UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

GRADO EN INGENIERÍA ELECTRÓNICA Y AUTOMÁTICA INDUSTRIAL




# "SISTEMA DE GESTIÓN DE ALMACENAMIENTO BASADO EN UN SISTEMA PICK-TO-LIGHT"

TRABAJO FIN DE GRADO

Enero-2022

AUTOR: Rafael Brotons Tolsa

DIRECTOR/ES: Roberto Gutiérrez Mazón

ÍNDICE

**AGRADECIMIENTOS** **1**

**1 INTRODUCCIÓN** **2**

1.1 INTRODUCCIÓN A LOS SISTEMAS PICK-TO-LIGHT. 2

1.2 TIPOS DE SISTEMAS PICK-TO-LIGHT. 4

1.3 OBJETIVOS MARCADOS EN EL TFG. 5

**2 IMPLEMENTACIÓN HARDWARE DEL SISTEMA** **6**

2.1 MICROPROCESADOR ESP32. 6

2.1.1 CARACTERÍSTICAS PRINCIPALES. 7

2.3 OTROS COMPONENTES. 9

2.4 DIAGRAMA DE BLOQUES. 11

**3 CREACIÓN SISTEMA.** **12**

3.1 QUINTA CAPA DEL SISTEMA (CLIENTES). 12

3.2 CUARTA CAPA DEL SISTEMA(SERVIDOR). 12

3.2.1 APLICACIONES UTILIZADAS 12

3.3 TERCERA CAPA DEL SISTEMA (MENSAJE JSON). 12

3.3.1 ESTRUCTURA JSON. 13

JSON matrices 14

3.4 SEGUNDA CAPA DEL SISTEMA (WLED).[5] 15

3.4.1 SOFTWARE WLED. 15

3.4.1.1 CARACTERÍSTICAS DE WLED. 16

3.4.2 DESCARGA. 22

3.4.3 FLASHEAR EL WLED/ESP32. 23

3.4.4 CONFIGURACIONES WLED. 24

3.4.5 IMPLEMENTACIÓN DEL SISTEMA. 33

3.4.5.1 GENERACIÓN DE MENSAJE JSON. 33

3.4.2.2 WLED 35

**4 PUESTA EN MARCHA DEL SISTEMA. 41**

4.1 CREACIÓN DE DIFERENTES ENTORNOS DE IMPLEMENTACIÓN. 41

4.2 ESCALABILIDAD DEL SISTEMA. 44

4.3 ESTABILIDAD DEL SISTEMA ANTE FALLAS DE HARDWARE. 45

4.4 LIMITACIONES DEL SISTEMA. 45

**5 CONCLUSIONES Y LÍNEAS FUTURAS. 47**

5.1 CONCLUSIONES. 47

5.2 LÍNEAS FUTURAS. 47

**6 BIBLIOGRAFÍA 48**

# AGRADECIMIENTOS

En primer lugar, agradecer a mi tutor de prácticas Roberto Gutiérrez Mazón por toda su ayuda durante todo el periodo de la creación del proyecto TFG.

Por supuesto agradecer a APLISEIN, la empresa donde he realizado las prácticas y donde he podido tener la posibilidad de realizar el desarrollo del TFG, más aparte de enseñarme a la programación en C++ al realizar proyectos.

# 1 INTRODUCCIÓN

## 1.1 INTRODUCCIÓN A LOS SISTEMAS PICK-TO-LIGHT.


Figura 1.1 Ejemplo de pick-to-light

Los almacenes dependen de sistemas de picking para recuperar artículos de los estantes y moverlos a las áreas apropiadas de embalaje y envío. La selección de almacén es una función vital, pero también puede contener una gran ineficiencia. Algunos almacenes están optimizando el proceso de picking con un sistema de picking llamado pick-to-light como el de la figura 1.1.

La técnica de pick-to-light es un método cuyo objetivo es guiar a los operarios, de forma visual hacia las ubicaciones exactas de los productos dentro del almacén, o donde se puede depositar dichos productos, para la preparación de pedidos.

Se basa en que cada posición posee una luz o un grupo de luces, la cual indican la dicha posición y con un botón u otra tecnología para confirmar cuando se han recogido y depositado una de esas unidades.

El operario lee el código de barras del pedido en cuestión y automáticamente se iluminan las posiciones ya comentadas. Se trata de un sistema claro, sencillo e intuitivo, que permite gestionar los pedidos con gran rapidez y exactitud.

Las luces están en cada unidad de almacenamiento. Estas unidades pueden ser bolsas reutilizables, estantes o contenedores de envío completos. Cada unidad tiene al menos dos luces LED de diferentes colores y un código de barras.

Dependiendo del sistema que quieran implantar, habrá un grupo de colores u otros de igual forma para detectar que se ha cogido un artículo o se ha dejado un artículo.

Toda la iluminación se adapta a las necesidades de la organización. Una vez que el empleado llega a la unidad correcta, generalmente hay un código de barras para escanear, que actualiza el sistema, para saber que se ha eliminado el elemento deseado.

En una empresa donde hay muchos almacenes la contratación de trabajadores temporales y una rotación regular es muy común. La capacitación de nuevos empleados requiere tiempo, esfuerzo y cuesta dinero. Un sistema de selección basado en la luz es más fácil y visual para prácticamente cualquier persona para enseñar y aprender.

Tener un sistema de stock que está conectado directamente con las estanterías donde se realizan los pedidos, permite una selección más rápida y actualizaciones en tiempo real. Cuando se necesita depositar algún producto para la realización del pedido, mediante indicaciones de luz se indicará inmediatamente la posición de la caja donde se está realizando dicho pedido.

También otra ventaja es que existe una reducción de tiempo, dado que el operario no tiene que buscar a través del almacén.

Pick-to-Light está directamente conectado con el inventario para garantizar la precisión, pero también para proporcionar información valiosa para mejorar aún más el rendimiento de las operaciones de almacén a lo largo del tiempo.

Los empleados que tienen que caminar de un lado a otro a través de las estanterías del almacén solo para imprimir una nueva lista y comenzar de cero pueden generar tiempo de inactividad, caminatas innecesarias.

Pick-to-light permite que los trabajadores se organicen en zonas. Una vez que los artículos están listos para ser recogidos y cuando se completa el pedido comienzan de nuevo.

## 1.2 TIPOS DE SISTEMAS PICK-TO-LIGHT.

Los tipos de Pick-to-light dependen de las necesidades o de la tecnología que utilicen para el picking.

Figura 1.2 Ejemplo de Pick-to-light

En principio el básico estaría compuesto de uno o dos ledes y un display que indica el número de unidades a coger o entregar como en la figura 1.2.

También dispone de un código QR donde determina el picking del producto es decir las unidades que ha cogido o ha depositado.

He de decir que el pick-to-light se puede crear de cualquier manera mientras satisfaga las necesidades de la empresa.

En mi caso, el tipo seleccionado para el Pick-to-light tratará de una tira de ledes donde mediante un código QR determinara el producto y se le otorgarán 5 ledes para el uso de cada caja, ya que en la misma caja pueden depositar hasta 4 trabajadores un producto, de la forma que cada uno de los trabajadores se identificara con un color, esto nos permite ser más veloces.

## 1.3 OBJETIVOS MARCADOS EN EL TFG.

- El objetivo de este proyecto es el diseño y desarrollo de un proceso Pick-to-light optimizado cuyo método mejora la eficiencia de la ubicación de los artículos en el almacén.
- Una gran cantidad de artículos puede dificultar la visibilidad de donde están ubicados, pero utilizando el método de Pick-to-light facilita la identificación de donde está el artículo deseado o de donde se debe depositar.
- Una prioridad de este proyecto es ser lo más económico posible teniendo en cuenta toda la instalación de hardware y software, para la transferencia de datos se intenta trabajar con software libre y plataformas abiertas.
- Otro objetivo importante es la escalabilidad del sistema ya que este sistema puede ser ampliado en naves industriales más grandes, o con los elementos en el almacén, los elementos seleccionados permiten ampliar el sistema sin ningún problema .

# 2 IMPLEMENTACIÓN HARDWARE DEL SISTEMA

## 2.1 MICROPROCESADOR ESP32.

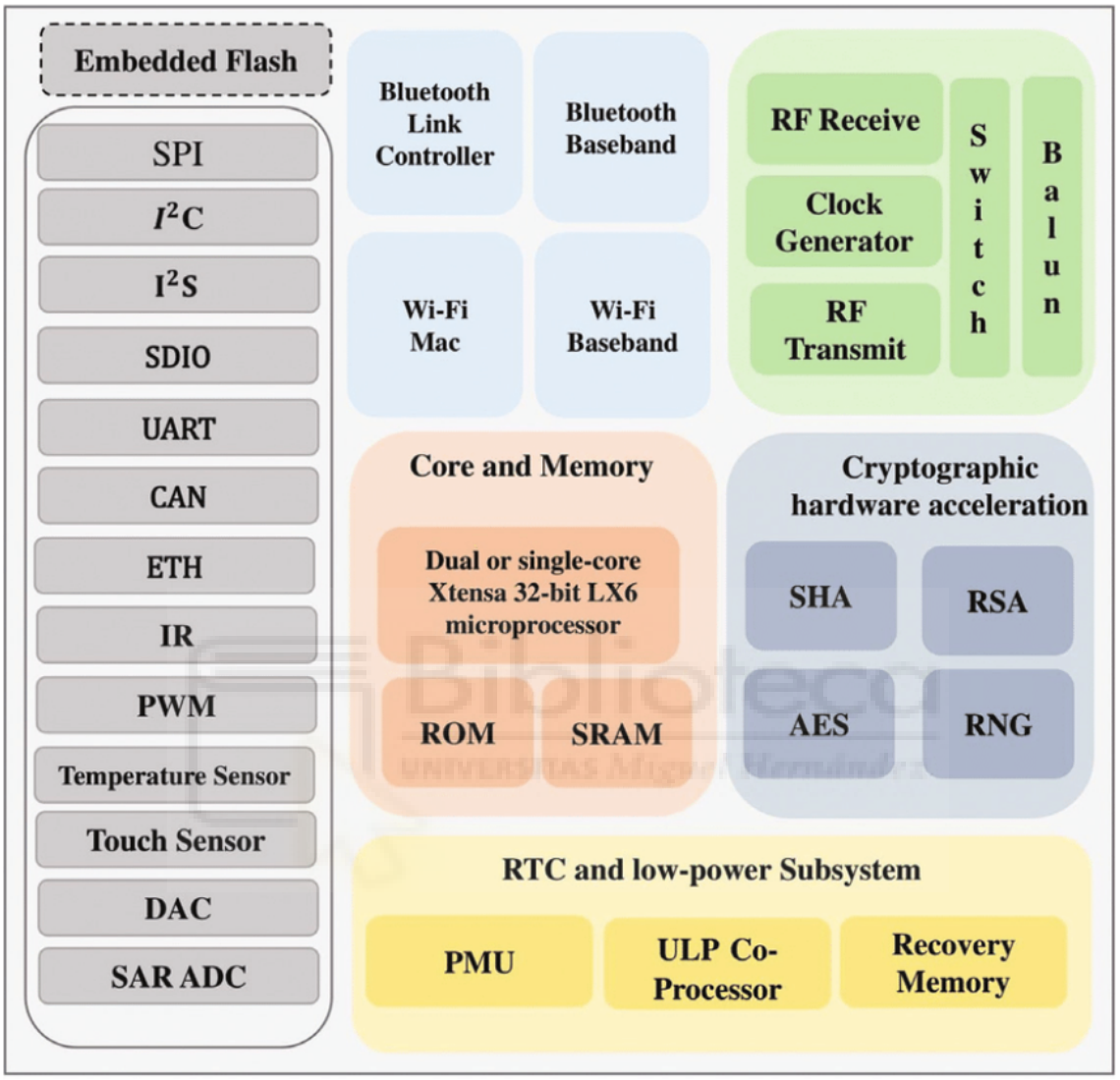


Figura 2.1 Periféricos del ESP32.

ESP32 es una serie de microcontrolador de sistema en un chip de bajo costo y bajo consumo, con Wi-Fi integrado y Bluetooth de modo dual. La serie ESP32 emplea un microprocesador Tensilica Xtensa LX6 en variaciones de doble núcleo y de un solo núcleo e incluye conexión wifi, amplificador de potencia, amplificador de recepción de bajo ruido, filtros y módulos de administración de energía, ESP32 es un sucesor del microcontrolador ESP8266.

### 2.1.1 CARACTERÍSTICAS PRINCIPALES.

En este apartado vamos a especificar las características más importantes del ESP32 como las conexiones inalámbricas y los distintos periféricos[1].

-Procesador: CPU: Xtensa® Dual-Core LX6 de 32 bits.

-Memoria ROM: 448 Kbytes.

-Reloj: 240 MHz.

-Memoria RAM: 520 Kbytes.

-Memoria flash: 4 MB.

Conectividad inalámbrica:

-Wi-Fi: 802.11 b/g/n.

-Bluetooth: v4.2 BR/EDR y BLE.

Las versiones 802.11b y 802.11g utilizan la banda ISM de 2,4 GHz.

Pueden sufrir interferencias con electrodomésticos tan comunes como el microondas, el horno o con dispositivos Bluetooth. Es por eso que deben controlar dicha susceptibilidad a las interferencias mediante métodos de señalización de espectro ensanchado por secuencia directa (DSSS) y de multiplexación por división de frecuencia ortogonal (OFDM).

La versión 802.11a utiliza la banda U-NII de 5 GHz

Ofrece al menos 23 canales que no se superponen en lugar de la banda de frecuencia ISM de 2,4 GHz que ofrece sólo tres canales que no se superponen.

Periféricos:

Los periféricos están representados en la figura 2.1 y a continuación se nombrará cada uno de ellos.

-34 × GPIO programables.

-SAR ADC (Conversor analógico/digital) de 12 bits hasta 18 canales.

-DAC (Conversor digital/analógico) de 2 × 8 bits.

-10 × sensores táctiles (GPIO de detección capacitiva).

- 4 × SPI.

-2 × interfaces I²S.

-2 × interfaces I²C.

-3 × UART Controlador de host SD / SDIO / CE-ATA / MMC / eMMC.

-Controlador esclavo SDIO / SPI.

-Interfaz Ethernet MAC con DMA dedicado y compatibilidad con el protocolo de tiempo de precisión IEEE 1588.

-CAN bus 2.0.

-PWM del motor.

-LED PWM (hasta 16 canales).

-Sensor de efecto Hall.

-Preamplificador analógico de potencia ultra baja.

A continuación en la figura 2.2 podremos ver todas las posibilidades que tenemos en nuestro ESP32, es decir cada pin para que está configurado para trabajar como para la transmisión SPI,I2c,CLK,VIN,GND y más configuraciones.

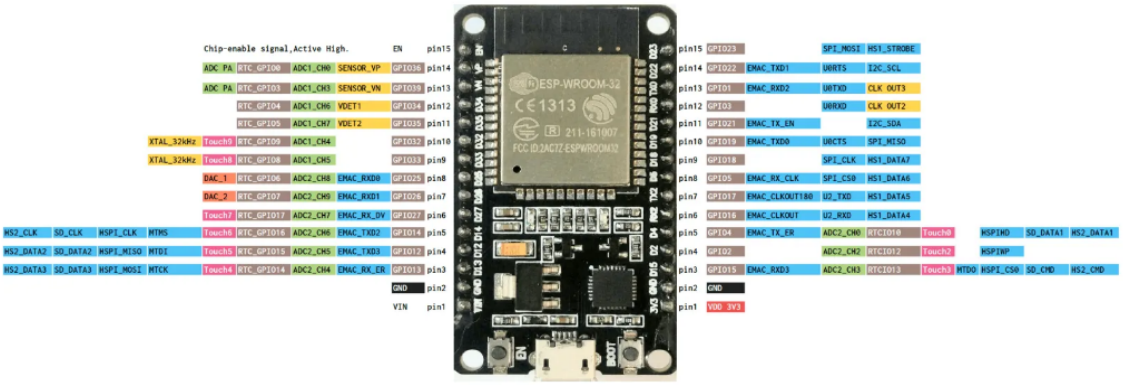


Figura 2.2 Pinout del ESP32.

## 2.3 OTROS COMPONENTES.

-Fuente de alimentación [2].

Tipo: Baiyouli Fuente de Alimentación Conmutada AC 110/220V a DC 5V 30A 150W Transformador Convertidor para Pantalla LED como se muestra en la figura 2.3.

Fuente de alimentación conmutada, entrada AC 100V-120V / 200-240V, salida DC 5V 30A 150W.

Proporciona mucha protección, alta eficiencia y alta confiabilidad. Tales como protecciones automáticas: protección contra sobrecarga, protección contra sobretensión, sobretemperatura y corriente de entrada.


Figura 2.3 Fuente de alimentación.

-Tiras de ledes [3].

Tipo: WS2812B /300 píxeles /60 ledes, DC 5V.

Cada LED se controla por separado y se puede configurar individualmente en cualquier color deseado , la tiras de ledes serán como en la figura 2.4.

IP67 a prueba de agua: la cinta de luz está cubierta con una funda protectora de silicona, que puede protegerla bien y facilitar su limpieza.

Cortable: la tira se puede cortar entre cada 1 Led sin dañar el resto de ledes, si se quiere acortar o alargar, también se puede volver a conectar mediante los conectores suministrados.

Seguro de usar: el voltaje de trabajo de 5 V CC es extremadamente bajo. Es tangible y seguro para los niños. Muy adecuado para iluminación y decoración de interiores y exteriores.

Figura 2.4 Tiras led.

## 2.4 DIAGRAMA DE BLOQUES.

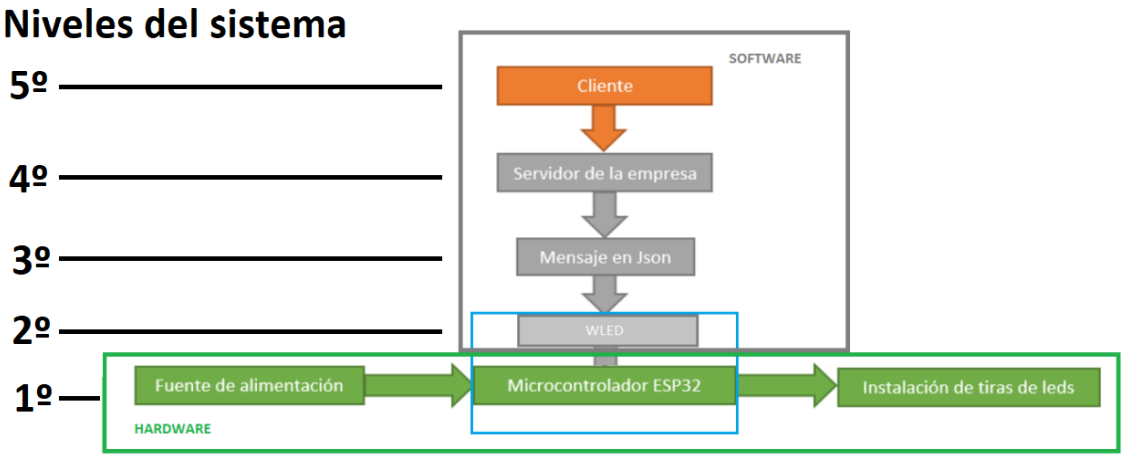


Figura 2.5 Diagrama de bloques.

El diagrama de bloques del sistema que podemos observar en la figura 2.5 consta de las partes el Hardware y el Software.

-Quinto nivel del sistema, tenemos los clientes que serán los realicen los pedidos mediante un servidor Web.

-Cuarto nivel del sistema, tenemos los servidores de la empresa, estos servidores almacenan la información leída de los clientes, normalmente son bases de datos y otras aplicaciones para convertir la información de las bases de datos y enviarla.

-Tercer nivel del sistema, tenemos el mensaje Json, este mensaje será enviado desde nuestro servidor hasta el microprocesador (ESP32).

-Segundo nivel del sistema, tenemos el programa WLED que estará dentro del microcontrolador(ESP32), este programa identifica mediante el mensaje Json que segmentos de los ledes tiene que encender.

-Primer nivel de la capa tenemos el Hardware donde consta de una fuente de alimentación, el microcontrolador(ESP32) y los dispositivos de iluminación las tiras de ledes.

# 3 CREACIÓN SISTEMA.

## 3.1 QUINTA CAPA DEL SISTEMA (CLIENTES).

En la quinta capa del sistema se encuentran los clientes donde mediante una página web podrán solicitar pedidos, cuyos productos están registrados en la base de datos de la empresa.

## 3.2 CUARTA CAPA DEL SISTEMA(SERVIDOR).

Una vez el servidor recibe la información del cliente solicitando una cantidad de productos, verifica que se encuentran en la base de datos.

### 3.2.1 APLICACIONES UTILIZADAS

La aplicación utilizada en el servidor para la base de datos es MySQL.

MySQL es un sistema de gestión de bases de datos de código abierto.

Código abierto significa que es posible para cualquier persona usarlo y modificarlo. Cualquier persona puede bajar el código fuente de MySQL y usarlo sin pagar. Cualquier interesado puede estudiar el código fuente y ajustarlo a sus necesidades.

MySQL es un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido.

## 3.3 TERCERA CAPA DEL SISTEMA (MENSAJE JSON).

Una vez el servidor localiza en la base de datos los productos mediante una aplicación, relaciona dicho producto con el lugar donde se debe de depositar, dicha caja está localizada por un segmento de ledes que tienen una enumeración [4].

Para transmitir la información de apagado o encendido  al programa WLED se envía esa información mediante un mensaje JSON.

JSON es un sistema de código abierto por ese motivo se ha decidido utilizar este sistema.

### 3.3.1 ESTRUCTURA JSON.

JSON es un formato de texto estándar para el intercambio de datos, donde se puede guardar información para luego usar los datos y darle una funcionalidad al archivo, se emplea habitualmente en entornos donde el tamaño del flujo de datos entre cliente y servidor es de vital importancia.

JSON sigue la sintaxis de objeto de JavaScript.

El tipo de variables que se pueden utilizar son strings, integer, arrays, booleans.

A un bajo nivel un JSON consta de tres componentes: 1º el nombre, 2º los dos puntos y 3º el valor.

Ejemplo Básico:


Figura 3.1 Ejemplo de JSON.

En este ejemplo de la figura 3.1 vemos que tenemos 5 campos, los dos primeros campos solamente tienen un valor, en el caso del campo ubicación tenemos una array con dos valores, uniforme contiene un objeto con 3 campos y cada uno un valor y por último tenemos jugadores que contiene una array de objetos donde cada objeto tiene 3 campos con 3 valores.

Los datos JSON - Un nombre y un valor.

Un par nombre / valor consiste en un nombre de campo (entre comillas), seguido de dos puntos, seguido por un valor como el la figura 3.2.

```
"firstName":"John"
```

Figura 3.2  Como se escribe un valor a un nombre.

**Los valores de JSON pueden ser:**

- Un número (entero o de coma flotante).
- Una cadena (entre comillas dobles).
- Un booleano (verdadero o falso).
- Una matriz (entre corchetes).
- Un objeto (entre llaves).
- nulo.

**Objetos JSON:**

- Se escriben dentro de llaves como en la figura 3.3.
- Al igual que JavaScript, objetos JSON pueden contener nombre múltiple / valores pares.

```
{"firstName":"John", "lastName":"Doe"}
```

Figura 3.3 Objetos JSON.

**JSON matrices:**

- Los arrays JSON se escriben entre corchetes como el la figura 3.4.
- Al igual que JavaScript, una matriz JSON puede contener varios objetos.

```
"employees":[
    {"firstName":"John", "lastName":"Doe"},
    {"firstName":"Anna", "lastName":"Smith"},
    {"firstName":"Peter","lastName":"Jones"}
]
```

Figura 3.4 Matrices JSON.

En el ejemplo anterior, el objeto "employees" es una matriz que contiene tres objetos. Cada objeto es un registro de una persona (con un nombre y un apellido).

Los límites de JSON aunque es un formato flexible con el que es fácil trabajar en muchos lenguajes de programación, hay algunos inconvenientes a tener en cuenta:

-Sólo un tipo de número: este límite significa simplemente que no se pueden aprovechar los diferentes y variados tipos de números disponibles en muchos lenguajes de programación.

-Sin comentarios: esto hace imposible anotar los campos en línea, lo que requiere documentación adicional y aumenta la probabilidad de malentendidos.

## 3.4 SEGUNDA CAPA DEL SISTEMA (WLED).[5]

En la segunda capa nos encontramos el Software WLED que se encuentra dentro del microcontrolador (ESP32), en esta capa se recibe el mensaje JSON enviado por el servidor hacia el programa WLED mediante la dirección IP donde identifica que segmentos de ledes quiere encender.

### 3.4.1 SOFTWARE WLED.

WLED es un programa dedicado al control de ledes, donde nos da la posibilidad de controlarlos de diferentes maneras y admite varios formatos para el intercambio de datos se puede tener un control de brillo como de transición de la iluminación, el

tratamiento de ledes individualmente como por segmentos y ofrece una gran cantidad de efectos para poder visualizar que todos los ledes funcionan perfectamente.

#### 3.4.1.1 CARACTERÍSTICAS DE WLED.

Características:

- Biblioteca WS2812FX integrada para más de 100 efectos especiales.
- Interfaz de usuario moderna con controles de color, efectos y segmentos.
- Segmentos para establecer diferentes efectos y colores en partes de los LED.
- Página de configuración: configuración a través de la red.
- Modo de punto de acceso y estación: AP automático a prueba de fallas.
- Hasta 10 salidas LED por instancia.
- Soporte para tiras RGBW.
- Hasta 250 ajustes preestablecidos de usuario para guardar y cargar colores efectos fácilmente, admite ciclos a través de ellos.
- Límite de brillo automático configurable para un funcionamiento más seguro.

Interfaces de control de luz compatibles:

- Aplicación WLED para Android e iOS.
- API de solicitud JSON y HTTP.
- MQTT.

MQTT es un protocolo de red de publicación y suscripción que transporta mensajes entre dispositivos. El protocolo generalmente se ejecuta sobre TCP / IP.

MQTT se ejecuta sobre TCP/IP utilizando una topología PUSH/SUBSCRIBE. En la arquitectura MQTT, existen dos tipos de sistemas: clientes y brókeres. Un bróker es el servidor con el que se comunican los clientes: recibe comunicaciones de unos y se las envía a otros. Los clientes no se comunican directamente entre sí, sino que se conectan con el bróker. Cada cliente puede ser un editor, un suscriptor o ambos.

También MQTT es un protocolo controlado por eventos, donde no hay transmisión de datos periódica o continua. Así se mantiene el volumen de transmisión al mínimo. Un cliente sólo publica cuando hay información para enviar, y un bróker sólo envía información a los suscriptores cuando llegan nuevos datos.

Otra forma en que MQTT minimiza sus transmisiones es con un tamaño de mensaje pequeño y bien definido. Cada mensaje tiene un encabezado fijo de apenas 2 bytes. Se puede utilizar un encabezado opcional, pero eso incrementa el tamaño del mensaje. La carga útil del mensaje está limitada a únicamente 256 MB. Tres niveles diferentes de calidad de servicio (QoS) permiten a los diseñadores de redes elegir entre minimizar la transmisión de datos y maximizar la fiabilidad.

- QoS 0: ofrece la cantidad mínima de transmisión de datos. Con este nivel, cada mensaje se entrega a un suscriptor una vez, sin confirmación, por lo que no hay forma de saber si los suscriptores recibieron el mensaje. Este método a veces se denomina “lanzar y olvidar” o “una entrega como máximo”. Debido a que este nivel asume que la entrega está completa, los mensajes no se almacenan para entregarlos a los clientes desconectados que luego se vuelven a conectar.

- QoS 1: el bróker intenta entregar el mensaje y, luego, espera una respuesta de confirmación del suscriptor. Si no se recibe una confirmación dentro de un período de tiempo especificado, el mensaje se envía de nuevo. Usando este método, el suscriptor puede recibir el mensaje más de una vez si el bróker no recibe el acuse de recibo del suscriptor a tiempo. Esto se suele denominar “entregado al menos una vez”.

- QoS 2: el cliente y el bróker utilizan un protocolo de enlace de cuatro pasos para garantizar no sólo que el mensaje se reciba, sino que lo haga una única vez. También se conoce como “entrega exactamente una vez”.

QoS 0 puede ser la mejor opción para situaciones donde las comunicaciones son fiables pero limitadas. En aquellos casos en que las comunicaciones no sean fiables, pero donde las conexiones tampoco gozan de recursos limitados, QoS 2 sería la mejor opción. QoS 1 proporciona una solución a caballo entre ambos mundos, pero requiere que la aplicación que recibe los datos sepa cómo gestionar los duplicados.

Tanto en QoS 1 como en QoS 2, los mensajes se guardan o se ponen en cola para los clientes que están desconectados y que tienen una sesión persistente establecida. Estos mensajes se envían (de acuerdo con el nivel de QoS apropiado) una vez que el cliente vuelve a conectarse.

- Blynk IoT

Blynk es la plataforma de IoT más popular para conectar dispositivos a la nube, diseñar aplicaciones para controlarlos y supervisarlos de forma remota, y administrar miles de productos implementados. El software Blynk ayuda a individuos y organizaciones a avanzar fluidamente desde un prototipo de un producto conectado hasta su lanzamiento comercial.

Puede usar la nube IoT Blynk gratuita para controlar sus luces WLED con la aplicación Blynk para Android y iOS.

- E1.31, Art-Net, DDP y TPM2.net.

E1.31 es un protocolo para enviar datos DMX512 a través de la familia de protocolos ACN (E1.17).

- Hyperion.
- UDP en tiempo real.
- Control de voz de Alexa (incluida la atenuación y el color).
- Adalight (PC ambilight a través de serie) y TPM2.
- Sincronizar el color de varios dispositivos WLED (notificador UDP).
- Mandos a distancia por infrarrojos (RGB de 24 teclas, se requiere receptor.

- API de solicitud JSON y HTTP.

  Los datos pueden ser enviados usando el método HTTP POST y recibidos usando el método HTTP GET. Echemos un vistazo y hagamos una solicitud GET. Utilizaré JSON Placeholder (insomnia), una API REST online gratuita para desarrolladores que devuelve datos aleatorios en formato JSON.

- MQTT.

  Desde las versiones WLED de 0.8.0 en adelante pueden conectarse a un agente MQTT para un control inteligente del hogar. La conexión a dominios y servidores IP se admite en el puerto 1883.

  Actualmente, no se admiten conexiones seguras. Recomiendan conectarse solo con corredores locales de MQTT. Solo en v0.8.4-0.8.6, WLED admite el descubrimiento automático de MQTT por parte del software HomeAssistant. Esto se ha eliminado debido a problemas de bootloop y a favor de la integración nativa de HomeAssistant.

- E1.31, Art-Net, DDP y TPM2.net.

  E1.31 es un protocolo para enviar datos DMX512 a través de la familia de protocolos ACN (E1.17).

  ACN es un conjunto de documentos que especifica una arquitectura, incluidos los protocolos y el idioma, que puede configurarse y combinarse con otros protocolos estándar para formar sistemas de control de audio, iluminación u otros sistemas flexibles y en red. Se puede implementar en redes que admiten UDP, IP y protocolos relacionados. No está vinculado a Ethernet como medio de transporte, pero Ethernet es una opción obvia.

- TPM2.NET.

  Compatible con el último maestro y estaba disponible desde WLED 0.10.1. Configure el puerto UDP de transmisión WLED en 65506 en la configuración de sincronización para habilitar la recepción de datos TPM2.NET.

- Hyperion.

  Ahora puede utilizar WLED con el popular software Hyperion de Ambilight. Simplemente configure Hyperion para usar un dispositivo UDP con protocolo 0 en el puerto 19446. El número máximo de LED admitidos en este modo es 490.

  Hyperion proporciona una poderosa API para escribir sus propios efectos, junto con posibles opciones e interfaz de usuario para ajustarlos.

  Sistemas compatibles:

  Raspberry Pi (ver también HyperBian).

  Debian 9 | Ubuntu 16.04 o superior.

  Mac OS.

  Windows 10.

- UDP en tiempo real.

WLED ofrece una forma de controlar directamente los LED conectados a través de UDP.

El protocolo se conoce como WLED Audio-Reactive-Led-Strip (WARLS), ya que el soporte de ese proyecto era su objetivo principal. Sin embargo, también se puede utilizar para otras aplicaciones en tiempo real como un ambilight.

WARLS usa el mismo puerto UDP que usa el notificador (por defecto 21324, se puede cambiar en la configuración). Por el momento, la interfaz de usuario web estará deshabilitada mientras esté activa, la API HTTP, Alexa y el control de botones seguirán funcionando. Utiliza la configuración actual de brillo y corrección de gamma.

El byte 0 del paquete UDP le dice al servidor qué protocolo en tiempo real usar.

Utilizaremos la versión (WLED 0.12.0).

Los aspectos más destacados en esta versión son:

- Soporte para múltiples salidas LED (10 en ESP32, 3 en ESP8266). Agregue aún más luces a su instalación.
- Listas de efectos facilita la búsqueda de sus favoritos.
- Lista de instancias: simplemente salte de un dispositivo WLED a otro sin recordar las direcciones IP.
- Se agregaron cuatro efectos nuevos, Aurora, Dynamic Smooth, Tetrix y simulador de TV.
- Indicador agregado para cuadros por segundo.

Las tiras de ledes compatibles están representadas en la figura 3.5.

| Type | Voltage | Comments |
|---|---|---|
| WS2812B | 5v | |
| WS2813 | 5v | |
| SK6812 | 5v | RGBW |
| APA102 | 5v | C/D |
| WS2801 | 5v | C/D |
| LPD8806 | 5v | C/D |
| TM1814 | 12v | RGBW |
| WS2811 | 12v | 3-LED segments |
| WS2815 | 12v | |
| GS8208 | 12v | |
| Analog/non-addressable | any | Requires additional circuitry |

Figura 3.5 Diferentes tipos de tiras ledes.

### 3.4.2 DESCARGA.

En primer lugar, necesitamos descargar el archivo (.bin) de WLED en este proyecto he seleccionado la versión 0.12.0 que es la más actualizada [5].

Primer paso descargar el archivo (.bin) figura 3.6.

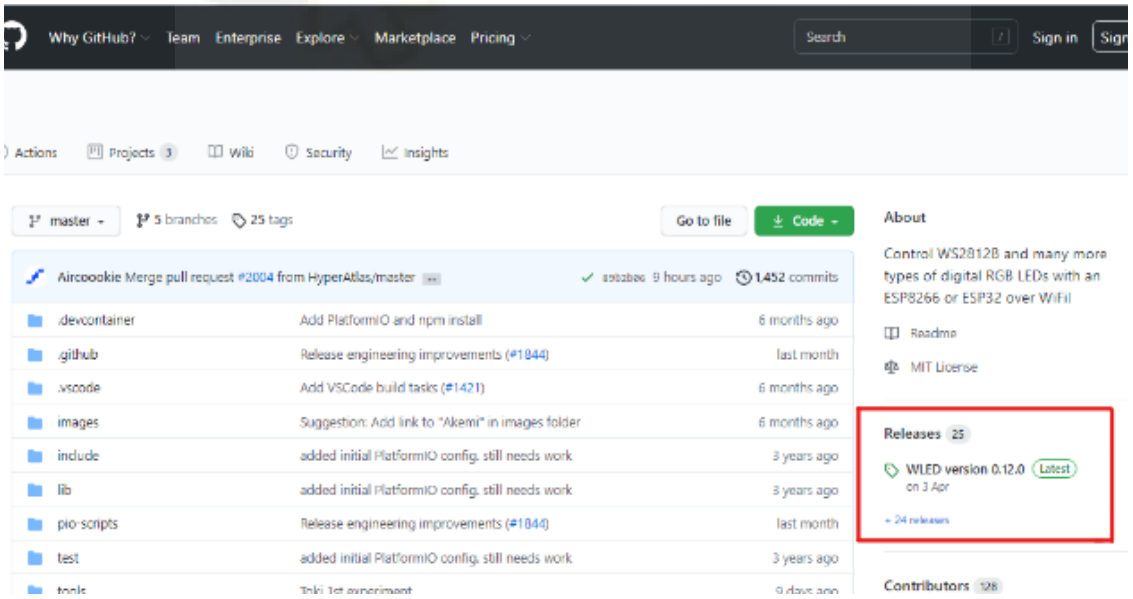


Figura 3.6 Búsqueda de WLED mediante Github.

Descargamos el archivo de WLED releases 0.12.0 como indica la figura 3.7.


Figura 3.7 Elección de versión de WLED.

### 3.4.3 FLASHEAR EL WLED/ESP32.

Para poder transferir el programa WLED en el microcontrolador (ESP32) tenemos que flashearlo [6].

En este caso se ha utilizado el programa de ESPHome-Flasher 1.3.0, aunque también podemos flashear mediante comando utilizando el esptool.

Actualmente he seleccionado la versión indicada en la figura 3.8.


Figura 3.8 Búsqueda de ESPHome-Flasher mediante Github.

Una vez completada la descarga es muy fácil saber utilizarlo ya que necesitamos tener el ESP32 conectado a un puerto serie del nuestro PC e indicar en SERIAL PORT cual es el puerto seleccionado, después en FIRMWARE tenemos que introducir la dirección donde se encuentra el archivo (.bin) del WLED 0.12.0. y haciendo clic en Flash ESP como se puede visualizar en la figura 3.9.

Una vez se está flasheando el programa necesitaremos pulsar durante unos segundos el botón de “boot” del ESP32 para poner al ESP32 en modo flasheo.

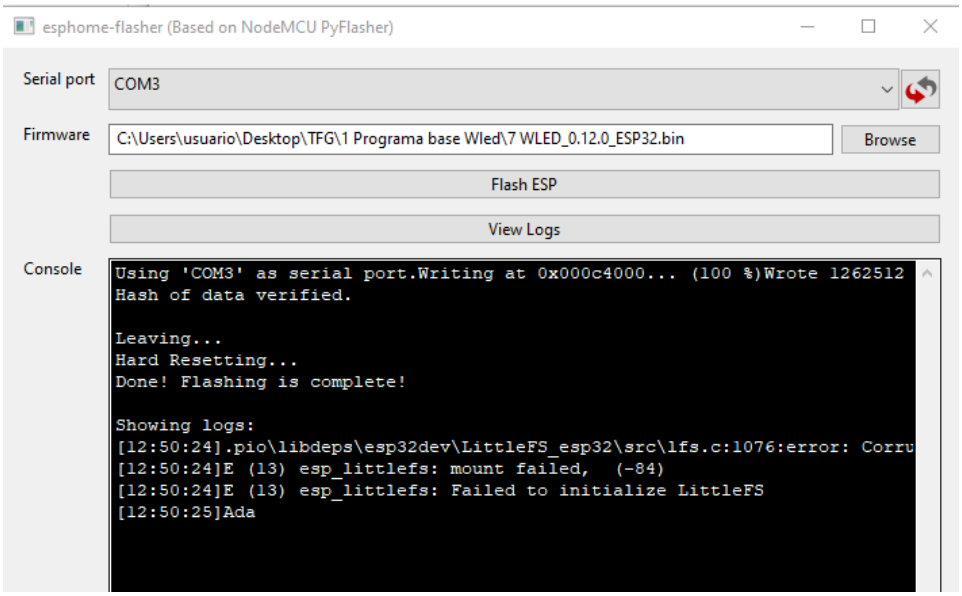


Figura 3.9 Aplicación ESPHome-Flasher.

### 3.4.4 CONFIGURACIONES WLED.

Una vez flasheado el programa WLED se crea un access point mediante una red wifi por defecto la red se llama WLED-AP como se ve en la figura 3.10 donde podemos modificar la configuración asociada a nuestro ESP32.


Figura 3.10 Access Point de WLED.

Una vez conectados la red WLED-AP se abrirá una ventana donde podemos hacer un tratamiento de ledes a modo de prueba y también podemos modificar la configuración de WLED, donde determinaremos a que red wifi se tiene que conectar el ESP32 para poder utilizarlo o modificar parámetros como cuantos ledes trabajan en cada pin de salida, con un máximo de 10 pines de salida, también podemos determinar segmentos, efectos o intensidad de laminación de los ledes, estos cambios se encuentran en WIFI SETTING como se puede observar en la siguiente Figura 3.11.

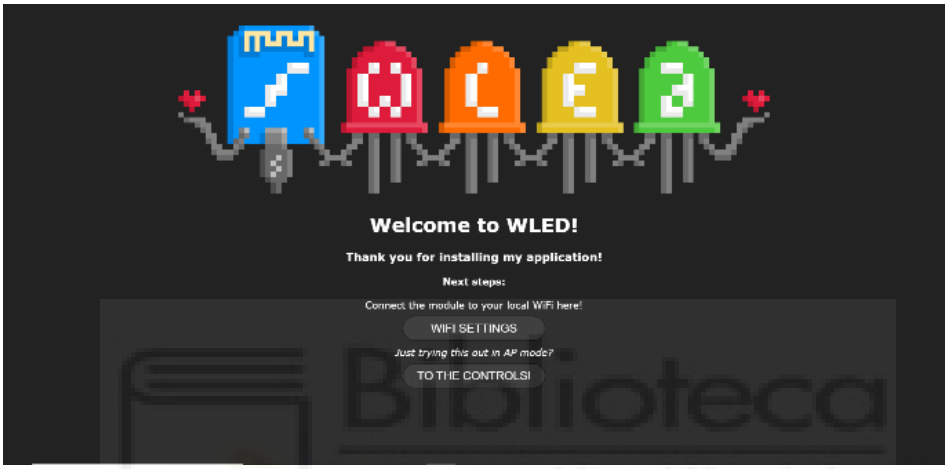


Figura 3.11 Pantalla principal del WLED

En la Figura 3.12 observamos el apartado de Network name (SSID), indicaremos el nombre de nuestra red wifi, exactamente como se escribe, en Network password indicaremos la contraseña e indicaremos el (mDNS address ) para una vez conectados a la red wifi poder modificar cualquier parámetro necesitaremos entrar a dicha dirección , en este caso (http://wled-ledes.local).

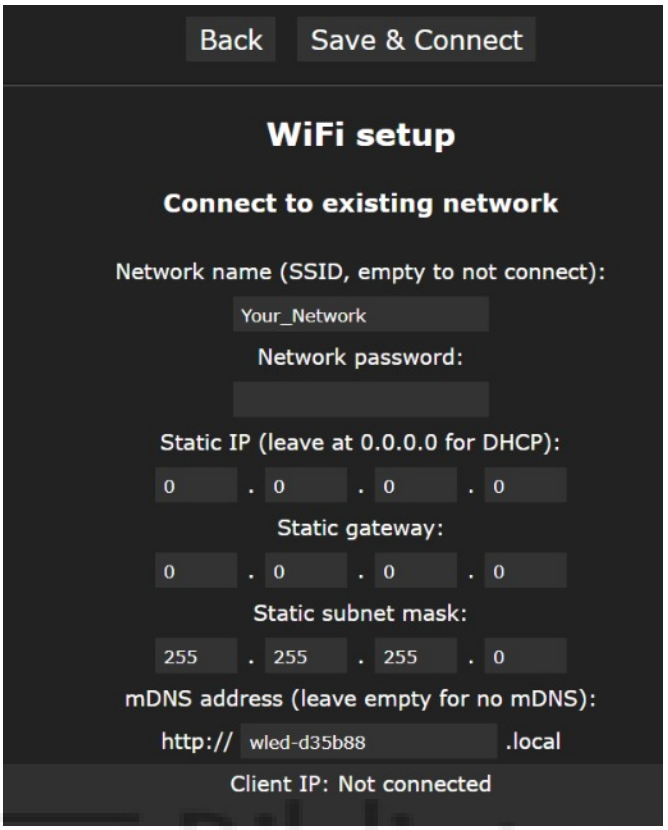


Figura 3.12 Setting del WLED.

Y finalmente guardamos dicha configuración para trabajar en la misma red wifi (Save &Connect) que está ubicada en el final de la figura 3.13.

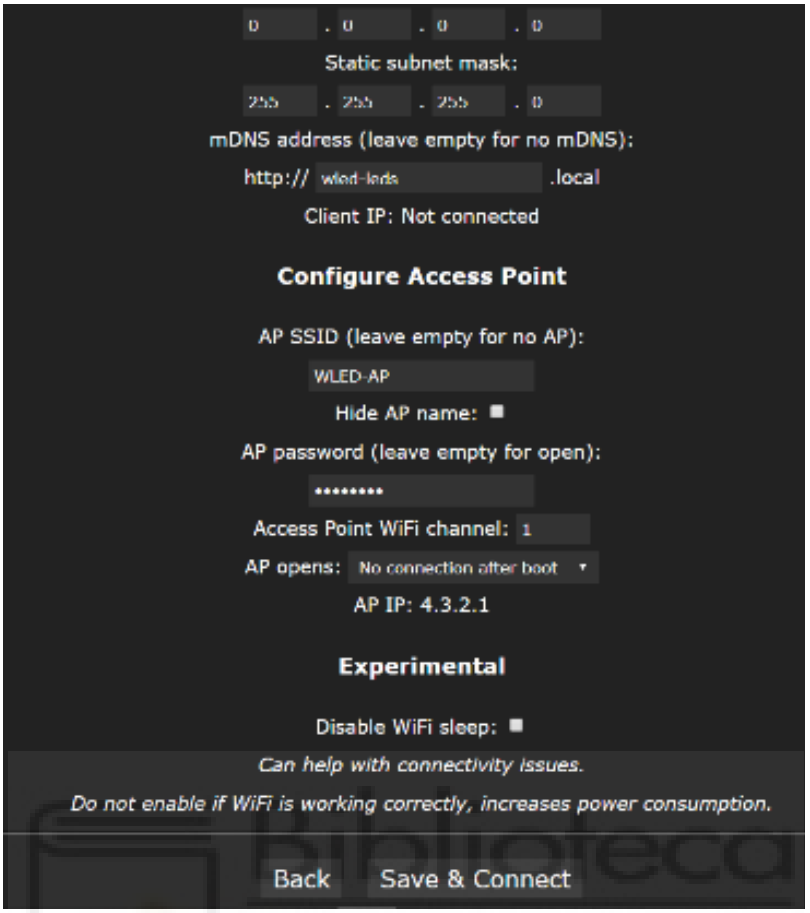

Figura 3.13 Configuración del software WLED.

También mediante programas de redes podemos saber la IP del ESP32 conectada a la red y podemos acceder a la misma configuración insertando el número de la IP del dispositivo.

La configuración de los ledes se modificará en el apartado LED Preferences donde podemos observar en la Figura 3.14 determinaremos cuantos ledes están vinculados con cada pin de salida de nuestro ESP32, led inicial y led final, el nivel de brillo de los ledes, el tipo de ledes que tiene nuestro proyecto y el máximo de corriente disponible para el brillo.

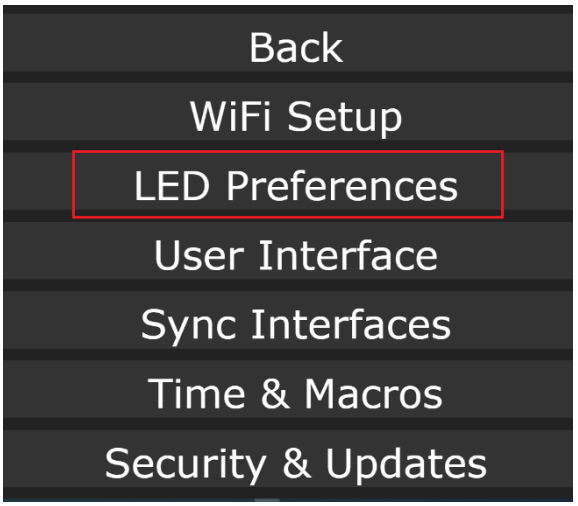

Figura 3.14 Menú de configuración.

En (Total LED cound) situaremos el total de ledes independientemente si están ubicados en diferentes pines como se puede ver en la figura 3.15.

Tenemos una opción llamada (Enable automatic brightness limiter) que su función es delimitar la intensidad que le suministra a los ledes por lo tanto su iluminación será la máxima posible.

En el apartado de (LED voltage) tendremos que seleccionar el voltaje de trabajo de nuestros ledes y también su consumo de intensidad.

Posteriormente tenemos la seccion de HARWARE SETUP donde en este apartado determinaremos que tipo de ledes son en este caso WS281x , el tipo de orden de color ya que muchos no son RGB sino GRB,BRG,RBG,BGR,GBR.

En la Figura 3.15 modificamos la configuración de la cantidad de ledes que hay asociados a cada pin de salida del ESP32, seleccionando primero el pin de salida , y posteriormente el número de inicio del pin y el número del último led.

Podremos añadir tantos pines de salida hasta que el WLED nos limite. En la actualización 0.12.0 el máximo de pines a utilizar serán de 10 .

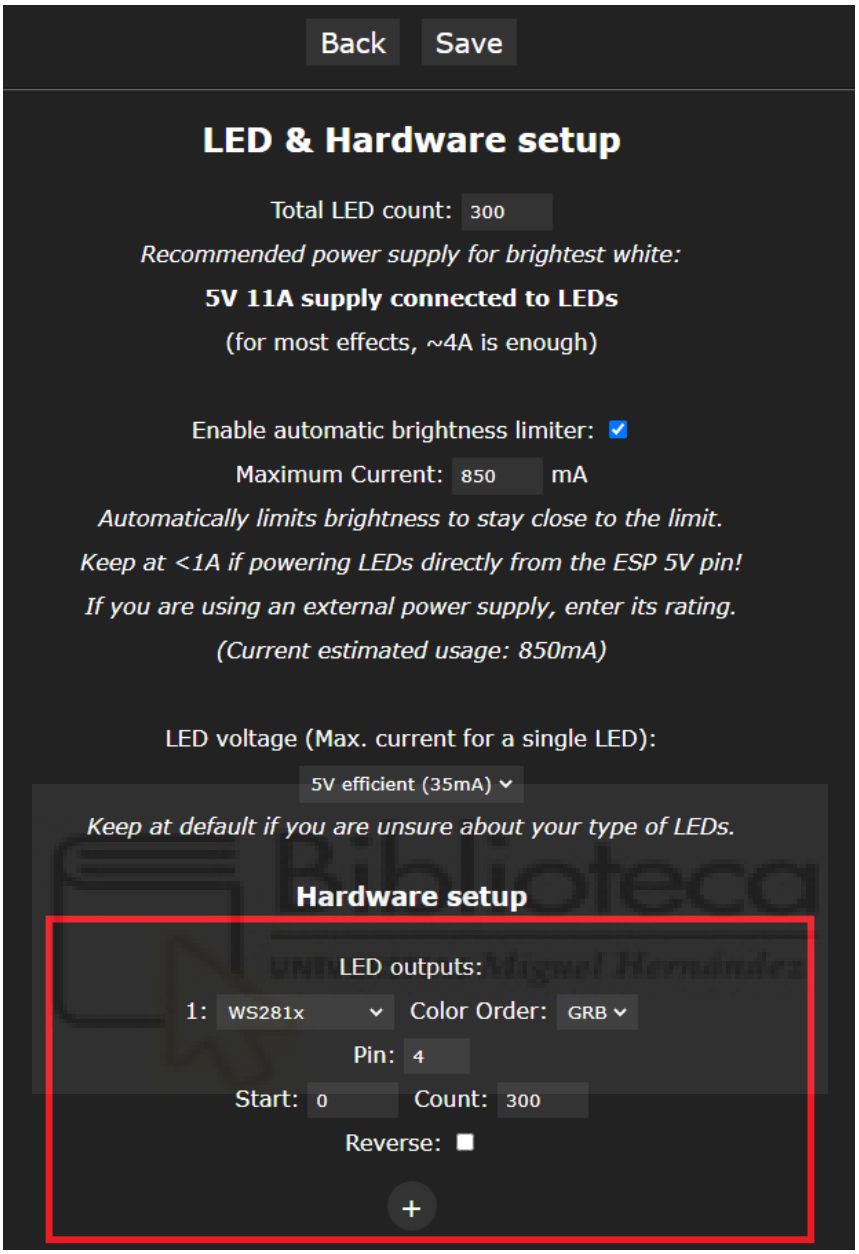


Figura 3.15 Configuración de las tiras ledes.

Por defecto podemos marcar un brillo que está entre los valores (0 y 255) como indica en la Figura 3.16, pero nosotros el nuestro proyecto no lo vamos a utilizar ya que nosotros decidimos el nivel de brillo dependiendo de la luminosidad que hay en la zona de trabajo es decir en el almacén, así podremos adaptarlo a otro almacén diferente cambiando este parámetro.

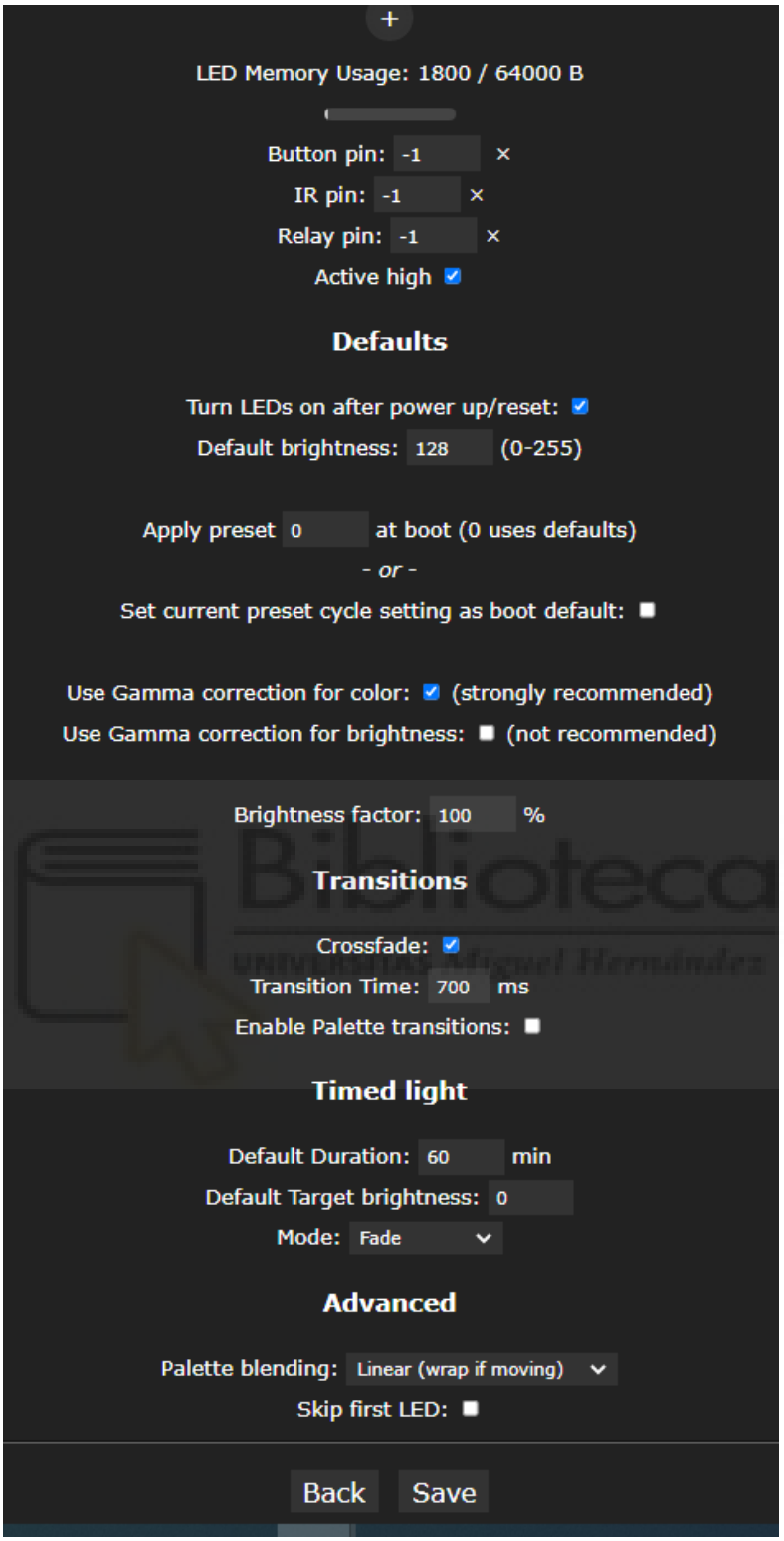


Figura 3.16 Configuración de las tiras ledes.

A continuación se mostrarán las formas de configurar la sincronización de las diferentes interfaces, en el menú principal seleccionaremos la opción de SYNC INTERFACES Figura 3.17.

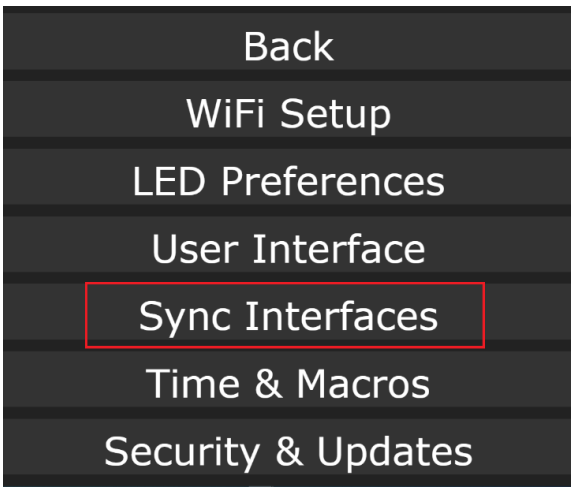


Figura 3.17 Menú de configuración.

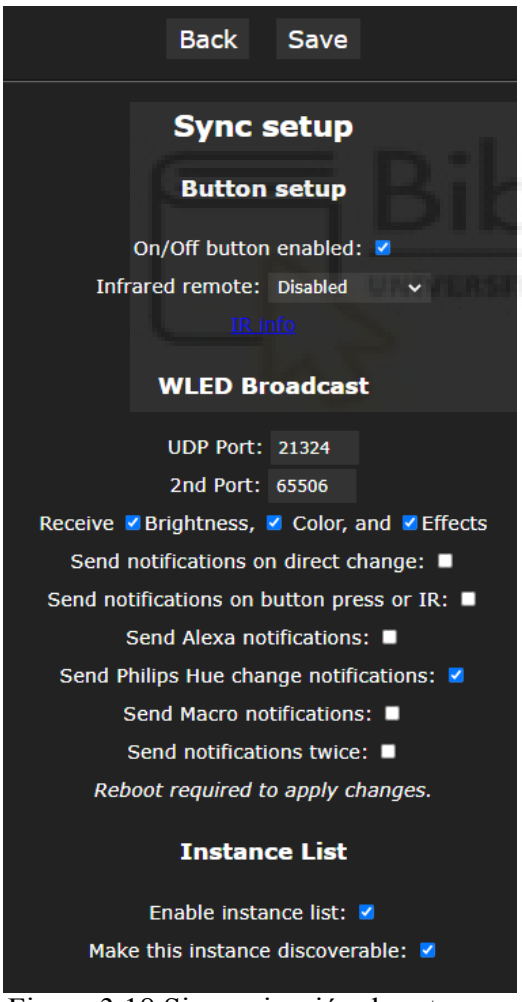


Figura 3.18 Sincronización de setup.

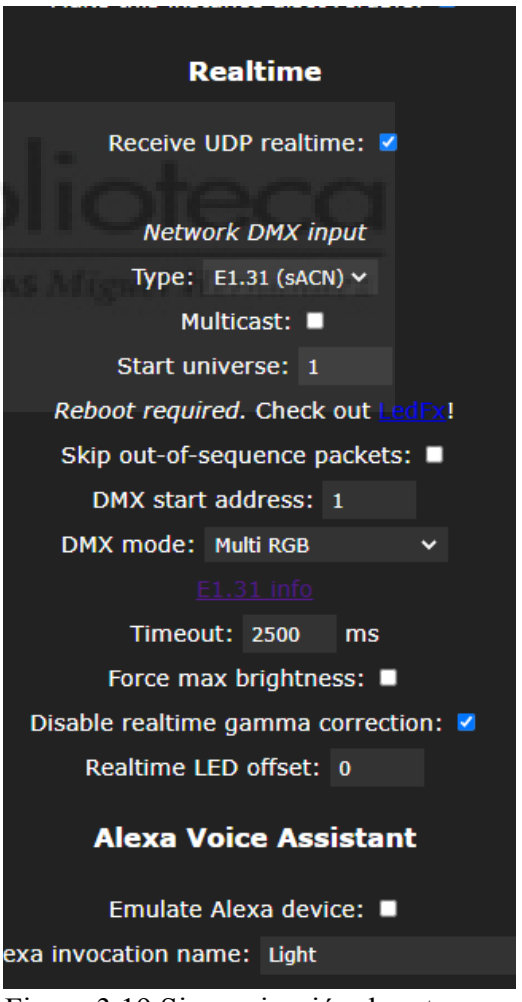


Figura 3.19 Sincronización de setup.

En las Figuras 3.18 y 3.19 muestran diferentes formas de sincronizar nuestra instalación de ledes.

Las más utilizadas son Blynk y MQTT Figura 3.20 y Figura 3.21.

En caso de Blynk habrá que definir el nombre del host donde lo tenemos ubicado y el número del puerto normalmente es el número 80.

Puede ser controlada mediante una app Android o Ios.

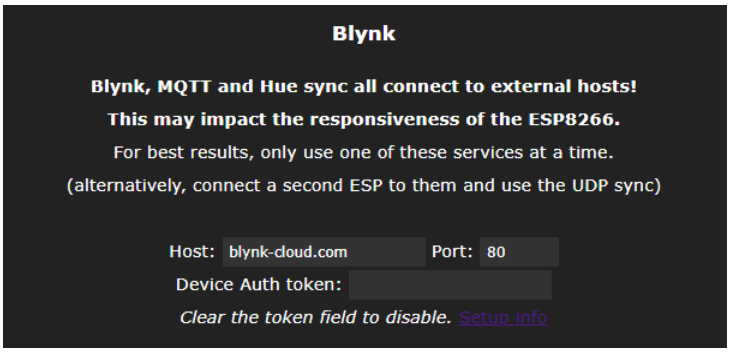


Figura 3.20 Sincronización de Blynk.

Otra interface sería MQTT donde mediante un Broker ubicado normalmente en un servidor se suscribe o publica información, este Broker tiene un host y un número de puerto siendo normalmente el 1883.

Definiremos el nombre del cliente y su contraseña Figura 3.21.

Y dependiendo de en qué tópico trabaje podremos leer la información.

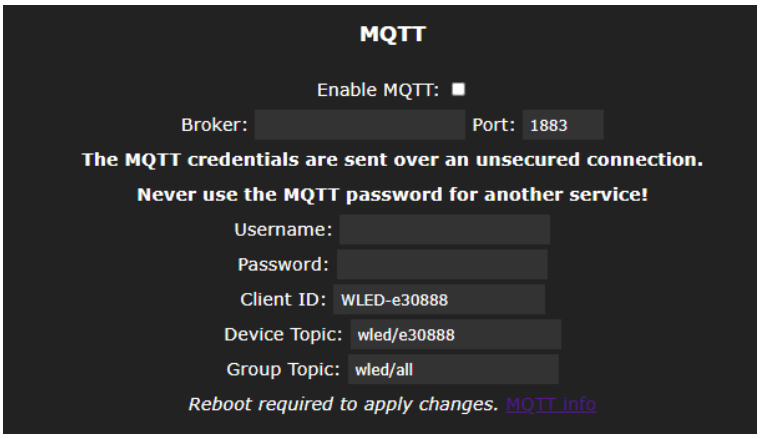


Figura 3.21 Sincronización de MQTT.

Guardaremos todas los cambios realizados para el desarrollo de nuestro proyecto, he utilizado el envió de datos y tramas mediante (Json).

Para el desarrollo de este proyecto utilizaremos la API JSON sobre HTTP, que desde la versión 0.8.4 WLED se ha sido implementada.

### 3.4.5 IMPLEMENTACIÓN DEL SISTEMA.

#### 3.4.5.1 GENERACIÓN DE MENSAJE JSON.

El primer mensaje debe de establecer unos parámetros que aparecen en la figura 3.22 que posteriormente también se pueden modificar, si no se modifican quedarán como constantes.


Figura 3.22 Mensaje inicial de la API JSON.

**-on** --Es un valor booleano --Estado encendido / apagado de la luz

**-bri** --Es un valor de 0 a 255 --Brillo de la luz. Si "on" es falso, contiene el último brillo cuando la luz estaba (también conocido como brillo cuando "on" se establece en true). Se admite la configuración de bri en 0, pero se recomienda usar el rango 1-255 y usar on: false para apagar. La respuesta del estado nunca tendrá el valor 0 de bri.

**-transition**-- Es un valor de 0 a 255 -- Duración del fundido cruzado entre diferentes colores/niveles de brillo. Una unidad es 100 ms, por lo que un valor de 4 da como resultado una transición de 400 ms.

**- udpn.send** --Es un valor booleano --Envíe el paquete de difusión WLED (sincronización UDP) en caso de cambio de estado.

**-udpn.recv**--Es un valor booleano --Recibir paquetes de difusión.

**-seg**--Matriz de objetos de segmento -- Los segmentos son partes individuales o agrupaciones de la tira de LED.

En la sección donde tratamos los segmentos "**seg**" tratamos de inicializar a 0 todos los ledes, "i";[0,( nº total de ledes) ,[0,0,0]].

Este mensaje solo se enviará al principio del día de trabajo, posteriormente enviaremos otros mensajes solamente definiendo los segmentos de los ledes, que deben de encenderse y de qué color o también podemos modificar las características de ledes específicas sin que sean segmentos.

### 3.4.2.2 WLED

Figura 3.23 Almacén.

SIMULACIÓN DE PEDIDO.

En este apartado se simulara una serie de pedidos realizados en la empresa, cada color está asociado a un empleado del almacén como en la figura 3.23, cuando realice el picking de un producto, se iluminarán los ledes asociados a las cajas donde debe de depositar dicho producto, este sistema nos permite tener una gama de colores, de momento solamente en esta sección trabajan 1 persona, del cual será el color Rojo, posteriormente explicó en cada fase, que está sucediendo.

En primer lugar debemos de inicializar el programa como se muestra en la Figura 3.24 determinando los parámetros generales como el brillo ,transición , "udpn" {"send"y "recv"}, y segmento que es el global de ledes , estos parámetros han sido explicados en el apartado anterior.

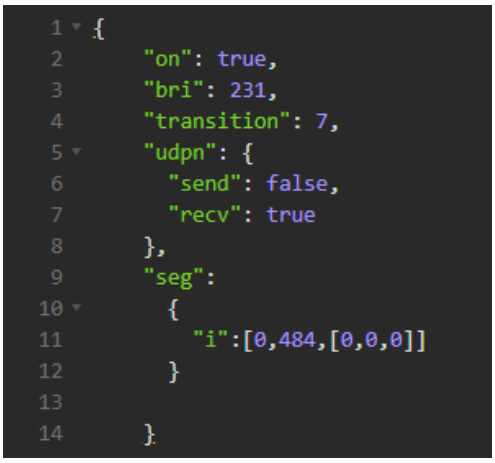


Figura 3.24 Primer mensaje.

Al iniciar el día, debemos verificar el funcionamiento de todos los ledes, por lo tanto hacemos que todos los ledes se iluminen en rojo como en la figura 3.27, verde como en la figura 3.26 y azul como en la figura 3.25 verificando los colores del RGB.

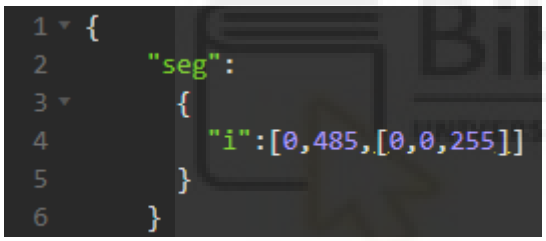


Figura 3.25 Color azul


Figura 3.26 Color verde

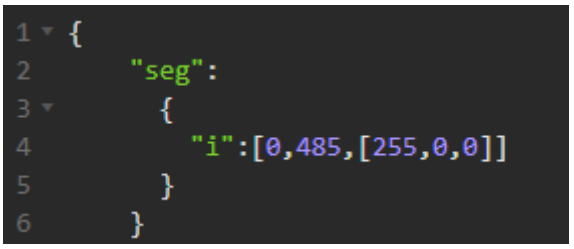


Figura 3.27 Color rojo

A continuación de este testeo se situarán todos los ledes en 0 (apagado) como indica la Figura 3.28 para ya estar preparados para trabajar.

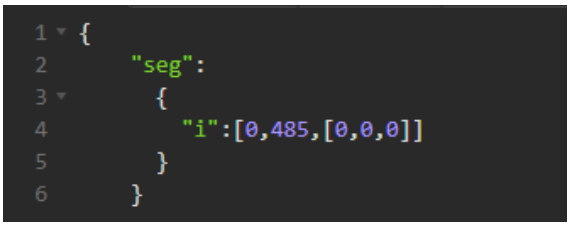

Figura 3.28 Todos los ledes están apagados.

Una vez verificados todos los ledes funcionan perfectamente, podemos proceder a trabajar.

En este ejemplo veremos cómo trabaja el operario que tiene asignado el color (Rojo). Figura 3.29, en este método podemos activar o desactivar ledes de manera independientemente o en segmentos.

Figura 3.29 Pedido de color rojo.

A medida que el trabajador de almacén van depositando los productos en las cajas que están indicadas con su correspondiente color, se van apagando esos ledes como se puede ver en la Figura 3.30, Figura 3.31, Figura 3.32, Figura 3.33 en este ejemplo se apagan en agrupaciones para recortar el programa.

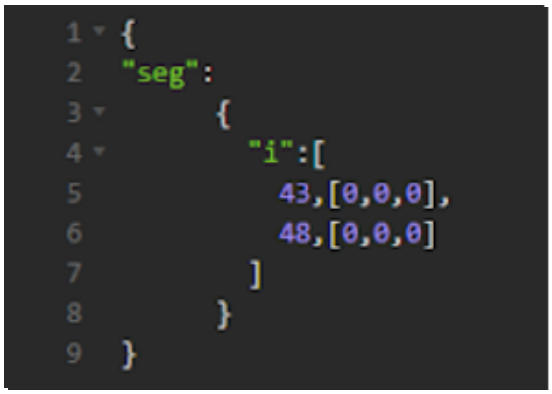


Figura 3.30 Apagado ledes 43 y 48.


Figura 3.31 Apagado ledes 83 y 88.


Figura 3.32 Apagado ledes 208 y 213.

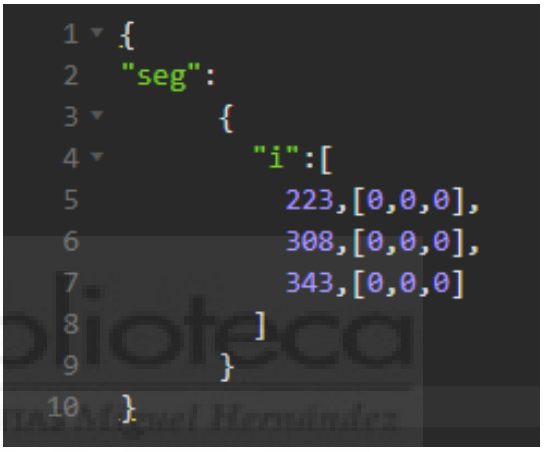


Figura 3.33 Apagado ledes 223,308 y 343.

El procedimiento será exactamente el mismo pero la desactivación será de forma independiente, ya que cada led determina una posición en el almacén que a la vez una determinada caja (pedido), se desactiva el led cuando el operario deposita el objeto y haga un pick del producto en la caja seleccionada.

Cuando un trabajador termina la colocación de todos los productos que tenía para distribuir en la preparación de pedidos, inicia otro pedido.

Una vez el pedido está completo de todos los productos deseados, los ledes encargados de la deposición de productos a dicho pedido pasaran a estar en amarillo como se muestra en la Figura 3.34 determinado que esta caja está completada.

Hay una persona que se dedica a retirar los pedidos donde su interior ya está completado.

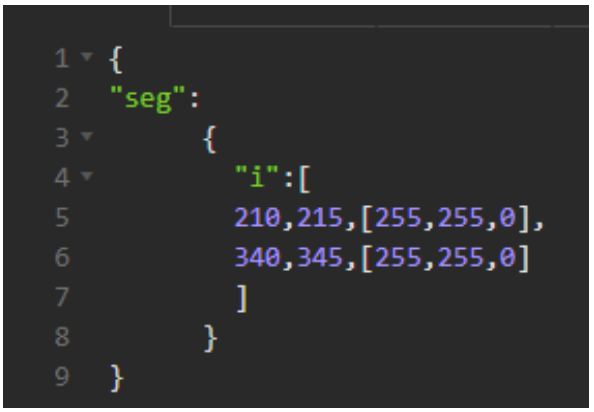


Figura 3.34 Caja completada.

Cuando se retire el pedido, el trabajador amarillo indicara que no hay caja (pedido) porque ha sido retirada para empaquetar y enviar, por lo tanto el segmento de ledes asociados a dicha caja pasarán a ser blanco, indicando (falta caja), como muestra en la Figura 3.35.

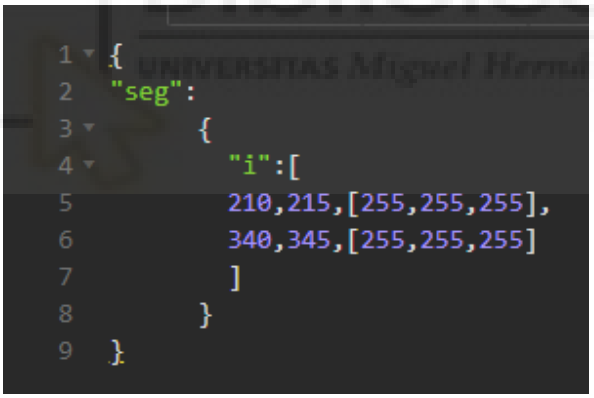


Figura 3.35 Posición sin caja.

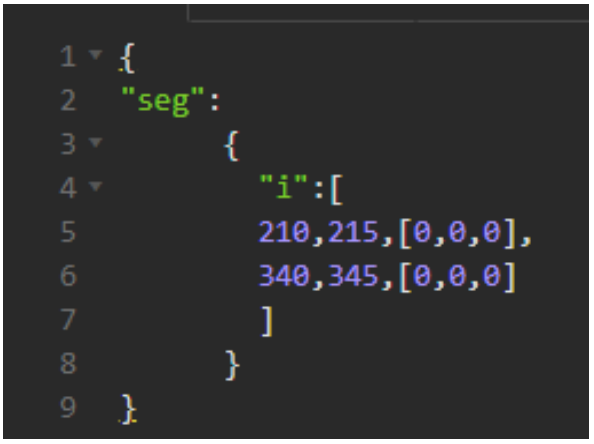


Figura 3.36 Posición lista para operar.

Por último una vez se sitúe otra caja se posicionarán todos los ledes en apagado como se puede observar en la Figura 3.36 esperando nuevas señales para activarse o desactivarse dependiendo de un nuevo pedido.

# 4 PUESTA EN MARCHA DEL SISTEMA.

## 4.1 CREACIÓN DE DIFERENTES ENTORNOS DE IMPLEMENTACIÓN.

Para la elección del modo de montaje del esquema eléctrico que en el diagrama de bloques se representa como la primera capa del sistema, he tenido en cuenta distintos esquemas eléctricos, para que el sistema trabaje más eficiente, el desequilibrio entre tiras de la iluminación será similar durante toda la instalación.

1 º Esquema eléctrico.

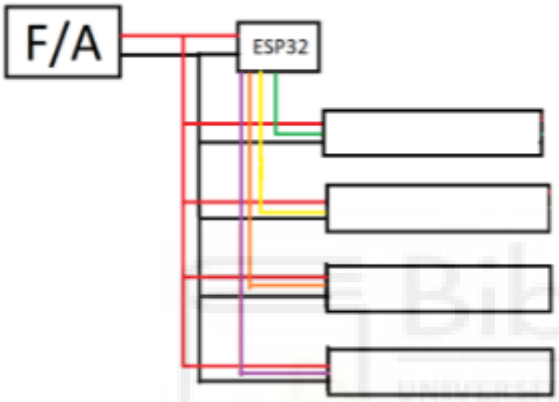


Figura 4.1 Esquema eléctrico con cuatro tiras ledes simples.

En este esquema eléctrico la alimentación se proporciona al principio de cada línea de tiras ledes y a nuestro microprocesador ESP32.

El microprocesador ESP32 dispondrá de 4 salidas de datos para la activación o desactivación de los ledes, la entrada del cable datos de cada línea que determinara cual es el orden de los ledes, previamente configurado en la app de WLED.

Este sistema funciona perfectamente ya que estamos trabajando con 300 ledes en cada línea.

Este esquema eléctrico fue implementado para visualizar cómo funcionaba al determinar 4 salidas del ESP32, con 4 tiras ya que anteriormente se probó un esquema eléctrico con solamente un cable de datos, esto fue debido a la versión WLED que solo

permitía la salida de datos con un pin de OUTPUT, en esta última versión 0.12.0 al permitirnos tener 10 puntos de salidas podemos aprovecharnos de esta ventaja se puede visualizar en la figura 4.1.

2º Esquema eléctrico.

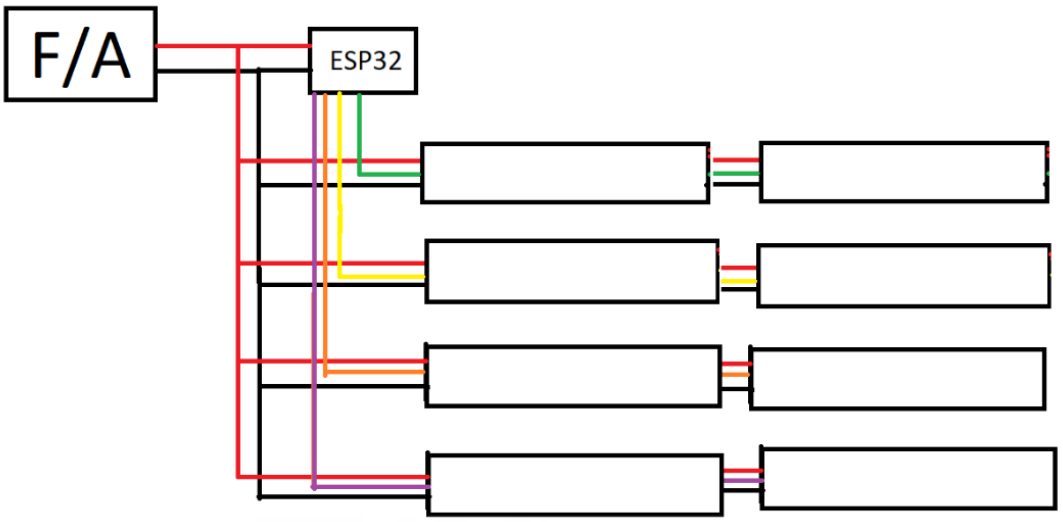


Figura 4.2 Esquema eléctrico con ocho tiras ledes simples.

En este esquema eléctrico la alimentación se proporciona igual que el esquema anterior.

El microprocesador ESP32 dispondrá de 4 salidas de datos también para la activación o desactivación de los ledes, pero le añadimos una tira de ledes a cada línea, aumentando el número de ledes es decir de pedidos en las estanterías del almacén.

Este sistema funciona perfectamente si trabajamos con un número de ledes muy pequeño, cuando trabajamos con un número elevado de ledes, no existe equilibrio de brillo entre el primer led de la tira del último, por lo tanto, debemos modificar el esquema eléctrico para equilibrarlo lo mejor posible el esquema de la Figura 4.2.

Una prueba fue llevar la alimentación directamente a la mitad de las dos tiras y distribuirlas a las tiras, pero nos producía otro error ya que las tiras tienen que estar alimentadas al principio de la tira +Vin y GND.

3º Esquema eléctrico.

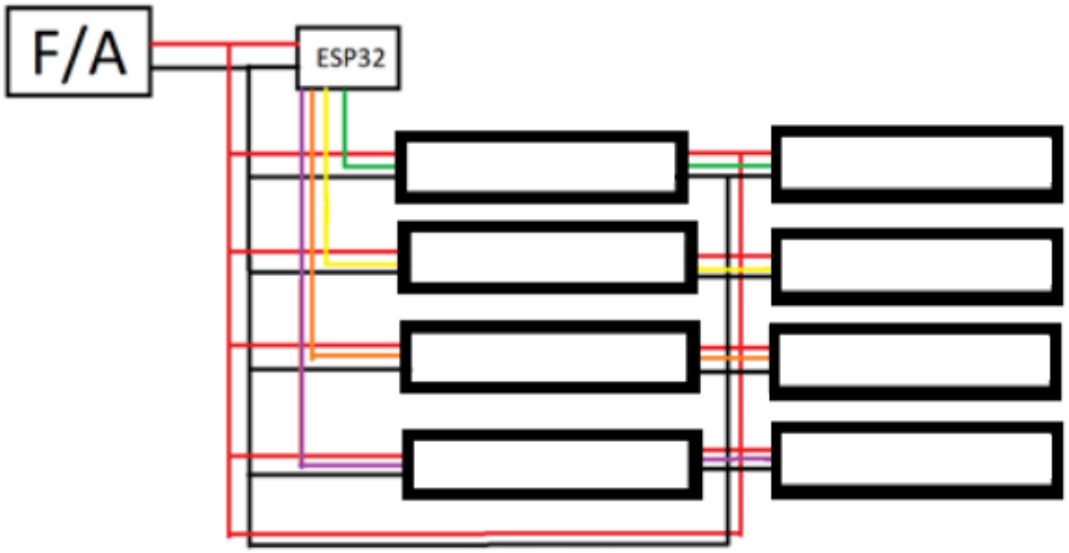


Figura 4.3 Esquema eléctrico con ocho tiras ledes simples con realimentación.

En este esquema eléctrico la alimentación se proporciona igualmente que el primer esquema.

En este esquema eléctrico Figura 4.3 lo principal era solucionar el problema de la atenuación del brillo, que depende de la cantidad de ledes hay en una línea, hacemos una realimentación desde la fuente de alimentación hasta el inicio de la siguiente tira de ledes igualando las tensiones de inicio.

Hay que tener en cuenta la longitud de las tiras de ledes, es preferible que tenga mayor longitud el cableado de alimentación que el cableado de datos, ya que con las largas distancias del cableado de datos perdemos información, o existen parpadeos en nuestros ledes y por lo tanto no tendría la calidad deseada.

## 4.2 ESCALABILIDAD DEL SISTEMA.

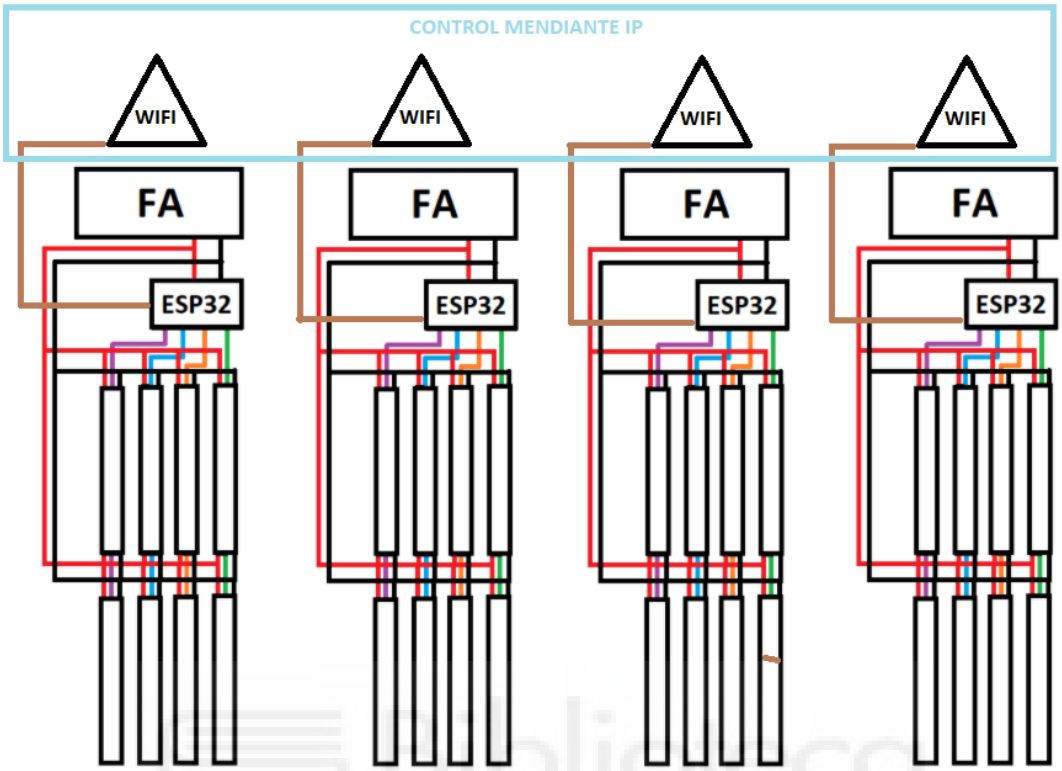


Figura 4.4 Esquema de cableado ampliado.

El sistema Pick-To-Light que he desarrollado puede tener la escalabilidad que deseamos, ya que podemos formar varios grupos de microcontroladores ESP32 con su fuente de alimentación, donde cada uno de los microcontroladores como se muestra en la figura 4.4 tendrán una IP y el servidor tendrá identificado cada microprocesador que productos tiene en sus ledes, pero tenemos que tener en cuenta que tenemos la condición de tener a los trabajadores bien organizados para detectar los cambios de colores en las estanterías del almacén, en muchas ocasiones la instalación de pick-to-light en grandes dimensiones viene acompañada de una plataforma que se puede desplazar por todas las estanterías del almacén y elevarse a cierta altura , este método de desplazamiento automatiza el recorrido, por lo tanto tiene mejor eficiencia pero tiene un mayor coste tanto para la construcción como para el mantenimiento .

Para el problema se pueden tener métodos o mejoras para que trabaje junto con el pick-to-light para organizar mejor los pedidos en cuanto a la posición de los trabajadores, este punto se desarrolla mejor en el apartado de líneas futuras.

## 4.3 ESTABILIDAD DEL SISTEMA ANTE FALLAS DE HARDWARE.

Para esta instalación hay que tener en cuenta las distancias de las estanterías del almacén, en mi caso las distancias son cortas por lo tanto no tenía problemas de caída de tensión, pero sí que podía apreciar una pequeña pérdida de intensidad lumínica en la tira de ledes en comparación a los primeros ledes de la tira ledes, por ese motivo opte por realimentar a mitad de la tira ledes para que esté más equilibrada la instalación y no haya diferencia lumínica, todos estos procesos se probaron en el almacén para ver si los ledes con esa iluminación eran bastante visibles por los trabajadores.

Otro fallo que podemos tener es que la IP del ESP32 se modifique, ya que si una IP de un dispositivo en la línea WIFI puede auto modificarse, por lo tanto, tiene que tener un mantenimiento para que no suceda esto ya que cuando queramos activar un producto que está asociado a este ESP32 no se activará, también si ocurre esto nos dirá que hay un error y se puede solucionar mirando la IP que ocupa en nuestra red, pero si se puede evitar, nos ahorramos posibles pérdidas de tiempo en solucionarlo.

## 4.4 LIMITACIONES DEL SISTEMA.

El desarrollo de este Pick-To-Light tiene la principal limitación generada por el ESP32 ya que solo disponemos de 10 pines de salida para el control de ledes.

Si deseamos trabajar con estanterías muy largas esto produce una caída de tensión en la línea , donde deberemos de realimentar la línea de ledes con la fuente de alimentación.

Si queremos ampliar a más ledes asociados a mas estanterias para una mejor gestión es aconsejable poner otro ESP32 ya que son económicos y se pueden asociar determinados productos a diferentes microprocesadores (ESP32).

Tenemos otra limitación que es, si un determinado trabajador está lejos de un producto y está cerca de la caja donde debe de depositar dicho producto, hay una pérdida de tiempo que se podría mejorar dándole esa acción de recoger ese dispositivo mediante otro trabajador que esté más cerca, esta limitación se puede solucionar mediante otro sistema comentado en líneas futuras.

# 5 CONCLUSIONES Y LÍNEAS FUTURAS.

## 5.1 CONCLUSIONES.

Mi punto de vista sobre este proyecto es que puede ser un gran avance para la empresa ya que cuanto menos tiempo tarden en realizar los pedidos más dinero ganarán y más sincronizados estarán todos.

Este sistema se va a utilizar en la empresa donde desarrolle las prácticas, dado que las dimensiones del almacén son muy grandes, verificaremos que podemos aplicar la escalabilidad, el sistema para la empresa consta de 30 unidades de microprocesadores con varias tiras leds y fuentes de alimentación instaladas como están indicadas en la Figura 4.4 y funciona perfectamente.

Por lo que me incumbe a mí en este proyecto me ha gustado enseñarme a cómo trabaja con el microprocesador ESP32 y qué con buenas ideas puedes mejorar procesos que realmente son importantes.

No conocía nada sobre JSON y al desarrollar proyectos como este voy adquiriendo esos conocimientos que son necesarios para tener más recursos a la hora de hacer un proyecto.

## 5.2 LÍNEAS FUTURAS.

Un método que se está desarrollando es la implementación de una baliza BLE.

BALIZA BLE : es un dispositivo transmisor que se utiliza una señal bluetooth de baja energía (LE) a dispositivos móviles que se encuentren cerca de él sin necesidad de sincronización previa. Esta tecnología se permite utilizar con teléfonos inteligentes.

Este sistema se situaría en varios ESP32, distribuidos por toda la nave del almacén, de forma estratégica para determinar qué operario está más cerca de determinados productos.

Cada operario trabaja con un móvil que lógicamente tendrá activado el bluetooth, mediante un programa que gestione las señales recibidas por los distintos microcontroladores la empresa mediante la aplicación calcula que trabajador está más cerca de dicho producto, mejorando el tiempo de pedido completado.

# 6 BIBLIOGRAFÍA


[1] ESP32.

- https://www.circuitos-electricos.com/esp32-especificaciones-y-disenos/

[2] Baiyouli Fuente de Alimentación Conmutada AC 110/220V a DC 5V 30A 150W.

- https://www.amazon.es/Baiyouli-transformador-convertidor-vigilancia-televisi%C3%B3n/dp/B07D6SJ7P2/ref=sr_1_12?__mk_es_ES=%C3%85M%C3%85%C5%BD%C3%95%C3%91&dchild=1&keywords=convertidor+de+220v+a+5v&qid=1618310945&sr=8-12

[3] Tiras LED.

- https://www.amazon.es/individualmente-adressier-300-Pixeles-5050-RGB-impermeable/dp/B01MZGXY21/ref=sr_1_8?__mk_es_ES=%C3%85M%C3%85%C5%BD%C3%95%C3%91&dchild=1&keywords=led+rgb+5v&qid=1618386350&s=lighting&sr=1-8

[4] JSON.

- https://github.com/Aircoookie/WLED/wiki/JSON-API

[5] WLED.

- https://github.com/Aircoookie/WLED

[6] ESPHome-Flasher.

- https://github.com/esphome/ESPHome-Flasher