF

```

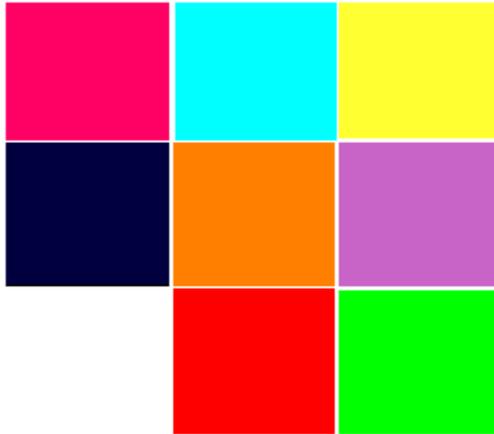



Imagen 35: resultado de iluminación del módulo según la trama planteada. obtención propia.

FW del dispositivo

A continuación, se procederá a explicar el FW del dispositivo.

En primer lugar, comenzamos definiendo los pines que utilizaremos, estos son los que se aprecian en la Imagen 36.

Hay definiciones denominadas HWSNX que son los pines de la dirección del módulo, definidos por HW; pines de control de la alimentación de los leds denominados PWRX, por último, los pines de datos, que son definidos con DATAX.

```

/*****PIN DEFINES*****/
#define HWSN0 23 // HW direction number pins 23,25-27,32-35
#define HWSN1 25
#define HWSN2 26
#define HWSN3 27
#define HWSN4 32
#define HWSN5 33
#define HWSN6 34
#define HWSN7 35

#define DATA1 25 // data pins
#define DATA2 35
#define DATA3 24
#define DATA4 34
#define DATA5 26
#define DATA6 29
#define DATA7 14
#define DATA8 16
#define DATA9 13

#define PWR1 27 //power pins |
#define PWR2 31
#define PWR3 31
#define PWR4 27
#define PWR5 23
#define PWR6 30
#define PWR7 28
#define PWR8 28
#define PWR9 30

```

Imagen 36: definición de pines FW

También se incluyen las librerías que serán empleadas en el desarrollo. Estas son:

```
#include <WiFi.h>
```

```
#include <Arduino.h>
```

```
#include <stdio.h>
```

A continuación, se definen algunas constantes y algunas variables. (Imagen 37)

```
/******General defines*****  
#define DATA_LEDS_SIZE 193 //3 bytes * 64 leds = 192 bytes + \0  
#define SW_SERIAL_NUMBER_SIZE 9 //8+1  
#define GLOBAL_SIZE 1736 //1735 + \0  
#define DEFINED_MS_TIME 16 /// 16ms 16,67 ms para 60fps  
  
/******Definicion de variables*****  
int HWDirectionNumber = 0;  
  
char data_in[GLOBAL_SIZE] = {};  
char data_leds_1[DATA_LEDS_SIZE] = {};  
char data_leds_2[DATA_LEDS_SIZE] = {};  
char data_leds_3[DATA_LEDS_SIZE] = {};  
char data_leds_4[DATA_LEDS_SIZE] = {};  
char data_leds_5[DATA_LEDS_SIZE] = {};  
char data_leds_6[DATA_LEDS_SIZE] = {};  
char data_leds_7[DATA_LEDS_SIZE] = {};  
char data_leds_8[DATA_LEDS_SIZE] = {};  
char data_leds_9[DATA_LEDS_SIZE] = {};  
char data_serial_number_in[SW_SERIAL_NUMBER_SIZE] = {};  
char SW_serial_number[SW_SERIAL_NUMBER_SIZE] = {"12345678"};//en ASCII (8Bytes)  
  
char data_In_Aux = '0';  
int data_counter = 0;  
int correct_Dir = 0;  
int correct_SN = 0;  
unsigned long currentTime = 0;  
unsigned long previousTime = 0;  
  
int i = 0; //loop variable  
  
const char* ssid = "Redmi Note 9";  
const char* password = "PASSWORD";
```

Imagen 37: variables FW modulo.

La mayoría de las constantes y variables son autodescriptivas, sin embargo, se procede a explicar cada una.

-DATA_LEDS_SIZE → 193, recoge el tamaño que ocupa toda una trama de leds de un submódulo, compuesto por 64 leds, con 3 bytes de información requerida por cada uno más en carácter nulo.

-SW_SERIAL_NUMBER_SIZE → 9, define el tamaño del número de serie más el carácter nulo.

-GLOBAL_SIZE → 1736, define el número completo de bytes de la trama más el carácter nulo.

-DEFINED_MS_TIME → 16, define el número de ms que debe haber entre cada imagen procesada para que la tasa de actualización de la imagen sea de 60 fps.

-HWDirectionNumber → inicializada a 0, aquí se recogerá más adelante el número de dirección definido por HW del dispositivo.

-char data_in [GLOBAL_SIZE] = {}; → un array de caracteres para contener la trama recibida por el dispositivo.

-char data_leds_1[DATA_LEDS_SIZE] = {}; → arrays más pequeños donde contener la subtrama específica de cada submódulo, hay 192 espacios más el carácter nulo. (64*3 +1). Hay 9 arrays, uno por cada submódulo.

-char data_serial_number_in [SW_SERIAL_NUMBER_SIZE] = {}; → array para guardar el número de serie enviado en la trama.

-char SW_serial_number [SW_SERIAL_NUMBER_SIZE] = {"12345678"}; → array para guardar el número de serie definido por SW.

-char data_In_Aux = '0'; → carácter auxiliar para procesar el mensaje

-int data_counter = 0; → contador de caracteres

-int correct_Dir = 0; → variable para indicar si la dirección es correcta o no.

-int correct_SN = 0; → variable para indicar si el número de serie es correcta o no.

-unsigned long currentTime = 0; y unsigned long previousTime = 0; → variables para medir el tiempo que ha transcurrido desde la última interacción y asegurarse que ha pasado el tiempo requerido.

-int i = 0; //loop variable → variable para los bucles.

-const char* ssid = "Redmi Note 9"; → Nombre de la red wifi.

-const char* password = "PASSWORD"; → Contraseña de la red Wifi.

A continuación, se explicará el funcionamiento de la función setup ().

```

void setup()
{
  Serial.begin(115200);
  delay(100);

  WiFi.begin(ssid, password);//Start connection

  while (WiFi.status() != WL_CONNECTED) { // Waiting for set the connection
    delay(500);
    Serial.print("Waiting for WiFi connection");
  }

  Serial.println("");
  Serial.println("WiFi connected.");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());//print IP
  server.begin();

  HWDirectionNumber = getHWDirectionNumber();//Get hardware direction from the GPIOs
}

```

Imagen 38: función setup () del FW

En primer lugar, se inicializa el puerto serie, a continuación, se comienza el proceso de conexión wifi utilizando el ssid y contraseñas antes definidas. Se permanece a la espera hasta que se obtiene por el puerto serie la expresión: “Wifi connected. IP address: IP.IP.IP.IP”. Por último, se lee con la función getHWDirectionNumber() (Imagen 39) la dirección asociada al dispositivo.

Esta función, obtiene el valor de cada pin de dirección y lo almacena en una variable int que devuelve y cuyo valor termina por guardarse en la variable HWDirectionNumber. Esto lo hace sumando la parte significativa de peso que tiene cada bit en caso de detectar un 1 en el GPIO de entrada.

```

int getHWDirectionNumber (void)
{
  int aux_HW_Direction_Number = 0;

  if(digitalRead(HWSN0))
    aux_HW_Direction_Number += 1;
  if(digitalRead(HWSN1))
    aux_HW_Direction_Number += 2;
  if(digitalRead(HWSN2))
    aux_HW_Direction_Number += 4;
  if(digitalRead(HWSN3))
    aux_HW_Direction_Number += 8;
  if(digitalRead(HWSN4))
    aux_HW_Direction_Number += 16;
  if(digitalRead(HWSN5))
    aux_HW_Direction_Number += 32;
  if(digitalRead(HWSN6))
    aux_HW_Direction_Number += 64;
  if(digitalRead(HWSN7))
    aux_HW_Direction_Number += 128;

  return aux_HW_Direction_Number;
}

```

Imagen 39: función getHWDirectionNumber ().

A continuación, explicaremos el funcionamiento del loop(). (Imagen 40)

```
void loop(){

  WiFiClient client = server.available(); // listen for incoming clients

  if (client) {
    while (client.connected()) {
      if (client.available()) {
        data_In_Aux = client.read();

        if (data_In_Aux == HWDirectionNumber && data_counter == 0) //Same direction
        {
          data_counter++;
          correct_Dir = 1;
        }
        if (data_counter < GLOBAL_SIZE && correct_Dir)//Save data
        {
          data_counter++;
          strcat(data_in,data_In_Aux);
        }
        if(data_counter == GLOBAL_SIZE)
        {
          break; // salir del while
          data_counter = 0;
        }
      }
    }

    client.stop();
    Serial.println("Client Disconnected.");
  }
}
```

Imagen 40: loop() parte 1 del FW.

En el loop, se comienza esperando recibir una petición de un cliente, en este caso la aplicación asociada. En caso de recibirla, comienza la conexión y se comienza a leer el mensaje recibido. Se utiliza la variable auxiliar data_In_Aux para comparar si el primer valor del mensaje coincide con la dirección del dispositivo. A continuación, se almacena el mensaje en el array de datos principal, data_in[], utilizando la variable de carácter auxiliar. Se almacena la información en el array utilizando la función strcat () de la librería inicializada #include <stdio.h>. Una vez se ha recogido todo el mensaje se procede a desconectar la petición.

```

for(i = 1; i < 1+SW_SERIAL_NUMBER_SIZE ; i++)
{
  if (data_in[i] == SW_serial_number[i] && data_counter < SW_SERIAL_NUMBER_SIZE)//compare SW SN with data SN
  {
    data_counter++;
  }
  else if (data_counter == SW_SERIAL_NUMBER_SIZE)
  {
    correct_SN = 1;
  }
  else
  {
    break; // error en el SN. NO es para mi.
  }
}

previousTime = currentTime;
currentTime = millis();
while(currentTime - previousTime >= DEFINED_MS_TIME)//waiting to 16ms
{
  currentTime = millis();
}

```

Imagen 41:loop() parte 2 del FW

A continuación, se procede a comparar si el número de serie del dispositivo, definido anteriormente en el propio programa, coincide. Si lo hace, el programa permanece a la espera usando las variables previousTime y currentTime junto con la función millis() hasta que hayan pasado al menos 16ms desde la última vez que se ha pasado por esta parte del programa. Esto es para mantener la tasa de refresco de imagen en 16ms, ya que como anteriormente mencionamos, debemos refrescar la imagen cada 16,67ms para mantener la tasa de refresco optima, pero no más.

```

if (correct_SN && correct_Dir && (currentTime - previousTime >= DEFINED_MS_TIME))/
{
  for( i=7; i < DATA_LEDS_SIZE+7 ;i++)//7 to save the first values.
  {
    strcat(data_leds_1,data_in[i]);
    strcat(data_leds_2,data_in[i+192]);//192bytes for every submodule message
    strcat(data_leds_3,data_in[i+192*2]);
    strcat(data_leds_4,data_in[i+192*3]);
    strcat(data_leds_5,data_in[i+192*4]);
    strcat(data_leds_6,data_in[i+192*5]);
    strcat(data_leds_7,data_in[i+192*6]);
    strcat(data_leds_8,data_in[i+192*7]);
    strcat(data_leds_9,data_in[i+192*8]);
  }

  send_data (FWR1,DATA1,(char *) data_leds_1);
  send_data (FWR2,DATA2,(char *) data_leds_2);
  send_data (FWR3,DATA3,(char *) data_leds_3);
  send_data (FWR4,DATA4,(char *) data_leds_4);
  send_data (FWR5,DATA5,(char *) data_leds_5);
  send_data (FWR6,DATA6,(char *) data_leds_6);
  send_data (FWR7,DATA7,(char *) data_leds_7);
  send_data (FWR8,DATA8,(char *) data_leds_8);
  send_data (FWR9,DATA9,(char *) data_leds_9);
}
}

```

Imagen 42: loop() parte 3, del FW.

Por último, si la dirección y el número de serie de la trama coinciden con los esperados y ha transcurrido el tiempo requerido, se procede a enviar la trama a los leds. En primer lugar, se utiliza un bucle “for” para almacenar los datos de cada submódulo desde el array principal en un array específico para cada uno. Cada uno de los arrays cuenta con 192 caracteres (3 x 64) y se almacenan los 9 a la vez.

Una vez almacenados los datos, se envían a los leds a través de los pines reservados para tal efecto empleando la función `send_data()`.

```
void send_data (int pin_on_off,int pin_data,char * array)
{
    int i = 0;//loop variable

    ////////////RESET THE LINE//////////
    digitalWrite (pin_on_off,LOW);
    NOP();
    digitalWrite (pin_on_off,HIGH);

    for(i = 0; i < DATA_LEDS_SIZE; i++)
    {
        if(array[i] & 0x80)//
            send_l_or_0(pin_data,1);
        else
            send_l_or_0(pin_data,0);

        if(array[i] & 0x40)//
            send_l_or_0(pin_data,1);
        else
            send_l_or_0(pin_data,0);

        if(array[i] & 0x20)//
            send_l_or_0(pin_data,1);
        else
            send_l_or_0(pin_data,0);
    }
}
```

Imagen 43: función `send_data()`.

La función `send_data()` (Imagen 43) sigue el siguiente proceso:

En primer lugar, reinicia los leds, eliminando la tensión mediante el pin indicado que a su vez desactiva el transistor que desconecta los leds de la alimentación. Este proceso se hace para sincronizar el comienzo de la trama enviada a los leds, ya que, de otro modo, en ocasiones se producen errores en la interpretación de los datos enviados y por tanto del color de iluminación de los leds.

A continuación, se procesan los datos del array interpretando con máscaras la posición de cada bit de la cadena. En el caso de que el bit corresponda con un 1 se envía un 1 siguiendo las indicaciones del datasheet de los leds, anteriormente explicadas, y un 0 si corresponde un 0.

Se utilizan las máscaras: & 0x80, & 0x40, & 0x20, & 0x10, & 0x08, & 0x04, & 0x02, & 0x01; para obtener los bits de mayor a menos significativo y son procesados en ese orden.

Tabla 2: mascarar empleadas para decodificar los bits.

DECIMAL	HEXADECIMAL	BINARIO
1	0x01	0b00000001
2	0x02	0b00000010
4	0x04	0b00000100
8	0x08	0b00001000
16	0x10	0b00010000
32	0x20	0b00100000
64	0x40	0b01000000
128	0x80	0b10000000

```
void send_1_or_0(int pin_data,int state)
{
  if(state)
  {
    digitalWrite(pin_data,HIGH);//1; long time to high level, short time to low level
    NOP();NOP();NOP();NOP();NOP();NOP();

    digitalWrite(pin_data,LOW);
    NOP();NOP();
  }
  else
  {
    digitalWrite(pin_data,HIGH);//0; short time to high level, long time to low level
    NOP();NOP();

    digitalWrite(pin_data,LOW);
    NOP();NOP();NOP();NOP();NOP();NOP();
  }
}
```

Imagen 44: send_1_or_0() función.

La función, send_1_or_0 () transmite un 0 o un 1 a los leds según el método indicado en el datasheet de los leds. Este consiste en enviar un 1 manteniendo una señal larga a nivel alto y una señal más corta a nivel bajo; y un 0 manteniendo una señal corta a nivel alto y a continuación una señal más larga a nivel bajo.

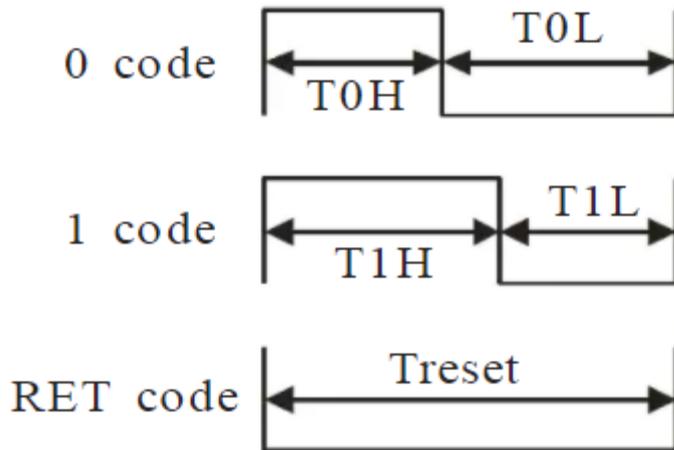


Imagen 45: protocolo indicado en datasheet de leds WS2812B

La función NOP() mantiene sin hacer nada al microcontrolador durante un ciclo de reloj. Dado que la frecuencia, definida por el datasheet del microcontrolador es de 40MHz, un ciclo de reloj tiene una duración de 25ns. Ideal para cumplir con nuestro objetivo.

Como se puede apreciar, el resultado de la compilación es positivo.

```

Compilado
El Sketch usa 8048 bytes (3%) del espacio de almacenamiento de programa. El máximo es 253952 bytes.
Las variables Globales usan 2195 bytes (26%) de la memoria dinámica, dejando 5997 bytes para las variables locales. El máximo es 8192 bytes.
  
```

Imagen 46: Resultado compilación FW.

COMPONENTES DEL PROYECTO

SELECCIÓN DE COMPONENTES Y ESQUEMAS ELECTRICOS.

Fuente de alimentación:

Para cumplir las especificaciones marcadas por el pliego de condiciones hemos seleccionado una fuente de alimentación de las siguientes características:



Imagen 47:Fuente de alimentación. Imagen obtenida de aliexpress.com

5V de ddp

Una intensidad de corriente de 5A, superior a los 4 A requeridos y que proporciona la potencia necesaria para el elevado consumo de los leds.

Medidas: 36x79x110 mm

Peso 0.165 Kg

Certificado CE

Material: aluminio

1 única salida a 12 V

Y una entrada a 110V-240V

Esta fuente de alimentación es PSU-60W con un precio de 6,95€ en la web de www.efectoled.com

Regulador:

Para cumplir las especificaciones marcadas por el pliego de condiciones hemos seleccionado un regulador de las siguientes características:



Imagen 48:regulador de tensión, obtenida de mouser.com

DC input: 5-18V

Encapsulado TO-20

VO: 4.95V-5.45V

Este regulador es un L7805 de la compañía STMicroelectronics con un precio por unidad de 0,68€ en Mouser

Resistencias:

Para cumplir las especificaciones marcadas por el pliego de condiciones hemos seleccionado unas resistencias de las siguientes características:

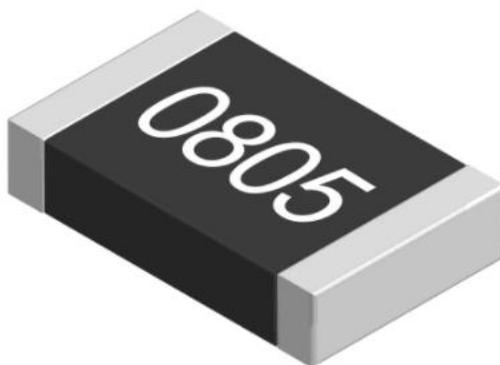


Imagen 49:resistencia 0805, obtenida de mouser.com

Encapsulado: 0805 SMD

La tolerancia de las resistencias debe ser igual o menor al 10%.

Los valores de las resistencias deben estar normalizados en la serie E12 de la normativa UNE-EN 60063:2015.

Potencia soportada 1/4W

Las resistencias utilizadas serán de la marca Vishay con un precio aproximado de 0,03€ la unidad en RS, dependiendo del valor de la misma.

Leds WS2812B:

Para cumplir las especificaciones marcadas por el pliego de condiciones hemos seleccionado los leds de las siguientes características:

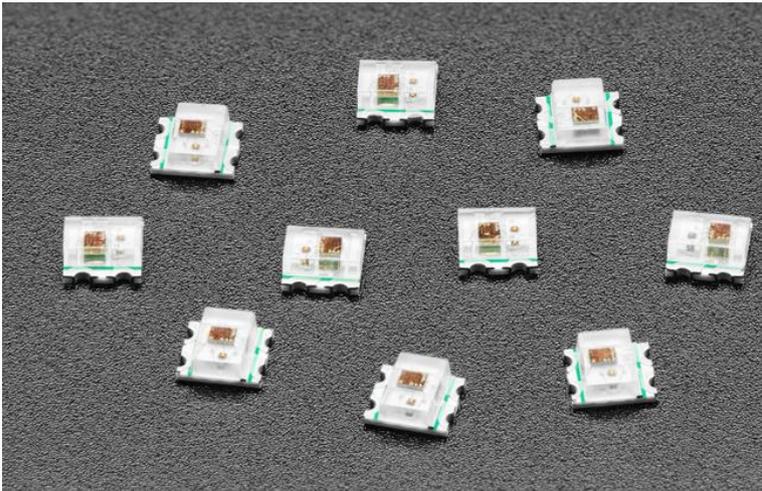


Imagen 50:pixel leds, obtenida de digikey.es

Tipo RGB inteligentes, modelo WS2812B

4 pines (I,O, VCC Y GND)

Dimensiones 5mmx5mm

VDD: 3.5V-5.3V

Los leds utilizados serán los WS2812B de la marca WorldSemi con un precio aproximado de 7.85€ el set de 100 en RS.

Microcontrolador:

Para cumplir las especificaciones marcadas por el pliego de condiciones hemos seleccionado un microcontrolador de las siguientes características:



Imagen 51: microcontrolador ESP_WROOM_32, obtenida de mouser.com

Microcontrolador ESP32

Xtensa® single-/dual-core 32-bit LX6

Wi-Fi-and-Bluetooth

Modo de bajo consumo (ultra-low-power)

448 KB ROM

Encapsulado: ESP32-pico-D4

La marca utilizada será ESP con un precio de 2,33€ en Mouser

Condensadores:

Para cumplir las especificaciones marcadas por el pliego de condiciones hemos seleccionado los condensadores de las siguientes características:



Imagen 52:condensador electrolítico, mouser.com

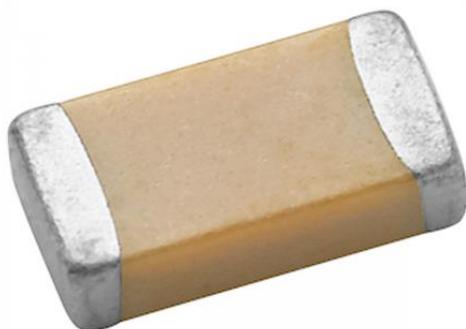


Imagen 53:condensador cerámico, digikey.es

Los valores de las capacitancias deben estar normalizados en la serie E12 de la normativa UNE-EN 60063:2015

Los de encapsulado SMD tienen un tamaño 0805

Los de encapsulado DIP tienen un paso entre pines de 2,5mm

Los condensadores utilizados serán de la marca Vishay para ambos encapsulados con un precio aproximado de 1.30 € los DIP y 0,13€ los SMD en Mouser.

Mosfet:

Para cumplir las especificaciones marcadas por el pliego de condiciones hemos seleccionado el mosfet de las siguientes características:

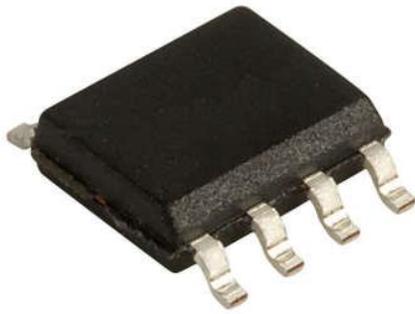


Imagen 54: integrado con 2 IRF7306, mouser.com

Doble canal P

Encapsulado: SO-08

Hemos escogido el mosfet IRF7306 de la marca Infineon con un precio en Mouser de 1.27€

Diodos:

Para cumplir las especificaciones marcadas por el pliego de condiciones hemos seleccionado diodos de las siguientes características:



Imagen 55: diodos encapsulados minimelf, mouser.com

Encapsulado: smd minimelf

De la marca Onsemi con un precio medio de 0,32€ el diodo, dependiendo de las características del mismo.

Conectores pines macho y hembra:

Utilizaremos conectores macho hembra para unir las PCBs de control entre ellas y con las PCBs de los leds. A través de ellos se transmitirán las señales de datos y la alimentación.

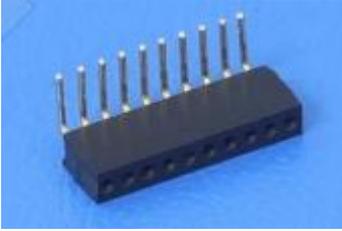


Imagen 56:pines hembra. Obtenidos de conexcon.com

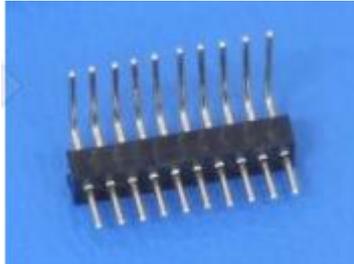
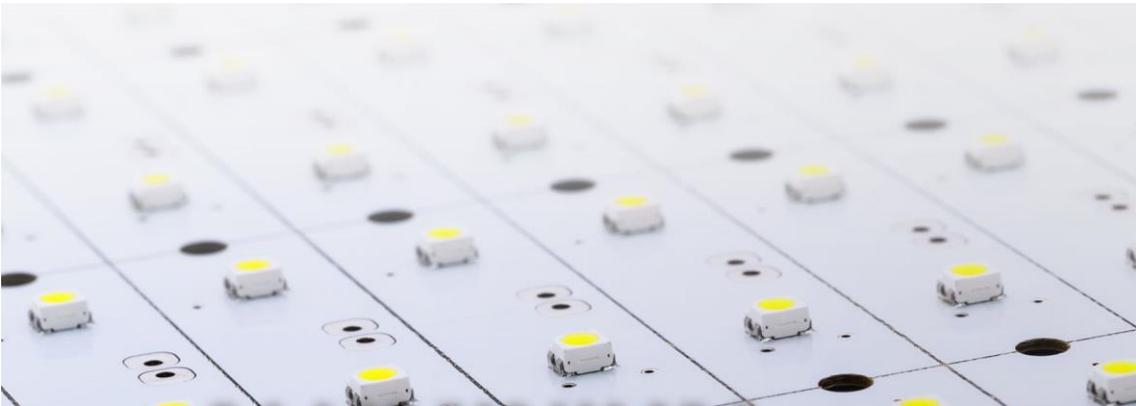


Imagen 57:pines macho, obtenidos de conexcon.com

PCBs:

Las PCBs serán PCBs en color blanco, con serigrafía en negro, material FR-4, anchura de 1,6mm y con 2 capas.



Serán compradas al proveedor JLCPCB a un precio de 15,86€ el set de ambas PCBs necesarias para el módulo, nueve de cada tipo.

La PCB de LEDs tendrá unas dimensiones de 96mmx96mm, mientras que la PCB de CONTROL tendrá unas dimensiones de 92mmx92mm.

Ambas PCBs deberán ser tropicalizadas una vez soldados los componentes.

Selección de componentes estructurales.

En cuanto a los elementos estructurales, estos no forman parte del objeto de desarrollo de este proyecto, sin embargo, el diseño de las PCBs del mismo se ha llevado a cabo teniendo en cuenta que el módulo debe estar anclado en un *housing*.

En esa línea, se ha planteado que las medidas de la PCB de leds sean de 96mm x 96mm. Dejando una distancia intra PCBs de 2mm a cada lado. Esta distancia está calculada al mm, para que la distancia entre un led de un módulo y otro módulo sea la misma que entre dos leds del mismo modulo. Sin embargo, las PCBs de control tienen una medida de 92mm, dado que estas deben ir interconectadas mediante pines macho-hembra como los de la Imagen 56, los cuales ocupan 8mm de ancho. Permitiendo así, que la distancia entre los centros de dos módulos conectados sea de 100mm. Permitiendo así formar módulos de 300mm x 300mm.

El housing que contenga el módulo, deberá tener en cuenta que las PCBs de control tienen 4 taladros de 2mm de diámetro, uno en cada esquina, a una distancia de 7mm x 7mm de cada lado. El anclaje de la PCB de leds con la PCB de control se llevará a cabo mediante la propia resistencia de los pines de conexión y con la sujeción externa de un elemento transparente, como puede ser metacrilato el cual deberá tener unas dimensiones de 296mm x 296mm e ir anclado en el housing mediante encaje.

Esquemas eléctricos.

Se han diseñado dos esquemas eléctricos, uno para cada PCB, ambos en el programa gratuito Kicad 6.

PCB de control: Imagen 58

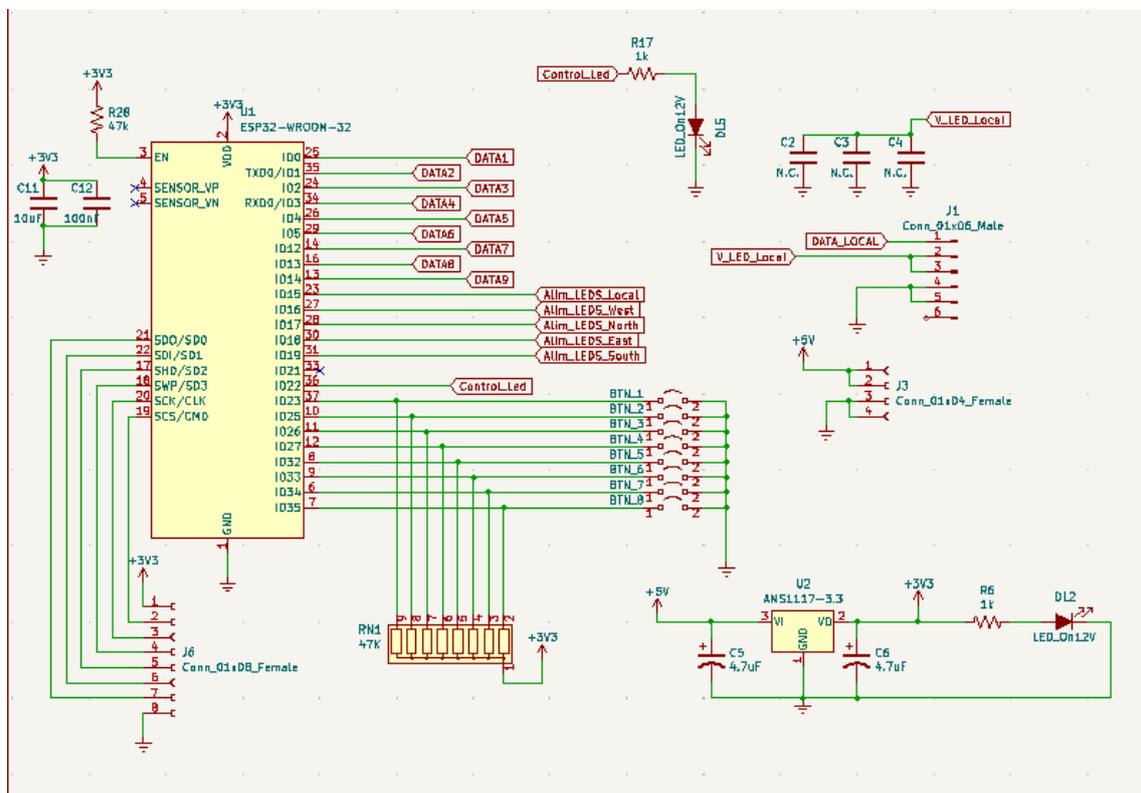


Imagen 58:esquemático PCB de control. Diseño propio.

- Se puede apreciar el microcontrolador, U1, con las entradas de los jumpers de dirección HW (BTN_1-BTN_8), puestas a pull-up mediante el array de resistencias RN1.
- Las salidas de control de alimentación y de datos: DATA1-DATA9 y Alim_Leds_Local, Alim_Leds_West, Alim_Leds_East, Alim_Leds_South, Alim_Leds_North.
- Un led de control, DL5, utilizado para labores de desarrollo.
- Un regulador de tensión, U2, de 12V a 3,3V, con sus respectivos condensadores
- Un diodo led de señalización de corriente, DL2, empleado en labores de debug.
- Condensadores de desacoplo para el microcontrolador, C11, C12.
- Condensadores para almacenar carga, que finalmente no serán utilizados, C2, C3 y C4.
- Conectores de alimentación, de conexión a la PCB leds y conector de programación.

Además, cuenta con los conectores de transmisión de datos y alimentación a las demás PCBs de control como se ha visto y explicado en las imágenes: Imagen 26, Imagen 27, Imagen 18.

En cuanto al esquema de la PCB de los leds:

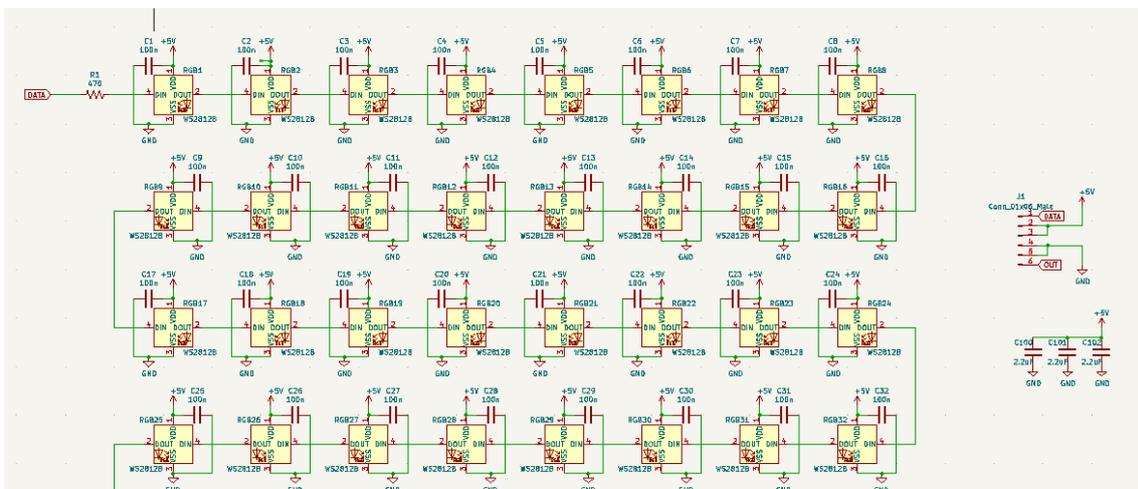


Imagen 59: Esquema PCB Leds, creación propia.

- 64 leds WS2812B con sus respectivos conectores de desacoplo. Están unidos en serie recibiendo un input de entrada por el pin 4 y devolviendo uno de salida por el pin 2.
- Conector J1, de unión con la PCB de control.
- Condensadores de estabilización de alimentación, C100, C101, C102.

Documentación adjunta.

Entre la documentación adjunta se encuentran los archivos de gerber de la pcb de control en la carpeta gerber_PCB_CONTROL.rar y los de la pcb de leds en la carpeta

gerber_PCB_LEDS.rar. Ambos gerbers pueden ser analizados con el programa gratuito KiCad 6.

También se encuentran los esquemáticos en los documentos respectivos Esquemas_PCB_LEDs.pdf (Imagen 60) y Esquemas_PCB_Control.pdf (Imagen 61)

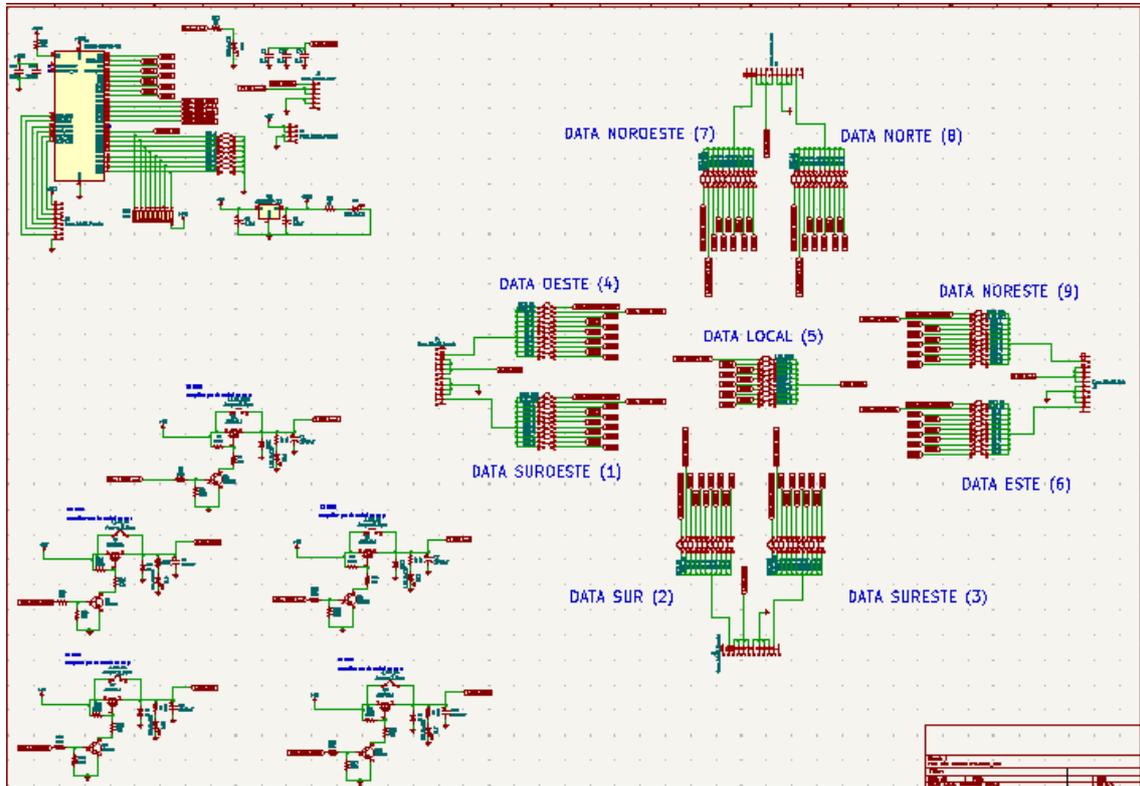


Imagen 60:Imagen del esquemático de la pcb de control. creación propia.

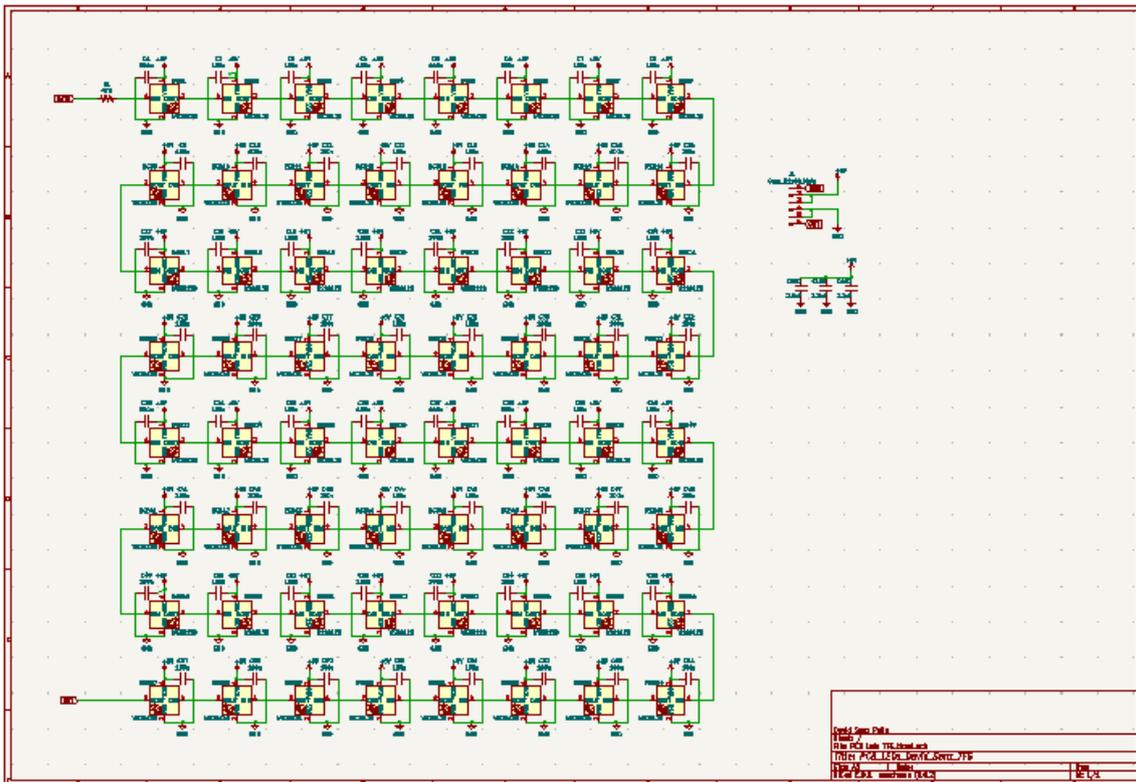


Imagen 61: Esquemático PCB de leds. creación propia.

OBJETIVO DESARROLLO SOSTENIBLE (ODS)

La tecnología siempre ha sido de utilidad para promover el avance de la sociedad, podemos ver un ejemplo de esto en la automatización de las actividades agrícolas, hecho el cual facilita el acceso a la alimentación de todo el mundo, trabajos de calidad y desarrollo personal del individuo. En esta línea, el proyecto de módulo de leds esta enfocado desde la perspectiva de avance social que cualquier proyecto tecnológico debería tener.

El principal aspecto relacionado con los objetivos de desarrollo sostenible es el ODS 11, ciudades y comunidades sostenibles, ya que el uso de este tipo de iluminación led, frente a otros tipos de iluminación, cuenta con ventajas de consumo, de eficiencia en el uso de los materiales de construcción de los leds frente a bombillas y permite que sea mas asequible el acceso a este tipo de productos fomentando de tal manera el crecimiento económico, ODS 8.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Grado en Ingeniería Electrónica Industrial y Automática

DESARROLLO DE UN SISTEMA EMBEBIDO PARA PANTALLA LED MODULAR

DOCUMENTO: PRESUPUESTO

Desarrollado por: D. David Sanz Peña

Dirigido por: D. Francisco Rodríguez Ballester

Curso 2022-2023

INTRODUCCIÓN

A continuación, se hace un planteamiento de los costes requeridos para completar el desarrollo del proyecto. Además, se añade un planteamiento económico del coste de fabricación del producto.

COSTES DE DESARROLLO

En estos costes se engloban los recursos empleados en el proyecto, tanto humanos, como costes en equipos, licencias o recursos generales.

En cuanto a los costes humanos se han empleado 300 horas de tiempo para lograr desarrollar el proyecto. Estas horas han sido repartidas en distintas actividades, entre las que destacan el tiempo dedicado al estudio del proyecto, el dedicado al desarrollo del circuito electrónico y la PCB, el tiempo dedicado a la programación del FW y el tiempo dedicado a pruebas, entre otros.

Se ha tenido en cuenta un precio medio por hora de desarrollo de 15€ brutos. Seguidamente, se desglosa el tiempo empleado y sus costes en la siguiente tabla.

Tabla 3: costes recursos humanos

COSTES DE RECURSOS HUMANOS				
Uds.	Denominación	Cantidad	Precio (€)	Total
h	Tiempo dedicado al estudio del proyecto	36	15,00 €	540,00 €
h	Tiempo dedicado al desarrollo del circuito y la PCB	68	15,00 €	1.020,00 €
h	Tiempo dedicado al desarrollo del <i>housing</i> y planos	30	15,00 €	450,00 €
h	Tiempo dedicado a la programación del FW	86	15,00 €	1.290,00 €
h	Tiempo dedicado al montaje del dispositivo	10	15,00 €	150,00 €
h	Tiempo dedicado a la redacción de documentación	20	15,00 €	300,00 €
h	Tiempo dedicado a pruebas durante el desarrollo	38	15,00 €	570,00 €
h	Tiempo dedicado a otras actividades secundarias	12	15,00 €	180,00 €
			TOTAL	4.500,00 €

A continuación, se desarrollarán los distintos puntos:

- Tiempo dedicado al estudio del proyecto: en este campo se han añadido las horas dedicadas a plantear el proyecto valorando la viabilidad económica, la utilidad como solución frente a otras alternativas y la dificultad técnica requerida para lograr cumplir las especificaciones planteadas. Además, se cubre el tiempo dedicado a hacer un plan de desarrollo que sirva como referencia inicial para el mismo.
- Tiempo dedicado al desarrollo del circuito y la PCB: este campo cubre las horas dedicadas a, como el propio título indica, el desarrollo del diseño electrónico y el desarrollo de la PCB. En este campo se añaden también, el tiempo dedicado

a la valoración de las distintas soluciones posibles, lectura de documentación como *datasheets*, el tiempo dedicado a la revisión del circuito y la revisión de los gerbers o la generación de documentación adjunta.

- Tiempo dedicado al desarrollo del housing y planos: en este campo se añade el tiempo empleado en diseñar una estructura para contener la electrónica del proyecto que sirva como carcasa del mismo. Incluye el tiempo empleado en encontrar la solución más idónea, el planteamiento del montaje y el tiempo empleado en generar documentación específica como planos.
- Tiempo dedicado a la programación del FW: en este apartado se incluye todo el tiempo de desarrollo de la programación asociada al producto. También se incluye el tiempo empleado en la lectura de documentación necesaria para el desarrollo como puede ser *application notes* del microcontrolador empleado o documentación del SDK (*software development kit*). También se incluye el tiempo empleado en acciones de depuración de código, necesarias para obtener un FW que cumpla con las necesidades planteadas.
- Tiempo dedicado al montaje del dispositivo: tiempo empleado en el montaje de los componentes con el fin de obtener un producto físico. Se incluye tiempo de soldadura de componentes electrónicos y acoplamiento con la estructura de housing.
- Tiempo dedicado a la redacción de documentación: en este apartado se incluye el tiempo dedicado a la generación de la documentación asociada al dispositivo. Desde este propio documento hasta el documento donde se define el protocolo de comunicación empleado.
- Tiempo dedicado a pruebas durante el desarrollo: se incluye el tiempo empleado en comprobar que el dispositivo cumple con las necesidades planteadas y corroborar que el funcionamiento de este es el esperado.
- Tiempo dedicado a otras actividades secundarias: en este apartado se incluye el tiempo empleado en actividades de menor relevancia, pero fundamentales para desarrollar el producto. Por ejemplo, seleccionar e instalar los distintos softwares utilizados en el desarrollo. Ej: fusión 360, Kicad, etc.

En el campo de los costes empleados en de equipos, licencias y recursos generales se va a hacer una división entre los costes que son de único pago y amortización a lo largo del tiempo y los que su coste depende directamente del tiempo en el que han sido utilizados los recursos.

Entre los elementos de único pago que se amortizan a lo largo del tiempo se incluye la compra de equipos empleados en el desarrollo, como un PC, o equipo de laboratorio, y la compra de licencias de por vida. En estos gastos, dado que podrán continuar siendo utilizados en el futuro para proyectos posteriores, se va a plantear una amortización a cinco años, y se añadirá como gasto el periodo de uso de un año. Es decir, se contabilizará como gasto el 20% del precio de compra planteando la amortización del 80% restante como un gasto para futuros proyectos donde también se empleen estos equipos o softwares.

A continuación, podemos ver el desglose de gastos de recursos generales, equipos y licencias:

Tabla 4: costes equipos, licencias y recursos generales.

COSTES DE EQUIPOS, LICENCIAS Y RECURSOS GENERALES				
Uds.	Denominación	Cantidad	Precio (€)	Total
u.	Amortización PC empleado	20%	1.500,00 €	300,00 €
u.	Amortización fuente de alimentación programable	20%	475,00 €	95,00 €
u.	Amortización osciloscopio	20%	355,00 €	71,00 €
u.	Amortización soldador y soldador de aire caliente	20%	188,85 €	37,77 €
u.	Licencia SW diseño electrónico y PCBs (Kicad)	20%	0,00 €	0,00 €
u.	Licencia SW diseño housing (FreeCad)	20%	0,00 €	0,00 €
u.	Amortización equipos de laboratorio restantes	20%	100,00 €	20,00 €
u.	Componentes electrónicos de desarrollo y test	1	100,00 €	100,00 €
u.	Material de oficina	20%	30,00 €	6,00 €
kWh	Luz	60,58	0,21 €	12,66 €
u.	Oficina	20%	0,00 €	0,00 €
km	Desplazamiento a oficina	0	0,00 €	0,00 €
			TOTAL	138,66 €

A continuación, se procede a desarrollar los diferentes costes:

- Amortización PC empleado: para el desarrollo de este proyecto se ha empleado un TOSHIBA SATELLITE PRO A50-D-1FZ cuyo coste de compra fue de 1500€. Posee un procesador Intel CORE i7 de 7th Gen un disco duro SSD de 256 GB, una memoria RAM de 8GB y una pantalla de 15,6 in.

- Amortización fuente de alimentación: se ha utilizado como soporte de electricidad en laboratorio una fuente de alimentación programable SIGLENT SPD3303X de 3 canales y potencia máxima de salida 220 W cuyo coste ha sido de 475€.
- Amortización osciloscopio: se ha empleado como soporte para poder desarrollar el producto un osciloscopio Rigol DS1054Z de 4 canales de 50MHz de ancho de banda cuyo coste ha sido de 355€.
- Amortización soldador y soldador de aire caliente: para el montaje de componentes y las pruebas en laboratorio es necesario contar con un soldador eléctrico de estaño. Además, por las características generales del proyecto, los leds utilizados deben soldarse utilizando aire caliente. Por este motivo se ha utilizado para estos fines un soldador eléctrico con pistola de aire NEW CALOX 8786D cuyo precio es de 188,85€.
- Licencia SW diseño electrónico y PCBs (Kicad): para el desarrollo de las PCBs se ha utilizado el software libre KICAD 5. El coste de esta herramienta ha sido de 0€.
- Licencia SW diseño housing (FreeCad): para el desarrollo del housing del dispositivo se ha empleado el software FREECAD. Al tratarse de un SW libre el coste de la licencia ha sido de 0€.
- Amortización equipos de laboratorio restantes: además de la fuente de alimentación de laboratorio, el soldador eléctrico y el osciloscopio es necesario contar con otras herramientas o elementos. Algunos ejemplos de estos elementos son estaño para soldar, cables cocodrilo para la fuente de alimentación, cables Dupont para conexiones estándar, un cortacables o un polímetro.
- Componentes electrónicos de desarrollo y test: para el desarrollo del producto ha sido necesario emplear una serie de componentes electrónicos como resistencias, condensadores o microcontroladores utilizados para hacer pruebas, validar el código, y probar los equipos. Este coste no puede añadirse al coste de producción ya que estos elementos han sido utilizados para el desarrollo.
- Material de oficina: incluye, desde libreta para tomar notas o hacer cálculos a bolígrafos, tijeras, etc.
- Luz: el consumo medio de luz anualizada por m² de oficina en España es de 52,5kWh/m². Como coste aproximado, teniendo en cuenta que las horas que se han empleado para el desarrollo del producto son 300 y que la jornada de oficina anual media son 2080h (40h*52 semanas). Teniendo en cuenta que la oficina empleada tiene una superficie de 8m² se puede sacar una estimación del consumo de electricidad empleado.

$$\frac{52.5\text{kWh}}{\text{m}^2} \times 8\text{m}^2 + \frac{300\text{h}}{2080\text{h}} = 60.58 \text{ kWh}$$

Teniendo en cuenta un precio medio del kWh en 2022 de 0,209€/kWh coste es el indicado en la tabla: 12.06€.

- Oficina: la oficina donde se ha desarrolla el producto ha sido una sala de 8m2 ubicada en la residencia habitual del desarrollador, por lo tanto, para el desarrollo de este proyecto se va a incluir un coste de 0€ en este concepto.

- Desplazamiento a oficina: en relación con lo comentado en el punto anterior el coste de este apartado se va a considerar de 0€.

Seguidamente, en la Tabla 5: resumen final presupuesto, se puede ver el desglose global y el coste final del desarrollo. En él se recoge la suma de los gastos de recursos humanos, en equipos y licencias y se añade además un sobrecoste del 10% de gastos globales derivados de la actividad económica, un 6% de beneficio industrial y, por último, el 21% de I.V.A.

Tabla 5: resumen final presupuesto

COSTE GLOBAL DE DESARROLLO			
Denominación		Coste (€)	
Coste de recursos humanos		4.500,00 €	
Coste de equipos, licencias y recursos generales		138,66 €	
TOTAL		4.638,66 €	
Denominación	%	Coste (€)	Sumatorio coste(€)
Gastos globales derivados de la actividad económica	10	468,87€	5102,53€
Beneficio Industrial	6	306,15€	5408,68€
I.V.A.	21	1135,82€	6544,50€
TOTAL PRESUPUESTO GENERAL		6544,50€	

El presupuesto general asciende a la cantidad de seis mil quinientos cuarenta y cuatro euros con cincuenta céntimos.

PLANTAMIENTO ECONÓMICO DEL COSTE DE FABRICACIÓN DEL PRODUCTO

En este apartado se plantea el coste de fabricar un módulo ya desarrollado sin tener en cuenta los costes de desarrollo. Se tiene en cuenta que cada módulo está compuesto por 9 submódulos que a su vez están compuestos por 9 PCBs de control y 9 PCBs de LEDs.

Estos costes que se recogen en la Tabla 6: costes de fabricación., se desglosan en costes de componentes electrónicos de carácter genérico, el coste de la fuente de alimentación, el coste de montaje, el coste de la PCB y el coste de los conectores de unión de las PCBs. Además, se tienen en cuenta costes generales por la actividad económica (10%) el beneficio industrial (6%) y el I.V.A. (21%).

Tabla 6: costes de fabricación.

COSTES DE FABRICACIÓN				
Uds.	Denominación	Cantidad	Precio (€)	Total
u.	Resistencias SMS	20	0,03 €	0,60 €
u.	Condensadores TH	36	0,40 €	14,40 €
u.	Condensadores SMD	606	0,02 €	12,12 €
u.	Diodos	35	0,15 €	5,25 €
u.	Leds de desarrollo	18	0,11 €	1,98 €
u.	Mosfet IRF7606	35	0,25 €	8,75 €
u.	Transistor BJT	35	0,24 €	8,40 €
u.	Conectores IDC macho y hembra	16	0,32 €	5,12 €
u.	Pixel LEDS WS2812B	576	0,03 €	14,40 €
u.	Microcontrolador ESP32_WROOM	1	2,33 €	2,33 €
u.	Fuente de alimentación	1	4,33 €	4,33 €
u.	PCB led	9	0,66 €	5,94 €
u.	PCB control	9	0,66 €	5,94 €
h	Tiempo de montaje	1	14 €	14,00 €
			TOTAL	99,23 €
COSTE GENERAL DE FABRICACIÓN				
Denominación	%	Coste (€)	Sumatorio coste(€)	
Gastos globales derivados de la actividad económica	10	9,92€	109,15€	
Beneficio Industrial	6	6,55€	115,70€	
I.V.A.	21	24,30€	140€	
TOTAL COSTE DE FABRICACIÓN				140€

El coste final de fabricación de cada módulo es de ciento cuarenta euros.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Grado en Ingeniería Electrónica Industrial y Automática

DESARROLLO DE UN SISTEMA EMBEBIDO PARA PANTALLA LED MODULAR

DOCUMENTO: PLIEGO DE CONDICIONES

Desarrollado por: D. David Sanz Peña

Dirigido por: D. Francisco Rodríguez Ballester

Curso 2022-2023

Contenido

1. DEFINICIÓN Y ALCANCE DEL PLIEGO	74
2. CONDICIONES Y NORMAS DE CARÁCTER GENERAL	74
3. CONDICIONES PARTICULARES	74
3.1 TÉCNICAS	75
3.2 FACULTATIVAS	77
3.3 ECONÓMICAS	78
3.4 LEGALES	78

1. DEFINICIÓN Y ALCANCE DEL PLIEGO

El objeto de este documento es definir las condiciones técnicas que deberá tener el sistema de iluminación led basado en módulos. Su aplicación por tanto se extiende a todos los sistemas de soporte, de desarrollo software y de diseño hardware, junto a otros posibles sistemas, tales como mecánicos, hidráulicos, que conforman este proyecto.

Se podrán adoptar, durante el desarrollo, nuevas e innovadoras soluciones, siempre y cuando no disminuya la calidad del producto.

2. CONDICIONES Y NORMAS DE CARÁCTER GENERAL

Este pliego de condiciones, con sus documentos, tiene carácter obligatorio al constituir un acuerdo entre las partes y deberá cumplirse durante la fabricación y hasta la terminación del proyecto.

En el ámbito nacional de España, la fabricación, desarrollo y gestión ambiental, tienen una normativa concreta especificada en distintos decretos.

Algunos de estos decretos son el Real Decreto 188/2016, del 6 de mayo, donde se especifican los requisitos para la comercialización y uso de equipos radioeléctricos. El real decreto 27/2021, del 19 de enero, donde se concreta como actuar en relación a la gestión de los residuos electrónicos, pilas o baterías. La Directiva 2011/65/UE que es de aplicación a los aparatos que necesitan corriente eléctrica o campos electromagnéticos para funcionar y que están diseñados para utilizar una tensión nominal no superior a 1000 V en corriente alterna y 1500 V en corriente continua, donde se restringe la utilización de determinadas sustancias peligrosas como el plomo, el mercurio, el cadmio o el cromo hexavalente.

Estos requerimientos legales deberán de ser llevados a cabo por el contratista y los poseedores futuros de los dispositivos, que deberán cumplir sus obligaciones.

Los responsables de tomar a cabo ciertas decisiones, así como dirigir el diseño, serán los responsables de la dirección del proyecto.

Destacar que durante el desarrollo del proyecto podrán llevarse a cabo las modificaciones necesarias siempre que estas sean aprobadas por los directores del proyecto estando en consonancia con los requerimientos de la entidad contratante.

3. CONDICIONES PARTICULARES

3.1 TÉCNICAS

Este documento hace referencia al subsistema electrónico del dispositivo pantalla modular led, los componentes estructurales del módulo serán especificados a continuación de la parte electrónica.

En términos de normativa, nos encontramos con la normativa de obligado cumplimiento anteriormente nombrada. Esta normativa se encuentra en el RD 188/2016, del 6 de mayo, la cual hace referencia a los términos de comercialización y uso del dispositivo; en el RD 186/2016, donde trata los asuntos relacionados con la compatibilidad de elementos electrónicos o el decreto 219/2013, donde define ciertas limitaciones de materiales disponibles para utilizar como pueden ser las soldaduras de estaño aleado con plomo.

En relación a la normativa aplicable y a las necesidades de uso se obtiene que utilizaremos la siguiente lista de materiales para nuestro subsistema electrónico. Debemos prestar especial atención en que los componentes que utilicemos cuenten con marcado CE, ya que de otro modo nuestro producto no podrá contar con él y es fundamental para cumplir con la normativa vigente. A continuación, mostraremos la lista de componentes requeridos.

SUBSISTEMA ELECTRÓNICO

-Fuente de alimentación, que cuente con al menos estos requisitos:

Una salida de corriente.

Dimensiones menores a 40x80x110 mm.

Peso inferior a 200g.

Potencia ≥ 60 W

Ddp de 12 V

Una corriente suministrada mínima de 4A para satisfacer las necesidades elevadas de corriente de los leds y también al circuito de control.

Marcado CE

Entrada a 230V

-Regulador, que cuente con al menos estos requisitos:

Encapsulado TO-20

DC Input: 12V

VO: 3.3V

-Resistencias, que cuenten con al menos estos requisitos:

Encapsulado: 0805 SMD

La tolerancia de las resistencias debe ser igual o menor al 10%.

Los valores de las resistencias deben estar normalizados en la serie E12 de la normativa UNE-EN 60063:2015.

Potencia soportada 1/4W

-LEDS, que cuenten con al menos estos requisitos:

Tipo RGB inteligentes, modelo WS2812B

4 pines (I,O, VCC Y GND)

Dimensiones 5mmx5mm

VDD: 3.3V

-Microcontrolador, que cuente con al menos estos requisitos:

Microcontrolador ESP32

Procesador: Xtensa® single-/dual-core 32-bit LX6

Módulos Wi-Fi-and-Bluetooth

Modo de bajo consumo (ultra-low-power)

Memoria: 448 KB ROM

Frecuencia de reloj: 40 MHz

-Condensadores, que cuente con al menos estos requisitos:

Los valores de las capacitancias deben estar normalizados en la serie E12 de la normativa UNE-EN 60063:2015

Los de encapsulado SMD deben tener un tamaño 0805

Los de encapsulado DIP deben tener un paso entre pines de 2,5mm

-Diodos mosfet, que cuenten con al menos estos requisitos:

De canal P

Encapsulado: smd SO-08

-Diodos, que cuente con al menos estos requisitos:

Encapsulado smd minimelf

-Conectores, que cuente con al menos estos requisitos:

El conector que une las dos PCBs, PCB_CONTROL con la PCB_LEDS. Y que una las PCB_CONTROL entre sí.

No se permite la utilización de cableado interno en el módulo.

De tipo IDC

-Microswitch, que cuente con al menos estos requisitos:

8 vías

Conexión SIP

-**PCBs**, que cuente con al menos estos requisitos:

Deberán existir al menos dos PCBs, una destinada a la alimentación y el control del dispositivo y otra que sea la que contenga la matriz de leds WS2812B.

Las PCBs serán PCBs en color blanco, con serigrafía en negro, material FR-4, anchura de 1,6mm y con 2 capas.

Estas deberán ser tropicalizadas para mayor protección contra los agentes climáticos.

Una vez implementados los componentes sobre las PCBs deberá someterse a un test de prueba para garantizar la funcionalidad del sistema y la funcionalidad de los componentes.

En la PCB de control deberán instalarse aquellos componentes encargados de la funcionalidad del módulo, tales como microcontroladores, microswitch, regulador, y la conexión con la fuente de alimentación.

En la PCB de la matriz de leds deberán estar fundamentalmente los LEDs, y un conector con la PCB de control a través de la cual enviar la corriente a los leds y un bus de datos.

SUBSISTEMA DE SOPORTE

Para el subsistema de soporte utilizaremos elementos diseñados con las siguientes características.

El peso del conjunto soporte no debe superar los 500g de peso.

Todas las piezas estructurales deberán ser fabricadas en aluminio por un proveedor con certificado CE.

Deberá poseer en la parte frontal un elemento de metacrilato que proteja el conjunto de la intemperie. Este elemento también deberá poseer el certificado CE de cumplimiento de las normas euro comunitarias.

Los planos empleados para definir el diseño deberán cumplir con la normativa vigente, tal normativa se encuentra en los siguientes documentos: UNE 1-027-95, UNE 1-032-82 ISO 128, UNE EN ISO 128-20:1996, UNE 1-135-89, UNE 1166-1, UNE-EN ISO 3098-0, UNE-EN ISO 5455, UNE-EN ISO 5457, UNE-EN ISO 6433, UNE-EN ISO 7200.

Además, deberá contar con la tornillería apropiada, siempre con elementos estándar que cumplan con la normativa vigente mostrada.

3.2 FACULTATIVAS

El director del proyecto será David Sanz, quien será el encargado de que el proyecto sea entregado conforme a lo acordado en este documento. Será él quien dirigirá los

distintos ámbitos de este tales como la programación de las tareas, la revisión de las mismas y asegurarse que todos los materiales estén en coherencia con lo especificado en el pliego de condiciones.

Ante cualquier modificación llevada a cabo en el desarrollo del proyecto, esta deberá ser apuntada y aclarada en el libro de órdenes, donde además se recogerán todos aquellos detalles que puedan ser relevantes para el mismo.

3.2 ECONOMICAS

El coste por unidad de panel no deberá superar los 100€ y deberán cumplirse las normas en cuanto a derechos laborales para los trabajadores cuyos derechos deberán ser respetados según convenio.

El plazo para la entrega de los paneles será de 3 meses desde la fecha de adjudicación del proyecto, estimando un tiempo de desarrollo de 300h, inquiriendo en una falta que deberá ser subsanada con una penalización en caso de implemento de los plazos.

Esta penalización vendrá marcada por una reducción en el coste de un 2% del valor pactado si los elementos son entregados durante el mes siguiente al vencimiento del plazo. En caso de superar ese límite se agravara un descuento del 5% por cada mes de retraso con un máximo de un 12% y 3 meses de retraso, tras lo cual será suspendida la recepción y el contratista deberá indemnizar al contratante por daños y perjuicios con una cuantía igual al 30% del valor del proyecto.

En caso de entrega de un proyecto que no cumpla las condiciones especificadas y acordadas en el presente documento, se incurrirá en un proceso judicial que de resolverse en contra del contratista este deberá indemnizar a la parte contratante con hasta el 30% del valor del proyecto.

3.3 LEGALES

La adjudicación del proyecto será entregada por concurso público al mejor proyecto que reúna mejor las condiciones priorizando en primer lugar el cumplimiento de las especificaciones, la calidad de los materiales y del producto final juzgado por un comité de expertos; el coste del producto y el porcentaje en el que el producto es *ecofriendly* y cumple con los objetivos de desarrollo sostenible en última circunstancia.

El contrato del proyecto será firmado ante notario y de obligado cumplimiento bajo las penalizaciones anteriormente mencionadas.

A la hora de entregar el proyecto, se deberán adjuntar los documentos requeridos para su ejecución como son la Memoria, los planos, el presupuesto y el presente documento. Estos deberán entregarse tanto en formato físico como digital y si ambas partes están conformes una vez culminada la entrega se firmará ante notario que el proyecto ha sido resuelto según lo acordado por el contratista.