



Universidad Michoacana de San Nicolás de Hidalgo

FACULTAD DE INGENIERÍA ELÉCTRICA

TESIS

Sistema de control de iluminación mediante Wi-Fi e interfaz web.

**Que para obtener el Título de
INGENIERO EN ELECTRÓNICA**

Presenta

Diter Abraham Canchola Romero

Asesor de Tesis

M.C. Félix Jiménez Pérez

Morelia Michoacán Agosto 2018

Agradecimientos

- *A mis padres Ma. de la Luz Romero García y Alfonso Canchola Martínez, que me han apoyado durante toda mi existencia y han dado siempre lo mejor de ellos, acompañándome en cada momento de mi vida escolar.*
- *A mi hermano Jessie Josue Canchola Romero, que ha sido mi cómplice y compañero de vida.*
- *Al Ing. Félix Jiménez Pérez, por su incondicional apoyo y los conocimientos que me ha aportado en gran parte de la Licenciatura.*
- *A todos mis maestros, por todo su conocimiento compartido.*
- *A mis amigos, por todas las horas de que han pasado junto a mí, y los infinitos momentos que me han ayudado, para ser una persona más feliz.*

Dedicatoria

- *A mis padres, que a pesar de mis equivocaciones nunca me han dejado de apoyar.*
- *A la memoria de todas las personas que algún día tuve el placer de conocer y quedaron marcados en mi vida.*

Resumen

Este proyecto presenta el diseño, implementación y control de un sistema de iluminación. El sistema puede controlar intensidad, color utilizando LEDs RGB y posición. La comunicación al sistema de iluminación se realiza mediante Wi-Fi y se puede efectuar desde cualquier dispositivo que tenga acceso a un navegador Web.

El hardware de este proyecto está constituido básicamente de seis elementos:

- Módulos ESP8266, encargados de crear la WLAN del sistema.
- LED's tienen la función de realizar los efectos básicos de iluminación.
- Módulo FC-102 RGB rainbow, para realizar los efectos de iluminación a color.
- Servomotores, para realizar el control de posición.
- Fuentes de alimentación, que suministran la energía a cada dispositivo.
- Dispositivo con Wi-Fi y acceso a navegador Web, para la función de interfaz entre el usuario y sistema.

Operación básica:

El sistema tiene capacidad de realizar control de iluminación y posición, desde cualquier dispositivo con Wi-Fi y navegador Web sin necesidad de internet.

Las características del sistema son:

- Comunicación Wi-Fi.
- Control de sistema por medio de una página web.
- Control LED: encendido y apagado, intensidad y frecuencia.
- Control de posición: movimiento en dos ejes.
- Control de color por medio de LEDs WS2812 RGB.
- Cuatro dispositivos activos en la red del sistema.

Palabras clave:

Wi-Fi: wireless field (campo sin cables).

Web: hace alusión a la expresión nominal World Wide Web (Red). Se escribe con minúscula, y significa que el sustantivo al cual modifica pertenece a la Red o a la Web. El ejemplo más habitual de este uso es “página web”.

WLAN: wireless local area network (red de área local inalámbrica).

Full dúplex: un sistema que es capaz de mantener una comunicación bidireccional.

Firmware: programa informático que establece la lógica de más bajo nivel que controla los circuitos electrónicos de un dispositivo.

Abstract

This project introduces design, implementation and control of an illumination system. The system can control intensity, color LEDs RGB and position. The system communication is by Wi-Fi and it is done with a device with Web browser.

The project's hardware is constituted by six elements.

- Modules Wi-Fi, they create system's WLAN.
- LEDs, they have function of doing basic effects illumination effects.
- Module FC-102 RGB rainbow to doing color effects.
- Servomotors, to doing position effects.
- Supply Sources to the devices.
- Device with Wi-Fi and access Web browser to doing the system's interface.

Basic function:

The system can do control of illumination from any device with Wi-Fi and Web browser, it without needs internet.

The characteristics system:

- Wi-Fi communication
- Control of system by web page.
- LED control: on-off, intensity and frequency.
- Position control: 2 axes.
- Color control by LEDs WS2812 RGB.
- Until 4 devices active in the WLAN.

Índice

Agradecimientos	I
Dedicatoria.....	II
Resumen	III
Abstract.....	IV
1. Capítulo 1. Introducción.....	1
1.1. Antecedentes.....	1
1.1.1. Iluminación Artificial	2
1.1.2. Comunicación inalámbrica.	7
1.2. Objetivo	13
1.2.1. Objetivo General.....	13
1.2.2. Objetivos Particulares.	13
1.3. Retos Personales	13
1.4. Justificación	15
1.5. Metodología.....	15
2. Capítulo 2. Marco Teórico.	16
2.1. Sistemas de iluminación	16
2.1.1. Sistemas de control de iluminación analógicos	17
2.1.2. Sistemas de control de Iluminación Digitales	17
2.1.3. Luminotecnia	17
2.2. Introducción a las redes Wi-Fi.....	18
2.2.1. Protocolos de Redes Inalámbricas	19
2.2.2. Tasas de transmisión.....	19
2.2.3. Capa física protocolo TCP/IP	20
2.2.4. Canales en 802.11 (Wi-Fi).....	21
2.2.5. Topología de redes inalámbricas	22
2.3. Microcontroladores.....	25
2.3.1. ¿Qué es un microcontrolador?	25
2.3.2. ¿Cómo es la escritura de los programas?.....	26
2.3.3. Consumo de Energía.....	27
2.4. ESP8266	27
2.4.1. Especificaciones de ESP8266.....	28

2.4.2.	Procesador.....	31
2.4.3.	Arquitectura del procesador.....	33
2.4.4.	Variantes	33
2.5.	Diodo LED	41
2.6.	SERVOMOTOR.....	43
2.6.1.	Características Generales	43
2.6.2.	Partes de un servomotor.....	43
2.6.3.	Consumo de energía.....	44
2.7.	WS2812	44
2.7.1.	¿Qué es WS2812?.....	45
2.7.2.	Driver	46
2.7.3.	Transferencia de datos.	47
2.7.4.	Consumo de Energía.....	48
2.7.5.	Diferencia entre familia WS28XX.....	48
2.8.	Socket TCP	50
2.9.	HTML	52
2.9.1.	¿Qué es HTML?	52
2.9.2.	Sintaxis.....	52
2.9.3.	Estructura interna de una página HTML.....	53
3.	Capítulo 3. Hardware.	57
3.1.1.	NodeMCU.....	57
3.1.2.	Versiones de NodeMCU	60
3.2.	Micro Servo Tower Pro MG90s	68
3.2.1.	Características:.....	69
3.2.2.	Cables de Conexión:	69
3.2.3.	Posicionamiento de un servomotor.....	70
3.3.	FC-102 rainbow LED RGB.....	71
4.	Capítulo 4. Diseño e Implementación.	72
4.1.	Procedimiento de diseño.....	72
4.2.	Firmware NodeMCU.....	73
4.3.	Herramienta de programación	78
4.4.	Red WiFi a través de NodeMCU.....	79

4.5.	Control de Actuadores	82
4.5.1.	Diodo LED.....	82
4.5.2.	Servo Motor.	86
4.5.3.	Rainbow FC-102.....	88
4.6.	Conexión entre dispositivos por medio de sockets TCP.	90
4.6.1.	Control desde el dispositivo con Wi-Fi y navegador web	90
4.6.2.	Conexión entre dos módulos NodeMCU.....	94
5.	Capítulo 5. Pruebas y Resultados.....	95
5.1.	Pruebas del prototipo funcional.....	96
5.2.	Control general del sistema por medio de un dispositivo con Wi-Fi y navegador google chrome.	101
5.3.	Observaciones y conclusiones	106
6.	Referencias	107

Lista de Figuras.

Figura 1.1-1 Homo erectus iluminando sus cuevas con fuego. [4]	1
Figura 1.1.1-1 Lámpara de aceite. [6]	2
Figura 1.1.1-2. Vela, artefacto de iluminación artificial más conocido y utilizado por el hombre. [9]	2
Figura 1.1.1-3 Lámpara Dietz y Lámpara Feuerhand Respectivamente. [10].	3
Figura 1.1.1-4 Lámpara de gas. [12]	4
Figura 1.1.1-5 Bombillo Incandescente. [14]	5
Figura 1.1.1-6. Diodo LED de luz blanca [15]	5
Figura 1.1.1-7.Gráfica de la evolución del LED de acuerdo a su eficiencia a través del tiempo. [16]	6
Figura 1.1.2-1 Telégrafo [18]	7
Figura 1.1.2-2.Temistocles Calzecchi fundador del cohesor. [19]	8
Figura 1.1.2-3 Cohesor. Detector de ondas electromagnéticas [21]	8
Figura 1.1.2-4.Guillermo Marconi. Inventor italiano a quien se atribuye el invento de la radio. [22]	9
Figura 1.1.2-5 Evolución de los dispositivos de comunicación inalámbrica. [25]	12
Figura 1.3-1.Cabeza robótica de iluminación LED RGB. [26]	13
Figura 1.3-2 Dispositivos de Iluminación. [27]	14
Figura 2.2-1. Logotipo de Wi-Fi. [33]	18
Figura 2.2.4-1. Espectro de frecuencias de transmisión en Wi-Fi. [31]	21
Figura 2.2.4-2.Canales sin solapamiento en la transmisión Wi-Fi (1, 6 y 11). [31]	21
Figura 2.2.5-1.Topología punto a punto. [31]	22
Figura 2.2.5-2.Topología punto a multipunto. [31]	22
Figura 2.2.5-3 Topología multipunto a multipunto. [31]	23
Figura 2.3.1-1. Arquitectura de un microcontrolador. [35]	26
Figura 2.4.1-1 ESP8266 Pinout. [37]	28
Figura 2.4.2-1 C.I. Procesador ESP8266EX. [36]	31
Figura 2.4.2-2 Interior de un ESP8266 y sus periféricos. [36]	32
Figura 2.4.2-3 Diagrama de bloques funcional. [37]	32
Figura 2.4.3-1 Arquitectura Harvard. [38]	33
Figura 2.4.4-1 Modulos ESP. [36]	33
Figura 2.4.4-2 Partes que constituyen el módulo Esp-01. [39]	34
Figura 2.4.4-3 ESP-01 pintout. [39]	34
Figura 2.4.4-4 ESP-02 pinout. [40]	35
Figura 2.4.4-5 ESP-03 pinout [41]	35
Figura 2.4.4-6 . ESP-04 pinout. [42]	36
Figura 2.4.4-7 ESP-05 pinout. [43]	36
Figura 2.4.4-8 ESP-06 y diagrama pinout. [36]	37
Figura 2.4.4-9 ESP-07 pinout. [44]	37
Figura 2.4.4-10 ESP-08 pinout. [45]	38
Figura 2.4.4-11 ESP-09 pinout. [46]	38

Figura 2.4.4-12 ESP-10 pinout. [46].....	38
Figura 2.4.4-13 ESP-11 pinout. [46]	39
Figura 2.4.4-14 ESP-12 pinout. [46]	39
Figura 2.4.4-15 ESP-13 pinout. [47]	40
Figura 2.4.4-16 ESP-14 pinout. [48]	40
Figura 2.5-1 Partes que constituyen un diodo LED. [50]	41
Figura 2.6.2-1 Partes de un servomotor. [51]	44
Figura 2.7-1.Encapsulado WS2812B RGB 5050. [54]	45
Figura 2.7.1-1 Variantes en la estructura de diseño de grupos de LEDs WS2812. [55].	46
Figura 2.7.2-1 Pinout y forma de estructura de conexión WS2812B. [55]	46
Figura 2.7.3-1 Diagrama de tiempos con respecto al flujo de datos. [55].....	47
Figura 2.7.5-1. Pinout LEDS 5050, WS2812B y WS2812, respectivamente. [57].....	49
Figura 2.8-1. Interfaz de un Socket TCP [58]	51
Figura 2.9.3-1 Código HTML para el método form. [64]	55
Figura 2.9.3-2. Resultado obtenido en una página Web, debida al código de la Fig. 2.4.3-1. [64]	55
Figura 3.1.1-1 NodeMCU. [65].....	57
Figura 3.1.1-2 Arquitectura de Node MCU. [65]	58
Figura 3.1.1-3 Versiones NodeMCU. [65]	59
Figura 3.1.2-1 ESP-12E. [65]	60
[Figura 3.1.2-2 Node MCU 1a generación V1. [65]	61
Figura 3.1.2-3 Node MCU 2a generación V2. [65]	61
Figura 3.1.2-4 Node MCU V2, colocado en un protoboard. [65]	62
Figura 3.1.2-5 Node MCU 2a generación V3. [65].....	62
Figura 3.1.2-6 Node MCU V3, pinout. [66]	64
Figura 3.1.2-7 Pinout NodeMCU haciendo énfasis en el pin analógico. [65].....	66
Figura 3.1.2-8 Node MCU BOTONES y LED. [67]	67
Figura 3.2-1 Tower Pro MG90S y accesorios. [68]	68
Figura 3.2.2-1 Pinout micro servo. [69]	69
Figura 3.2.3-1 Diagrama de pulsos con respecto al tiempo, para controlar la posición o giro del motor. [70]	70
Figura 3.3-1 Anillo RGB con 16 LEDS y controlador integrado. [71]	71
Figura 4-1 Diagrama a bloques del funcionamiento del sistema. [72]	72
Figura 4.1-1 Captura de imagen de la página mostrando la sección de los módulos. [73] ..	73
Figura 4.2-1 Captura de imagen de la página web para empezar a crear un propio firmware. [73]	74
Figura 4.2-2 Herramienta Node MCU Firmwate Programer. [74]	74
Figura 4.2-3 Flashear NodeMCU [74]	75
Figura 4.2-4 Cargando Firmware. [74].....	75
Figura 4.2-5 Construcción de firmware. [75]	76
Figura 4.2-6 Cargando Firmware creado con 15 módulos. [74]	77
Figura 4.2-7 Terminación de Carga de Firmware. [74].....	77
Figura 4.3-1 Ejemplo de IDE Esplorer. [36]	78

Figura 4.3-2 Lua Uploader. [76].....	78
Figura 4.4-1 Sintaxis para el modo de operación. [77]	79
Figura 4.4-2 Sintaxis para la configuración en modo punto de acceso. [77]	80
Figura 4.4-3 Conexión a la Red por medio de un celular con conexión Wi-Fi. [78]	81
Figura 4.5.1-1. Diagrama de conexión NodeMCU y Diodo LED. [80]	82
Figura 4.5.1-2 Encendido de LED. [81]	83
Figura 4.5.1-3 Apagado del LED. [81]	83
Figura 4.5.1-4 Sintaxis para una salida pwm [82]	84
Figura 4.5.1-5 Sintaxis para modificar ciclo de trabajo o frecuencia en una salida pwm. [82]	84
Figura 4.5.1-6 PWM al 25%. [97]	85
Figura 4.5.1-7 PWM 50%. [97]	85
Figura 4.5.1-8 PWM 75%. [97]	85
Figura 4.5.1-9 PWM 90%. [97]	85
Figura 4.5.2-1 Diagrama de conexión de un Servomotor. [83]	86
Figura 4.5.2-2 Servo Motor Posicionado a 0°. [84]	86
Figura 4.5.2-3 Servo Motor Posicionado a 90°. [84]	87
Figura 4.5.2-4 Servo Motor Posicionado a 180°. [84]	87
Figura 4.5.3-1 Diagrama de conexión para rainbow FC-102. [85]	88
Figura 4.5.3-2 Sintaxis para inicializar un módulo WS2812. [82]	89
Figura 4.5.3-3 Ejemplos de algunos colores para WS2812 RGB. [86]	89
Figura 4.5.3-4 Resultado de rainbow FC-102 utilizando el color gold. [87]	90
Figura 4.6.1-1 Sintaxis para poner a escuchar al servidor. [77]	91
Figura 4.6.1-2 Ejemplo de código para hacer que el servidor escuche. [77]	92
Figura 4.6.1-3. Script en HTML para la creación de una página web sencilla con dos botones. [88]	92
Figura 4.6.1-4 Página Web para controlar el encendido y apagado de un led. [89]	93
Figura 4.6.2-1 Script en Lua, para crear un cliente y conectarse a un servidor. [90]	94
Figura 5-1 Diagrama funcional del sistema de iluminación. [91]	95
Figura 5.1-1 Lua Uploader, cargando programa al servidor principal. [92]	96
Figura 5.1-2 Iniciando Servidor principal. [92]	97
Figura 5.1-3 Página WEB, selección de dispositivo. [92]	97
Figura 5.1-4 Respuesta por parte del servidor principal. [92]	98
Figura 5.1-5 Prueba de manipulación de deslizadores. [92]	98
Figura 5.1-6 Respuesta a la prueba de los deslizadores. [92]	99
Figura 5.1-7 Inicialización de dispositivo 2. [92]	100
Figura 5.1-8 Petición recibida en el dispositivo 2 por parte del servidor principal. [92] ...	100
Figura 5.2-1 página web visto desde una pc con google Chrome. [93]	101
Figura 5.2-2 Control de iluminación LED. [93]	102
Figura 5.2-3 Control Rainbow FC-102 RGB. [93]	103
Figura 5.2-4 Lista de Selección de color. [93]	104
Figura 5.2-5 Lista de selección de efectos. [93]	105
Figura 5.2-6 Control de Movimiento. [93]	105

Lista de Tablas.

Tabla 1. Protocolo TCP/IP	20
Tabla 2. Especificaciones ESP8266x	29
Tabla 3. Función de cada pin.....	30
Tabla 4 Características Generales ESP8266.....	31
Tabla 5. Versiones NodeMCU.	60
Tabla 6. Relación entre E/S y pines internos del ESP8266.....	65

1. Capítulo 1. Introducción.

En este capítulo se mencionan los antecedentes basados a la tecnología usada en este proyecto, así como, los objetivos, justificación, motivación y metodología para la realización de dicho sistema.

Como se sabe, la iluminación artificial tiene como objetivo remplazar a la natural cuando esta falta o es escasa.

La luz eléctrica, es la más cómoda, limpia, segura e higiénica respecto a otros tipos de luz artificial; sin embargo, requiere utilizarla en forma eficiente y económica, tomando en consideración que las fuentes primarias de producción de energía eléctrica que alimentan a las instalaciones y sistemas de alumbrado, consumen energéticos primarios, como el petróleo, que constituyen fuentes de energía no renovables. [1]

Debido a ello ha surgido la “luminotecnia” que estudia las distintas formas de producción de luz, así como su control y aplicación con fines domésticos, industriales o artísticos [2]. Un ejemplo de aplicación son los sistemas de iluminación, estos sistemas cada vez se tornan más complejos pero gracias a eso se obtiene mejor resultado en cuanto a los efectos o funciones que realizan, esta razón hace que estos sistemas sean costosos y complicados de operar si no se cuenta con conocimientos previos.

1.1. Antecedentes

Desde tiempo de la prehistoria el hombre descubrió el fuego aproximadamente hace 790.000 años. Nuestro antepasado el Homo erectus fue el primero en usar y dominar el fuego según recientes estudios de la Universidad Hebrea de Jerusalén y una de sus utilidades fue la iluminación de sus cavernas ver Figura 1.1-1. [3]



Figura 1.1-1 Homo erectus iluminando sus cuevas con fuego. [4]

De este modo la llama fue la primera forma de iluminación artificial utilizada por el hombre.

1.1.1. Iluminación Artificial

Lámpara de aceite.

Se tiene indicios de que el primer candil apareció hace 50.000 años, usando como combustibles aceite o grasa de origen animal del que además se utilizaba el cráneo para colocar en su interior estos combustibles con una mecha de trenza de pelos, Figura 1.1.1-1 [5]



Figura 1.1.1-1 Lámpara de aceite. [6]

La vela.

Consiste en una mecha que asciende por el interior de una barra de combustible sólido, como puede ser la cera, la grasa animal o la parafina (el más habitual en la actualidad). [7]

La vela ya era utilizada en Egipto alrededor del siglo XIV AC. [8], utilizando cera de abeja como combustible ver Figura 1.1.1-2. [7]



Figura 1.1.1-2. Vela, artefacto de iluminación artificial más conocido y utilizado por el hombre. [9]

Siendo éstas, artefactos portátiles de iluminación que han perdurado hasta nuestros días.

Introducción

Lámpara de petróleo.

En la Figura 1.1.1-3 se muestra este tipo de lámparas de queroseno o petróleo, tienen su origen íntimamente relacionado con el descubrimiento del oro negro en 1859.



Figura 1.1.1-3 Lámpara Dietz y Lámpara Feuerhand Respectivamente. [10].

A pesar del tiempo transcurrido y de los notables avances respecto al alumbrado portátil, las lámparas de petróleo siguieron fabricándose a lo largo de todo el siglo XX e incluso lo siguen haciendo en el XXI ciertos países orientales, como India o China. Pero si entre todos los fabricantes mundiales hubiese que destacar a dos solamente, sin lugar a dudas estos serían Dietz en Estados Unidos y Feuerhand en Alemania. [10]

Iluminación a gas.

En cuanto a las lámparas de gas, se sabe que los antiguos códigos de Egipto y Persia hablaban de explosiones de gases combustibles que brotaban a través de la tierra. Los chinos usaban gas como fuente de iluminación muchos siglos antes de la era cristiana.

En 1784, Jean Pierre Mincklers produjo luz por primera vez con gas mineral. La primera instalación de iluminación de gas, la uso William Murdock en 1784 para iluminar su casa. Desde ese momento comenzaron a difundirse las primeras lámparas de gas.

En el siglo XIX la invención obtuvo algunos cambios, para lograr luz blanca incandescente en las lámparas de gas. [11]

Así en 1890 la mayoría de las ciudades tenían luz de gas ver Figura 1.1.1-4.



Figura 1.1.1-4 Lámpara de gas. [12]

Se aprecia como esta lámpara de gas está colocada en una pared y en la cual la mano de un ser humano gira una perilla permitiendo o bloqueando el paso de gas para la producción de luz artificial.

Iluminación eléctrica.

Si una corriente eléctrica pasa a través de cualquier conductor que no sea perfecto, se consume una determinada cantidad de energía que aparece en forma de calor en el conductor. Recordando que cualquier cuerpo caliente despedirá una cierta cantidad de luz a temperaturas superiores a los 525°C , un conductor que se calienta por encima de dicha temperatura mediante una corriente eléctrica actuará como fuente luminosa.

1650: Otto von Guericke de Alemania descubrió que la luz podía ser producida por excitación eléctrica. Encontró que cuando un globo de sulfuro era rotado rápidamente y frotado, se producía una emanación luminosa. En 1706, Francis Hawsbee inventó la primera lámpara eléctrica al introducir sulfuro dentro de un globo de cristal al vacío. Después de rotarla a gran velocidad y frotarla, pudo reproducir el efecto observado por von Guericke. William Robert Grove en 1840, encontró que cuando unas tiras de platino y otros metales se calentaban hasta volverse incandescentes, producían luz por un periodo de tiempo. En 1809, usó una batería de 2000 celdas a través de la cual pasó electricidad para producir una llama de luz brillante de forma arqueada. De este experimento nació el término "lámpara de arco". La primera patente para una lámpara incandescente la obtuvo Frederick de Moleyns en 1841, Inglaterra. Aun cuando ésta producía luz por el paso de electricidad entre sus filamentos, era de vida corta. Durante el siglo XIX, finalmente, Tomas A. Edison produjo una lámpara incandescente con un filamento carbonizado que se podía comercializar, aunque esta lámpara producía luz constante durante un periodo de dos días. Su primer sistema de iluminación incandescente la exhibió en su laboratorio el 21 de diciembre de 1879. [13]

Desde entonces estas lámparas incandescentes se convirtieron en parte esencial para generar iluminación artificial ver Figura 1.1.1-5.



Figura 1.1.1-5 Bombillo Incandescente. [14]

Iluminación LED.

Un LED (siglas en inglés de Diodo Emisor de Luz) es un componente electrónico de dos terminales que permite la circulación de la corriente eléctrica a través de él sin intermediación de un gas, como sucede en los bombillos tradicionales.

El LED desplaza en eficiencia y versatilidad a cualquier otra tecnología. Su calidad de color lo hace una posibilidad inmediata para el ahorro energético y no utiliza materiales peligrosos en su fabricación, de tal manera que sustituyen a las tecnologías basadas en mercurio o incandescentes.

La Figura 1.1.1-6 muestra un LED de luz blanca los cuales están basados en LEDs que emiten luz azul recubiertos con fósforo y así generar ese tono de luz.



Figura 1.1.1-6. Diodo LED de luz blanca [15]

Antecedentes de la iluminación LED.

1907: Henry Joseph Round descubre el efecto físico de la electroluminiscencia. [16]

Mediados de la década de 1920: Oleg Vladimirovich Losev fabricó un LED de óxido de zinc y carburo de silicio. [16]

1927: Losev publicó los detalles del primer LED en una revista científica rusa. [16]

1962: Sale al mercado el primer diodo luminiscente rojo, tipo GaAsP. Nick Holonyak Jr. es considerado el inventor del primer LED que emitía en el espectro visible. [16]

1971: Están disponibles LEDs en otros colores: verde, naranja, amarillo. Hay continuas mejoras en el flujo luminoso y la eficacia de todos los LEDs. [16]

1993: Se obtienen diodos tipo InGaN con una gran eficacia en los colores azul y verde. Se han conseguido las condiciones para crear luz blanca. Shuji Nakamura es considerado el inventor de los primeros LEDs de estos colores. [16]

1995: Se lanza al mercado el primer LED de luz blanca creado a través de conversión luminiscente. [16]

La Figura 1.1.1-7 explica cómo fue evolucionando la tecnología LED.

Historia del LED

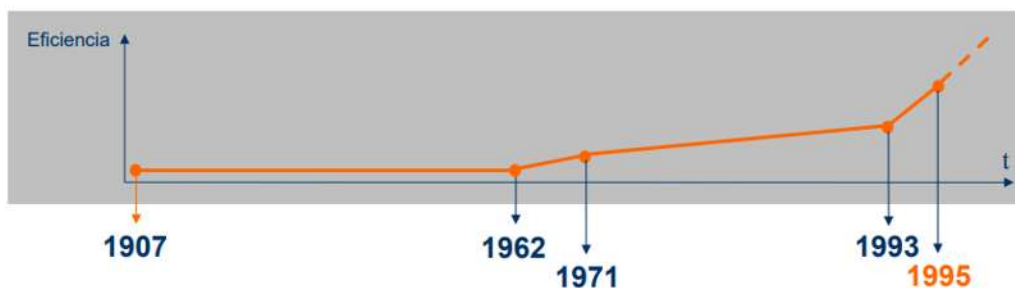


Figura 1.1.1-7. Gráfica de la evolución del LED de acuerdo a su eficiencia a través del tiempo. [16]

Se observa que desde 1907 se descubre el fenómeno físico el cual no tuvo gran crecimiento hasta 55 años después y a partir de 1993 esta tecnología creció exponencialmente hasta convertirse hoy en día en la tecnología de mayor tendencia para la producción de luz artificial.

1.1.2. Comunicación inalámbrica.

1729: Stephan Gray descubre que la electricidad puede ser transmitida a través de un cuerpo conductor. [17]

1853: Se inventa el Telégrafo por cable para transmisión simultánea en ambas direcciones (modo dúplex), se usa el método de compensación, propuesto por el físico austriaco Julius Wilhelm Gintl. [17]

Estos sistemas se basan en la transmisión de información a través de un conductor que transporta corriente eléctrica Figura 1.1.2-1. [18]

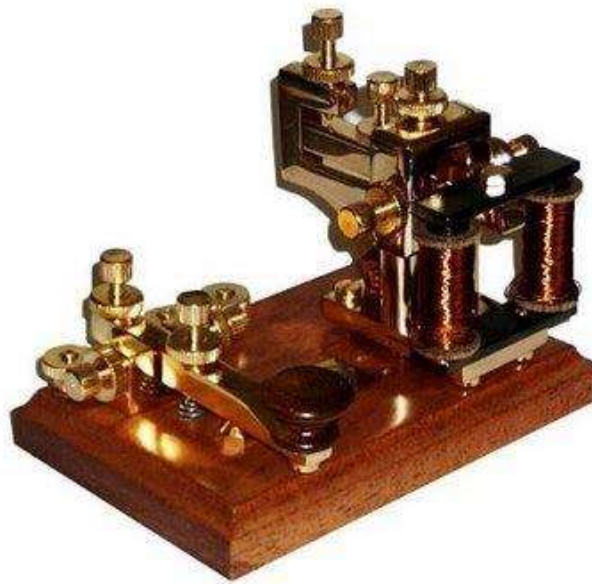


Figura 1.1.2-1 Telégrafo [18]

1861: Philip Reis demostró a varios profesores Alemanes su invento, el primer teléfono con posibilidad de transmisión de 90 metros. Reis uso una membrana animal excitada por un contacto eléctrico para producir sonidos, la recepción se lograba con un inductor galvánico oscilando de la misma forma que la membrana. [17]

1873: Se publicó la primera teoría unificada electromagnética, postulando que la luz era de naturaleza electromagnética y que era posible la radiación a otras longitudes de onda. [17]

1884: El investigador Italiano Temístocles Calzecchi Onesti establece los fundamentos científicos del cohesor. [17] Un cohesor es un dispositivo que permite la detección de ondas de radio ver Figura 1.1.2-2.

Introducción

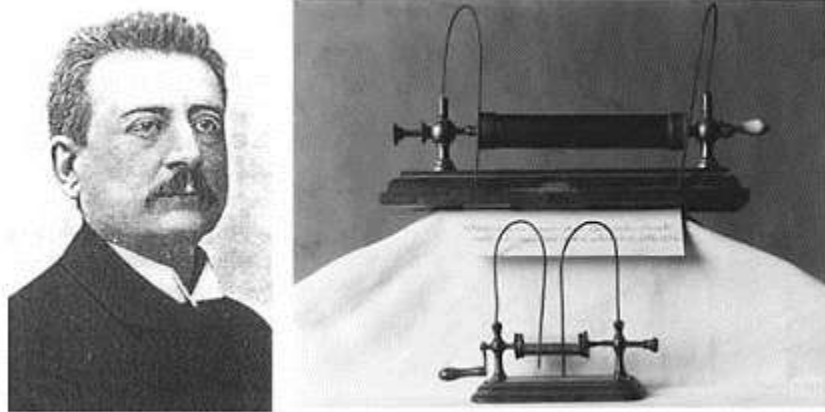


Figura 1.1.2-2. Temistocles Calzecchi fundador del cohesor. [19]

1887: El alemán Heinrich Hertz, profesor de la Universidad de Karlsruhe, da expresión matemática a la teoría de Maxwell y con su excitador y su resonador de construcción propia crea el primer detector radioeléctrico. En 1888 Hertz demuestra la existencia de ondas electromagnéticas producidas por una corriente eléctrica oscilante de gran frecuencia, demostrando que estas ondas tienen propiedades análogas a las de la luz. [20]

1891: El físico francés Edouard Branly construyó el primer receptor de ondas electromagnéticas al que denominó cohesor, logro fundamental para la radiocomunicación. Consistía en un tubo lleno de limaduras de hierro conectado a una pila y un galvanómetro ver Figura 1.1.2-3. [17]

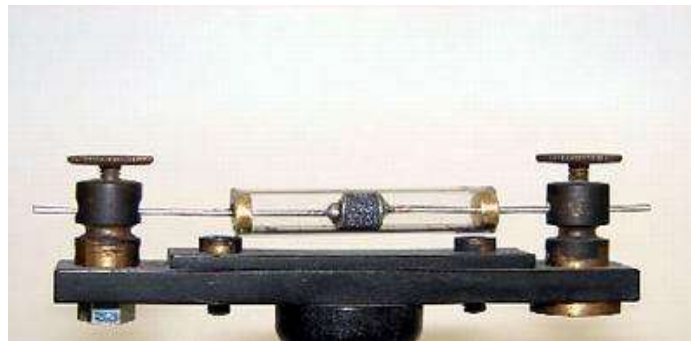


Figura 1.1.2-3 Cohesor. Detector de ondas electromagnéticas [21]

1894: El italiano Marconi ver Figura 1.1.2-4, quien efectúa la primer transmisión de señales inalámbricas a través de una distancia de 2 millas. [17]



Figura 1.1.2-4. Guillermo Marconi. Inventor italiano a quien se atribuye el invento de la radio. [22]

1894: El sabio inglés Lodge, en el Real Instituto de Londres, utilizando un excitador HERTZ y un cohesor Branly, establece la primera comunicación en morse a 36 metros de distancia. [17]

1895: El ingeniero italiano Guillermo Marconi realiza su primer experimento de transmisión de señales radioeléctricas a poca distancia. Marconi transmite señales Morse, sin ayuda de alambre de unión, a una distancia de milla y media. [17]

1896: Marconi patenta un dispositivo de perfeccionamiento en las transmisiones de impulsos y señales eléctricas. Con lo que se evoluciona a la radiotelegrafía. [17]

1899: El día 28 de marzo Marconi asombra con la primera comunicación por radio entre Inglaterra y Francia a través del Canal de la Mancha. Las primeras palabras fueron para Branly, por su contribución del cohesor. [17]

1902: Poulsen inventa su generador de arco que durante muchos años se utilizó en las emisoras de telegrafía sin hilos. Comunicaciones radioeléctricas para embarcaciones que navegaban alrededor del mundo usando código Morse. [17]

1906: Se construye en América el primer sistema para transmisión de voz a través de ondas electromagnéticas. Comienzo de la era Electrónica: rectificadores, tríodos, válvulas termoiónicas, amplificadores, etc. [17]

El período comprendido entre 1910 y 1919 se caracteriza por la construcción de transmisores con grandes antenas de baja frecuencia y elevada potencia. En la década también se introdujeron nuevas técnicas, que impactaron positivamente a la navegación y las comunicaciones con submarinos sumergidos y los sistemas de control a distancia. Nace la transmisión de amplitud modulada (AM) usando una frecuencia portadora modulada por una señal de voz. [17]

1910: Se inventa el tubo de Vacuum, dispositivo que permite transmitir voz a través de largas distancias y más de una conversación sobre el mismo cable. [17]

Introducción

1913: Meissner fabrica el primer oscilador. [17]

1918: Armstrong proyectó el circuito superheterodino, básico para receptores AM. [17]

1921: Surge la T.S.F. (por sus siglas en francés de la Telegrafía sin Hilos). [23]. Inician en París los primeros ensayos de programas de radio para el público utilizando la Torre Eiffel como antena. [17]

1926: Se descubre la Modulación en frecuencia (FM) con lo que se logra alta calidad del sonido para la radiodifusión. [17]

1928: El físico alemán Paul Nipkow, inventor de la televisión realiza la primera transmisión inalámbrica de imágenes. [17]

1930: Radioastronomía. Las interferencias que se producían en las comunicaciones de baja frecuencia (LF) fueron detectadas en los laboratorios de la Bell por el científico Karl G. Jansky, quien construyó una antena tipo cortina de Bruce de 8 elementos con reflector, funcionando en la banda de 14 metros, con función rotatoria. Con dicha antena se comprobó que el ruido estaba originado en las tormentas, pero descubrió además una fuente de ruido que estaba siempre presente, y que tenía una periodicidad de 24 horas. Tras meses de observación Karl G. Jansky determinó que estos ruidos provenían de la tierra, del sol y además había un ruido que provenía de la galaxia. [17]

1931: Allen Dumont inventa el osciloscopio. [17]

La segunda guerra mundial (1939-1945), supuso un esfuerzo considerable en el desarrollo de todas las tecnologías asociadas a las comunicaciones y a los sistemas de radar. Las investigaciones realizadas sentaron las bases para los desarrollos futuros de sistemas de aplicación civil. [17]

1940: Es instalado el primer servicio de radio teléfonos por Deutsche Reichspost entre Berlín y New York. [17]

1945: Arthur C. Clarke, propuso la utilización de los satélites geoestacionarios para los sistemas de comunicaciones de cobertura mundial. Un satélite en órbita circular ecuatorial de radio 42.242 vería siempre en la misma zona. Un satélite cubriría casi un hemisferio y con tres satélites espaciados a 120 grados se tendría una cobertura mundial. [17]

1949: Se inventan las primeras tarjetas de circuitos impresos con el fin de facilitar la localización de los componentes y abaratar los costos de los equipos electrónicos. [17]

1957: Fue lanzado al espacio el primer satélite por la URSS, era una esfera con un diámetro de 58 centímetros y un peso de 84 kilogramos, su nombre Sputnik. [17]

Introducción

1958: EE.UU. El 18 de diciembre de 1958 se lanzó el. SCORE (Signal Communicating by Orbiting Relay Equipment). La órbita era elíptica de baja altitud, con un período de 101 minutos. El satélite grababa el mensaje al pasar por una estación y lo reproducía frente a otra estación receptora. La longitud máxima del mensaje era de 4 minutos, equivalente a un canal vocal o setenta canales de teletipo de 60 palabras por minuto. La frecuencia del enlace ascendente era 150 MHz y el descendente de 132 MHz. Había un radiofaro a 108 MHz. [17]

1958: Desarrollo del circuito integrado. Primeras transmisiones de radio estereofónicas. [17]

1961: IBM Alemania introduce el concepto de Tele-Procesamiento. Los datos transmitidos serial o paralelamente a través de una línea telefónica pueden ser procesados directamente en un computador. [17]

1965: Se logran las primeras fotografías del planeta Marte transmitidas desde el satélite Mariner 4. [17]

1969: Nacimiento de Internet, gracias al desarrollo de la red de computadores ARPANET por VP Algore. [17]

1971: La Universidad de Hawaii creó el primer sistema de conmutación de paquetes mediante una red de comunicación por radio (ALOHA). Fue la primera red local inalámbrica (WLAN), y estaba formada por 7 ordenadores situados en distintas islas que se podían comunicar con un ordenador central. [24]

1979: Se publicaron los resultados de un experimento hecho por ingenieros de IBM que crearon una red local con infrarrojos en una fábrica. [25]

1982: European Telecommunications Standards Institute (ETSI) establece un patrón común: el Groupe Special Mobile (GSM), para una futura red celular de ámbito Europeo. [17]

1982: El nuevo sistema de teletipo llamado Telefax se introduce en Alemania, Suiza y Gran Bretaña; tiene capacidad de procesamiento digital y velocidad de transmisión 1200 bit/s. [17]

1992: Nace Internet comercialmente. [17]

1992: Empieza a funcionar el GSM. [17]

1994: Apareció el primer borrador del estándar IEEE 802.11. [25]

1996: Terry Wynne da la idea del más grande proyecto en cuanto a redes a nivel mundial el WWW (World Wide Web). Se desarrolla el software para transmitir voz telefónica y música de alta calidad a través de Internet. [17]

Los inicios de las tecnologías de comunicación inalámbrica fueron algo complicados, ya que cada fabricante desarrollaba sus propios modelos, incompatibles con los demás. [24]

Introducción

A finales de los 90 empresas como, Nokia o Symbol Technologies crearon la WECA (Wireless Ethernet Compatibility), que en 2003 pasó a llamarse Wi-Fi Alliance, cuyo objetivo era el fomento de la tecnología Wi-Fi y la creación de estándares para que los equipos fueran compatibles entre sí. [24]

2001: Se proporcionó servicio sobre GPRS en Europa. [25]

2002: Se hizo el lanzamiento comercial de 3G (UMTS) en la mayoría de los países de Europa. [25]

2006: Apareció el primer borrador de 802.11n, que soportaba velocidades próximas a 600 Mbps. [25]

En el año 2011 aparece el estándar Bluetooth 4.0, que destaca por la reducción significativa del consumo de batería y por una velocidad de transmisión máxima superior a 300 Mbps. [25]

En el año 2012 se empiezan a hacer las primeras pruebas sobre 4G. [25]

La Figura 1.1.2-5, muestra los avances de la comunicación inalámbrica, la cual comenzó desde que el hombre controlaba el fuego comunicándose a través de señales de humo. En 1896 Marconi invento la comunicación inalámbrica y con ello los fundamentos de la radio. En 1925 se tiene la modulación en frecuencia (FM) con lo que se logra una mejor calidad para la radiodifusión. Alrededor de 1947 se hace uso de antenas para una mejor transmisión y recepción. En 1983 se tienen los primeros inicios del teléfono móvil de 1995 a 2010 la tecnología celular tuvo grandes y constantes cambios tanto de hardware, software así como el método de transmisión vía inalámbrica tales como: GSM (mensajes multimedia), transmisión por radiación infrarroja, la tecnología bluetooth. Apartir del 2013 se volvió común el uso de dispositivos con tecnología inalámbrica.

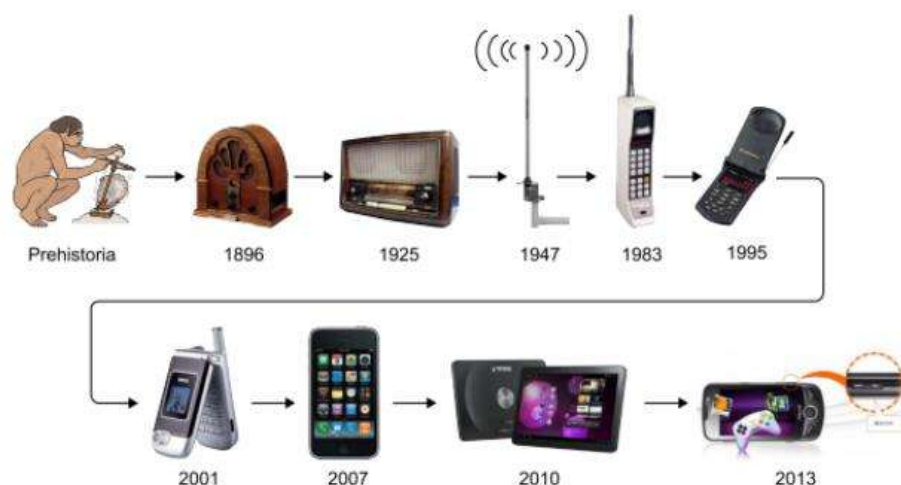


Figura 1.1.2-5 Evolución de los dispositivos de comunicación inalámbrica. [25]

1.2. Objetivo

1.2.1. Objetivo General

Incrementar las funciones de los sistemas de iluminación, mejorando las prestaciones para su control y permitiendo que sea realizado por cualquier usuario a través de un sitio web, utilizando herramientas conocidas hoy en día y de fácil acceso, creando así un sistema amigable con el usuario de fácil implementación y bajo costo.

1.2.2. Objetivos Particulares.

- Control de iluminación LED y servomotores por medio de una red inalámbrica Wi-Fi.
- Utilizar módulos Wi-Fi que conecten cada elemento del sistema.
- Crear una red de dispositivos Wi-Fi utilizando un módulo Wi-Fi, en modo servidor.
- Elaborar una interfaz WEB que sea accesible y amigable para cualquier usuario.

Los objetivos obtenidos en este proyecto, serán las bases para realizar posteriormente sistemas inalámbricos con mayor complejidad, con una interfaz de usuario más interactivo.

1.3. Retos Personales

Tengo una particular atracción por los efectos realizados con LEDs, y por ende los sistemas de iluminación. Uno de los dispositivos que utilizan estos sistemas son cabezas robot, el cual fue la inspiración en este proyecto, como se muestra en la Figura 1.3-1.



Figura 1.3-1. Cabeza robótica de iluminación LED RGB. [26]

Es interesante ver su capacidad y así realizar algo similar en funcionamiento y mejorar las características de operación.

Introducción

Existen dispositivos como el de la Figura 1.3-2 utilizados para iluminación en el hogar, los cuales permiten el cambio de posición en diferentes ángulos, sin embargo el ajuste de posición se realiza manualmente.

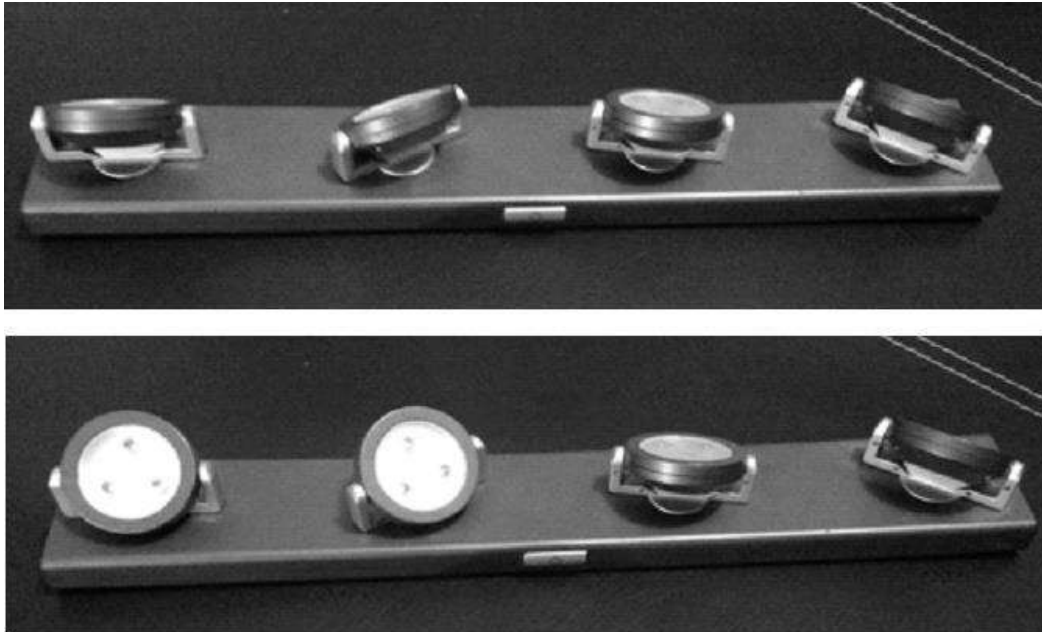


Figura 1.3-2 Dispositivos de Iluminación. [27]

De ahí nace la idea de enfocarme en un prototipo que permita el control de un dispositivo con características similares de los dispositivos de la Figura 1.3-1 y Figura 1.3-2, aplicando los conocimientos adquiridos en la carrera de electrónica y utilizar tecnología que se tiene al alcance para ofrecer un sistema que compita con cualquier sistema de iluminación para el hogar sin necesidad de realizar grandes costos.

1.4. Justificación

- La tecnología está en constante desarrollo y es importante adquirir nuevos conocimientos para aprovechar ésta al máximo, de esta manera desarrollar sistemas más complejos.
- La bombilla incandescente ha sido remplazada por iluminación LED debido a sus ventajas, una de ellas son los costos que tiene producir luz artificial mediante la bombilla incandescente por lo que este sistema debe reforzar el uso de tecnología LED.
- Hoy en día surge la tecnología llamada internet de las cosas (IoT). Aprovechare el proyecto para adentrarme más a ella.
- Satisfacer necesidades como:
 - Acceso remoto.
 - Control de nivel de iluminación.
 - Control de dirección de iluminación.
 - Ahorro en consumo energético.
 - Reducir el uso de cableado y dispositivos robustos, difíciles y tediosos de instalar.
 - Realizar el control de iluminación de todo un establecimiento u hogar desde un solo dispositivo.
- Ofrecer un sistema de iluminación accesible para todo público, de bajo costo, fácil implementación y amigable para su control.

1.5. Metodología

Investigar el funcionamiento de los módulos ESP8266. Consulta sobre NodeMCU y uso del lenguaje de programación LUA.

Desarrollo de scripts LUA que permiten el control de los actuadores por medio de los GPIO que tienen los módulos ESP8266.

Aprender a desarrollar páginas web con HTML, JavaScript y CSS.

Controlar desde la página web, LEDs, LEDs RGB y servomotores.

Diseño del sistema e implementación de Hardware y, posteriormente desarrollo de software considerando todo el sistema en conjunto.

2. Capítulo 2. Marco Teórico.

En éste capítulo se presenta los fundamentos teóricos para el desarrollo del proyecto, introduciendo los conocimientos previos tanto de hardware como de software entre otras definiciones que permiten generar las ideas básicas para el diseño, desarrollo y entendimiento del sistema.

2.1. Sistemas de iluminación

Los sistemas de control de iluminación no son más que el conjunto de elementos que regula el comportamiento de una o varias luminarias, en función de determinados parámetros. [28]

Regular y controlar el sistema de iluminación de forma activa, algo que va más allá del tradicional interruptor de encendido y apagado, nos permite adaptar la luz del espacio a cualquier situación. En otras palabras hay una mejora en el confort o la optimización del consumo energético, entre otros. [28]

En los sistemas de control de iluminación se encuentran varios dispositivos relacionados con el control lumínico, que se interconectan y nos permiten actuar sobre diferentes luminarias desde un único interfaz. [28]

Funciones y beneficios.

Algunas de las funciones y beneficios que tiene emplear un sistema de control en un sistema de iluminación son los siguientes:

- Automatización y programación.
- Ajuste de niveles de iluminación.
- Comunicación y conectividad.
- Ahorro y eficiencia en el sector energético.
- Reducción de contaminación lumínica.
- Vida más larga de lámparas
- Confort
- Flexibilidad
- Calidad de iluminación
- Seguridad. [29]

2.1.1. Sistemas de control de iluminación analógicos

El Triac es un dispositivo electrónico semiconductor que regula la intensidad de la iluminación mediante el control de la fase de la señal de corriente alterna que alimenta el sistema. [28]

La regulación 1-10V es uno de los primeros métodos de regulación y se basa en una señal de corriente continua, donde 10V corresponde al máximo de luz, 1V al mínimo y 0V indica que la luminaria está apagada. [28]

Para el control de escenarios o espacios similares se empleaban más complejos como AMX192 o D54 que se han ido remplazando por sus equivalentes digitales. [28]

2.1.2. Sistemas de control de Iluminación Digitales

Uno de los sistemas digitales más extensos es DALI (Digital Addressable Lighting Interface), que sitúa una posición intermedia entre funcionalidad y costo de implementación. Su funcionamiento se basa en un controlador y uno o más elementos esclavos. Se pueden controlar hasta 64 dispositivos de forma independiente, por grupos o todos a la vez mediante comandos broadcast. [28]

Otros protocolos de control es DMX512 que permite controlar hasta 512 canales que se traduce, normalmente, en hasta 32 dispositivos. También está el DSI, muy similar al DALI, pero requiere un cable de control por dispositivo. También se emplean estándares como KNX que no sólo es un sistema de iluminación, si no que permite controlar sistemas de ventilación, de alarma o de distribución de audio y video, entre otros. [28]

2.1.3. Luminotecnia

La Luminotecnia es la ciencia que estudia las distintas formas de producción de luz, así como su control y aplicación con fines domésticos, industriales o artísticos. [2]

Los sistemas de control de la iluminación son el conjunto de elementos que regula el comportamiento de una o varias luminarias. Con el desarrollo de la Domótica y el IoT (Internet of Things) este tipo de sistemas constituyen la base de la iluminación moderna. [2]

El estudio de la iluminación tiene una doble variante:

- Se debe plantear como un fenómeno físico.
- Se debe plantear como técnica aplicada.

2.2. Introducción a las redes Wi-Fi.

Hoy en día, existe en el mercado una gran cantidad de posibilidades para implementar una red inalámbrica. Cada una intenta responder a ciertas necesidades y normalmente cada posibilidad pertenece a una cierta compañía que apostó por ésta. Las compañías necesitan que sus productos sean compatibles con los de otras compañías, de ahí que surja la necesidad de tener un estándar que seguir. [30]

Que sea interoperable con equipos de diferentes fabricantes y puede ser extendida para ofrecer funcionalidades mucho más allá de las previstas originalmente por los fabricantes. [31]

La especificación IEEE 802.11 (ISO/IEC 8802-11) es un estándar internacional que define las características de una red de área local inalámbrica (WLAN). Wi-Fi (que significa "Fidelidad inalámbrica", a veces incorrectamente abreviado WiFi) es el nombre de la certificación otorgada por la Wi-Fi Alliance, anteriormente WECA (Wireless Ethernet Compatibility Alliance), grupo que garantiza la compatibilidad entre dispositivos que utilizan el estándar 802.11. Por el uso indebido de los términos (y por razones de marketing) el nombre del estándar se confunde con el nombre de la certificación. Una red Wi-Fi es en realidad una red que cumple con el estándar 802.11. [32]

A los dispositivos certificados por la Wi-Fi Alliance se les permite usar este logotipo ver Figura 2.2-1.



Figura 2.2-1. Logotipo de Wi-Fi. [33]

Esto se debe a que Wi-Fi utiliza estándares abiertos: enrutadores, tablet PCs, laptops y teléfonos Wi-Fi pueden interoperar porque todos se adhieren al estándar 802.11.

BANDAS ISM / UNII

La mayoría de los dispositivos inalámbricos comerciales (teléfonos móviles, televisión, radio, etc.) usan frecuencias de radio adjudicadas mediante una licencia. Las grandes organizaciones pagan elevados cánones por el derecho de utilizar esas frecuencias. Wi-Fi utiliza porciones del espectro que no requieren licencia. [31]

- Las bandas ISM (Industrial, Scientific and Medical) permiten el uso de las porciones 2.4-2.5 GHz, 5.8 GHz, y muchas otras frecuencias (no utilizadas en Wi-Fi).
- Las bandas UNII (Unlicensed National Information Infrastructure) permiten el uso sin licencia de otras porciones del espectro de 5 GHz.

Nota: Las bandas “UNII” están definidas y reguladas únicamente en EEUU, otros países utilizan una nomenclatura distinta y otras reglas para el uso de estas frecuencias. ISM es una recomendación de la UIT (Unión Internacional de Telecomunicaciones) de aplicación internacional. [31]

2.2.1. Protocolos de Redes Inalámbricas

La familia de protocolos 802.11 son la base de Wi-Fi.

- 802.11a permite hasta 54 Mbps en las bandas no licenciada a 5 GHz.
- 802.11b permite hasta 11 Mbps en las bandas no licenciada a 2.4 GHz.
- 802.11g permite hasta 54 Mbps en las bandas no licenciada a 2.4 GHz.
- 802.11n permite hasta 600 Mbps en las bandas no licenciada a 2.4GHz y 5 GHz.

Las tecnologías específicas utilizadas por los equipos Wi-Fi incluyen 802.11a, b, g, y n. 802.11n fue ratificado por IEEE en septiembre 2009, es un estándar reciente. [31]

802.11g es compatible con 802.11b, y 802.11n es compatible con 802.11a cuando opera a 5 GHz, y con b/g en la banda de 2.4 GHz. 802.11n puede utilizar dos canales adyacentes de 20 MHz, para un total de 40MHz lo que no está contemplado en los estándares anteriores, y de esta manera puede alcanzar rendimientos reales superiores a 100 Mbps. El estándar permite inclusive mejorar esta cifra usando múltiples flujos de datos y ya existen equipos que utilizan esta modalidad. [31]

802.11a, b, y g son ahora parte del estándar IEEE 802.11-2007 que comprende todas las enmiendas ratificadas hasta ese año, incluyendo 802.11e que permite QoS (calidad de Servicio). [31]

Obsérvese que WiMAX es una tecnología completamente diferente de Wi-Fi, está basada en estándares diferentes y puede operar tanto en bandas licenciadas como exentas de licencia. [31]

2.2.2. Tasas de transmisión

Note que las “tasas de transmisión” mencionadas en las especificaciones de equipos Wi-Fi se refieren a la tasa de transmisión total de los símbolos, no al verdadero caudal o rendimiento de la transmisión a nivel de TCP/IP. La diferencia es lo que se conoce como protocol overhead (tasa debida al protocolo) y es utilizada por el protocolo Wi-Fi para

manejar colisiones, retransmisiones y en general la gestión del enlace. Una regla general es que el caudal máximo a nivel TCP/IP es la mitad de la tasa de símbolos. [31]

Por ejemplo, un enlace 802.11 a 54 Mbps tiene un rendimiento máximo práctico de unos 25 Mbps. Un enlace 802.11b tiene un rendimiento máximo de transmisión de 5 Mbps. [31]

Wi-Fi usa el mismo canal para ambas direcciones de tráfico, así que cuando un radio transmite no puede recibir. Esto limita el rendimiento. [31]

2.2.3. Capa física protocolo TCP/IP

Los dispositivos Wi-Fi deben escoger ciertos parámetros antes de poder establecer la comunicación. Estos parámetros deben configurarse adecuadamente para poder establecer conectividad “a nivel de la capa uno”. [31]

Tabla 1. PROTOCOLO TCP/IP

Pila de protocolos TCP/IP	
5	Aplicación
4	Transporte
3	Internet
2	Enlace de datos
1	Física

Los parámetros para esta capa son:

- Canal de radio.
- Modo de operación del radio.
- Nombre de la red.
- Tipo de seguridad.

Todos los dispositivos deben compartir el mismo canal, si no, no podrán “escucharse” entre sí. El modo de operación del radio debe escogerse adecuadamente para que exista comunicación. El nombre de la red (también llamado ESSID) debe ser el mismo para todos los dispositivos que se quiere comunicar. Cualquier mecanismo de seguridad también debe configurarse adecuadamente.

Al igual que como se conoce la capa física de la red Ethernet que se refiere a un cable. Si los parámetros mencionados anteriormente de la red Wi-Fi no están establecidos correctamente es como si el cable estuviera desenchufado.

2.2.4. Canales en 802.11 (Wi-Fi)

Los dispositivos Wi-Fi utilizan una modalidad de transmisión llamada half-duplex. Envían y reciben en el mismo canal, por lo que sólo un dispositivo puede transmitir en un instante determinado.

Cómo se aprecia la Figura 2.2.4-1 los canales están separados cada 5 MHz, pero las señales 802.11 ocupan 22 MHz.

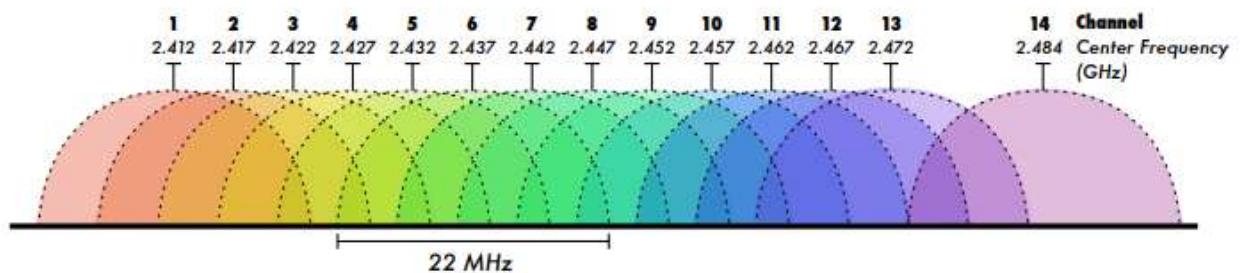


Figura 2.2.4-1. Espectro de frecuencias de transmisión en Wi-Fi. [31]

Para evitar interferencias se deben escoger canales que no se solapen, es decir que las respectivas señales no se superpongan en ninguna parte del espectro.

Transmitir en canales sin solapamiento permitirá no tener interferencia entre canales, estos tres canales nos permiten un amplio funcionamiento adecuado en los AP sin tener interferencia. Figura 2.2.4-2.

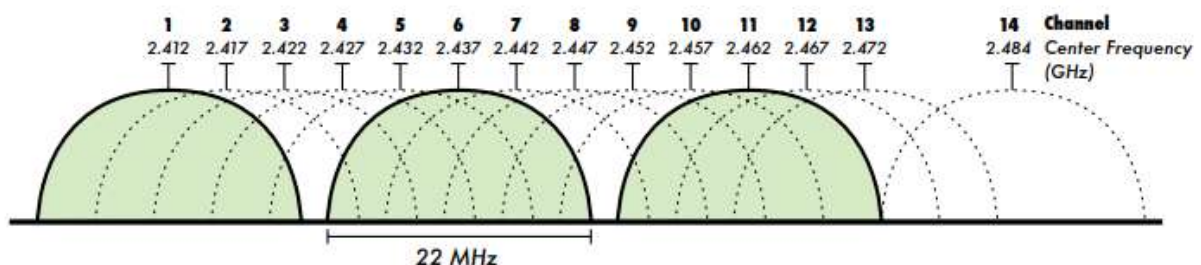


Figura 2.2.4-2. Canales sin solapamiento en la transmisión Wi-Fi (1, 6 y 11). [31]

2.2.5. Topología de redes inalámbricas

Cualquier tipo de red inalámbrica aunque no sea Wi-Fi, estará constituida por la combinación de estas configuraciones básicas.

- Punto a punto.
- Punto a multipunto.
- Multipunto a multipunto.

1.- Punto a Punto:

La conexión más simple es un enlace punto-a-punto como se muestra en la Figura 2.2.5-1. Estos enlaces pueden usarse para extender su red a grandes distancias. Los enlaces punto a punto ofrecen el mayor caudal posible.

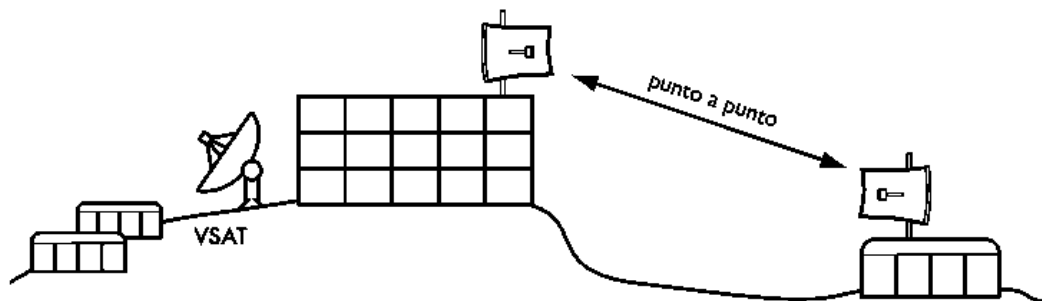


Figura 2.2.5-1. Topología punto a punto. [31]

2.- Punto a Multipunto:

Cuando más de un nodo debe comunicarse con un punto central se tiene una red punto-a-multipunto ver Figura 2.2.5-2.

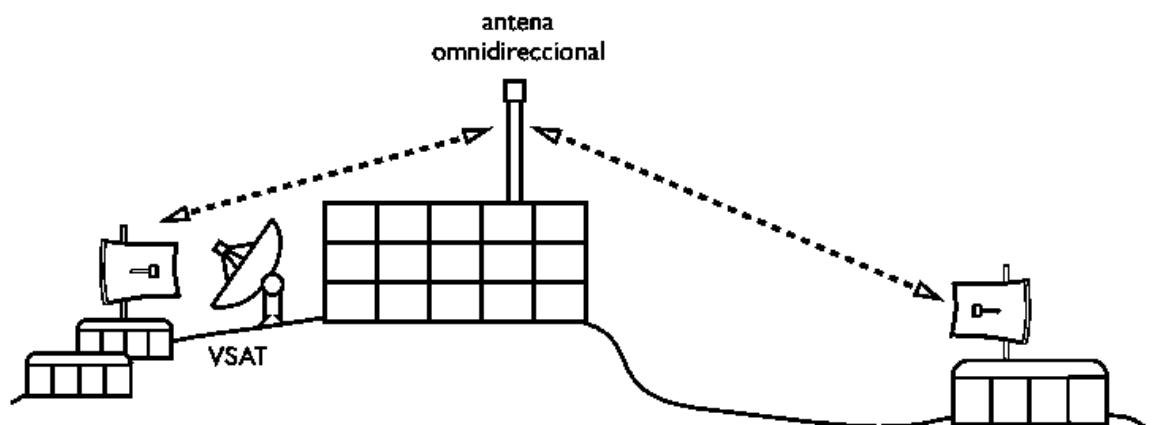


Figura 2.2.5-2. Topología punto a multipunto. [31]

La red punto a multipunto es la topología más común. Considere el caso de un AP con muchos clientes.

El diseño de redes punto a multipunto es muy diferente del de las redes punto a punto. No se puede simplemente reemplazar una antena parabólica por una omnidireccional y esperar que eso sea todo. La transición de punto a punto a multipunto aumenta la complejidad porque ahora se tienen múltiples nodos que compiten por los recursos de la red. El resultado neto es que el caudal total disminuye. [31]

3.- Multipunto a Multipunto:

Cuando cada nodo de una red puede comunicarse con cualquier otro tenemos una red multipunto a multipunto, también conocida como red en malla (mesh) o ad-hoc ver Figura 2.2.5-3.

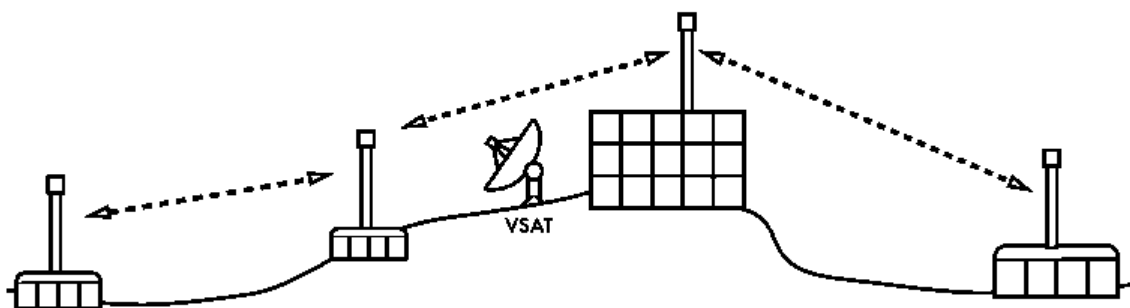


Figura 2.2.5-3 Topología multipunto a multipunto. [31]

Las redes Multipunto a Multipunto son considerablemente más complejas, pero también mucho más flexibles que las redes punto a multipunto. No hay una autoridad central en una red en malla. El protocolo de malla automáticamente añade nuevos nodos a la medida que se incorporan a la red, sin necesidad de cambiar la configuración de ninguno de los nodos existentes. Esta topología reduce significativamente el caudal al máximo posible.

Modos de funcionamiento en Wi-Fi:

Los dispositivos Wi-Fi pueden operar en alguno de los siguientes modos: Cada modo tiene restricciones de operación específicas, y los radios sólo pueden operar en un modo en un momento determinado.

- Master (AP -access point-)
- Managed (también llamado cliente o estación)
- Ad-hoc (usado en redes en malla)
- Monitor (no usado normalmente para comunicaciones)

Los radios Wi-Fi pueden operar en uno sólo de estos cuatro modos en un momento determinado. Esto significa que el mismo radio no puede funcionar simultáneamente como AP y como cliente. Pero existen enrutadores inalámbricos que aceptan más de un radio en cuyo caso se puede tener un radio funcionando como AP (Access Point) y otro como cliente. Esto se usa a menudo en redes en malla para aumentar el rendimiento. [31]

Modo master:

El modo master (también llamado modo AP o de infraestructura) se usa para instalar una red con un AP (punto de acceso) que conecta a diferentes clientes. El AP crea una red con un nombre específico (denominado SSID ó ESSID) y un canal sobre el cual se ofrecen los servicios de la red. Los dispositivos Wi-Fi en modo master pueden comunicarse sólo con los dispositivos asociados a ellos que estén en modo managed.

Los AP crean redes Wi-Fi punto a multipunto. Un radio operando en el modo master funciona como un AP, anunciando una red con cierto nombre en un determinado canal y permite que los clientes se le conecten. Puede haber limitaciones en el número máximo de clientes permitidos (el límite depende del modelo de AP usado).

Modo Managed:

El modo Managed es llamado también modo cliente. Los dispositivos inalámbricos en modo managed se unirán a una red creada por el master y automáticamente cambiarán el canal para ajustarse al del master.

Los radios en modo managed no pueden comunicarse directamente entre sí y sólo se pueden comunicar con el master al cual están asociados.

Modo ad-hoc:

El modo Ad-hoc mode se usa para crear redes en malla donde:

- No hay dispositivos en modo master (AP)
- Se realiza la comunicación directamente entre todos los nodos

Los dispositivos deben estar dentro de su rango de cobertura para poder comunicarse y deben escoger un nombre de red y canal común.

El modo Ad-hoc se usa para crear una red en malla, es decir una red multipunto a multipunto donde no hay ningún master.

Por ejemplo: El modo Ad hoc también puede usarse para conectar dos laptops equipados con Wi-Fi sin utilizar un AP.

Modo monitor:

El modo Monitor se usa para escuchar pasivamente todo el tráfico en un canal dado. Es útil para:

- Analizar los problemas en un enlace inalámbrico.
- Observar el uso del espectro en una zona.
- Realizar tareas de mantenimiento y de seguridad. [31]

2.3. Microcontroladores

Las aplicaciones de los microcontroladores son vastas, se puede decir que solo están limitadas por la imaginación del usuario. Es común encontrar microcontroladores en campos como la robótica y la automatización, en la industria del entretenimiento, en telecomunicaciones, en instrumentación, en el hogar, en la industria automotriz, etc.

2.3.1. ¿Qué es un microcontrolador?

Un microcontrolador (abreviado μC , UC o MCU) es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica. Un microcontrolador incluye en su interior las tres principales unidades funcionales de una computadora: unidad central de procesamiento (CPU), unidades de memoria (RAM y ROM) y periféricos de entrada/salida. Estas partes están interconectadas dentro del microcontrolador, y juntas forman lo que se conoce como microcomputadora. Vale la pena mencionar que sin un programa, los microcontroladores carecen de utilidad. [34]

El objetivo fundamental de los microcontroladores es leer y ejecutar los programas que el usuario les escribe, la programación es una actividad básica e indispensable al diseñar los circuitos y sistemas que los incluyen. La naturaleza programable de los microcontroladores simplifica el diseño de los circuitos electrónicos. Permiten modularidad y flexibilidad, ya que el mismo circuito se puede usar para realizar diferentes funciones simplemente cambiando el programa del microcontrolador. [34]

En la Figura 2.3.1-1 se muestran las partes que constituyen un microcontrolador, es recomendable profundizar en su arquitectura interna, con este conocimiento puede tomar más y mejor las capacidades de un microcontrolador.

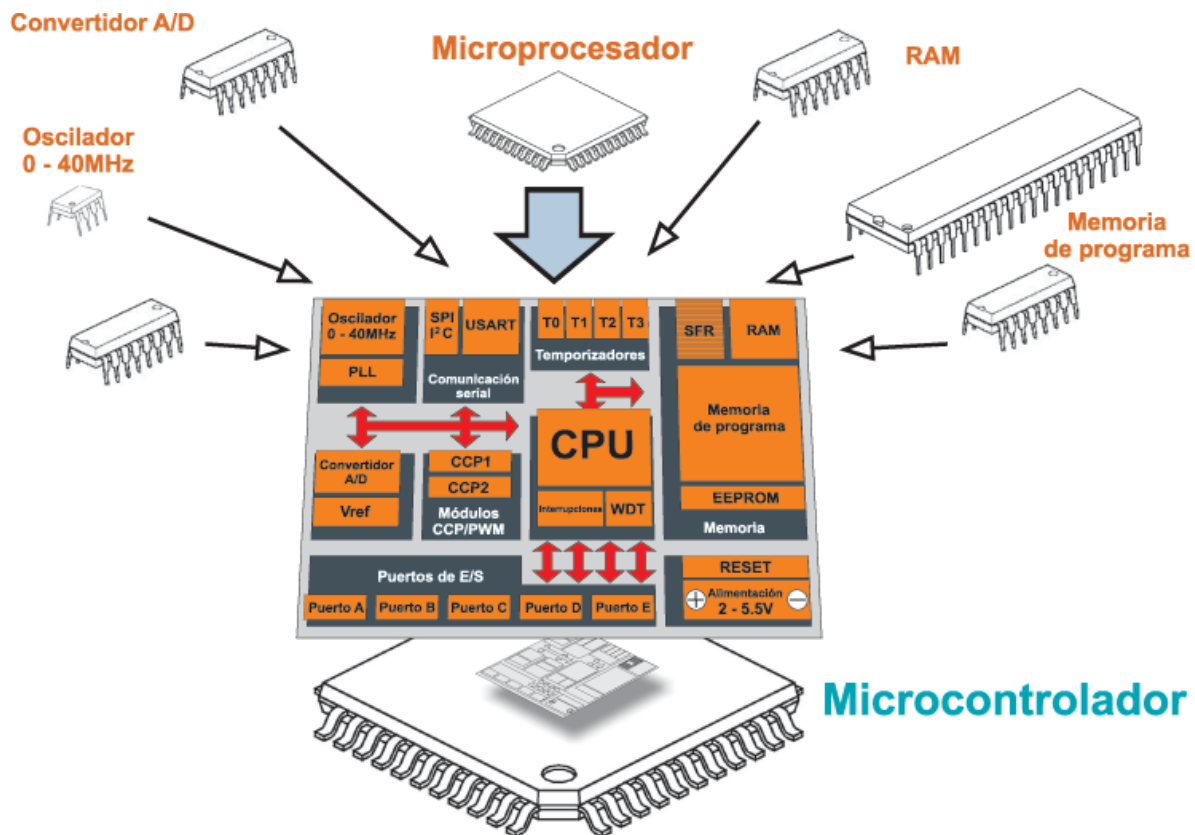


Figura 2.3.1-1. Arquitectura de un microcontrolador. [35]

2.3.2. ¿Cómo es la escritura de los programas?

Los microcontroladores están diseñados para interpretar y procesar datos e instrucciones en forma binaria. Los patrones de 1 y 0 conforman el lenguaje de máquina de los microcontroladores, y es lo único que son capaces de entender. Estos 1 y 0 representan la unidad mínima de información, conocida como bit, ya que solo puede adoptar uno de los dos valores posibles: 0 o 1. [34]

Conocido como lenguaje de bajo nivel. Debido a que las instrucciones no son tan específicas para el lenguaje humano.

Esta es la razón por la que la programación se lleva a cabo comúnmente en un lenguaje de alto nivel, es decir, un lenguaje que utiliza frases o palabras similares o típicas del lenguaje humano como lo son C o BASIC.

Otro tipo de lenguaje más especializado es el lenguaje ensamblador. El lenguaje ensamblador es una lista con un número limitado de instrucciones a las que un microcontrolador puede responder. Estas instrucciones son palabras o abreviaturas que representan las instrucciones del lenguaje de la máquina del microcontrolador.

Las instrucciones de lenguaje ensamblador, también conocidas como mnemónicos, son fáciles de entender y permiten operar directamente con los registros de memoria y con las instrucciones intrínsecas del microcontrolador. El lenguaje ensamblador es sin duda el lenguaje por excelencia en la programación de microcontroladores, permite un uso eficiente de la memoria y minimiza el tiempo de ejecución de un programa. [34]

2.3.3. Consumo de Energía.

El microcontrolador se caracteriza por un muy bajo consumo de energía durante su estado activo, requiere menores tiempos para el procesamiento de datos, al igual que para despertarse rápidamente (salir de modo Sleep), además de contar con la virtud de requerir muy baja corriente en modo de espera.

Algunos microcontroladores pueden utilizar palabras de cuatro bits y funcionan a velocidad de reloj con frecuencias tan bajas como 4 kHz, con un consumo de baja potencia (mW o microvatios). Por lo general, tendrá la capacidad de mantenerse a la espera de un evento como pulsar un botón o de otra interrupción; así, el consumo de energía durante el estado de reposo (reloj de la CPU y los periféricos de la mayoría) puede ser sólo de nanovatios, lo que hace que muchos de ellos sean muy adecuados para aplicaciones con batería de larga duración.

Otros microcontroladores pueden servir para roles de rendimiento crítico, donde sea necesario actuar más como un procesador digital de señal (DSP), con velocidades de reloj y consumo de energía más altos. [34]

Los consumos varían dependiendo del tipo de microcontrolador que se utilice.

2.4. ESP8266

ESP8266 es el nombre de un microcontrolador diseñado por Espressif Systems, una compañía china con sede en Shanghai. El volumen de producción de estos microcontroladores no empezó hasta principios de 2014. [36]

El ESP8266 se anuncia a sí mismo como una solución autónoma de redes Wi-Fi que se ofrece como un puente entre los microcontroladores que hasta ahora existían hasta los MCU con Wi-Fi, siendo además capaz de ejecutar aplicaciones independientes. [36]

Se da lugar a varias versiones de ESP8266 (Ej: ESP-01, ESP-02...) pero todas con el mismo procesador, lo que las diferencian son el número de pines GPIO expuestos, la cantidad de memoria flash, las dimensiones, la forma de exponer los pines, y otras consideraciones relativas a su construcción. Pero desde una perspectiva de programación todas son iguales. [36]

El circuito integrado ESP8266 viene en un pequeño paquete, tal vez cinco milímetros cuadrados. Obviamente, a menos que se sea un maestro soldador no va a hacer mucho con eso. Las buenas noticias son que una serie de proveedores han creado placas breakout que permiten el trabajo mucho más simple, ya que estas tarjetas contienen un único componente que, debido a sus reducidas dimensiones no pueden ser conectados directamente a una placa de desarrollo para sus salidas y por lo que son ampliados con el fin de tener acceso a sus terminales.

Esto permite trabajar con este componente único acoplado a un microcontrolador, para desarrollar un proyecto, para evaluar su funcionamiento, o para complementar un proyecto que requiere su presencia. Este es el caso de un circuito integrado que tiene sus pines accesibles de modo que puedan ser fácilmente conectados a un microcontrolador, es equivalente a una matriz de contacto, pero en la forma de una placa de circuito impreso con algún componente o componentes ya soldadas.

2.4.1. Especificaciones de ESP8266

En la Figura 2.4.1-1 se muestra el pinout del microcontrolador utilizado en ESP8266.

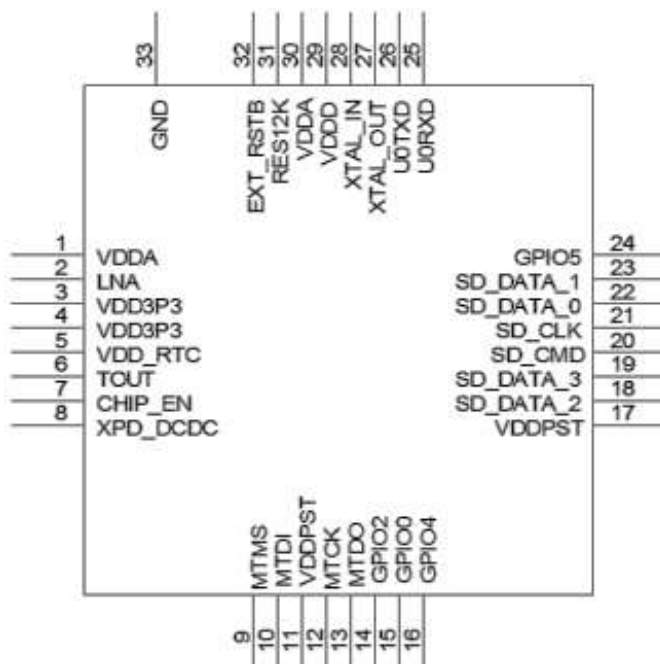


Figura 2.4.1-1 ESP8266 Pinout. [37]

Tabla 2. ESPECIFICACIONES ESP8266x

categorías	Items	Parámetros
Wi-Fi	Certificación	Wi-Fi Alliance
	Protocolos	802.11 b/g/n
	Rango de frecuencia	2.4G~2.5G (2400M ~ 2483.5M)
	Poder de transmisión (Tx power)	802.11 b: +20 dBm
		802.11 g: +17 dBm
		802.11 n: +14 dBm
		802.11 b: -91 dbm (11 Mbps)
	Sensibilidad de recepción (Rx Sensitivity)	802.11 g: -75 dbm (54 Mbps)
		802.11 n: -72 dbm (MCS7)
	Antena	PCB Trace, External, IPEX Connector, Ceramic Chip
Hardware	CPU	Tensilica L106 32-bit processor
	Interfaz periférica	UART/SDIO/SPI/I2C/I2S/IR Control Remoto
		GPIO/ADC/PWM/LED Iluminación y botón
	Voltaje de operación	2.5V ~ 3.6V
	Corriente de operación	Valor promedio: 80 mA
	Rango de temperatura de funcionamiento	-40°C ~ 125°C
	Rango de temperatura de almacenamiento	-40°C ~ 125°C
	Tamaño de encapsulado	QFN32-pin (5 mm x 5 mm)
Software	Modo Wi-Fi	Station/SoftAP/SoftAP+Station
	Seguridad	WPA/WPA2
	Encriptación	WEP/TKIP/AES
	Actualización de Firmware	Descarga, por medio de UART / OTA (via red)
	Desarrollo de software	Compatible con Cloud Server Development/ Firmware y SDK para una rápida programación en chip
	Protocolos de red	IPv4, TCP/UDP/HTTP/FTP
	Configuración de usuarios	AT Instruction Set, Cloud Server, Android/iOS App

Tabla 3. FUNCIÓN DE PINES.

Pin	Nombre	Tipo	Función
1	VDDA	P	Voltaje Análogo 2.5V ~ 3.6V
2	LNA	I/O	RF antena Chip interfaz de salida impedancia=39+j6 Ω . Se sugiere mantener la red de adaptación de tipo π para que coincida con la antena.
3	VDD3P3	P	Voltaje de Amplificación 2.5V ~ 3.6V
4	VDD3P3	P	Voltaje de Amplificación 2.5V ~ 3.6V
5	VDD_RTC	P	NC (1.1V)
6	TOUT	I	ADC pin Se puede usar para probar el voltaje de la fuente de alimentación de VDD3P3 (Pin3 and Pin4) y la entrada de voltaje de TOUT (Pin 6). Sin embargo, estas dos funciones no se pueden usar simultáneamente.
7	CHIP_PU		Habilita el chip en Alto: On, el chip trabaja apropiadamente. Bajo: Off, corriente de bajo consumo
8	XPD_DCDC	I/O	Deep-sleep wakeup (necesita ser conectado a EXT_RSTB); GPIO16
9	MTMS	I/O	GPIO 14; HSPI_CLK
10	MTDI	I/O	GPIO 12; HSPI_MISO
11	VDDPST	P	E/S Digital fuente de voltaje (1.8V ~ 3.6V)
12	MTCK	I/O	GPIO 13; HSPI_MOSI; UART0_CTS
13	MTDO	I/O	GPIO 15; HSPI_CS; UART0_RTS
14	GPIO2	I/O	UART Tx durante la programación de flash; GPIO2
15	GPIO0	I/O	GPIO0; SPI_CS2
16	GPIO4	I/O	GPIO4
17	VDDPST	P	Digital/IO Fuente de poder (1.8V ~ 3.6V)
18	SDIO_DATA_2	I/O	Conectado a SD_D2 (Series R: 200 Ω); SPIHD; HSPIHD; GPIO9
19	SDIO_DATA_3	I/O	Conectado a SD_D3 (Series R: 200 Ω); SPIWP; HSPIWP; GPIO10
20	SDIO_CMD	I/O	Conectado a SD_CMD (Series R: 200 Ω); SPI_CS0; GPIO11
21	SDIO_CLK	I/O	Conectado a SD_CLK (Series R: 200 Ω); SPI_CLK; GPIO6
22	SDIO_DATA_0	I/O	Conectado a SD_D0 (Series R: 200 Ω); SPI_MISO; GPIO7
23	SDIO_DATA_1	I/O	Conectado a SD_D1 (Series R: 200 Ω); SPI_MISO; GPIO8
24	GPIO5	I/O	GPIO5
25	5 U0RXD	I/O	UART Rx durante la programación de flash; GPIO3
26	U0TXD	I/O	UART Tx durante la programación de flash; GPIO1; SPI_CS1
27	XTAL_OUT	I/O	Conectado a la salida del oscilador de cristal, puede ser usado para proporcionar BT entrada de señal de reloj
28	XTAL_IN	I/O	Conectado a la entrada del oscilador de cristal
29	VDDD	P	Voltaje Análogo 2.5V ~ 3.6V

Pin	Nombre	Tipo	Función
30	VDDA	P	Voltaje Análogo 2.5V ~ 3.6V
31	RES12K	I	Conector serial con un resistor de 12 k Ω y conectado a tierra.
32	EXT_RSTB	I	Señal de reseteo externo (Nivel de bajo voltaje: active

Tabla 4 CARACTERISTICAS GENERALES DE ESP8266

Nombre	Parámetro
Voltaje	3.3 V
Consumo de corriente	10 μ A – 170Ma
Memoria Flash	16MB máx. (512 k normal)
Procesador	Tensilica L106 32bit
Velocidad del procesador	80 – 160 MHz
GPIOs	17
ADC	1 entrada con 10 bits de resolución (1024 valores)
Soporte de 802.11	b/g/n/d/e/i/k/r

2.4.2. Procesador

El sistema en el chip (SoC) ESP9266EX usa un microcontrolador Tensilica Xtensa L106, que contiene en su interior un procesador de 32 bit con instrucciones de 16 bit. [36] Ver Figura 2.4.2-1.



Figura 2.4.2-1 C.I. Procesador ESP8266EX. [36]

El SoC describe la tendencia cada vez más frecuente de usar tecnologías de fabricación que integran todos o gran parte de los módulos que componen un computador o cualquier otro sistema informático o electrónico en un único circuito integrado o chip. Como se puede ver en la Figura 2.4.2-2.

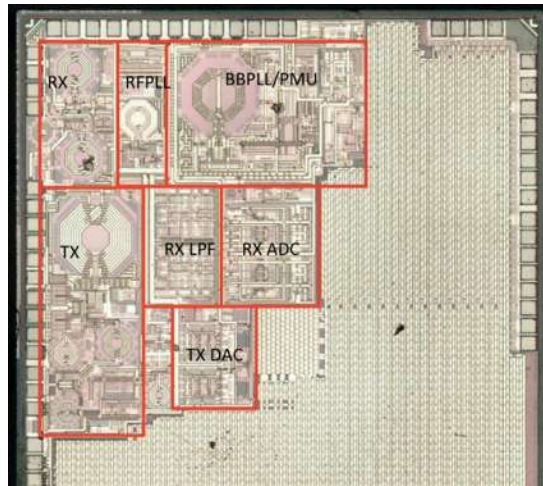


Figura 2.4.2-2 Interior de un ESP8266 y sus periféricos. [36]

Se trata de un SoC o Sistema en Chip. Básicamente consiste en un chip que tiene todo integrado (o casi todo) para que pueda funcionar de forma autónoma como si fuera un ordenador. En el caso del ESP8266 lo único que no tiene es una memoria para almacenar los programas. Esto supone un inconveniente ya que parte de los pines de entrada y salida, tendrán que ser utilizados para conectarse a una memoria Flash externa.

En la Figura 2.4.2-3 se aprecia cómo está constituido el ESP8266 lo que está pintado con azul es la parte de la tarjeta de red y la parte naranja el microcontrolador Tensilica Xtensa L106.

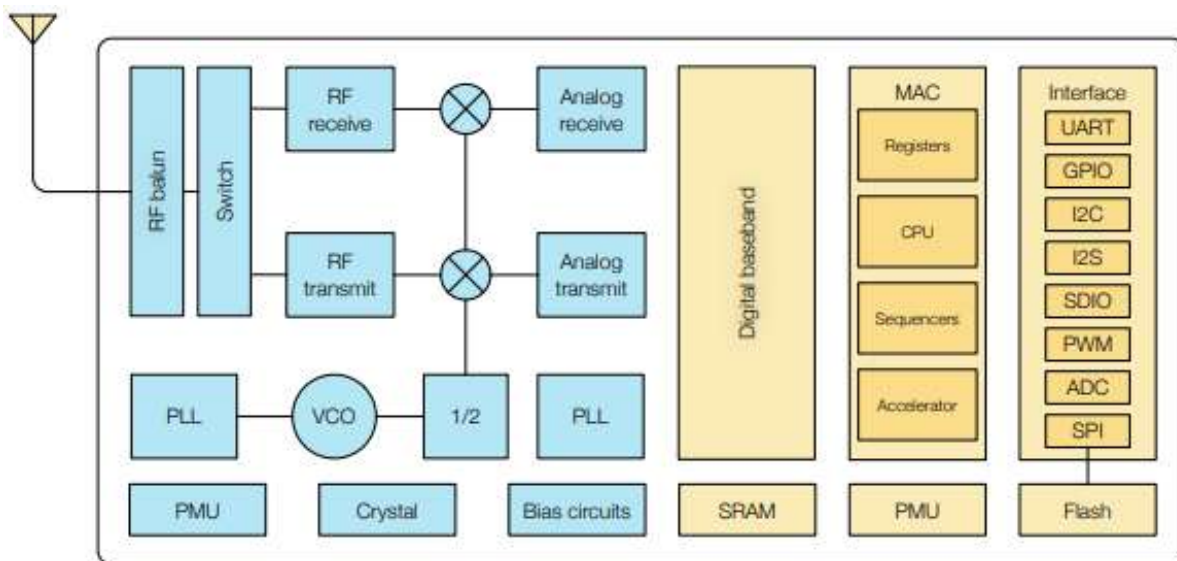


Figura 2.4.2-3 Diagrama de bloques funcional. [37]

El procesador funciona por defecto a 80 MHz pero puede ir hasta 160 MHz, tiene ~ 80kB de DRAM (Data RAM), y ~ 35kB IRAM (Instrucción RAM). La IRAM se carga en el arranque con lo que el usuario quiere mantener en el procesador, aunque el procesador puede ejecutar el código directamente fuera del flash externo a una velocidad más baja. [36]

2.4.3. Arquitectura del procesador.

Tiene una arquitectura Harvard, con lo cual la CPU puede tanto leer una instrucción como realizar un acceso a la memoria de datos al mismo tiempo, incluso sin una memoria caché. En consecuencia, una arquitectura de computadores Harvard puede ser más rápida para un circuito complejo, debido a que la instrucción obtiene acceso a datos y no compete por una única vía de memoria. [36] Ver Figura 2.4.3-1.

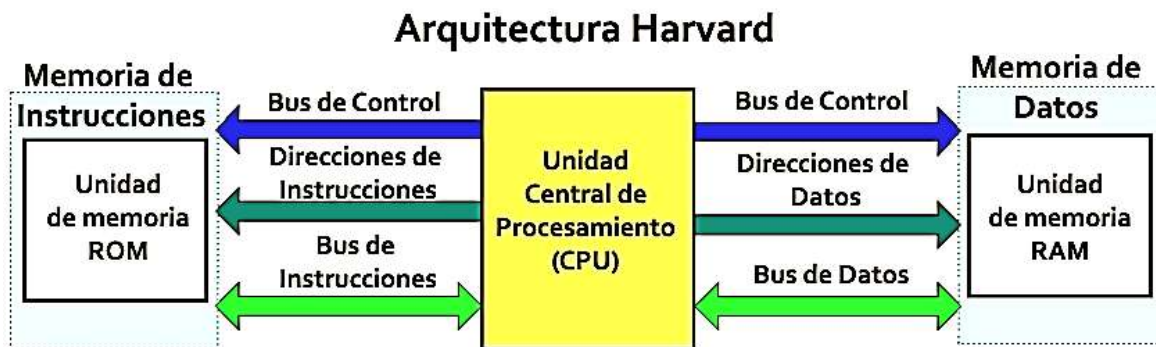


Figura 2.4.3-1 Arquitectura Harvard. [38]

2.4.4. Variantes

Cómo se aprecia en la Figura 2.4.4-1 Esp8266 cuenta con una variedad de versiones.

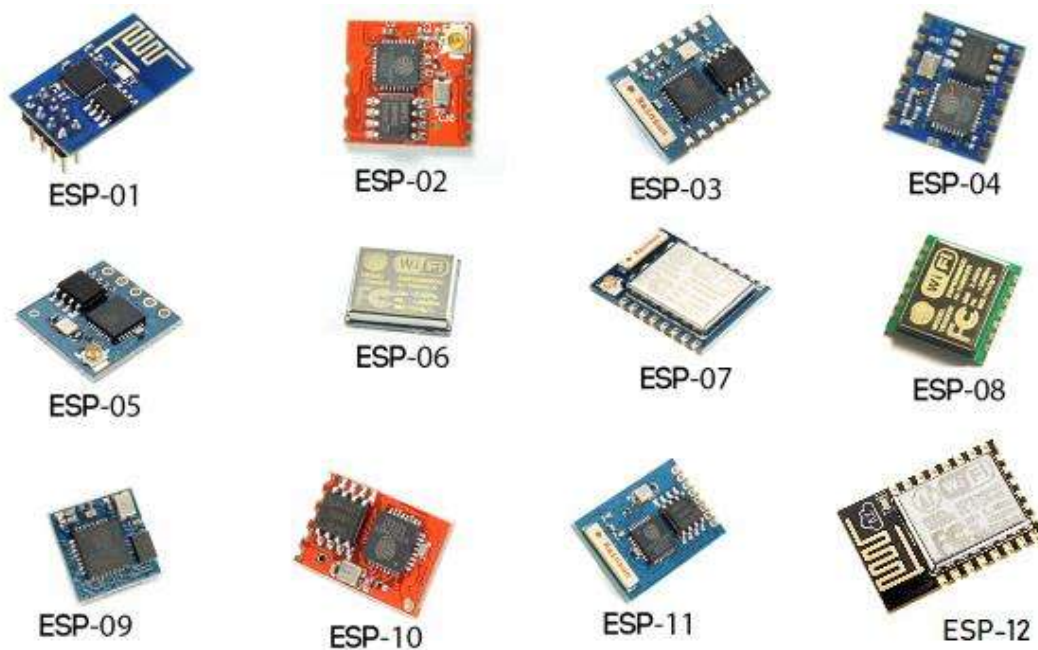


Figura 2.4.4-1 Módulos ESP. [36]

Tienen el mismo procesador, pero su versión varía a la hora de construirlo sobre una placa impresa ya que sus características de construcción difieren en otros aspectos; los cuales se muestran a continuación.

ESP-01 Figura 2.4.4-2 y Figura 2.4.4-3.

El primero de ellos y uno de los más populares,

- GPIOs: 2 Pines GPIO (GPIO 0/2)
- Dimensiones: 14,3 x 24,8mm
- Antena: Impresa

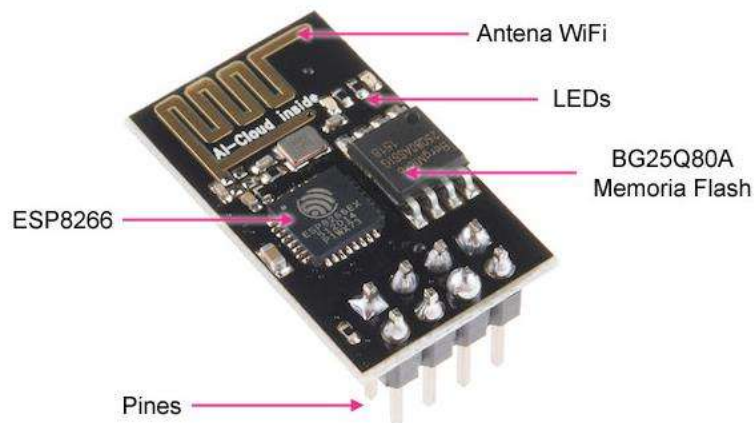


Figura 2.4.4-2 Partes que constituyen el módulo Esp-01. [39]

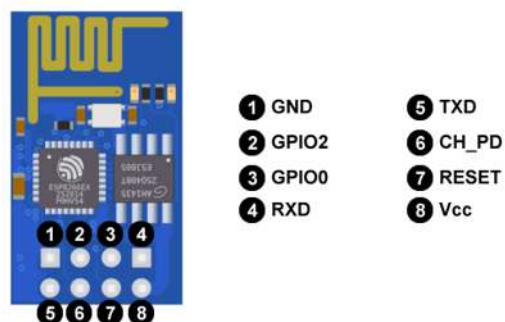


Figura 2.4.4-3 ESP-01 pintout. [39]

ESP-02 Figura 2.4.4-4.

- GPIOs: 3 (GPIO0/2/15)
- Dimensiones: 14,2x14,2 mm
- Antena: Externa (U-FL)

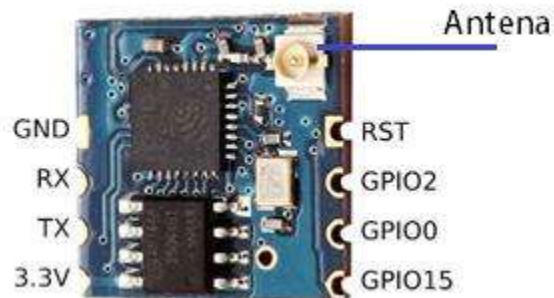


Figura 2.4.4-4 ESP-02 pinout. [40]

ESP-03 Figura 2.4.4-5.

Este módulo será el utilizado para el proyecto el cual cuenta:

- GPIOs : 7 Pines GPIO (GPIO0/2/12/13/14/15/16)
- Dimensiones: 17,3 × 12,1mm
- Antena: Cerámica



Figura 2.4.4-5 ESP-03 pinout [41]

Este es el módulo ESP8266 ESP-03 con software versión 0.90 que tiene una antena de cerámica a bordo para un alcance inalámbrico mejorado y se comunica a 115200 baudios. El módulo divide todos los pines del chip en un encabezado de pin SMD de 2x7 con un paso de 0,2 milímetros.

ESP-04 Figura 2.4.4-6.

- GPIOs : 7 Pines GPIO (GPIO0/2/12/13/14/15/16)
- Dimensiones: 14,7 × 12,1mm
- Antena: Ninguna

Este módulo es similar al anterior la diferencia está en la antena. Al igual que el modulo anterior cuentan con los pines de reset y GPIO16 los cuales se muestran en la ilustración 22.



Figura 2.4.4-6 . ESP-04 pinout. [42]

ESP-05 Figura 2.4.4-7.

- GPIOs: Ninguno
- Dimensiones: 25mm x 15mm
- Antena: Externa (U-FL)

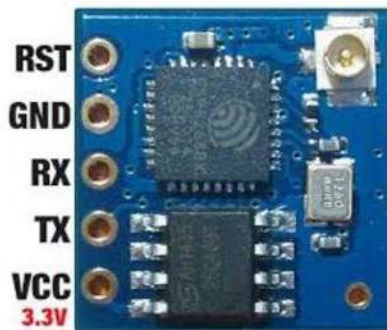


Figura 2.4.4-7 ESP-05 pinout. [43]

ESP-06 Figura 2.4.4-8.

- GPIOs: 3 (GPIO0/2/15)
- Dimensiones: 14,2x14,7 mm
- Antena: Ninguna

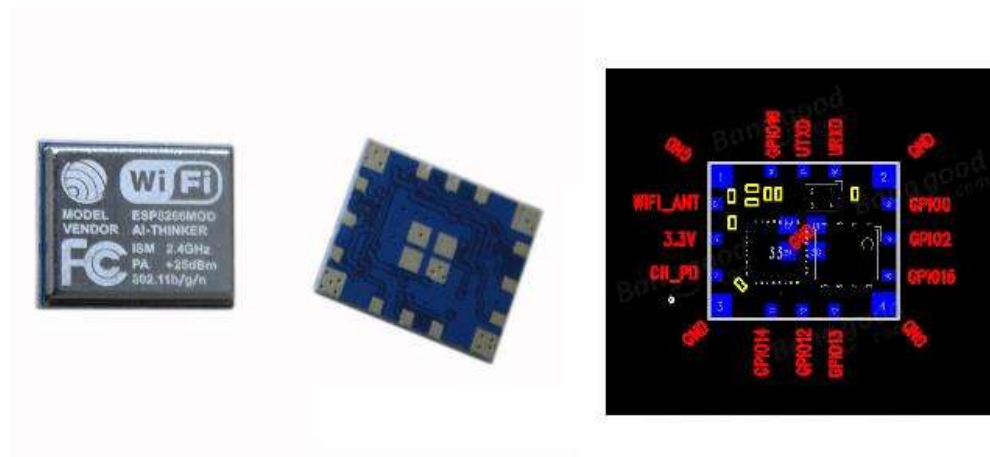


Figura 2.4.4-8 ESP-06 y diagrama pinout. [36]

ESP-07 Figura 2.4.4-9.

- GPIOs: 9(GPIO0/2/4/5/12/13/14/15/16)
- Dimensiones: 22 x 16mm
- Antena: Cerámica y Externa (U-FL)



Figura 2.4.4-9 ESP-07 pinout. [44]

ESP-08 Figura 2.4.4-10.

- GPIOs: 7 (GPIO0/2/12/13/14/15/16)
- Dimensiones: 17x16mm
- Antena: Ninguna

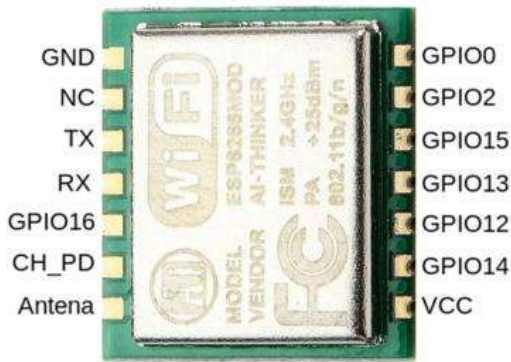


Figura 2.4.4-10 ESP-08 pinout. [45]

ESP-09 Figura 2.4.4-11.

- GPIOs: 6 (GPIO0/2/12/13/14/15)
- Dimensiones: 10x10mm
- Antena: Ninguna



Figura 2.4.4-11 ESP-09 pinout. [46]

ESP-10 Figura 2.4.4-12.

- GPIOs: Ninguno
- Dimensiones: 14,2x10 mm
- Antena: Ninguna



Figura 2.4.4-12 ESP-10 pinout. [46]

ESP-11 Figura 2.4.4-13.

- GPIOs: 2 (GPIO0/2)
- Dimensiones: 19,3x13 mm
- Antena: Cerámica



Figura 2.4.4-13 ESP-11 pinout. [46]

ESP-12 Figura 2.4.4-14.

- GPIOs: 9 GPIOs + ADC
- Dimensiones: 24x16 mm
- Antena: Impresa

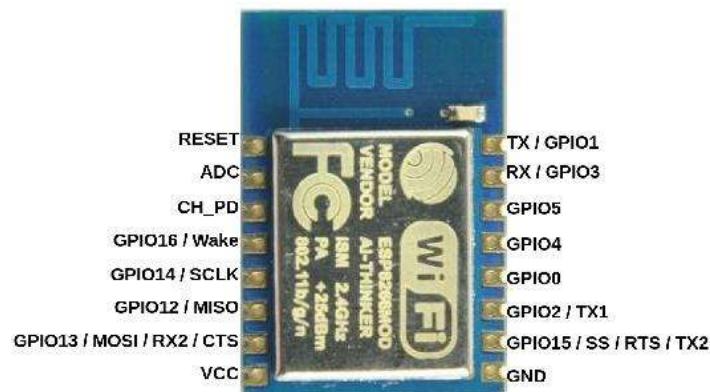


Figura 2.4.4-14 ESP-12 pinout. [46]

Este módulo incorpora la memoria Flash para almacenar los programas o sketches y la antena. Pero ya no sólo eso, aquí comienza la tarea de facilitar el acceso a los pines y demás conectores del SoC y del microcontrolador.

ESP-13 Figura 2.4.4-15.

- GPIOs: 9 GPIOs + ADC
- Dimensiones: 18x20 mm
- Antena: Impresa

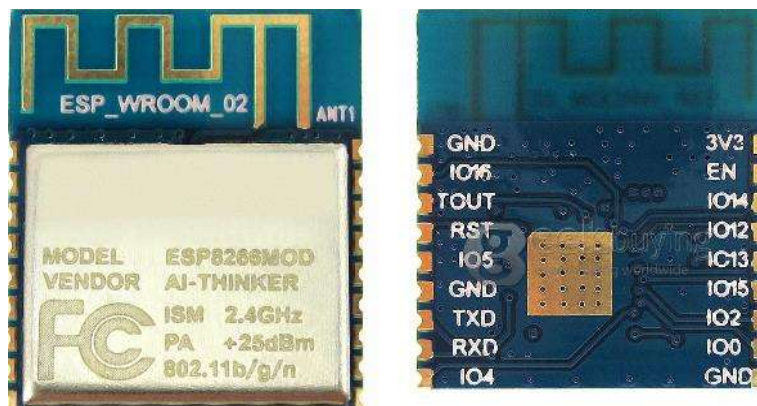


Figura 2.4.4-15 ESP-13 pinout. [47]

ESP-14 Figura 2.4.4-16.

- GPIOs: 11 GPIOs + ADC
- Dimensiones: 2.9 cm x 1.1 cm x 0.3 cm
- Antena: Impresa

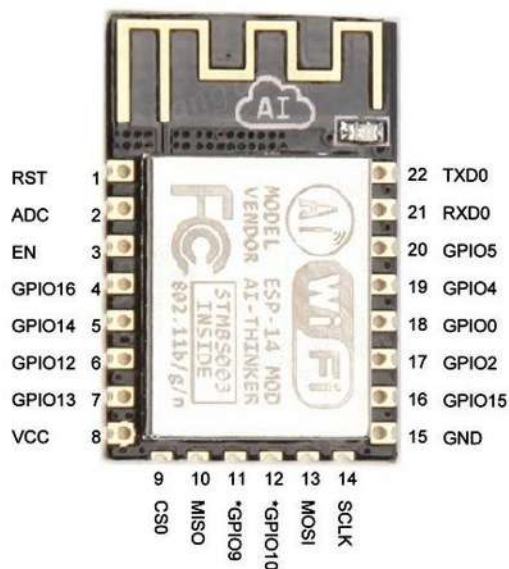


Figura 2.4.4-16 ESP-14 pinout. [48]

2.5. Diodo LED

El LED, acrónimo de “Light Emitting Diode”, o diodo emisor de luz de estado sólido, constituye un tipo especial de semiconductor, cuya característica principal es convertir en luz la corriente eléctrica de bajo voltaje que atraviesa su chip. Desde el punto de vista físico un LED común se presenta como un bulbo miniaturizado, carente de filamento o de cualquier otro tipo de elemento o material peligroso, con la ventaja sobre otras tecnologías que no contamina el medio ambiente. [49]

LED (más común de color rojo. Constituye un elemento semiconductor cuya característica principal es producir luz visible cuando la corriente eléctrica que atraviesa el chip lo polariza directamente ver . [49]

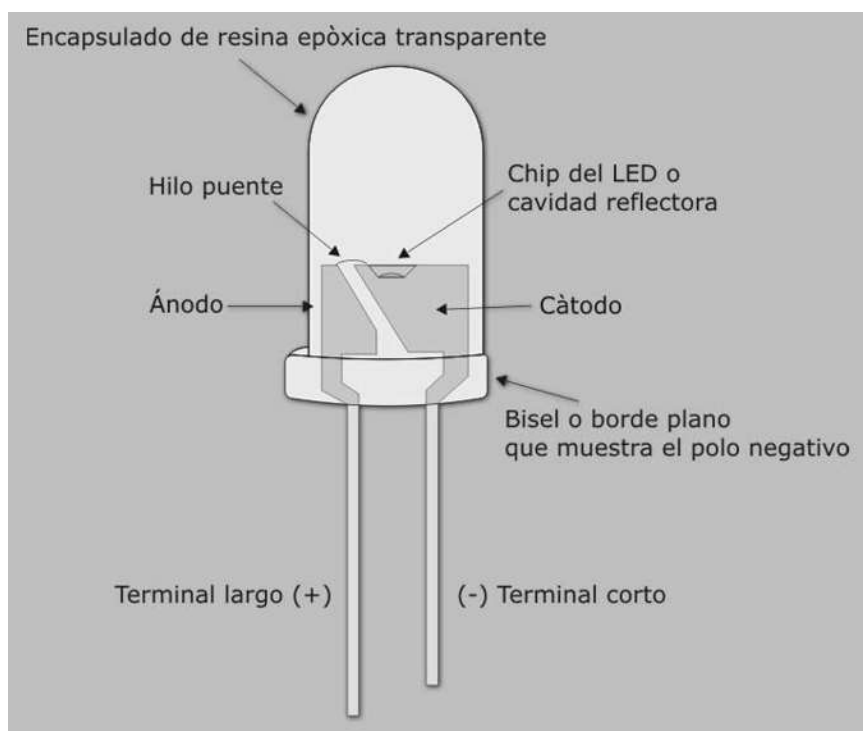


Figura 2.5-1 Partes que constituyen un diodo LED. [50]

El diodo emisor de luz tiene una terminal por la que entra la carga positiva y otra por la que entra la negativa. Ambas están separadas por un pequeño espacio donde se da la transición que produce la luz.

También se fabrican diodos LEDs emisores de rayos láser o luz coherente, de color rojo, verde o azul visible. Los dispositivos emisores de luz roja láser se utilizan ampliamente en función de punteros señalizadores en manos de profesores y conferencistas, así como en los equipos de grabación y reproducción de CDs y DVDs. El láser azul, por su parte, tiene un uso muy extendido en los equipos reproductores de películas en discos Blu-Ray, que pueden almacenar mucha más información que un DVD, incluso que los que tienen doble capa. [49]

Todos los diodos LEDs poseen una construcción sólida. La principal característica que los diferencia de otras fuentes de luz es que no poseen filamento ni partes frágiles de vidrio, lo que les permite absorber vibraciones extremas y golpes sin llegar a romperse ni estropearse. El chip emisor de luz es la parte principal de un LED y se encuentra encerrado en una envoltura o cápsula de resina epoxi transparente o también tintado del mismo color de la luz que emiten, pudiendo tener formas y tamaños diferentes. Desde el mismo chip que se encuentra encerrado en el interior de la cápsula parten dos terminales que atraviesan su base y salen al exterior para que se puedan conectar a un circuito eléctrico de corriente directa (C.D.), de forma tal que el LED quede polarizado directamente. Generalmente cuando el LED es nuevo, el terminal más corto corresponde al polo negativo (-) del chip, mientras el más largo corresponde al polo positivo (+). [49]

De acuerdo con las características de funcionamiento de cada LED, el chip puede operar con una tensión o voltaje de polarización directa entre 1 y 4 V aproximadamente, con una corriente que fluctúa entre 10 y 40 mA que puede variar también de acuerdo con el tamaño y color de luz que emite cada uno en particular. La aplicación de un voltaje o corriente de trabajo por encima del valor establecido por el fabricante para cada diodo LED en particular puede llegar a acortar su vida útil, o también destruirlo. [49]

A diferencia de las lámparas incandescentes, fluorescentes, halógenas y CFL que radian la luz en todas direcciones de forma uniforme, los LEDs la emiten a partir de la superficie superior del chip y la proyecta directamente en forma de cono, formando un ángulo entre 120 y 140 grados. [49]

Los diodos LEDs tienen una vida útil extensa, de unas 50 mil horas aproximadamente, comparadas con las mil horas que poseen las lámparas incandescentes y las 10 mil de las fluorescentes. [49]

Clasificación según la radiación emitida.

- LEDs Infrarrojos: Emiten radiación con una longitud de onda superior a los 780nm. Se suelen usar en artefactos electrónicos y elementos de control.
- LEDs Ultravioletas: Emiten radiación con una longitud de onda inferior a los 380nm.
- LEDs de Colores: Emiten radiación con una longitud de onda entre los 380nm y los 780nm. Se consigue la emisión de luz en una longitud de onda determinada del espectro según los materiales semiconductores usados. Principales colores usados: Azul, Verde, Amarillo, Naranja y Rojo.
- LEDs de Luz Blanca: Se obtiene luz blanca mediante conversión por luminiscencia usando LEDs azules o ultravioletas combinados con fósforo de distintos colores.
- LEDs RGB: Combinación de tres chips de color rojo (Red), verde (Green) y azul (Blue) que pueden generar cualquier color del espectro al regular su luminosidad. [16]

2.6. SERVO MOTOR

Un servomotor (o servo) es un tipo especial de motor con características especiales de control de posición. Al hablar de un servomotor se hace referencia a un sistema compuesto por componentes electromecánicos y electrónicos. [51]

Por lo general un servomotor no gira su eje 360° (aunque ahora hay algunos que si lo permiten), los servomotores comunes, solo giran 180° hacia la izquierda o hacia la derecha (ida y retorno). [51]

2.6.1. Características Generales

Las características principales de un servomotor son el par y la velocidad. [52]

El par es la fuerza que es capaz de hacer en su eje. El par también llamado torque se suele expresar en $N\cdot m$ (que equivale a un kilogramo-fuerza por distancia que recorre). A mayor par, mayor corriente de consumo del servo. [52]

Un servo normal o estándar proporciona un par de 3 a 4 $N\cdot cm$, la corriente que requiere depende del tamaño del servo. Normalmente el fabricante indica cual es la corriente que consume. La corriente depende principalmente del par. [53]

2.6.2. Partes de un servomotor

En la Figura 2.6.2-1 se observa las partes que constituyen un servomotor el cual es un sistema compuesto por:

- Un motor eléctrico: Genera el movimiento a través de su eje.
- Un sistema de regulación: Formado por engranajes que actúan sobre el motor para regular su velocidad y el par. Mediante estos engranajes, normalmente ruedas dentadas, podemos aumentar la velocidad y el par o disminuirlas.
- Un sistema de control o sensor: Circuito electrónico que controla el movimiento del motor mediante el envío de pulsos eléctricos.
- Un potenciómetro: conectado al eje central del motor que nos permite saber en todo momento el ángulo en el que se encuentra el eje del motor.

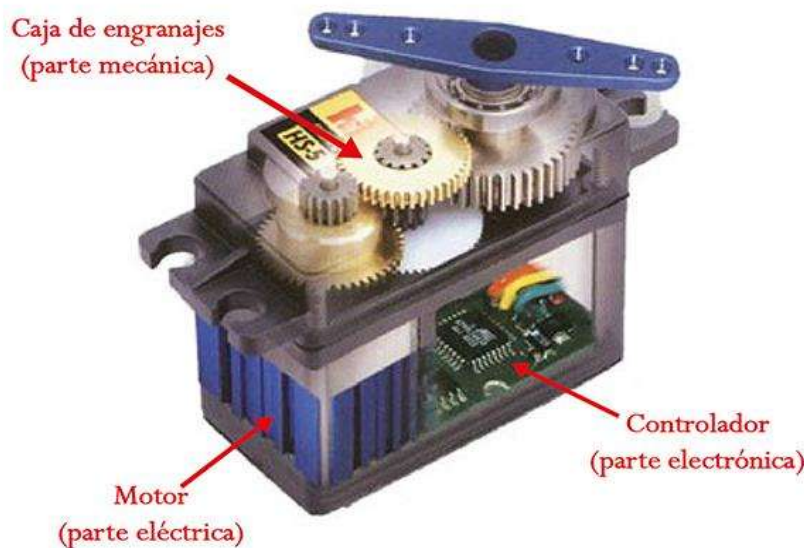


Figura 2.6.2-1 Partes de un servomotor. [51]

2.6.3. Consumo de energía

La energía consumida por una carga eléctrica será igual a la potencia (producto del voltaje por la corriente que entra a la carga) multiplicada por el tiempo de uso del motor. Debido a que los servomotores son alimentados entre 4 y 6 voltios, es posible asumir que el voltaje de alimentación es casi constante para la mayoría de los modelos. Lo que no es igual para uno u otro modelo de servomotor es el consumo de corriente. La corriente demandada por un servomotor depende de diferentes parámetros:

- Fabricante y modelo del servomotor.
- La inercia acoplada al eje del motor. Esto incluye la inercia del sistema de engranajes y la de la carga que esté acoplada al motor.
- La velocidad de rotación aplicada al eje del servomotor. [51].

2.7. WS2812

WS2812 es un led el cual contiene un chip (driver) que permite realizar control de encendido, apagado y diversos efectos ver Figura 2.7-1.



Figura 2.7-1. Encapsulado WS2812B RGB 5050. [54]

2.7.1. ¿Qué es WS2812?

WS2812x: Leds multicolor direccionables individualmente.

En adelante nos centraremos en el WS2812B, por ser el más moderno de la familia, (WS2801, WS2811, WS2812, WS2812B).

Los WS2811, WS2812 y WS2812B son LED's que disponen de lógica integrada, por lo que es posible variar el color de cada LED de forma individual (a diferencia de las tiras RGB convencionales en las que todos los LED cambian de color de forma simultánea). [55]

Estas tiras WS2812x pueden ser controladas desde un programador digital con tarjeta SD o por un autómata programable (MCU).

Están basados en el LED 5050, llamado así porque tiene un tamaño de 5.0 x 5.0 mm. Es un LED de bajo consumo y alto brillo, que incorpora en un único encapsulado los 3 colores RGB. [55]

La genial novedad del WS2812B (y el resto de familia) es añadir un integrado dentro de cada LED, que permite acceder a cada pixel de forma individual. Por este motivo este tipo de LED se denominan “individual addressable”. [55]

Podemos encontrar conjuntos de WS2812B en una gran variedad de formaciones, incluidas tiras, paneles, y anillos como se muestra en la Figura 2.7.1-1.

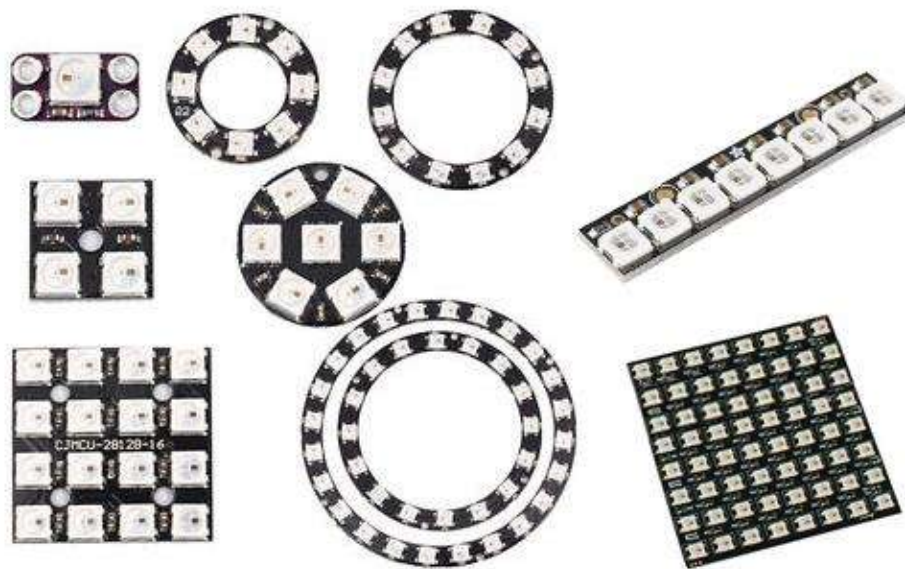


Figura 2.7.1-1 Variantes en la estructura de diseño de grupos de LEDs WS2812. [55].

A los LED WS2812B también se les denomina NeoPixel.

2.7.2. Driver

Cada LED dispone de un integrado que almacena 3 bytes (24 bits), que corresponden con los 3 colores del RGB. Cada pixel puede tener 256 niveles en 3 colores, lo que supone un total de 16.777.216 posibles colores. [55]

Cuando un LED recibe un flujo de bytes, almacena los últimos bytes recibidos y transmite los que contenía al siguiente LED esto es debido a la forma en la que se conecta un led de tras de otro ver Figura 2.7.2-1. Finalmente, con una señal de “resetcode” cada LED muestra el último valor almacenado. [55]

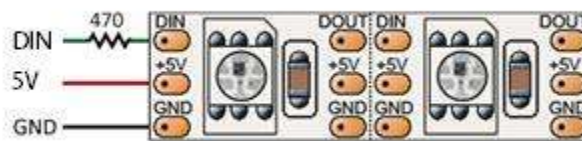


Figura 2.7.2-1 Pinout y forma de estructura de conexión WS2812B. [55]

Cada vez que un punto transmite al siguiente punto una señal, realiza una reconstrucción de forma que la distorsión y el ruido no se acumulan. Esto permite alimentar tiras de más de 5m sin necesidad de dispositivos adicionales. [55]

Estos leds se alimentan con 5V y el driver encapsulado proporciona la corriente constante necesaria para dar la luminosidad a cada uno de los colores, asegurando la consistencia del color del pixel de led. [56]

2.7.3. Transferencia de datos.

La transmisión de datos se realiza mediante le protocolo de comunicación NZR y el paquete de datos es cómo el que se muestra en la Figura 2.7.3-1.

La transmisión de 0 y 1 y resetcode se realiza mediante señales pulsadas temporizadas.

- Un 0 se realiza por un pulso HIGH de 0,35 us, seguido de un periodo LOW de 0,9.
- Un 1 se realiza por un pulso HIGH de 0,9us y LOW 0,35us.
- El “resetcode” se manda como una señal LOW de 50us.

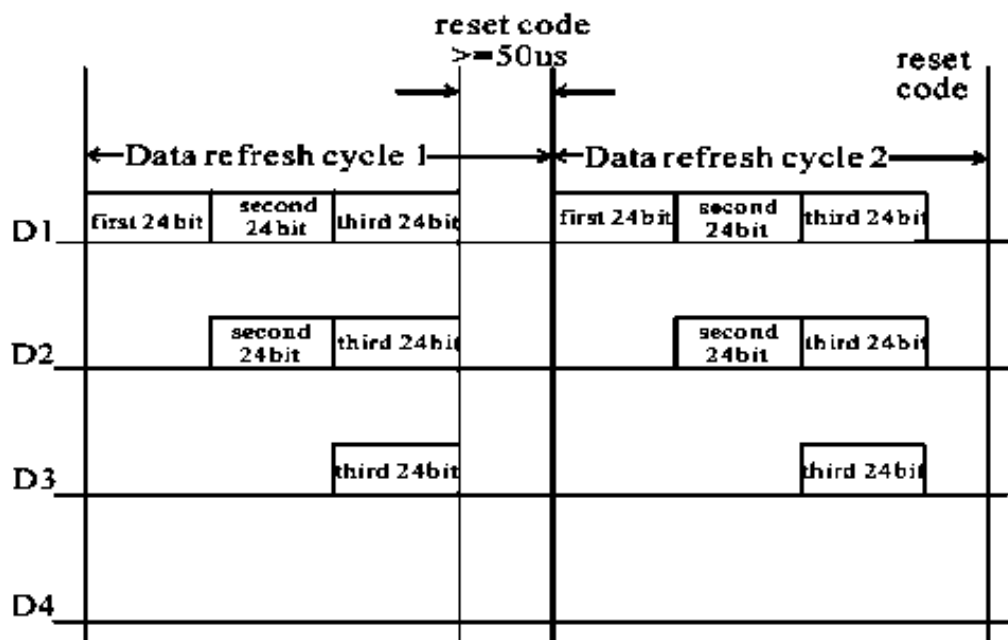


Figura 2.7.3-1 Diagrama de tiempos con respecto al flujo de datos. [55]

La frecuencia de funcionamiento es superior a 400Hz/s. Esto permite que se puedan animar más de 1024 puntos a una tasa refresco de 30fps. [55]

2.7.4. Consumo de Energía

La alimentación de la tira LED tiene que realizarse desde una fuente de alimentación externa de 5V.

Cada LED WS2182b consume unos 60mA (0,3W) dando color blanco intenso (20mA por cada componente de color). Esto supone un consumo de 9W para 30 LED, y 18W para 60 LED, lo que es mucha potencia en una fuente de 5V. [55]

Afortunadamente, no siempre se van a iluminar todos los LED de forma simultánea con blanco a máxima intensidad. Dependiendo del programa y efectos que se hagan, la potencia requerida será mucho menor. [55]

En casos de un alto consumo y gran número de LED's puede ser necesario alimentar por más de un punto a la vez (por ejemplo, cabeza y cola de la tira). De lo contrario, la caída de tensión hará que los últimos LED's tiendan a prender en color rojizo. [55]

Se aconseja emplear una resistencia de 470 ohmios entre el pin digital y el pin de señal del WS2818b, o podrán dañarse los primeros LED ver Figura 2.7.2-1. [55]

2.7.5. Diferencia entre familia WS28XX.

Una importante mejora de seguridad de las tiras de led WS2812B respecto de las WS2812 es que implementan un circuito de protección frente a conexiones inversas de corriente para incrementar la estabilidad y durabilidad de las tiras de led. [57]

A nivel arquitectónico las nuevas tiras constan de menos patas de conexión lo que aumenta el espacio entre patas de conexión y disminuye el riesgo de cortocircuitos por lo que aumenta la calidad de las nuevas tiras frente a las más antiguas. [57]

Pero desde que aparecieron estos leds, que empezaron a conocerse con el nombre de WS2811, hasta hoy, ha habido evoluciones. WS2801, WS2811, WS2812 y WS2812B.

WS2801 = Chip driver que se encuentra dentro de los leds y necesita de alimentación, línea de datos y línea de reloj (Obsoleto).

WS2811 = Chip driver que se encuentra dentro de los leds y necesita de alimentación y línea de datos.

WS2812 = WS2811 encapsulado con circuitería y electrónica Led.

El WS2801 usaba una versión antigua del protocolo de comunicaciones que necesitaba de una línea de reloj, lo que le hacía fallar con algunos controladores. La versión WS2811 ya no necesita de esa línea de reloj.

Como se ve en la Figura 2.7.5-1, el nuevo modelo WS2812B solo necesita de 4 patas: VDD, VSS, DIN y DOUT. Mientras que su modelo predecesor tenía 6 patas: VDD, VSS, VCC, DIN, DOUT y NC.

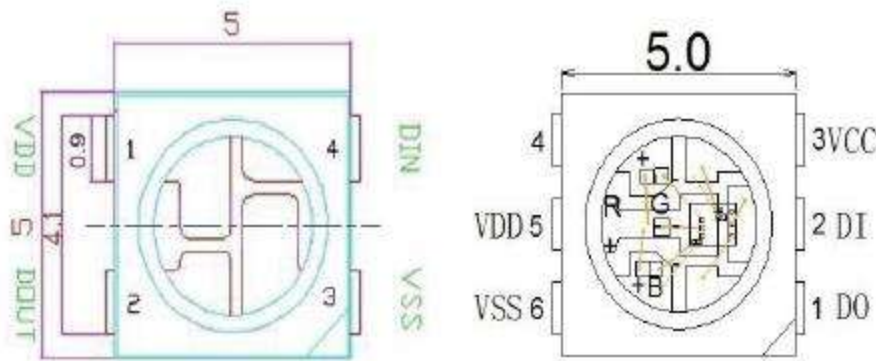


Figura 2.7.5-1. Pinout LEDs 5050, WS2812B y WS2812, respectivamente. [57]

Y como nota adicional, cuando se hable de WS2812S, es el mismo que el WS2812 normal.

El nuevo chip trabaja con el mismo protocolo que el anterior WS2812 pero tiene un par de mejoras en las características mecánicas que hacen que sea más sencillo trabajar con él.

Algunas de las características más importantes del nuevo modelo WS2812B son:

Rendimiento más seguro: Este modelo está basado en el WS2812. Este led, no solo ha heredado las buenas cualidades del WS2812S, sino que también han mejorado la disposición mecánica exterior de la estructura, mejorando de este modo la estabilidad y la eficiencia.

Nuevo empaquetado: El nuevo modelo solo tiene 4 pines en vez de 6. Esto ayuda a que sea un diseño más simple, con mejor uso de la tensión de alimentación y mejor disipación térmica.

Mejor brillo: El nuevo modelo WS2812B es capaz de producir un brillo mucho mayor que la WS2812.

Estructura interna mejorada: El WS2812B tiene una estructura interna mejor que el WS2812. Esta modificación ha hecho que el circuito de control y la luz RGB se hayan separado de tal modo que ahora se tiene una mejor disipación del calor. Los dos módulos trabajan de forma independiente y lo hacen más estable.

Protección contra inversión de alimentación: El nuevo modelo tiene protección frente a errores de alimentación en los que invertimos las conexiones de VDD y VSS.

2.8. Socket TCP

Un socket permite comunicarnos entre dos diferentes procesos ya sea desde la misma máquina o diferentes máquinas en el sistema de aplicación cliente-servidor. [58]

Tipos de Sockets

Existen cuatro tipos de sockets, Stream Sockets, Datagram Sockets, Raw Sockets, Sequenced Packet Sockets (SPS), los dos primeros tipos son más comúnmente usados que los otros. [58]

Socket de Flujo:

(Stream Socket), están libres de errores: Si por ejemplo, enviáramos por el socket de flujo tres objetos "A, B, C", llegarán al destino en el mismo orden -- "A, B, C" --. Estos sockets usan TCP ("Transmission Control Protocol") y es este protocolo el que nos asegura la transmisión adecuada de los objetos. [58]

Sockets de Datagramas:

Éstos usan UDP ("User Datagram Protocol") y no necesitan de una conexión accesible como los Sockets de Flujo, construirá un paquete de datos con información sobre su destino y se enviará afuera, sin necesidad de una conexión. [58]

Raw Socket:

Estos proporcionan acceso de usuario a los protocolos de comunicación subyacentes, que admiten abstracciones de socket. No están destinados para uso general. [58]

Sequenced Packet Sockets:

Similares a los Socket Stream, con la excepción de que se conservan los límites del registro. La interfaz SPP solo se proporciona como parte de la abstracción del socket del sistema de red y es muy importante en las aplicaciones más serias del sistema de red. [58]

SPS permite al usuario recibir el encabezado en los paquetes entrantes.

El proceso definido anteriormente es para comunicarse entre los mismos tipos de sockets. Pero no hay restricción en la comunicación entre los diferentes tipos de socket. [58]

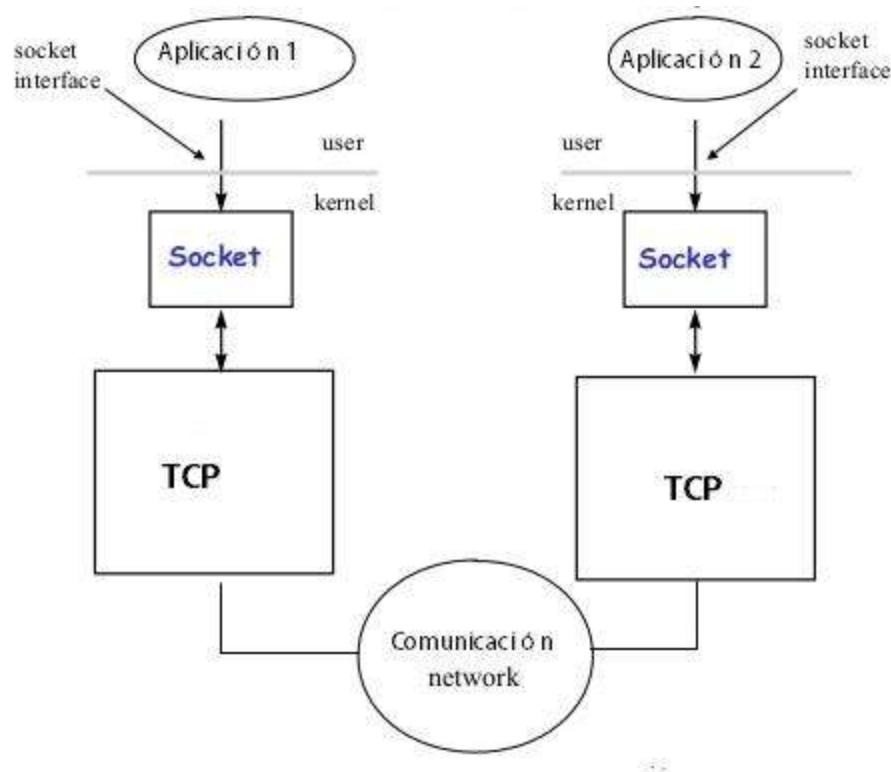


Figura 2.8-1. Interfaz de un Socket TCP [58]

SOCKET TCP

- El cliente debe contactar con el servidor.
- El servidor debe estar ejecutándose primero.
- El servidor tiene que crear un socket por el que espera que el cliente se contacte.
- El cliente crea su propio socket y especifica su dirección IP y el puerto por el cual el servidor está escuchando.
- Cuando el servidor es contactado crea un nuevo socket TCP para la comunicación con el cliente.
 - ❖ Permite que el servidor se comunique con varios clientes simultáneamente
 - ❖ La dirección IP del cliente y el puerto empleado por su aplicación sirven para distinguirlos (más cuando veamos TCP)

Un stream es una secuencia de bytes que fluye hacia o desde un proceso, un socket TCP ofrece un stream bidireccional full dúplex. [59]

2.9. HTML

2.9.1. ¿Qué es HTML?

Es el lenguaje más básico para la creación de páginas web. Es usado para definir la estructura y el contenido.

HTML, Lenguaje de Marcado para Hipertextos (HyperText Markup Language) es el elemento de construcción más básico de una página web y se usa para crear y representar visualmente una página web. Determina el contenido de la página web, pero no su funcionalidad. [60]. Las páginas web pueden ser vistas por el usuario mediante un tipo de aplicación llamada cliente web o más comúnmente "navegador". [61]. Este lenguaje está constituido de elementos que el navegador interpreta y las despliega en la pantalla de acuerdo a su objetivo.

HTML le da "valor añadido" a un texto estándar, Hiper Texto se refiere a enlaces que conectan una página web con otra, ya sea dentro de una página web o entre diferentes sitios web, los vínculos son un aspecto fundamental de la Web. [60]

Otro aspecto importante es que el lenguaje HTML no es sensible a mayúsculas y minúsculas, es decir podemos escribirlo como más nos guste, además no requiere que dispongamos cada marca en una línea (podríamos inclusive escribir toda la página en una sola línea). [62]

Una página HTML es un archivo donde está contenido el código HTML en forma de texto. Estos archivos tienen extensión .html o .htm (es indiferente cuál utilizar). De modo que cuando programemos en HTML lo haremos con un editor de textos y guardaremos nuestros trabajos con extensión .html, por ejemplo mipágina.html. Cabe remarcar que hoy todo el mundo usa la extensión ".html" y no ".htm". [61]

2.9.2. Sintaxis

HTML usa "markup" o marcado para anotar textos, imágenes, y otros contenidos que se muestran en el Navegador web. El lenguaje de marcado HTML incluye "elementos" especiales tales como: <head>, <title>, <body>, <header>, <article>, <section>, <p>, <div>, , , entre muchos otros más. [60]

Basa su sintaxis en un elemento base al que llamamos marca, tag o simplemente etiqueta. A través de las etiquetas vamos definiendo los elementos del documento.

La etiqueta presenta frecuentemente dos partes, su apertura y cierre, y se encierran ambas partes entre símbolos "menor que" y "mayor que". [63]

2.9.3. Estructura interna de una página HTML

Las instrucciones HTML están encerradas entre los caracteres: < y >.

Muchos elementos HTML requieren una marca de comienzo y otra de finalización. Todo aquello que está fuera de las marcas del lenguaje se imprime en la pantalla (dentro del navegador). [62].

La estructura básica de una página HTML es:

```
<!DOCTYPE html>
<html>
<head>
  <title>Título de la página</title>
  <meta charset="UTF-8">
</head>
<body>
  Cuerpo de la página.
</body>
</html>
```

Lo primero es el DOCTYPE que informa al navegador que el contenido siguiente se trata de un archivo HTML.

Luego continua con la marca: <html> y finaliza con la marca </html> al final del archivo.

Una página HTML tiene dos secciones bien definidas que son la cabecera:

```
<head>
</head>
```

Y el cuerpo de la página:

```
<body>
<p>Hola, estamos en el párrafo 1</p>
<p>Ahora hemos cambiado de párrafo</p>
</body>
```

En la cabecera es común inicializar el título de la página dentro de las marcas `<title></title>` (normalmente aparece en la barra superior de nuestro navegador y es utilizado por los motores de búsqueda como Google para indexar la página) El título debe hacer referencia al contenido en sí de la página. [62].

Dentro de la cabecera disponemos la etiqueta meta donde definimos la propiedad charset con el valor UTF-8 que es un formato de caracteres ampliamente empleado en internet (si no se dispone de este formato no se podrá disponer de caracteres acentuados por ejemplo). [62].

```
<meta charset="UTF-8">
```

Todo el texto que se disponga dentro del `<body>` aparece dentro del navegador tal cual lo hayamos escrito.

Todas las páginas tiene como mínimo esta estructura: cabecera y cuerpo. [62]

Formulario:

Un formulario permite que el visitante al sitio cargue datos y sean enviados al servidor.

HTML solo tiene el objetivo de crear el formulario. El HTML no tiene la responsabilidad de registrar los datos en el servidor, esta actividad está delegada a un lenguaje que se ejecute en el servidor (PHP, ASP, ASP.Net, JSP, NodeJS, etc. [64]

Para crear un formulario se debe utilizar el elemento form, que tiene marca de comienzo y fin. Dentro de la marca form veremos otros elementos para crear botones, editores de línea, cuadros de chequeo, radios de selección, etc. [64]

En la Figura 2.9.3-1 se muestra código en HTML para crear un botón tipo "sumit", y un cuadro de texto con tamaño de 20 caracteres.

```
<!DOCTYPE html>
<html>
<head>
  <title>Prueba de formulario</title>
  <meta charset="UTF-8">
</head>
<body>
<form action="registrardatos.php" method="post">
  Ingrese su nombre:
  <input type="text" name="nombre" size="20">
  <br>
  <input type="submit" value="enviar">
</form>
</body>
</html>
```

Figura 2.9.3-1 Código HTML para el método form. [64]

El resultado en el navegador es del código anterior es el que se muestra en la Figura 2.9.3-2

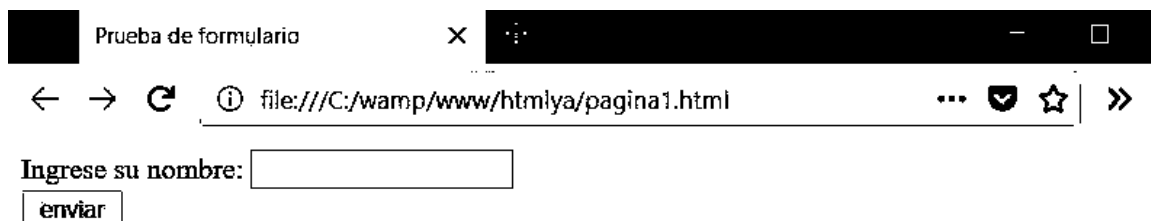


Figura 2.9.3-2. Resultado obtenido en una página Web, debida al código de la Fig. 2.4.3-1. [64]

La estructura de un formulario elemental, lo primero la apertura de la marca form donde debemos definir dos propiedades (action y method):

```
<form action="registrardatos.php" method="post">
```

La propiedad action se inicializa con el nombre de la página que contiene el programa que procesará los datos en el servidor. [64]

La segunda propiedad que debemos inicializar es method. Esta propiedad puede almacenar únicamente dos valores (post o get). [64]

Normalmente un formulario se envía mediante post (los datos se envían con el cuerpo del formulario). En caso de utilizar get los datos se envían en la cabecera de la petición de la página, utilizando el método get estamos limitados en la cantidad de datos a enviar, no así con el método post. [64]

Ahora veamos el cuadro de texto donde se ingresa el nombre:

Ingrese su nombre:

```
<input type="text" name="nombre" size="20">  
<br>
```

El mensaje "Ingrese su nombre:" es un texto fijo.

El elemento input permite definir un cuadro de texto (editor de línea) asignando le a la propiedad type el valor "text".

Por último se inicializa la propiedad size con el valor 20, esto significa que el cuadro de texto se dimensiona para permitir mostrar 20 caracteres (no se limita la cantidad de caracteres a ingresar por parte del visitante sino la cantidad de caracteres que se pueden visualizar). [64]

Seguidamente:

```
<input type="submit" value="enviar">
```

También mediante el elemento input se define un botón para el envío de datos al servidor. Se inicializa la propiedad type con el valor submit, con esto ya tenemos un botón para el envío de datos. [64]

La propiedad value almacena la etiqueta que debe mostrar el botón. [64]

Esta pequeña introducción de HTML servirá para realizar el control desde la página WEB, sin embargo este lenguaje es demasiado extenso. En internet se encuentran tutoriales para aprender más sobre el lenguaje HTML, o puedes consultar la referencia. [64]

3. Capítulo 3. Hardware.

En este capítulo se presenta el hardware utilizado en la implementación del proyecto así como las características de cada elemento.

3.1.1. NodeMCU



Figura 3.1.1-1 NodeMCU. [65]

NodeMCU es una plataforma de desarrollo para Internet de las cosas (IoT) muy similar a Arduino. Te permite crear rápidamente proyectos que se pueden conectar a Internet por Wi-Fi.

NodeMCU el kit de desarrollo para el IoT.

Es una placa de desarrollo totalmente abierta, a nivel de software y hardware. Al igual que ocurre con Arduino, en NodeMCU todo está dispuesto para facilitar la programación de un microcontrolador o MCU (del inglés Microcontroller Unit). [65]

No hay que confundir microcontrolador con placa de desarrollo. NodeMCU es una placa o kit de desarrollo que lleva incorporado un chip que se suele llamar SoC (System on a Chip) **ESP8266**. Que dentro tiene un microcontrolador o MCU. [65]

El esquema general de este tipo de placase es el de la Figura 3.1.1-2:

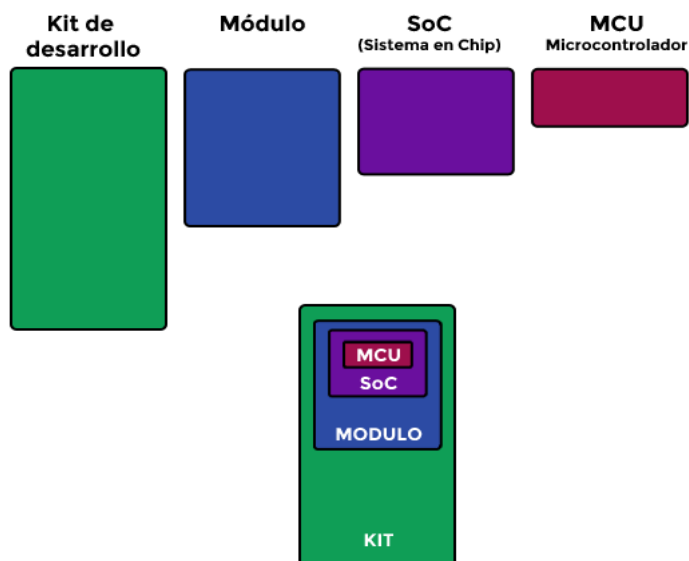


Figura 3.1.1-2 Arquitectura de Node MCU. [65]

El objetivo es programar el MCU o microcontrolador a través del kit o placa de desarrollo. Todo lo demás sirve de apoyo para crear nuestros proyectos lo más sencillo posible. [65]

ESP-12. Este es el módulo que utiliza NodeMCU.

Existen diferentes modelos de varias marcas con características y funcionalidades diferentes dependiendo del SoC o microcontrolador que utilizan. Pero todos tienen el mismo objetivo, facilitarnos el desarrollo de proyectos con microcontroladores.

NodeMCU es uno de ellos y sus características principales son:

- Convertidor Serie-USB para programar y alimentar a través del USB
- Fácil acceso a los pines
- Pines de alimentación para sensores y componentes
- LEDs para indicar estado
- Botón de reset

Ahora solo se necesita, un cable USB y un entorno de desarrollo para programar el MCU o microcontrolador.

Versiones de NodeMCU

Existe una gran confusión con respecto a las diferentes versiones que hay de NodeMCU. Debido a que se trata de una placa de hardware abierto y cualquier fabricante puede crear su propia distribución.

Pero todos los NodeMCU se basan en los mismos módulos el ESP-12 y ESP-12E que a su vez se basan en el SoC ESP8266.

Partiendo de esta premisa, las diferencias que vamos a encontrar son básicamente el número de pines a los que tenemos acceso y el tamaño de cada placa. A lo largo de este capítulo veremos tres versiones.



NodeMCU v0.9 / V1



NodeMCU v1.0/ V2



NodeMCU v1.0/ V3

Figura 3.1.1-3 Versiones NodeMCU. [65]

Actualmente existen 3 versiones ver Tabla 5.

Tabla 5. VERSIONES NodeMCU.

Generación	Versión	Común
1ª	V0.9	V1
2ª	V1.0	V2
3ª	V1.0	V3

La última columna es el nombre que encontrarás en diferentes tiendas online como Amazon, Baggood o AliExpress.

El problema que existe con respecto a los nombres es debido a que se trata de una placa de hardware abierto y cualquiera puede fabricar un NodeMCU y comercializarlo.

Principalmente vamos a encontrar 3 fabricantes:

- **Amica** es el distribuidor oficial
- **DOIT**
- **Lolin / Wemos**

3.1.2. Versiones de NodeMCU

Cuando adquieres un NodeMCU es importante reconocer la versión a la que pertenece según la tabla que hemos visto antes. Es fundamental a la hora de programar para sacar todo el potencial.

Las placas de 1ª y 2ª generación son fáciles de distinguir debido a su tamaño. Ambas utilizan el módulo ESP-12 con 4 MB de Flash, pero la 2ª generación V3 utiliza la más nueva y mejorada ESP-12E.



Figura 3.1.2-1 ESP-12E. [65]

1ª generación / v0.9 / V1 ver |Figura 3.1.2-2.



|Figura 3.1.2-2 Node MCU 1a generación V1. [65]

Quizás el mayor inconveniente que se encuentra con esta placa es su tamaño. Es muy incómodo cuando se conecta al protoboard ya que ocupa todo el espacio y no deja hueco para conectar los pines.

El módulo que utiliza es el ESP-12 basado en el ESP8266 con una memoria Flash de 4 MB.

2ª generación / v1.0 / V2 ver Figura 3.1.2-3.



Figura 3.1.2-3 Node MCU 2a generación V2. [65]

Con esta nueva generación de NodeMCU se produjo un salto sustancial de calidad y facilidad en la programación y prototipo. Esta 2ª generación sí que encaja perfectamente en la protoboard y permite conectar cables a los pines de una forma sencilla como en la Figura 3.1.2-4

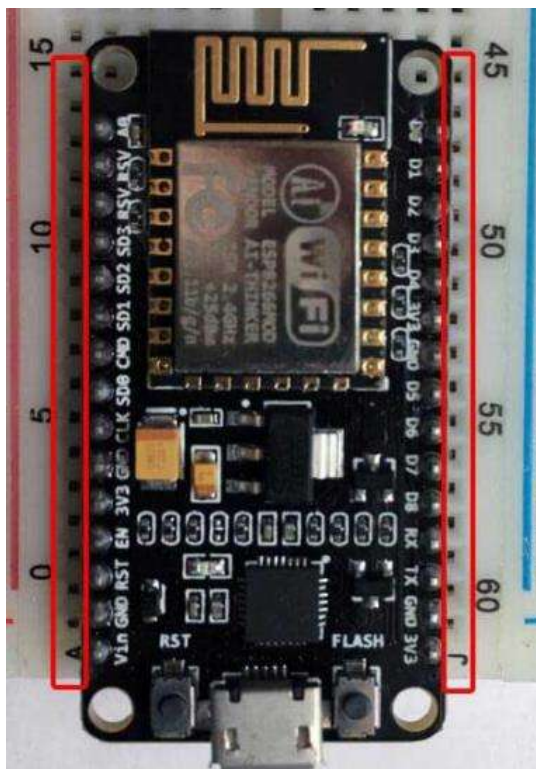


Figura 3.1.2-4 Node MCU V2, colocado en un protoboard. [65]

Otra diferencia es que el chip fue actualizado del ESP-12 al ESP-12E y esto permite tener algunos pines extra.

2ª generación / v1.0 / V3 ver Figura 3.1.2-5.



Figura 3.1.2-5 Node MCU 2a generación V3. [65]

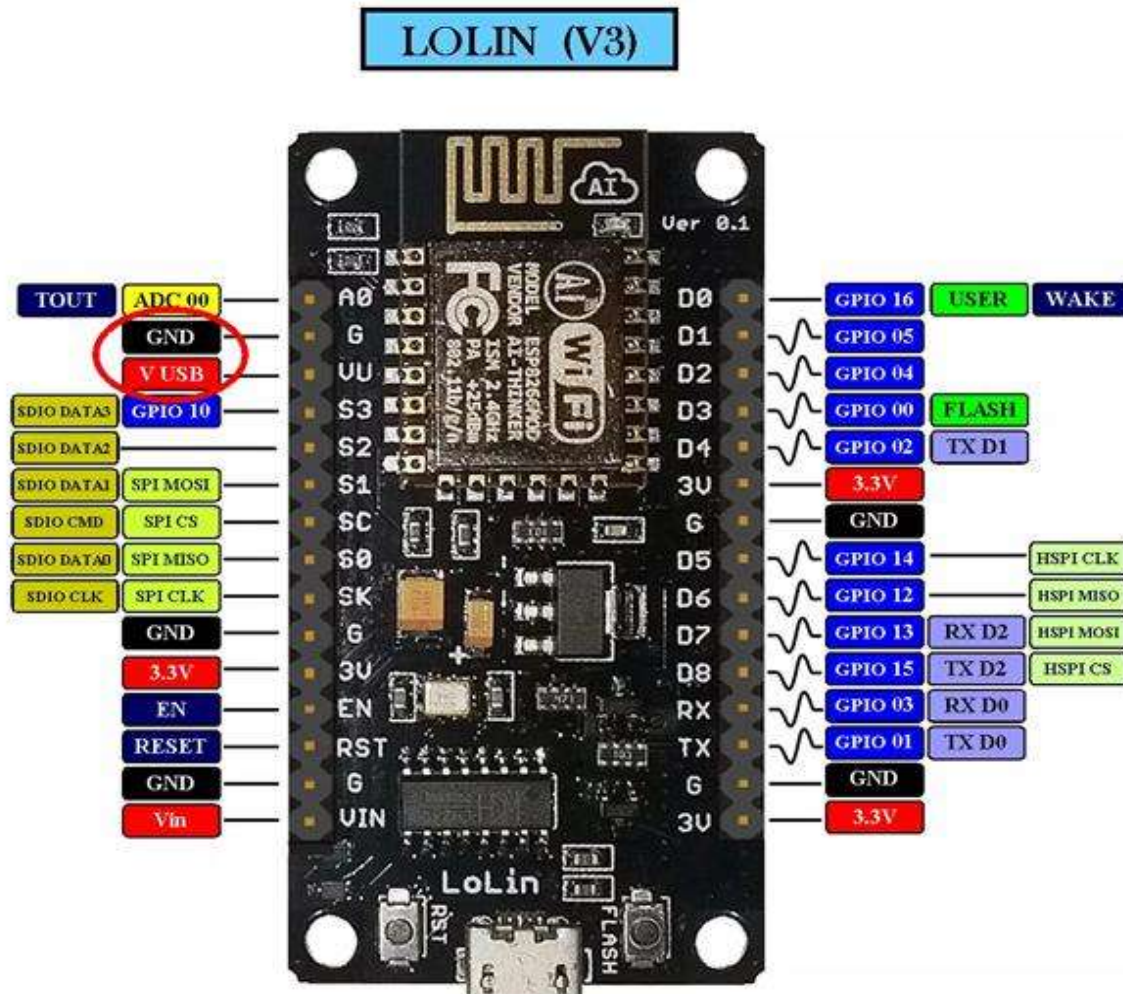
Hardware

Características V3:

- Lo primero que hay que decir de esta nueva versión es que no es una especificación oficial por parte de NodeMCU.
- Se trata de una versión creada por Lolin que aporta ciertas mejoras como que el puerto USB parece ser más robusto y dos de los pines que no se utilizan en la versión V2 se han utilizado como salida de 5V directa del USB y un GND adicional.
- Otra diferencia mínima es el tamaño ya que la V3 es un poco más ancha que la V2.

Aun así, es recomendable ceñirse a las especificaciones de NodeMCU V2 ya que es la última versión oficial.

La placa de la Figura 3.1.2-5 es la placa seleccionada para trabajar en este proyecto y en la Figura 3.1.2-6 se muestra el pinout de esta.



ADC	PIN DE SALIDA ANALÓGICA (el rango es entre 0V y +3.3V dividido en 1023 intervalos).
SPI	BUS SPI (Serial Peripheral Interface).
HSPI	BUS HSPI (Hardware Serial Peripheral Interface).
SDIO	PINES PARA INICIO DEL ESP8266 DESDE UNA TARJETA SD. Para activar el modo SDIO el pin GPIO 15 debe estar en tensión cuando se enciende la placa.
TX/RX	COMUNICACIÓN SERIE TX/RX. Los pines GPIO01 y GPIO02 están conectados al puerto USB a través del conversor UART.
Vin	ALIMENTACIÓN EXTERNA (de +5V a +20V).
3.3V	ALIMENTACIÓN INTERNA (desde la placa a dispositivos).
V USB	ALIMENTACIÓN INTERNA PROVENIENTE DEL MiniUSB (desde la placa a dispositivos +5V).
GND	TIERRA (GND Ground).
GPIO	PIN DE ENTRADA/SALIDA +3.3 V (GPIO General Purpose Input/Output). Entrada digital —. Entrada analógica ~. (Todas las salidas son digitales).

Figura 3.1.2-6 Node MCU V3, pinout. [66]

Se encuentran dos nomenclaturas para nombrar los pines, los que aparecen en la placa escritos y en ocasiones verás el nombre asociado a cada Dx con GPIOx.

Todo el acceso se basa en el número de índice de E / S en los kits de desarrollo de NodeMCU, no en el pin de GPIO interno. Por ejemplo, el pin D0 en el kit de desarrollo está asignado al pin 16 interno de GPIO.

Tabla 6. RELACIÓN ENTRE E/S Y PINES INTERNOS DE ESP8266.

IO index	ESP8266 pin	IO index	ESP8266 pin
0 [*]	GPIO16	7	GPIO13
1	GPIO5	8	GPIO15
2	GPIO4	9	GPIO3
3	GPIO0	10	GPIO1
4	GPIO2	11	GPIO9
5	GPIO14	12	GPIO10
6	GPIO12		

Pines digitales útiles de NodeMCU

Como se sabe, el ESP8266 no tiene memoria flash para almacenar el código de ejecución y datos, por lo tanto necesita de una memoria externa. Para conectarse a ella, necesita utilizar pines del propio ESP8266. [65]

Algunos de los pines de NodeMCU no pueden ser utilizados. Dependerá de la versión y el fabricante con el que estemos trabajando pero lo recomendable es no utilizar los pines GPIO6, GPIO7, GPIO8, GPIO9 (D11), GPIO10 (D12) y GPIO11. [65]

Solo dos de ellos son accesibles a través de la placa NodeMCU, el pin GPIO9 (D11) y el pin GPIO10 (D12). En tu placa aparecerá probablemente como SD2 y SD3. Evita, en la medida de lo posible, utilizar estos pines para conectar sensores y componentes. [65]

De los pines restantes es aconsejable no utilizar tampoco los pines D9 y D10 correspondientes al Rx (recibir) y Tx (transmitir). [65]

Estos pines se utilizan para transmitir un programa o sketch o para la comunicación entre el NodeMCU y el PC a través del puerto serie. [65]

Si se utilizan pueden ocasionar interferencias a la hora de cargar un programa.

Por lo tanto quedan los pines del D0 al D8, 9 pines digitales de entrada y salida.

Pin analógico de NodeMCU

Si antes hablábamos de que en los pines digitales solo tenemos dos estados LOW y HIGH, en el pin analógico vamos a poder tener un rango de valores. Este rango vendrá determinado por la resolución del conversor ADC (del inglés Analog Digital Converter) Conversor Analógico Digital. [65]

El NodeMCU tiene solo un pin analógico ver Figura 3.1.2-7 que admite un rango de valores de 0 a 3,3V con una resolución de 10-bit.

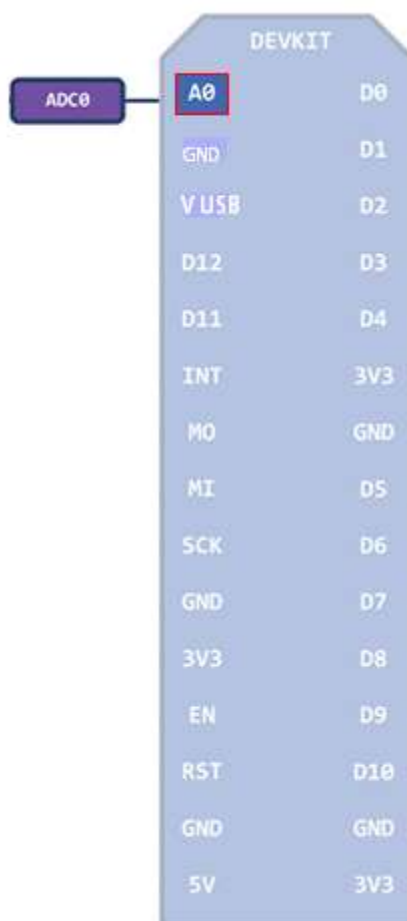


Figura 3.1.2-7 Pinout NodeMCU haciendo énfasis en el pin analógico. [65]

Esto implica que dentro del código tendremos un valor entre 0 y 1023 que se mapea con el voltaje entre 0 y 3,3V.

Botones y LED integrados dentro de la placa

Esta placa nos ofrece unos componentes extra para controlar su funcionamiento.

Hardware

Dentro de la propia placa de desarrollo vamos a encontrar 1 LED integrado y 2 pulsadores ver Figura 3.1.2-8.

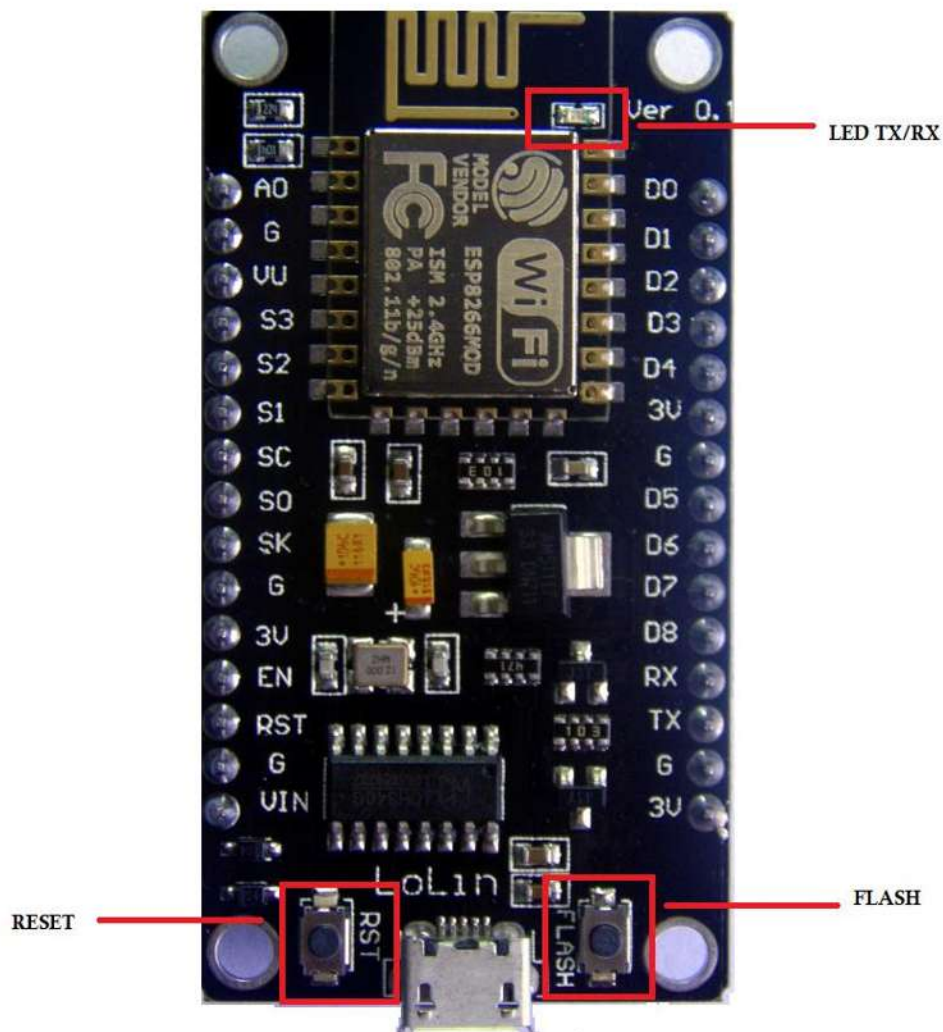


Figura 3.1.2-8 Node MCU BOTONES y LED. [67]

El LED nos indica si se está cargando un programa, ya que está asociado a TX/RX. Al igual el LED está conectado al D4 y pertenece al módulo ESP12-E, lo que nos permite controlarlo a través de código.

Además de los LEDs la placa tiene dos pulsadores. El típico botón de RESET (RST) que si lo pulsamos se resetea la placa y comienza la ejecución de cero y otro botón que coloca al código en modo de programación.

El botón de RESET hace eso, resetear. Esto no quiere decir que elimine el código, lo único que hace es comenzar la ejecución desde el principio pasando por la función **setup ()**.

El botón de FLASH que permite cargar un programa o firmware. Esto no es algo específico de NodeMCU o del ESP8266, todos los microcontroladores tienen como mínimo dos estados.

El estado de ejecución es cuando el microcontrolador ejecuta el programa que se ha cargado. El estado carga de programa o de firmware nos permite subir un programa al microcontrolador.

3.2. Micro Servo Tower Pro MG90s



Figura 3.2-1 Tower Pro MG90S y accesorios. [68]

Micro servo motor MG90S, con engranaje de metal, El servo puede rotar aproximadamente 180 grados (90 en cada dirección) y funciona igual que otros servos de tipo estándar, pero más pequeños. Puede usar cualquier código de servo, hardware o biblioteca para controlar estos servos. Tamaño compacto, incluye accesorios. [69]

3.2.1. Características:

- Voltaje de operación: 4.8 V a 6 V
- Velocidad de operación: 0.1 s/60° (4.8 V), 0.08 s/60° (6 V)
- Torque detenido: 1.8 kgf·cm (4.8 V), 2.2 kgf·cm (6 V).
- Con cojinete.
- Banda muerta: 5 μ s.
- Peso ligero: 13.4 g.
- Dimensiones compactas: Largo 22.5 mm, ancho 12 mm, altura 35.5 mm aprox.
- Largo del cable: 25 cm aprox.
- Engranaje metálico.
- Incluye 3 brazos o cuernos (horns) y su tornillo de sujeción, 2 tornillos para montaje del servo y cable de conexión con conector.
- Conector universal tipo “S” compatible con la mayoría de receptores incluyendo Futaba, JR, GWS, Cirrus, Blue Bird, Blue Arrow, Corona, Berg, Spektrum y Hitec, entre otros. [69]

3.2.2. Cables de Conexión:

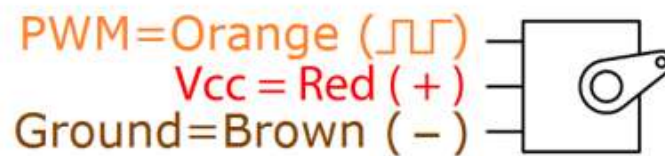


Figura 3.2.2-1 Pinout micro servo. [69]

- Cable rojo es para alimentación, Vcc ($\sim$ +5volts);
- Cable café para conexión a tierra (GND);
- Cable naranja es la línea de control de la señal (pwm) codificada para controlar el ángulo en el que se debe posicionar.

3.2.3. Posicionamiento de un servomotor

Para posicionar un servomotor tenemos que aplicar un pulso eléctrico, cuya duración determinará el ángulo de giro del motor. Un pulso es sencillamente enviar corriente eléctrica al motor durante un tiempo determinado. En la Figura 3.2.3-1 se muestran los ciclos de trabajo y el ángulo de salida del servomotor.

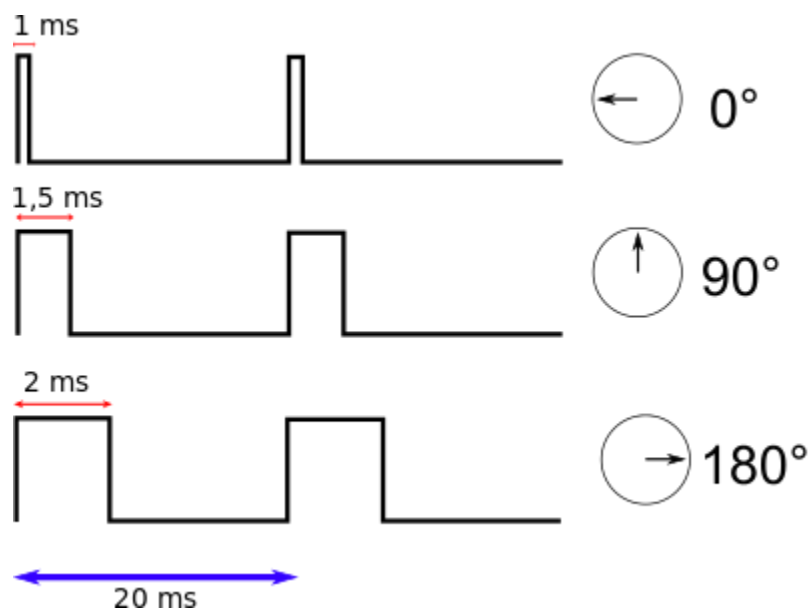


Figura 3.2.3-1 Diagrama de pulsos con respecto al tiempo, para controlar la posición o giro del motor. [70]

3.3. FC-102 rainbow LED RGB

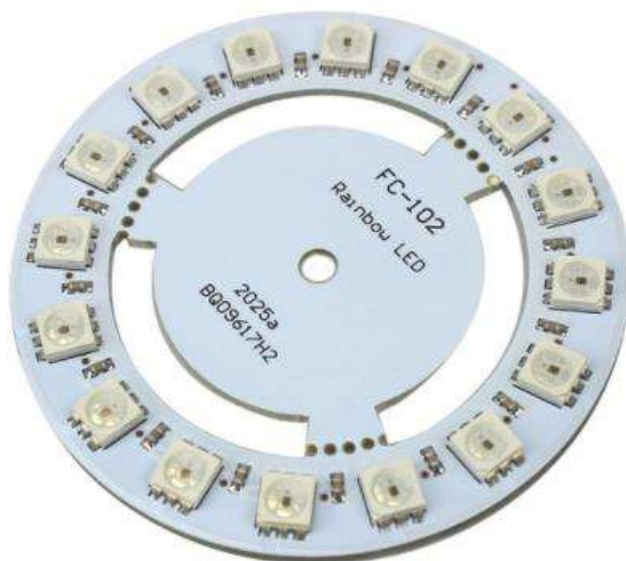


Figura 3.3-1 Anillo RGB con 16 LEDs y controlador integrado. [71]

16 Bits LED WS2812 WS2811 5050 RGB con conductores integrados FC-102

- Modelo: Módulo 5050 Redondo 16 full-color LED
- Dimensión: 60mm Diámetro
- Chip: WS2811 (built-in LED)
- LED: 5050 RGB full color paquete de alta luminosidad.
- Voltaje: 5V.
- Puertos: Digital.
- Plataforma: Arduino microcontrolador.
- 5050 alta luminosidad LED's, built-in control chip, solo un puerto IO para controlar los múltiples LED's.
- Circuito de configuración de chip integrado, la distorsión de la señal no es acumulativa, pantalla estable.
- Brillo ajustable # colores, 256, 160,000-color TFT display, frecuencia de operación no menos que 400Hz / S
- Interfaz serie de nivel uniforme, a través de una línea de señal para completar la recepción y decodificación de datos.
- Frecuencia de actualización de 30 cuadros /seg. Incluso series de no menos de 512 puntos.
- Envía y recibe datos hasta una velocidad de 800Kbps.
- LED de alta luminosidad, color claro uniformidad de alto brillo.
- Conexión directa para ambos extremos. [71]

4. Capítulo 4. Diseño e Implementación.

En la Figura 4-1 se muestra la forma en que se deben comunicar cada dispositivo o elemento del sistema.

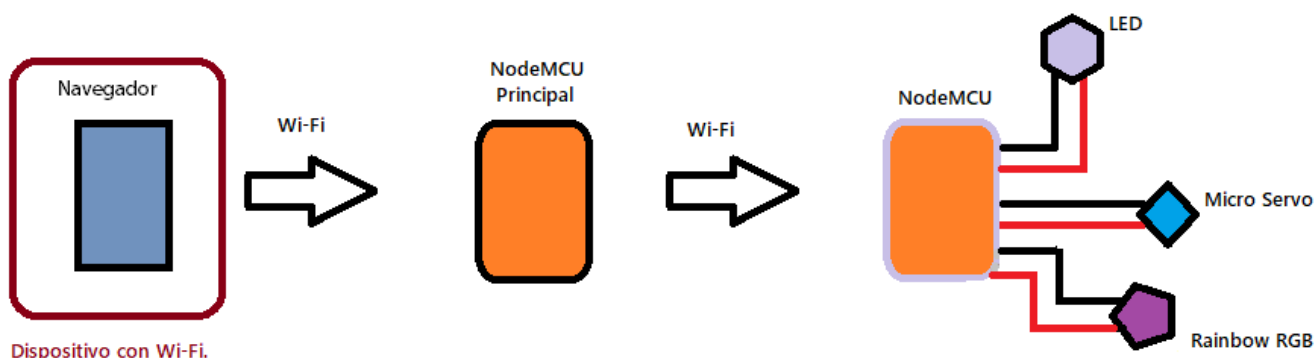


Figura 4-1 Diagrama a bloques del funcionamiento del sistema. [72]

El funcionamiento del sistema de la Figura 4-1 comienza con la conexión Wi-Fi y un navegador. Al conectarse al NodeMCU principal por medio del navegador se visualizan los controles del sistema y así interactuar con cada uno de ellos. NodeMCU principal recibe las peticiones solicitadas por usuario, para posteriormente transmitir dicha información a otro NodeMCU en el cual están conectados los dispositivos a controlar.

4.1. Procedimiento de diseño.

1. Cargar el Firmware adecuado a NodeMCU para la interpretación de programas en lenguaje Lua.
2. Realizar scripts en NodeMCU en servidor y módulos NodeMCU clientes.
 - a. Crear la WLAN a través de NodeMCU y conectarse a dicha red con algún otro dispositivo con Wi-Fi (celular, tableta portátil, pc, etc.)
 - b. Controlar los actuadores del sistema por medio de los GPIOs de NodeMCU.
3. Realizar control en los actuadores a través de un dispositivo con navegador
 - a. Realizar conexión entre los dispositivos de la red por medio de sockets TCP.
 - b. Crear una página Web en HTML que permita controlar los actuadores.
4. Realizar las conexiones de funcionamiento de la Figura 4-1.
 - a. Modo punto de acceso y modo estación.
 - b. Conexión entre dos módulos NodeMCU.

NodeMCU cuenta con un sitio Web donde se tiene la información del funcionamiento del módulo NodeMCU y la creación de scripts en lenguaje LUA. Se desglosa en módulos que explican las funciones que tiene el kit de desarrollo. La aplicación más popular de Lua para el ESP8266 se conoce como el NodeMCU Lua está disponible en su repositorio github. <https://nodemcu.readthedocs.io>, ver Figura 4.1-1.

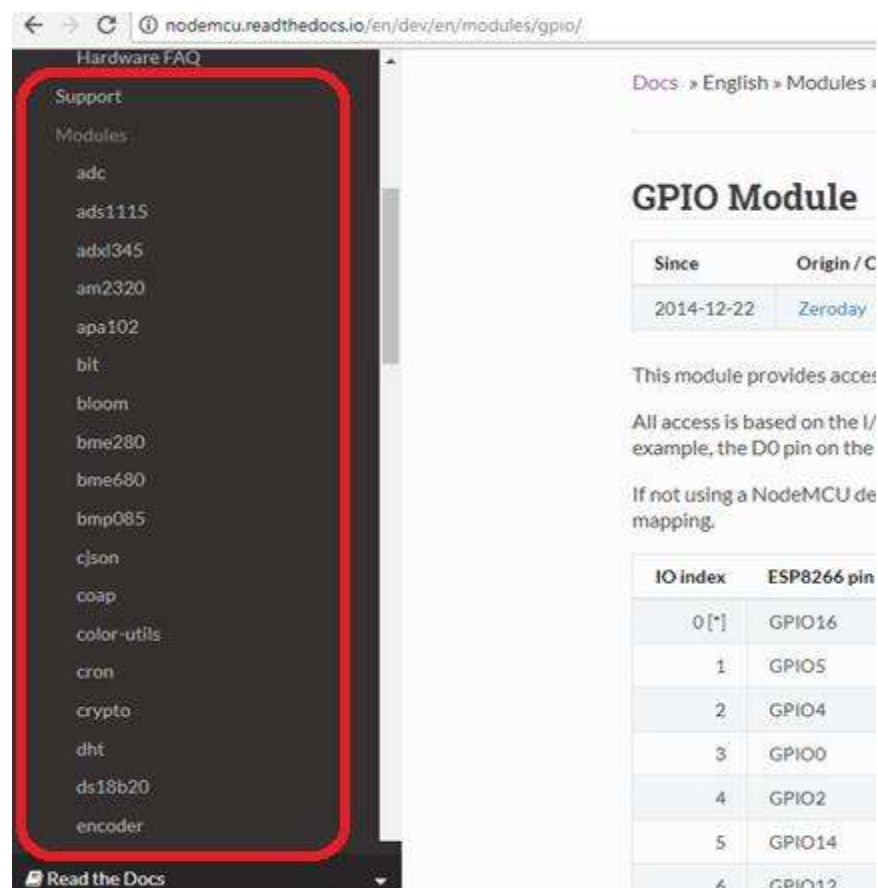


Figura 4.1-1 Captura de imagen de la página mostrando la sección de los módulos. [73]

4.2. Firmware NodeMCU

Plataformas:

Existen varias plataformas para programar ESP8266.

- Arduino.
- Lua NodeMCU.
- JavaScript.
- ESP8266 Basic.

Este proyecto está basado en la plataforma Lua NodeMCU

Lua NodeMCU es un firmware para el ESP8266 basado en el Espressif Non-OS SDK y usa el lenguaje de programación Lua, es un potente lenguaje de scripting disponible en entornos de ESP8266

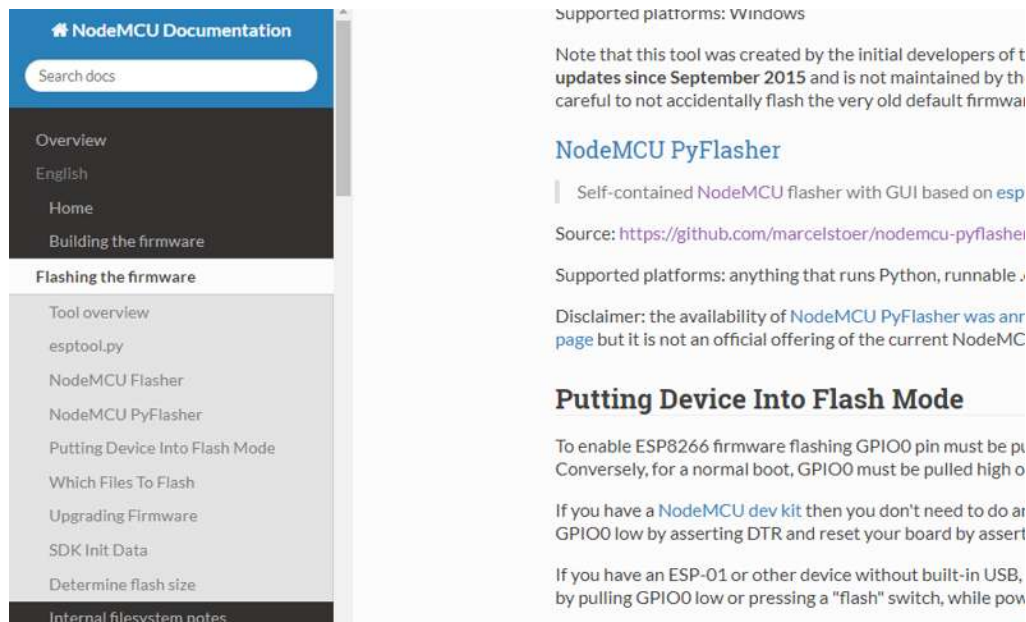


Figura 4.2-1 Captura de imagen de la página web para empezar a crear un propio firmware. [73]

En la Figura 4.1-1 en la parte izquierda se puede consultar los diferentes archivos y módulos para LuaNodeMCU. Comenzaremos con el Firmware tal como lo muestra en la Figura 4.2-1. En este caso la programación del firmware se realiza de manera gráfica. Ahí mismo página web nos explica otra manera la cual se realiza por medio de comandos.

Una vez teniendo la copia del firmware, se flashea usando una herramienta de programación para ESP8266 “NODEMCU FIRMWARE PROGRAMMER” ver Figura 4.2-2.

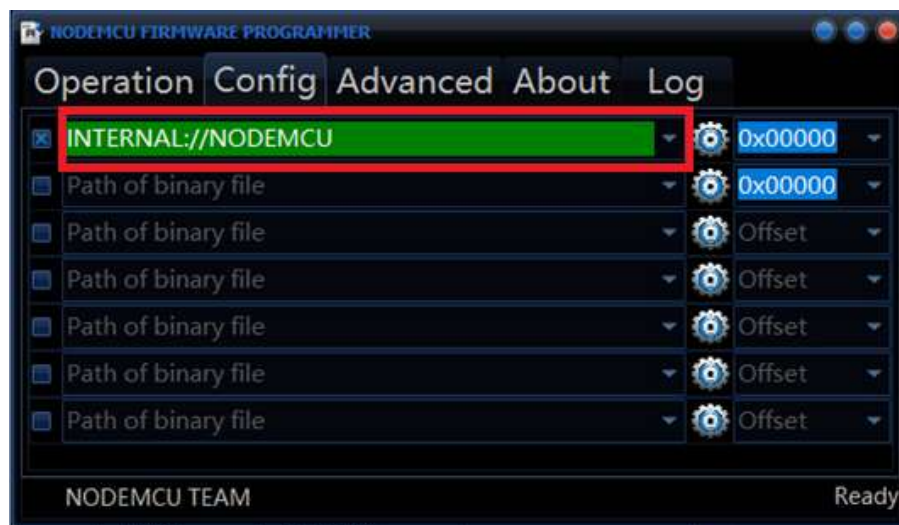


Figura 4.2-2 Herramienta Node MCU Firmware Programmer. [74]

Se trata de un programa ejecutable, el programa que viene de manera predeterminada con un Firmware precargado mostrado en la Figura 4.2-2 se trata de un bootloader el cual se encuentra en la pestaña “Config”.

Luego de conectar nuestro Kit de desarrollo NodeMCU a la PC. Presionamos Click en Flash (F), asegurándonos que este se encuentre en el puerto correcto en este caso el dispositivo se encuentra en el puerto COM9.



Figura 4.2-3 Flashear NodeMCU [74]

Cuando comience a cargar el firmware nos aparecerá un código QR como en la Figura 4.2-4, de mismo modo el LED RX/TX que se encuentra en el ESP-12 estará parpadeando.

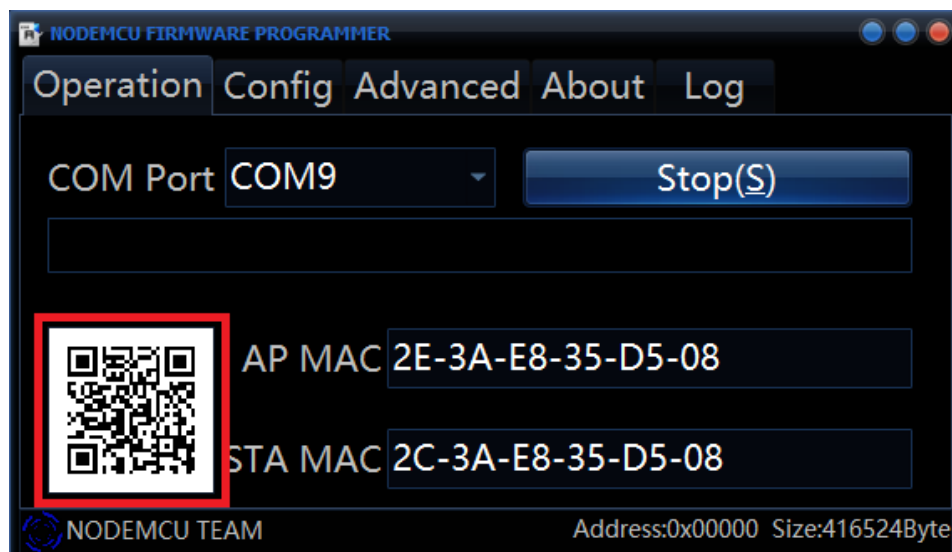


Figura 4.2-4 Cargando Firmware. [74]

En internet, se encuentran varios firmwares que podremos cargar en nuestro ESP. Sin embargo la página proporcionada también nos permite construir nuestro propio firmware con los módulos que son requeridos para nuestro propósito ver Figura 4.2-5.

Select modules to include

☐ ADC	☐ end user setup	☐ perf	☐ Switec
☐ ADS1115	☒ file	☒ PWM	☐ TCS34725
☐ ADXL345	☐ gdbstub	☒ RC (no docs)	☐ TM1829
☐ AM2320	☒ GPIO	☐ rfswitch	☒ timer
☐ APA102	☐ HDC1080	☐ rotary	☐ TSL2561
☐ bit	☐ HMC5883L	☐ RTC fifo	☐ U8G
☐ Bloom filter	☒ HTTP	☐ RTC mem	☒ UART
☐ BME280	☐ HX711	☐ RTC time	☐ UCG
☐ BME680	☐ I ² C	☐ Si7021	☒ websocket
☐ BMP085	☐ L3G4200D	☐ Sigma-delta	☒ WiFi
☐ CoAP	☐ MCP4725	☐ SJSON	☐ WiFi monitor
☐ color utils	☐ mDNS	☐ SNTP	☐ WPS
☐ Cron	☐ MQTT	☐ Somfy	☐ WS2801
☐ crypto	☒ net	☐ SPI	☒ WS2812
☐ DHT	☒ node	☐ SQLite 3	☒ WS2812 effects
☐ DS18B20	☐ 1-Wire	☐ struct	☐ XPT2046
☐ encoder	☐ PCM		

Figura 4.2-5 Construcción de firmware. [75]

Después de diseñar y descargar el firmware creado, procedemos a programarlo en el NodeMCU de igual manera que el bootloader a gregando este firmware como se muestra en la Figura 4.2-6.

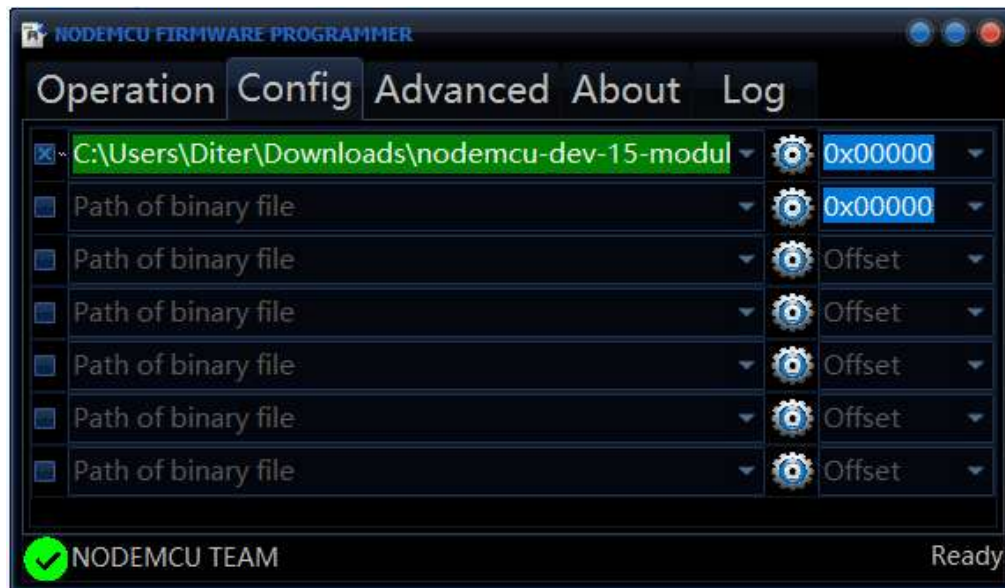


Figura 4.2-6 Cargando Firmware creado con 15 módulos. [74]

Volviendo a la pestaña Operation seleccionar el botón Flash(F) y por último se muestra cuando el firmware ya se ha cargado ver Figura 4.2-7.

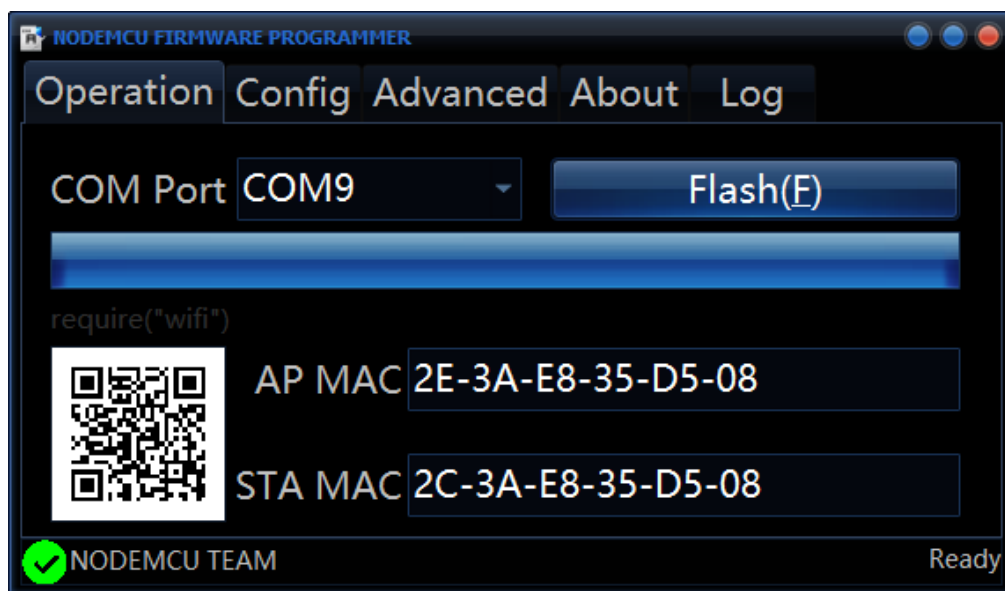


Figura 4.2-7 Terminación de Carga de Firmware. [74]

Teniendo el firmware cargado en nuestro kit de desarrollo NodeMCU. Proseguimos con el paso 2 de nuestro proceso de implementación.

4.3. Herramienta de programación

El IDE de ESPlorer es un entorno de desarrollo para la construcción de aplicaciones Lua funcional para el ESP8266 ver Figura 4.3-1.

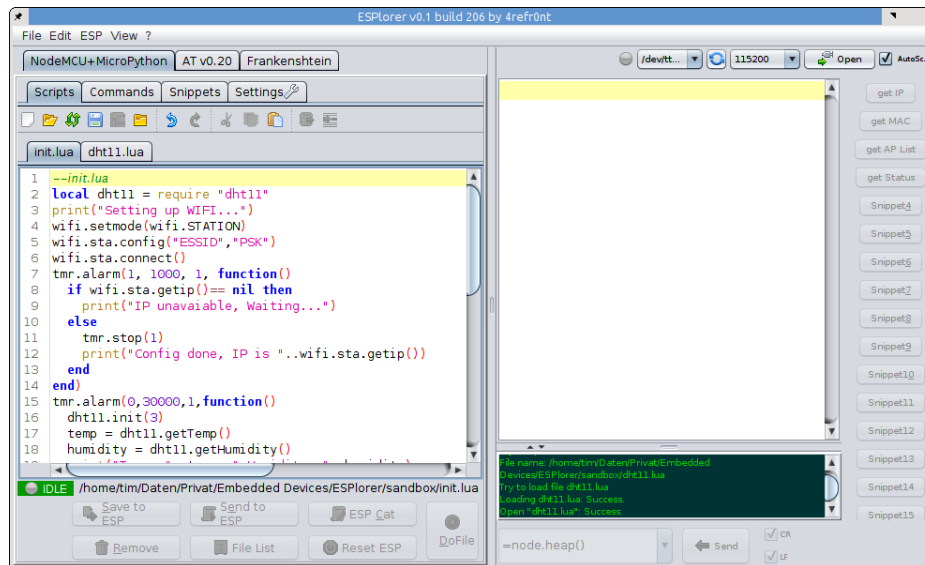


Figura 4.3-1 Ejemplo de IDE Esplorer. [36]

En la Figura 4.3-2 se muestra otro IDE funcional para ESP8266 de nombre LuaUploader, siendo este el usado para realizar el proyecto y cargar los programas a NodeMCU.

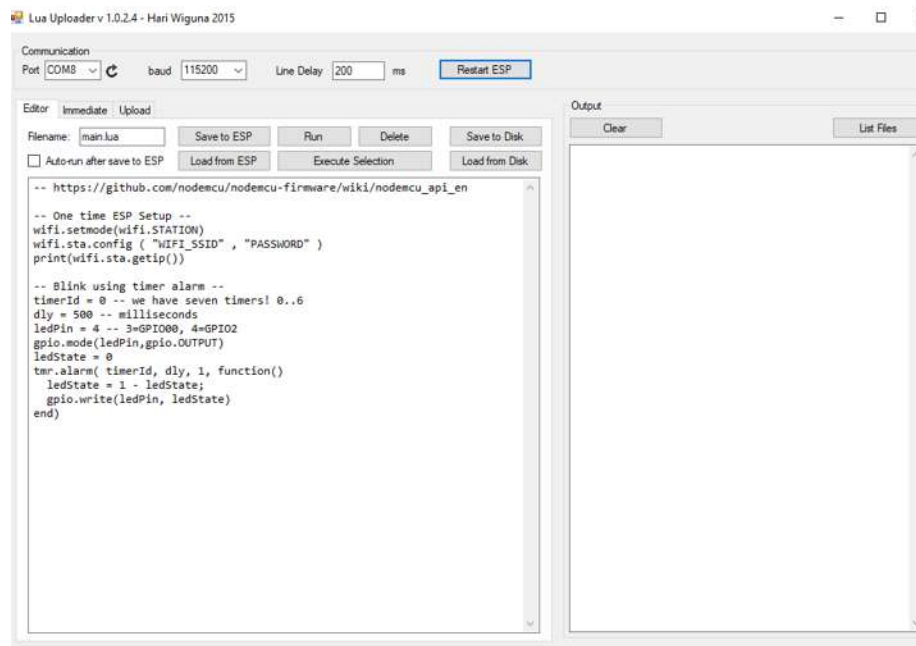


Figura 4.3-2 Lua Uploader. [76]

4.4. Red WiFi a través de NodeMCU.

En la página de NodeMCU se encuentra la información del funcionamiento y programación del módulo Wi-Fi. El cual permite tener los 4 modos de operación antes vistos en el Capítulo 2.2.

Un punto de acceso (AP) es un dispositivo que proporciona acceso a la red Wi-Fi a dispositivos (estaciones) y los conecta a una red cableada. ESP8266 puede proporcionar una funcionalidad similar, excepto que no tiene interfaz para una red cableada. Tal modo de operación se llama punto de acceso suave (soft-AP). La cantidad máxima de estaciones conectadas de manera simultánea al soft-AP es cinco. [77]

Este será el modo en el que opera nuestro primer módulo NodeMCU para conectarnos por primera vez a una red Wi-Fi a través de él. Wifi.setmode permite asignar el modo de operación Wi-Fi. Como se observa en la Figura 4.4-1.

Syntax

```
wifi.setmode(mode[, save])
```

Parameters

- **mode** value should be one of
 - **wifi.STATION** for when the device is connected to a WiFi router. This is often done to give the device access to the Internet.
 - **wifi.SOFTAP** for when the device is acting *only* as an access point. This will allow you to see the device in the list of WiFi networks (unless you hide the SSID, of course). In this mode your computer can connect to the device, creating a local area network. Unless you change the value, the NodeMCU device will be given a local IP address of 192.168.4.1 and assign your computer the next available IP address, such as 192.168.4.2.
 - **wifi.STATIONAP** is the combination of **wifi.STATION** and **wifi.SOFTAP**. It allows you to create a local WiFi connection *and* connect to another WiFi router.
 - **wifi.NULLMODE** changing WiFi mode to NULL_MODE will put wifi into a low power state similar to MODEM_SLEEP, provided **wifi.nullmodesleep(false)** has not been called.
- **save** choose whether or not to save wifi mode to flash
 - **true** WiFi mode configuration **will** be retained through power cycle. (Default)
 - **false** WiFi mode configuration **will not** be retained through power cycle.

Figura 4.4-1 Sintaxis para el modo de operación. [77]

En este caso será: `wifi.setmode(wifi.SOFTAP)`.

Después de configurar en modo de punto de acceso, se asigna un SSID y una Contraseña principalmente. Existen otras opciones adicionales como canal de transmisión, tipo de seguridad, etc, ver Figura 4.4-2.

Syntax

```
wifi.ap.config(cfg)
```

Parameters

- `cfg` table to hold configuration
 - `ssid` SSID chars 1-32
 - `pwd` password chars 8-64
 - `auth` authentication method, one of `wifi.OPEN` (default), `wifi.WPA_PSK`, `wifi.WPA2_PSK`, `wifi.WPA_WPA2_PSK`
 - `channel` channel number 1-14 default = 6
 - `hidden` false = not hidden, true = hidden, default = false
 - `max` maximum number of connections 1-4 default=4
 - `beacon` beacon interval time in range 100-60000, default = 100
 - `save` save configuration to flash.
 - `true` configuration will be retained through power cycle. (Default)
 - `false` configuration will not be retained through power cycle.
 - Event callbacks will only be available if `WIFI_SDK_EVENT_MONITOR_ENABLE` is uncommented in `user_config.h`
 - Please note: To ensure all SoftAP events are handled at boot time, all relevant callbacks must be registered as early as possible in `init.lua` with either `wifi.ap.config()` or `wifi.eventmon.register()`.
 - `staconnected_cb`: Callback executed when a new client has connected to the access

Figura 4.4-2 Sintaxis para la configuración en modo punto de acceso. [77]

Configuración básica que necesita el punto de acceso son ssid y pwd. Sin embargo considere configurar el canal para evitar interferencias entre otras redes que se encuentren cerca. Los demás parámetros se dejaron con los valores predeterminados.

Realizamos el código y se carga al ESP8266.

Nota: el nombre del programa que será cargado debe de ser “init.lua” y deberá ser respetado ya que el Firmware siempre espera leer un código con ese nombre.

En la Figura 4.4-3 se aprecia la captura de imagen tomada desde el celular utilizado para la conexión e interacción con los módulos NodeMCU.

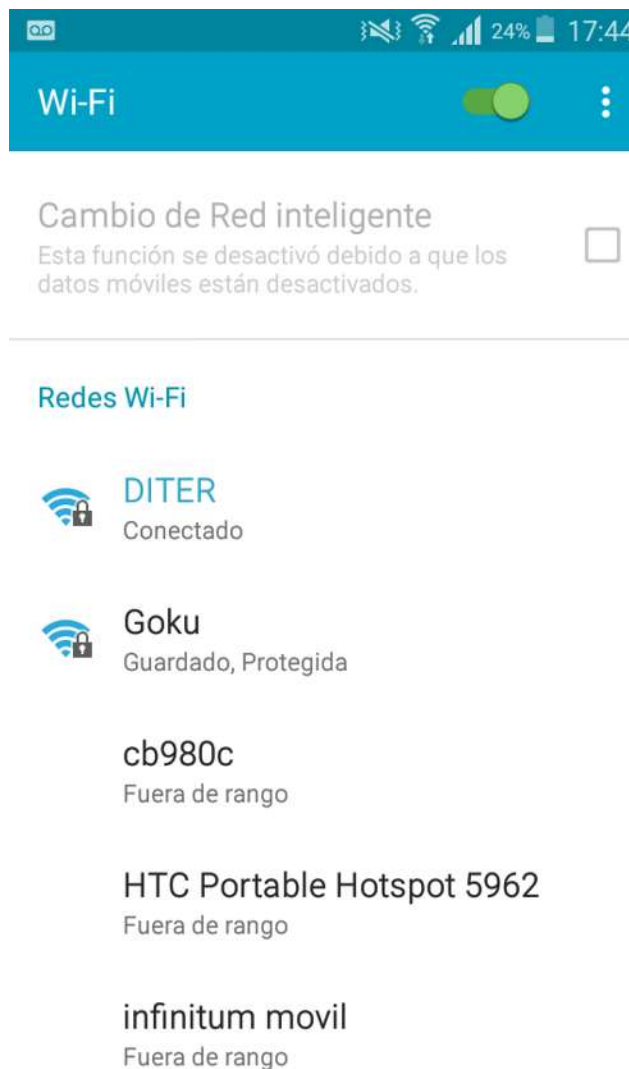


Figura 4.4-3 Conexión a la Red por medio de un un celular con conexión Wi-Fi. [78]

La red creada por el módulo con SSID de nombre DITER en la cual aparece que existe conexión a la red. Cabe mencionar que por medio de esta red no se puede conectar a internet sólo se trata de una WLAN y existe comunicación con los dispositivos que se encuentren conectados a la misma red.

4.5. Control de Actuadores

Un **actuador** es un dispositivo capaz de transformar energía eléctrica en energía hidráulica, neumática o eléctrica con la finalidad de generar un efecto sobre un proceso automatizado. Este recibe la orden de un regulador o controlador y en función a ella genera la orden para activar un elemento final de control. [79]

Los actuadores considerados para este sistema son:

- Diodo LED.
- Servo Motor.
- Rainbow FC-102 RGB.

Los cuales estarán conectados a los pines GPIOs del módulo. Encargados de realizar control de iluminación en intensidad, frecuencia para el actuador tipo led. El cambio de posición en dos ejes es realizado por dos servomotores y por último rainbow FC-102 es quien tiene más interacción en el sistema ya que nos permite realizar los efectos de color.

4.5.1. Diodo LED.

La conexión de este dispositivo es como se muestra en la Figura 4.5.1-1.

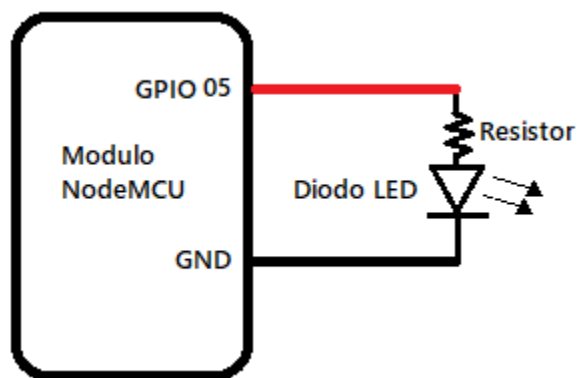


Figura 4.5.1-1. Diagrama de conexión NodeMCU y Diodo LED. [80]

Debido a que un diodo led enciende con corriente continua y polarización directa, este puede ser controlado de dos maneras:

1. Como ON-OFF, que consiste en suministrar voltaje y cortar el suministro de voltaje respectivamente.
2. Control de brillo de encendido y encendido intermitente, lo cual se realiza por medio de señales PWM, el ciclo de trabajo determinará la intensidad del brillo con que encenderá el led y la frecuencia de la señal permite un encendido intermitente.

Diseño e Implementación

Las funciones del Módulo GPIO. Describe cómo utilizar los pines como puertos generales de entrada y salida permitiendo leer, escribir, entre otras funciones; todo dependerá el modo de funcionamiento del pin.

Este código nos permite encender un led por medio del pin 1 que corresponde al GPIO05 como se muestra en la Figura 4.5.1-2.

-- Se agrega valor a la variable "pin" la cual será utilizada para identificar que número de IOindex correspondiente, se coloca el pin en modo salida, se pone el pin el alto.

```
pin=1
```

```
gpio.mode(pin, gpio.OUTPUT)
```

```
gpio.write(pin, gpio.HIGH)
```

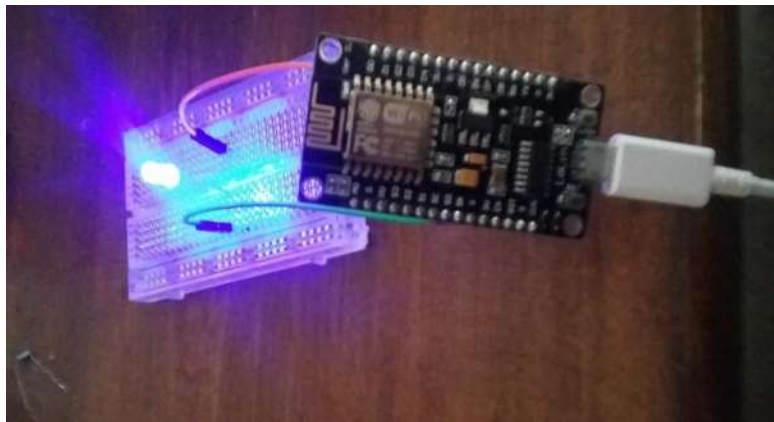


Figura 4.5.1-2 Encendido de LED. [81]

La siguiente línea de comando apagará el LED como se aprecia en la Figura 4.5.1-3.

```
gpio.write(pin, gpio.LOW)
```



Figura 4.5.1-3 Apagado del LED. [81]

De esta manera se controla el encendido y apagado de un LED.

Los GPIOs también pueden ser usados como PWM. A través del módulo PWM, hay que considerar que cuando el GPIO es puesto en este modo deja de funcionar el módulo GPIO el cual es el modo que permite encender y apagar el LED.

Para colocar el pin en modo PWM la sintaxis es la que se muestra en la Figura 4.5.1-4.

Syntax

```
pwm.setup(pin, clock, duty)
```

Parameters

- **pin** 1~12, IO index
- **clock** 1~1000, pwm frequency
- **duty** 0~1023, pwm duty cycle, max 1023 (10bit)

Figura 4.5.1-4 Sintaxis para una salida pwm [82]

NodeMcu tiene un rango de frecuencia para una señal PWM de 1Hz a 1000Hz y un ciclo de trabajo de 0 a 1023 valores en este ultimo solo se necesita realizar una sencilla regla de 3 para asignarle el valor en porcentaje, ejemplo: 0=0% ; 1023=100%

También podemos modificar la frecuencia o el ciclo de trabajo individualmente con la sintaxis que se muestra en la Figura 4.5.1-5 .

Syntax

```
pwm.setclock(pin, clock)
```

Syntax

```
pwm.setduty(pin, duty)
```

Figura 4.5.1-5 Sintaxis para modificar ciclo de trabajo o frecuencia en una salida pwm. [82]

Diseño e Implementación

En las siguientes Figuras sólo se incluyó el efecto que se realiza a través del ciclo de trabajo, ya que, observar la variación de frecuencia por medio de imágenes hace que no se aprecie, debido a esto no se proporcionan ejemplos de variación de frecuencia. Ver Figura 4.5.1-6, Figura 4.5.1-7, Figura 4.5.1-8 y Figura 4.5.1-9.

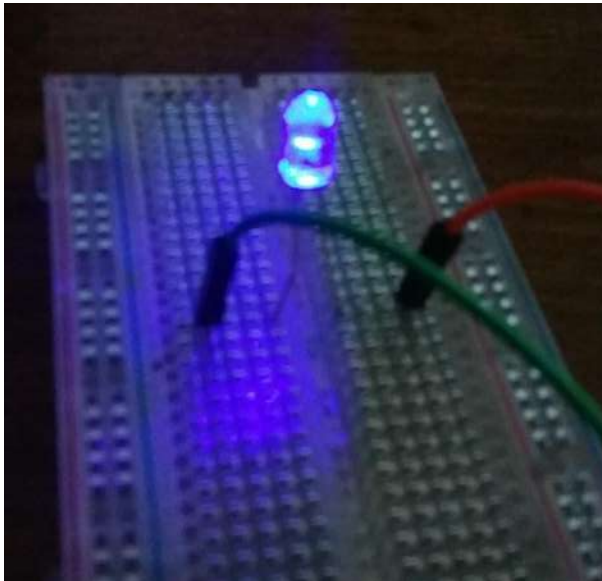


Figura 4.5.1-6 PWM al 25%. [97]

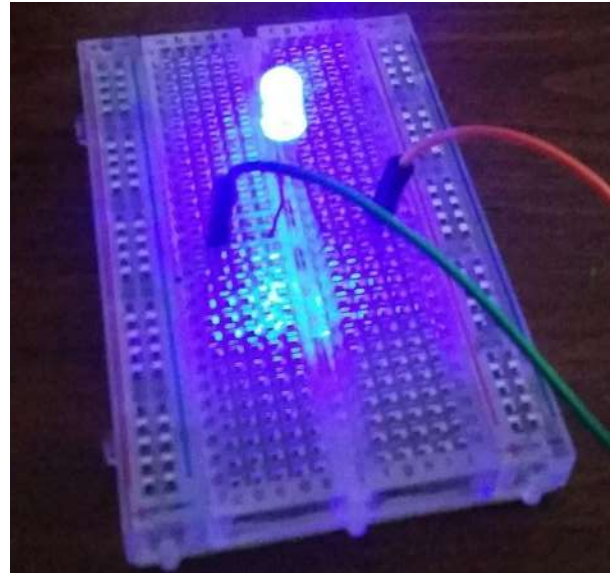


Figura 4.5.1-7 PWM 50%. [97]

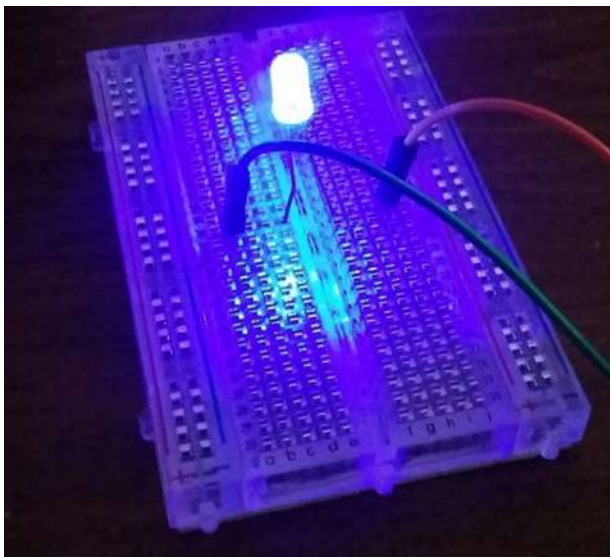


Figura 4.5.1-8 PWM 75%. [97]

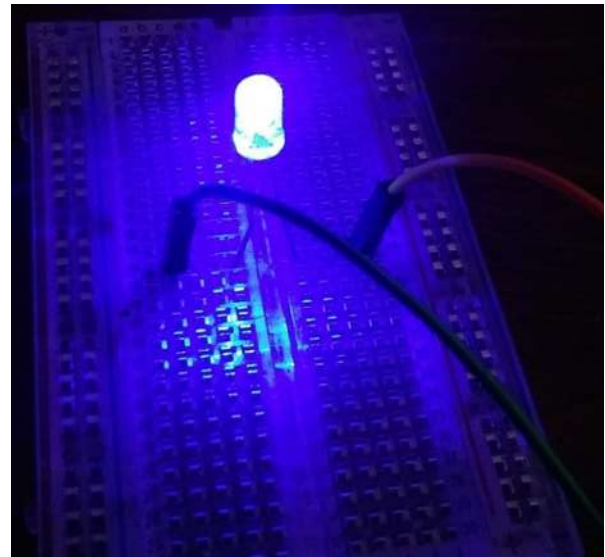


Figura 4.5.1-9 PWM 90%. [97]

4.5.2. Servo Motor.

En la Figura 4.5.2-1 se muestra el diagrama de conexión para un servomotor el cual utiliza una fuente alterna para el suministro de potencia., esto se debe a que nodeMCU no proporciona la suficiente corriente. Nota: no olvidar conectar ambas tierras.

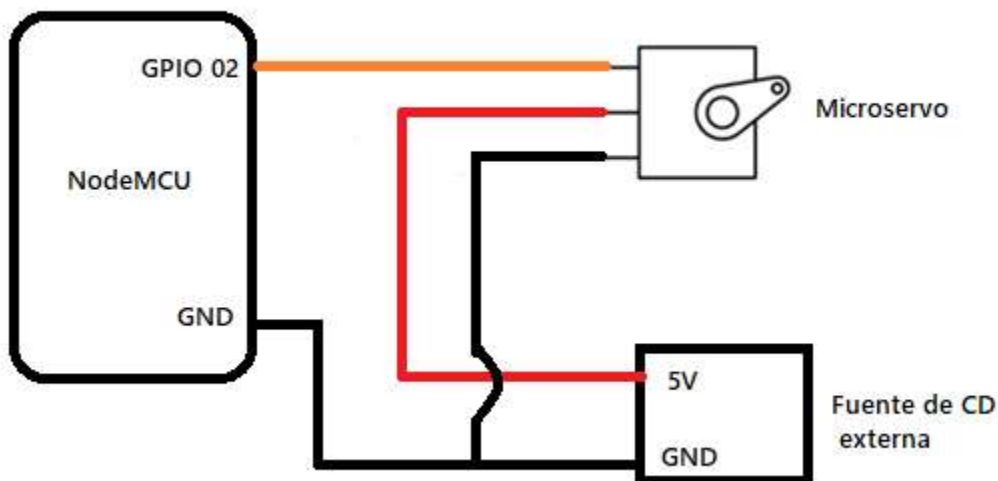


Figura 4.5.2-1 Diagrama de conexión de un Servomotor. [83]

La señal PWM de NodeMCU tiene valores de 0 a 1023 correspondientes al valor del ciclo de trabajo (0 = 0%, 1023=100%) al igual, realizando una sencilla regla de 3, se pueden ajustar los valores adecuados para el ciclo de trabajo requerido para posicionar el servomotor. Visto en el capítulo 3.2.3 sobre el posicionamiento del servomotor Figura 3.2.3-1. El kit de desarrollo NodeMCU V3 permite obtener a la salida 5V, por medio del pin VUSB el cual será usado para las pruebas de control del Servo Motor ver Figura 4.5.2-2.



Figura 4.5.2-2 Servo Motor Posicionado a 0°. [84]

En la Figura 4.5.2-3 se posiciono el servomotor a 90°.

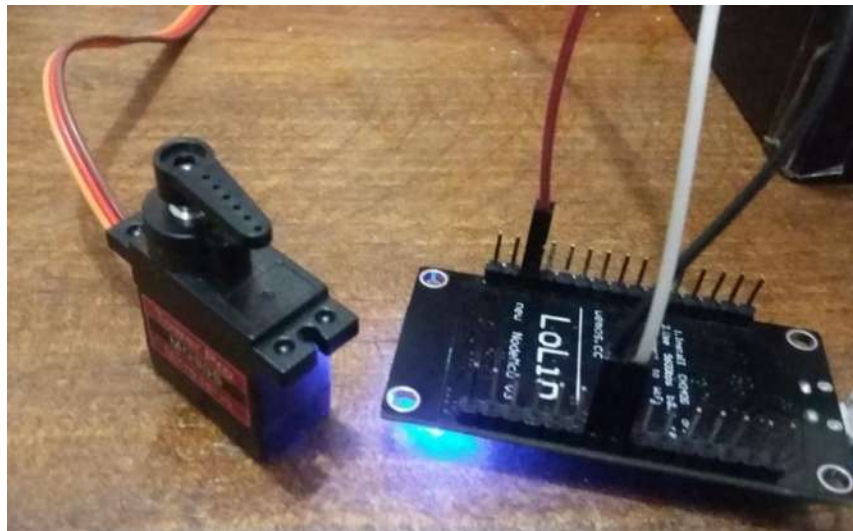


Figura 4.5.2-3 Servo Motor Posicionado a 90°. [84]

Y por ultimo en la Figura 4.5.2-4 se posiciono a 180° que son el maximo de grados de libertad que el servo puede moverse.

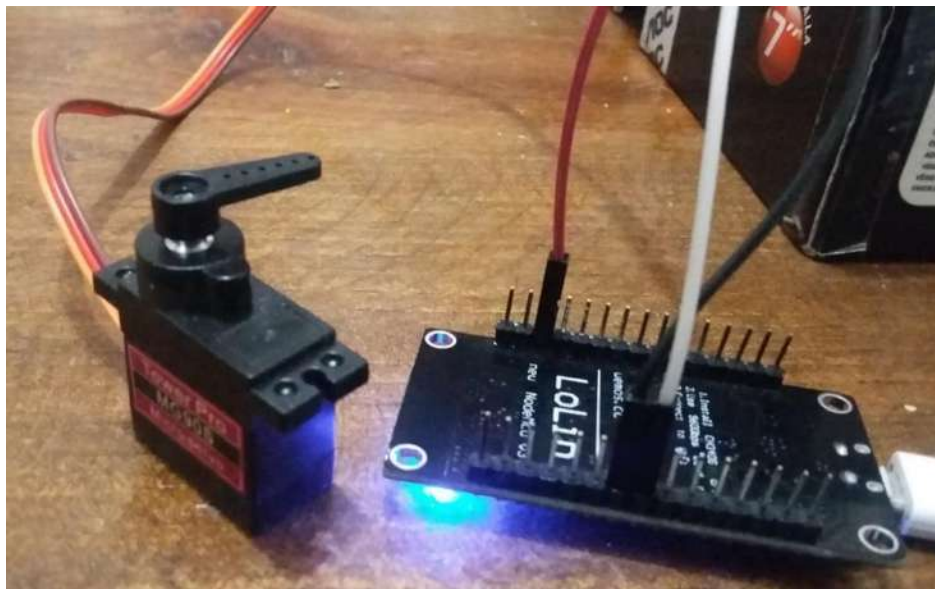


Figura 4.5.2-4 Servo Motor Posicionado a 180°. [84]

Se realiza lo mismo con el otro servo motor sólo colándolo en otro GPIO. Para tener movimiento en dos ejes sólo es necesario que un servo motor este orientado en sentido de un eje (por ejemplo eje vertical) y el otro servo motor orientado en el eje horizontal.

4.5.3. Rainbow FC-102

NodeMCU también permite realizar efectos con WS2812. Por medio de una biblioteca la cual permite realizar varios efectos precargados, para controlar los LED RGB o RGBW. El trabajo que nos toca realizar es programar la combinación de colores y los diversos efectos. Para fines prácticos y de prototipo solo se muestra la realización de algunos colores.

Para la alimentación a los leds WS2812B es necesario utilizar una fuente externa debido al consumo de corriente, la conexión se realiza como en la Figura 4.5.3-1.

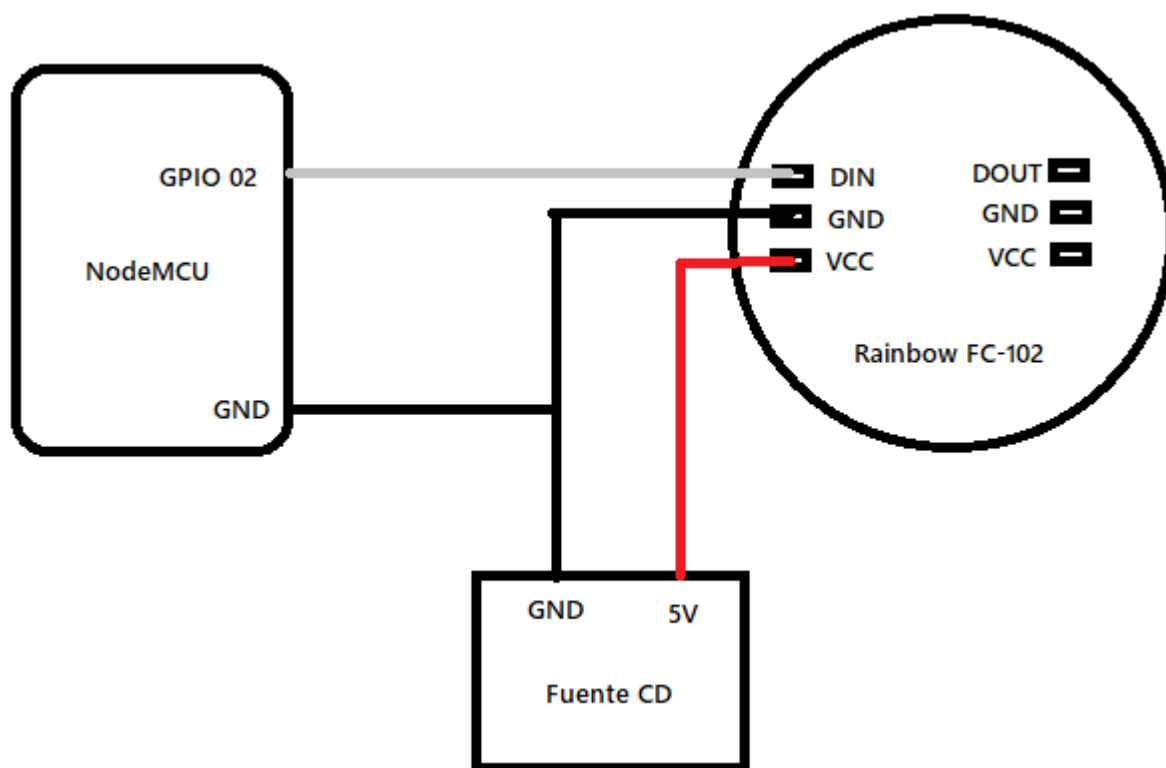


Figura 4.5.3-1 Diagrama de conexión para rainbow FC-102. [85]

DOUT nos permite conectar otro dispositivo de LED's WS2812 conectado en serie (DOUT con DIN del siguiente dispositivo) y así controlar varios dispositivos WS2812 desde un solo pin del microcontrolador tomando en consideración el número de LED's WS2812 en serie.

En la documentación NodeMCU se encuentra WS2812 effects Module que es una biblioteca que tiene efectos programados para ser utilizados en este tipo de leds ver Figura 4.5.3-2 un ejemplo en la sintaxis que nos permite inicializar nuestro dispositivo WS2812.

```
-- init the ws2812 module
ws2812.init(ws2812.MODE_SINGLE)
-- create a buffer, 60 LEDs with 3 color bytes
strip_buffer = ws2812.newBuffer(60, 3)
-- init the effects module, set color to red and start blinking
ws2812_effects.init(strip_buffer)
ws2812_effects.set_speed(100)
ws2812_effects.set_brightness(50)
ws2812_effects.set_color(0,255,0)
ws2812_effects.set_mode("blink")
ws2812_effects.start()
```

Figura 4.5.3-2 Sintaxis para inicializar un módulo WS2812. [82]

En la Figura 4.5.3-2 se aprecia que el primer paso es iniciar el módulo se crea un buffer en el cual se le asignan los parámetros de cantidad de LEDs a controlar en esta parte es donde pueden existir varios módulos conectados en serie y lo único que se modifica sería este parámetro de acuerdo a la cantidad de LEDs, el segundo es la cantidad de bytes en este caso los LEDs son RGB por eso el número 3, luego se inician los efectos en este caso prendera de color rojo con brillo de 50 a una velocidad de 100 utilizando el efecto blink ya teniendo los parámetros de efectos se inicia el dispositivo con el comando start. En la Figura 4.5.3-3 se muestran algunas combinaciones de los 3 bytes para diferentes colores.

```
white = string.char(255, 255, 255)
off = string.char(0,0,0)
blue=string.char(0, 0, 255)
pink=string.char(255, 0, 255)
purple=string.char(20, 0, 20)
red =string.char(100, 0, 0)
green =string.char(0, 100, 0)
yellow =string.char(100, 100, 0)
gold =string.char(180, 100, 0)
silver = string.char(60, 60, 100)
orange = string.char(255/4,127/4,0)
violet = string.char(138,43,226)
indigo = string.char(75,0,130)
```

Figura 4.5.3-3 Ejemplos de algunos colores para WS2812 RGB. [86]

En la Figura 4.5.3-4 se observa el resultado de este dispositivo utilizando el color gold que fue obtenido de la lista de colores de la Figura 4.5.3-3 y así asumiendo que el dispositivo funciona correctamente.



Figura 4.5.3-4 Resultado de rainbow FC-102 utilizando el color gold. [87]

4.6. Conexión entre dispositivos por medio de sockets TCP.

Esta parte es el corazón del sistema. Está encargado de conectar cualquier dispositivo conectado a la red y hacer que estos se comuniquen por medio de TCP/IP. De esta forma nos permite realizar el control desde cualquier dispositivo con Wi-Fi e interfaz web y acceder a cualquier módulo NodeMCU y realizar el control.

4.6.1. Control desde el dispositivo con Wi-Fi y navegador web

Socket TCP

El módulo “net” explica cómo crear conexión servidor - un cliente.

El kit de desarrollo NodeMCU hace la función de servidor y el dispositivo es el cliente.

1. Para comenzar la conexión primero debemos crear un servidor el cual escuche las peticiones que uno solicite.

```
net.createServer(net.TCP, 30) -- 30s timeout
```

Socket TCP con 30 segundos en espera si el cliente está inactivo después de ese tiempo lo desconecta.

2. Creado el servidor le asignamos un puerto en el cual escuchará al cliente que quiera conectarse en la Figura 4.6.1-1 se muestra la sintaxis que permite que el servidor escuche.

net.server:listen()

Listen on port from IP address.

Syntax

```
net.server.listen([port],[ip],function(net.socket))
```

Parameters

- `port` port number, can be omitted (random port will be chosen)
- `ip` IP address string, can be omitted
- `function(net.socket)` callback function, pass to caller function as param if a connection is created successfully

Figura 4.6.1-1 Sintaxis para poner a escuchar al servidor. [77]

El servidor escucha por el puerto asignado cuando un cliente se conecta por el puerto el servidor hace una llamada de función a `net.socket`, el socket nos permite una variedad de acciones entre servidor-cliente unas de las acciones es enviar y recibir datos. Los cuáles serán los que utilizaremos para la comunicación entre los dispositivos y el control de actuadores.

En la Figura 4.6.1-2 se muestra un ejemplo donde el servidor escucha por el puerto 80 cuando algún cliente se conecta a él, este recibe la petición en seguida ejecuta una función para posterior mente enviarle al cliente la cadena de caracteres “hello world”.

Example

```
-- server listens on 80, if data received, print data to console and send "hello world" back
-- 30s time out for a inactive client
sv = net.createServer(net.TCP, 30)

function receiver(sck, data)
  print(data)
  sck:close()
end

if sv then
  sv:listen(80, function(conn)
    conn:on("receive", receiver)
    conn:send("hello world")
  end)
end
```

Figura 4.6.1-2 Ejemplo de código para hacer que el servidor escuche. [77]

Uso de HTML para la interacción entre el servidor y el dispositivo con Wi-Fi e interfaz web.

En la Figura 4.6.1-3 se muestra código HTML la idea en este momento es crear una página web simple. Se observa el encabezado de la página, el tipo de botón y el método de envío, al ser presionado manda la cadena o valor que tiene que tiene asignado "value" por medio de la variable name="mcu_do"

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <meta content="text/html; charset=utf-8">
5  <title>PROYECTO DITER</title></head>
6  <body><h1>PROYECTO DITER!</h1>
7
8
9  -- Buttons
10 <p> SELECCIONAR EFECTO </p>
11 <br>
12 <form action="" method="POST">
13 <br>
14 <input type="submit" name="mcu_do" value="ON LED">
15 <input type="submit" name="mcu_do" value="OFF LED">
16 </body>
17 </html>
18
```

Figura 4.6.1-3. Script en HTML para la creación de una página web sencilla con dos botones. [88]

De esta manera nuestro dispositivo se conectara como cliente del servidor principal por medio del navegador en este caso se utiliza Google Chrome. El resultado del código de la Figura 4.6.1-3 es el que se obtienen en la Figura 4.6.1-4.

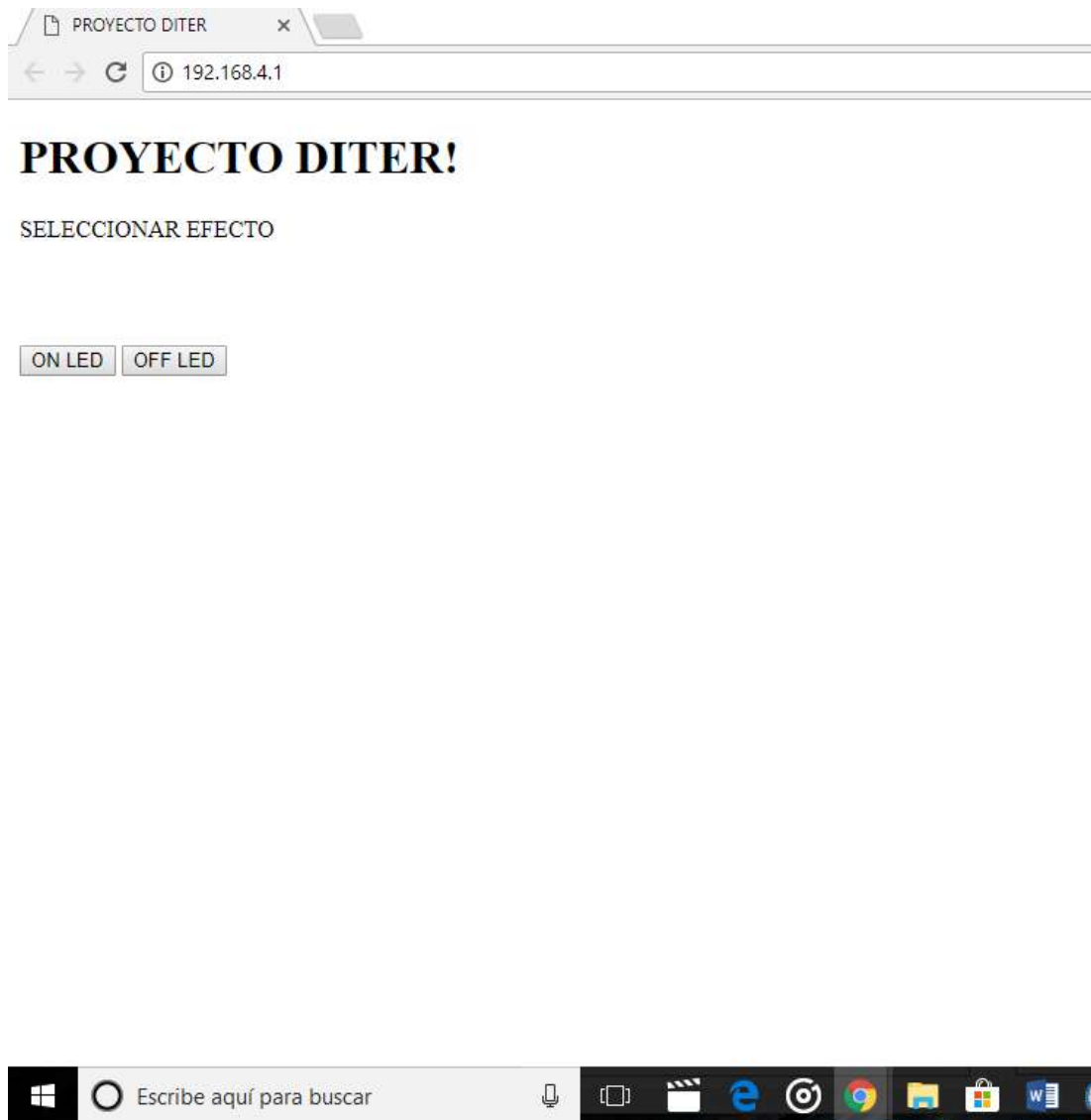


Figura 4.6.1-4 Página Web para controlar el encendido y apagado de un led. [89]

4.6.2. Conexión entre dos módulos NodeMCU.

El protocolo de transmisión de datos TCP/IP se encarga de la comunicación en todo el sistema. De esta forma el servidor principal es responsable de comunicarse con los demás servidores los cuales son los encargados de realizar el proceso de recibir información y transformarla en los efectos de iluminación o movimiento requeridos por el usuario.

La forma en que se comunican dos ESP8266 es la siguiente:

Ambos dispositivos deben estar conectados a la misma red Wi-Fi. El módulo programado como servidor principal crea una red Wi-Fi. Los servidores que son los periféricos del sistema deben conectarse a esta red. El servidor principal funciona como punto de acceso y los demás servidores estarán en modo estación.

Como mencionamos todos dispositivos serán servidores TCP cada uno escucha en su propio canal y una IP específica para cada dispositivo y de esta manera identificaremos a cada dispositivo en la red.

En la página web de Documentación NodeMCU en el módulo net nos explica cómo crear sockets TCP, recibir o enviar datos entre otras funciones.

En la Figura 4.6.2-1 el servidor principal se comunica con los demás servidores, al momento de establecer una conexión el servidor principal crea un socket que trabaja como cliente, en este caso el nuevo socket se nombró como conn2 una vez creado el socket se conecta con el servidor por medio de su puerto de escucha y su IP seguido de eso se envía la información que desea y se cierra la conexión.

```
conn2 = net.createConnection(net.TCP, 0)
conn2:on("receive", function(conn2, payload)
    print(payload)
    conn2:close()
end)
conn2:connect(110, "192.168.4.6") -- server/AP IP
conn2:send(mcu_do)
```

Figura 4.6.2-1 Script en Lua, para crear un cliente y conectarse a un servidor. [90]

5. Capítulo 5. Pruebas y Resultados.

El diseño está basado para realizar control de varios dispositivos los cuales se encuentren distribuidos en un área local, conectados entre sí por una red Wi-Fi ver Figura 5-1.

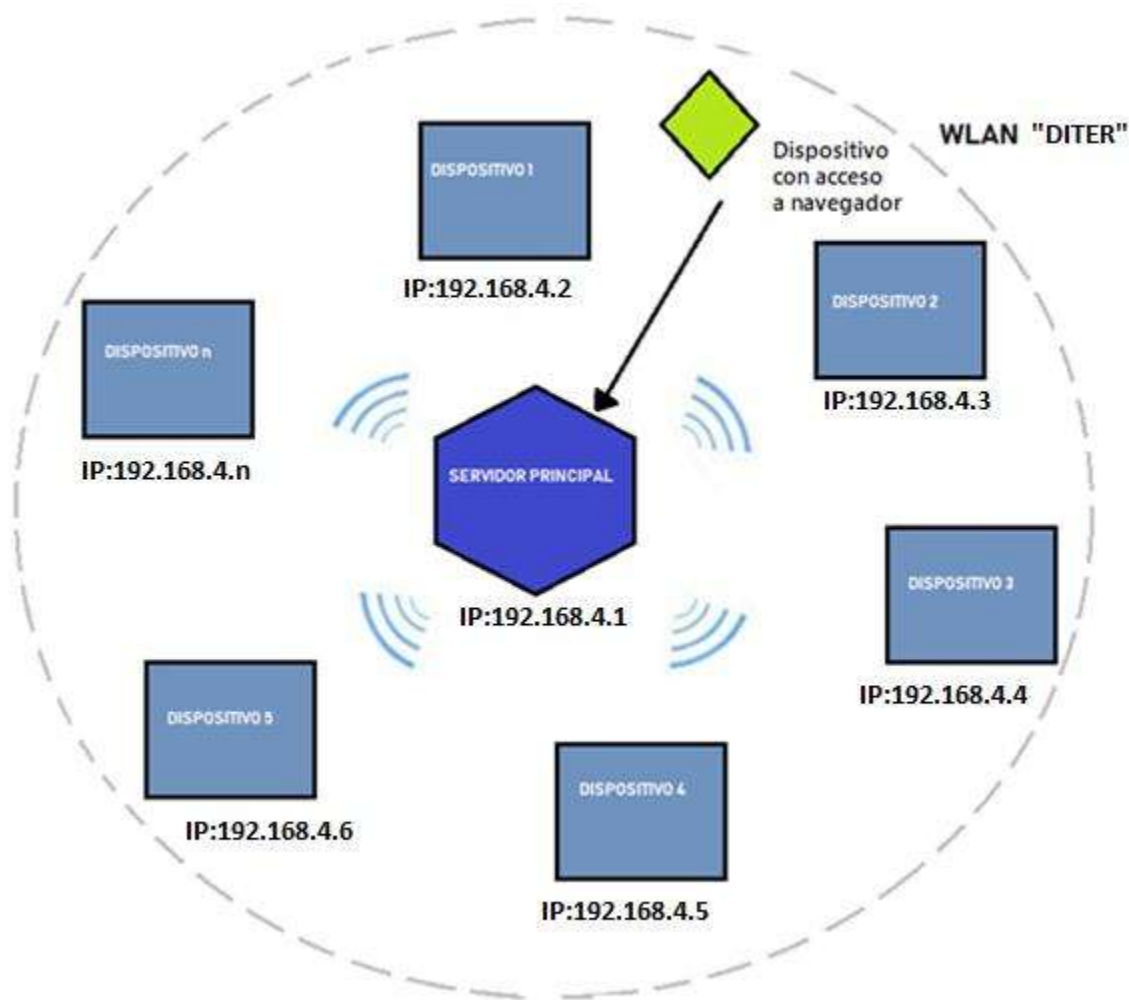


Figura 5-1 Diagrama funcional del sistema de iluminación. [91]

En la Figura 5-1. Se muestra el sistema completo y su función, en el cual todos los dispositivos se encuentran conectados a la WLAN “DITER” creada por el servidor principal, desde cualquier dispositivo con Wi-Fi nos conectamos a la red y por medio de un navegador se accede como cliente al servidor principal a través de su dirección IP. Los dispositivos 1, 2, 3,..., n. son los módulos NodeMCU que realizan el control de los actuadores, también cuentan con su IP específica así permitiendo la comunicación con el dispositivo y actuador deseado, estos módulos también funcionan como servidores, al realizar una petición al servidor principal desde el navegador este determina a que servidor en específico realiza la transferencia de información.

5.1. Pruebas del prototipo funcional

En capítulo cuatro se explica como crear una página web en el modulo NodeMCU programado como servidor principal. En la página aparecen dos botones y al presionar alguno de estos botones se pueden observar por medio del IDE la información que es enviada desde la página web al servidor principal. Este fue el procedimiento para las pruebas y observar primero el funcionamiento corecto de los modulos Wi-Fi y de la red WLAN antes de conectar cualquier actuador.

Lua Uploader tiene dos secciones en la Figura 5.1-1 se puede apreciar dos recuadros, el recuadro de la izquierda permite cargar el programa al módulo NodeMCU y el recuadro derecho es para la interacción con el módulo, donde se observan las respuestas a estas pruebas.

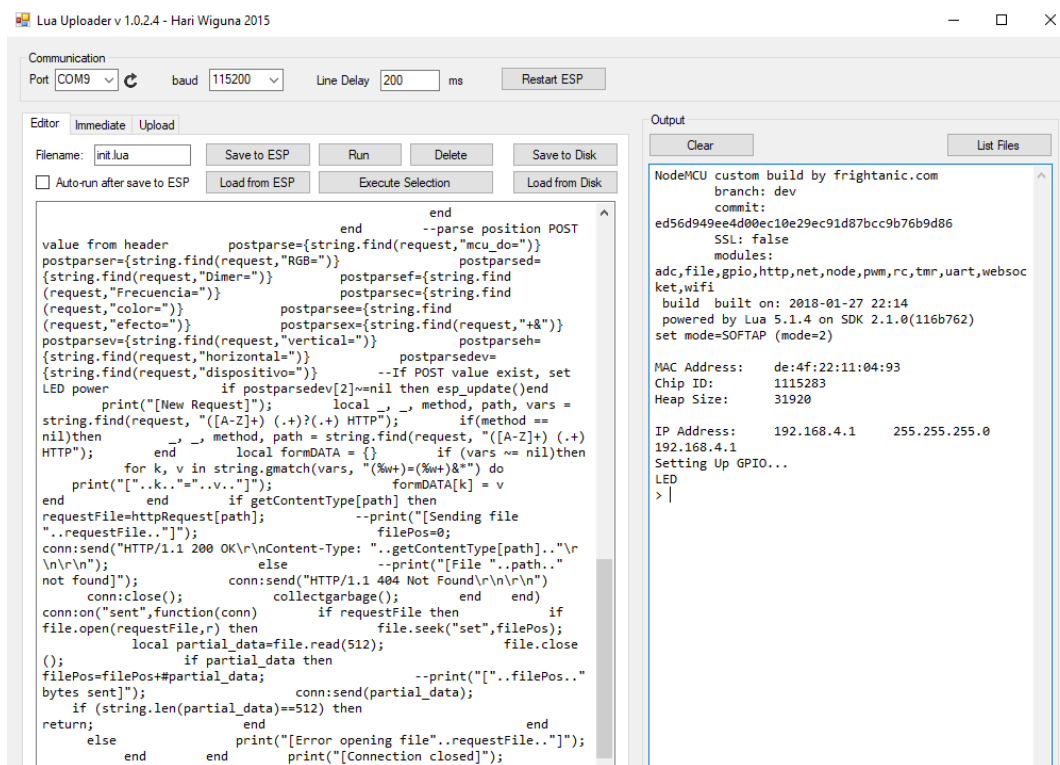


Figura 5.1-1 Lua Uploader, cargando programa al servidor principal. [92]

Al programar el servidor principal, el programa permite monitorear ciertos parametros que indiquen que el módulo ha iniciado correctamente ver Figura 5.1-2.

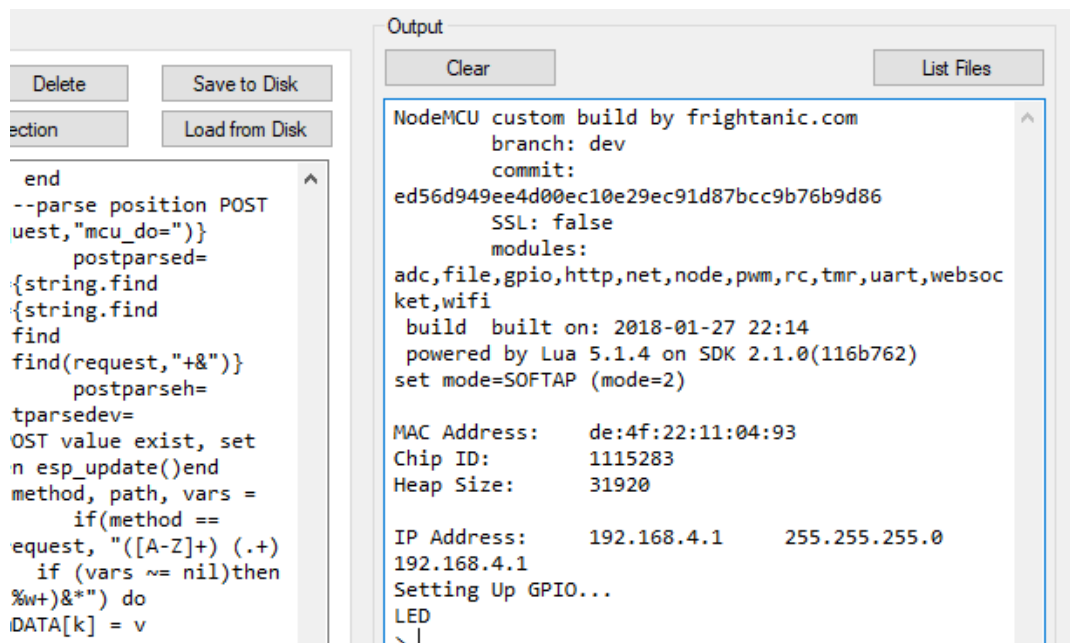


Figura 5.1-2 Iniciando Servidor principal. [92]

En los parametros de interes se observa la direccion MAC y dirección IP, se debe tomar nota de la direccion IP que permite el acceso por medio de la página web, también se observa el inicio de los perifericos GPIOs del modulo en este caso solo se inicializa un led para indicar si el servidor principal actua al momento de recibir peticiones desde la página web y para la transmision de información a otros dispositivos.

El principal objetivo del sistema es controlar cualquier dispositivo conectado a la WLAN, esto se realiza mediante botones tipo radio que permite la acción de selección para un grupo en el que no más de una opción puede ser seleccionada al mismo tiempo de acuerdo a la Figura 5.1-3.

Selección de dispositivo:

Control de Iluminacion!

☒ Dispositivo1 ☐ Dispositivo2 ☐ Dispositivo3

Figura 5.1-3 Pagina WEB, selección de dispositivo. [92]

En la Figura 5.1-4 se muestran las peticiones junto con el dispositivo seleccionado y en seguida se cierra la conexión entre servidor principal y dispositivo seleccionado, esto para permitir una mayor cantidad de conexiones futuras.

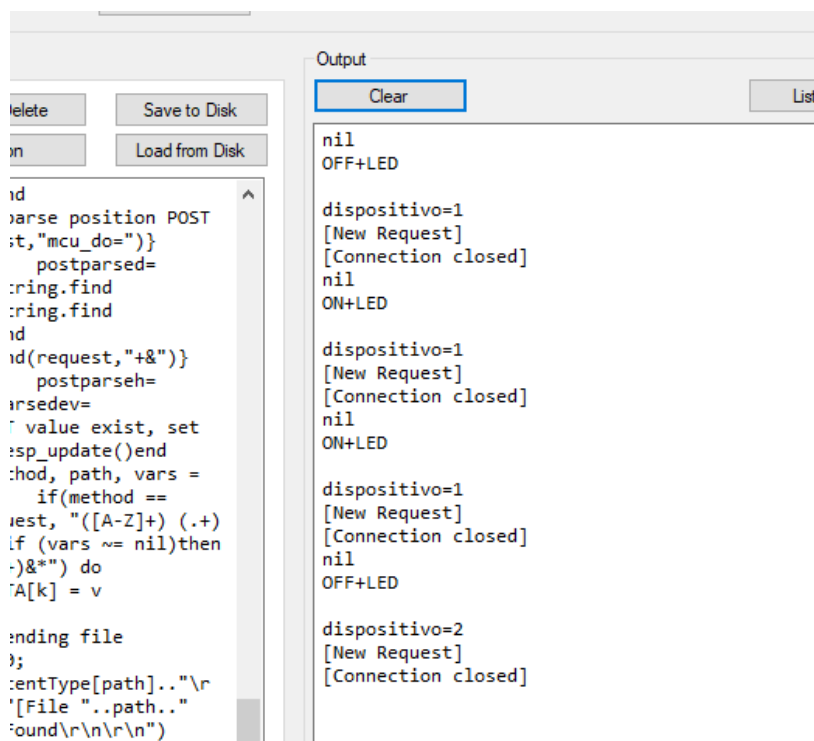


Figura 5.1-4 Respuesta por parte del servidor principal. [92]

Al seleccionar el dispositivo que se desea controlar, se envía junto con la petición que se desea realizar a dicho dispositivo.

Manipulación de deslizadores:

En la Figura 5.1-5 se observa la selección de dispositivo, tres deslizadores y un botón de envío. Primero se selecciona el dispositivo, se manipulan los deslizadores y se envía la información.

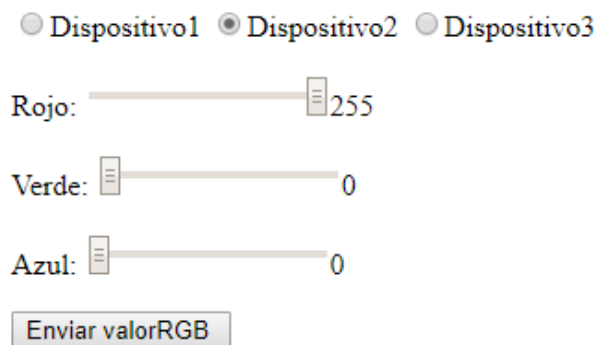


Figura 5.1-5 Prueba de manipulación de deslizadores. [92]

En la Figura 5.1-6 la respuesta obtenida por el servidor principal indica la selección del dispositivo dos y la información recibida es: Un valor de 255 para el primer deslizador y cero para los deslizadores siguientes tal como se ve en la Figura 5.1-5.

<pre> jest, "&")}] tparseh= /= = exist, set date()end >path, vars = method == '([A-Z]+) (.+) ~s ~= nil)then do = v </pre>	<pre> dispositivo=1 [New Request] [Connection closed] nil ON+LED </pre>
<pre> file >e[path].."\"r "..path.." ~\n\"r\n") end) if ',filePos); file.close ["..filePos.." </pre>	<pre> dispositivo=1 [New Request] [Connection closed] nil OFF+LED </pre>
<pre> file >e[path].."\"r "..path.." ~\n\"r\n") end) if ',filePos); file.close ["..filePos.." </pre>	<pre> dispositivo=2 [New Request] [Connection closed] nil OFF+LED </pre>
<pre> file >e[path].."\"r "..path.." ~\n\"r\n") end) if ',filePos); file.close ["..filePos.." </pre>	<pre> dispositivo=2 [New Request] [Connection closed] nil 255&G=0&B=0 </pre>

Figura 5.1-6 Respuesta a la prueba de los deslizadores. [92]

Estas pruebas solo han sido para el funcionamiento del servidor principal, ahora las pruebas se enfocaran a la comunicación entre servidor principal y otro dispositivo que aparezca en la página web para ser controlado como se realizo en la Figura 5.1-4, pero esta vez para comprobar que se recibe dicha petición en el dispositivo seleccionado.

Para entender mas sobre la conexión entre dispositivos se recuerda que se asigno la dirección IP 192.168.4.1. para el servidor principal, de forma manual asignó la dirección IP 192.168.4.2. al dispositivo 1 y la dirección IP 192.168.4.3. al dispositivo 3 como en la Figura 5-1.

Ahora conectamos el dispositivo 2 a Lua Uploader y aplicamos un reset.

En la Figura 5.1-7. Se aprecia que este módulo se conecta en modo estación, se activan los GPIOs del módulo en los cuales están conectados los actuadores y en seguida observamos la dirección IP que corresponde al dispositivo 2.

Ahora conectamos el dispositivo 2 a Lua Uploader y aplicamos un reset.

En la Figura 5.1-7. Se aprecia que este módulo se conecta en modo estación, se activan los GPIOs del módulo en los cuales están conectados los actuadores y en seguida observamos la dirección IP que corresponde al dispositivo 2.

P

Run

Delete

Save to Disk

SP

Execute Selection

Load from Disk

```

end
--parse position POST
se={string.find(request,"mcu_do=")}
"RGB="}) postparsed=
postparsef={string.find
postparsec={string.find
postparsee={string.find
postparsex={string.find(request,"+&")}
"vertical="}) postparseh=
="}) postparsedev=
="}) --If POST value exist, set
arsedev[2]~nil then esp_update()end
local _,_, method, path, vars =
.+)?(.) HTTP"; if(method ==
sth = string.find(request,"([A-Z]+) (.+)
formData = {} if (vars ~= nil)then
stch(vars, "(%w+)=(%w+)&") do
formData[k] = v
contentType[path] then
--print("[Sending file
filePos=0;
tent-Type: "..getContentType[path].."r
--print("[File "..path.."
nd("HTTP/1.1 404 Not Found\r\n\r\n")
collectgarbage(); end end)
if requestfile then if
file.seek("set",filePos);
file.close()

```

Output

Clear

List Files

```

SENT: node.restart()
{1
NodeMCU custom build by frightanic.com
branch: dev
commit:
c8d2937fd55b1234ae8af7a84ae41560acebcda
SSL: false
modules:
adc,file,gpio,http,net,node,pwm,rc,sjson,tmr,uart,
websocket,wifi,ws2812,ws2812_effects
build built on: 2018-02-28 17:40
powered by Lua 5.1.4 on SDK 2.1.0(116b762)

DITER

set mode=STATION (mode=1)

MAC Address: dc:4f:22:11:39:0c
Chip ID: 1128716
Heap Size: 41288

Setting Up GPIO...
LED
Setting Up GPIO pin 4,5 y rainbow...
> IP Info:
IP Address: 192.168.4.3
Netmask: 255.255.255.0
Gateway Addr: 192.168.4.1

```

Figura 5.1-7 Inicialización de dispositivo 2. [92]

Se realizan pruebas similares a las que fueron realizadas para el servidor principal, pero solo cuando se selecciona el botón de radio para el dispositivo 2, la información es recibida de la manera que se muestra en la Figura 5.1-8, primero se envía la manipulación de los rangos rgb donde rojo es el único que tiene su máximo valor, en seguida se alternaron valores para los tres deslizadores y por último se envió la petición ON LED.

```

:h].."r
ith.."
,n")
id)
if
:Pos);
.e.close

.lePos.."

end
.e.."l"):

```

```

Setting Up GPIO...
LED
Setting Up GPIO pin 4,5 y rainbow...
> IP Info:
IP Address: 192.168.4.3
Netmask: 255.255.255.0
Gateway Addr: 192.168.4.1

255&G=0&B=0
128&G=255&B=67
ON+LED

```

Figura 5.1-8 Petición recibida en el dispositivo 2 por parte del servidor principal. [92]

Viendo que el sistema de comunicación funciona de manera adecuada se continúa con la conexión de los actuadores, dependiendo de la información recibida en el módulo ESP8266 procesa la información y realiza el control del actuador correspondiente a la petición, habilitando o no los GPIOs activados.

5.2. Control general del sistema por medio de un dispositivo con Wi-Fi y navegador google chrome.

En este caso se utilizó una laptop ya que se aprecia mejor la página sin embargo la mayoría de las pruebas fueron realizadas desde un dispositivo celular.

La página web resultante para el control del sistema es la que se visualiza en la Figura 5.2-1 a la cual se accedió desde el navegador google Chrome.

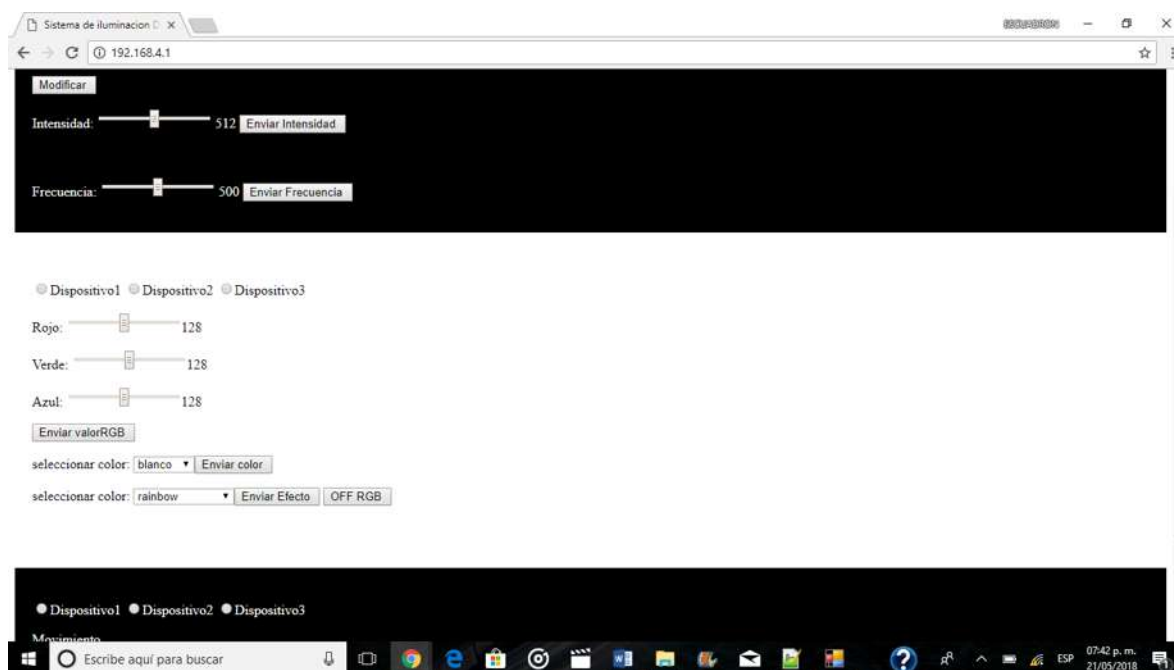


Figura 5.2-1 página web visto desde una pc con google Chrome. [93]

La página cuenta con: título, encabezado, estilos a través de css, indicador de posición de los deslizadores a través de javascript, todo esto para darle mejor estética y mejor interacción con el usuario, a continuación se explicara el funcionamiento de la página web.

Acceso a la página web:

En la Figura 5-1 existe un servidor principal el cual crea la red Wi-Fi y se encarga de comunicarse con todos los dispositivos y es en ese servidor principal donde se encuentra la página web. Accediendo a esta a través de su dirección IP como se aprecia en la Figura 5.2-2

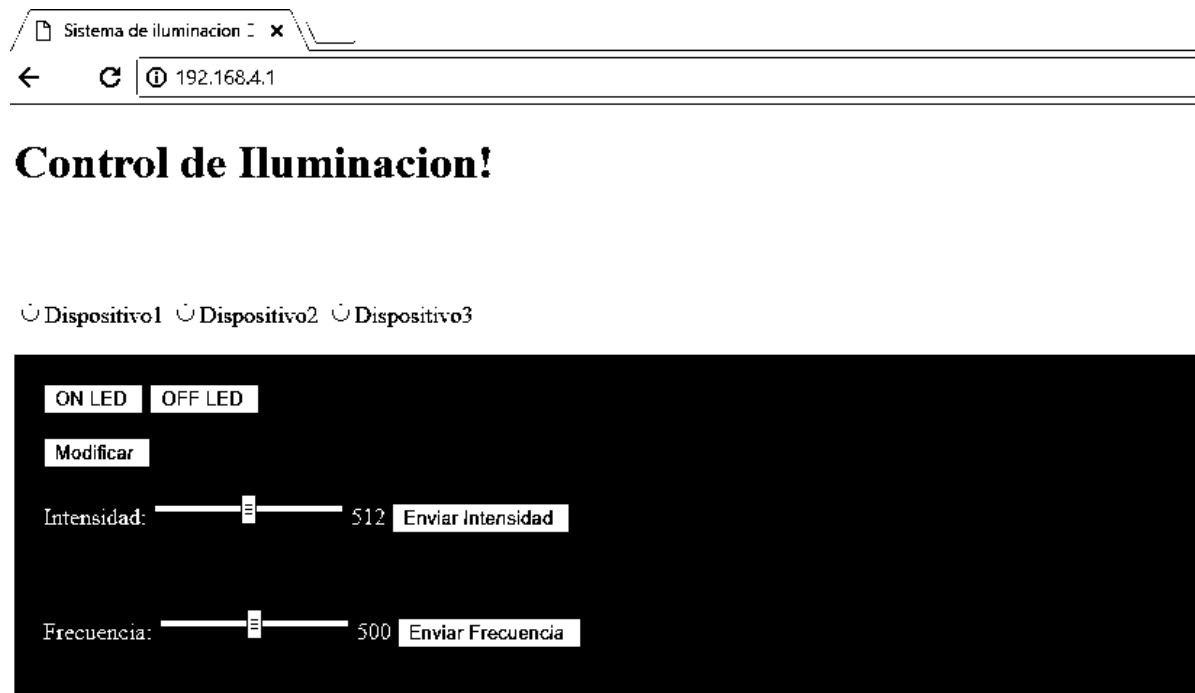


Figura 5.2-2 Control de iluminación LED. [93]

En la Figura 5.2-2 se observa la IP del servidor principal en la parte donde se asigna la URL para acceder a la página, también se aprecia el título y el encabezado.

Función de la página: Lo primero que se debe hacer antes de realizar una modificación es seleccionar el dispositivo que se quiere controlar. Solo se han programado 3 dispositivos para fines del prototipo.

La página Web está dividida en tres secciones:

- Control LED.
- Control Rainbow FC-102.
- Control de Movimiento.

Control del LED:

Existen 2 botones los cuales permiten el encendido y apagado de un LED; El botón Modificar permite cambiar el modo al módulo PWM visto anteriormente en la sección 4.5.1 al modificar la intensidad o frecuencia primero debe ser presionado el botón modificar, luego los deslizadores permiten seleccionar un rango. Una vez modificado el rango se envía el dato al presionar el botón de enviar intensidad o enviar frecuencia dependiendo que parámetro es el que se desee modificar.

Control Rainbow FC-102 RGB:

Lo primero es seleccionar el dispositivo que deseamos controlar, luego existen deslizadores para cada color (rojo, verde, azul). Cada uno tiene un rango de 0 a 255, al combinar los rangos obtendrá un color formado por la combinación de los tres valores para darnos una mejor idea podemos ver la Figura 4.5.3-3 que explica cómo formar colores con la combinación de estos valores. Después de seleccionar los rangos se envía los valores de los tres deslizadores presionando el botón Enviar valorRGB de acuerdo a la Figura 5.2-3.

● Dispositivo1 ● Dispositivo2 ● Dispositivo3

Rojo:

Verde:

Azul:

seleccionar color:

seleccionar color:

Figura 5.2-3 Control Rainbow FC-102 RGB. [93]

Hay otra opción la cual ya tiene colores predeterminados como observamos en la Figura 5.2-4. Una vez seleccionado el color enviamos el dato con el botón Enviar color.

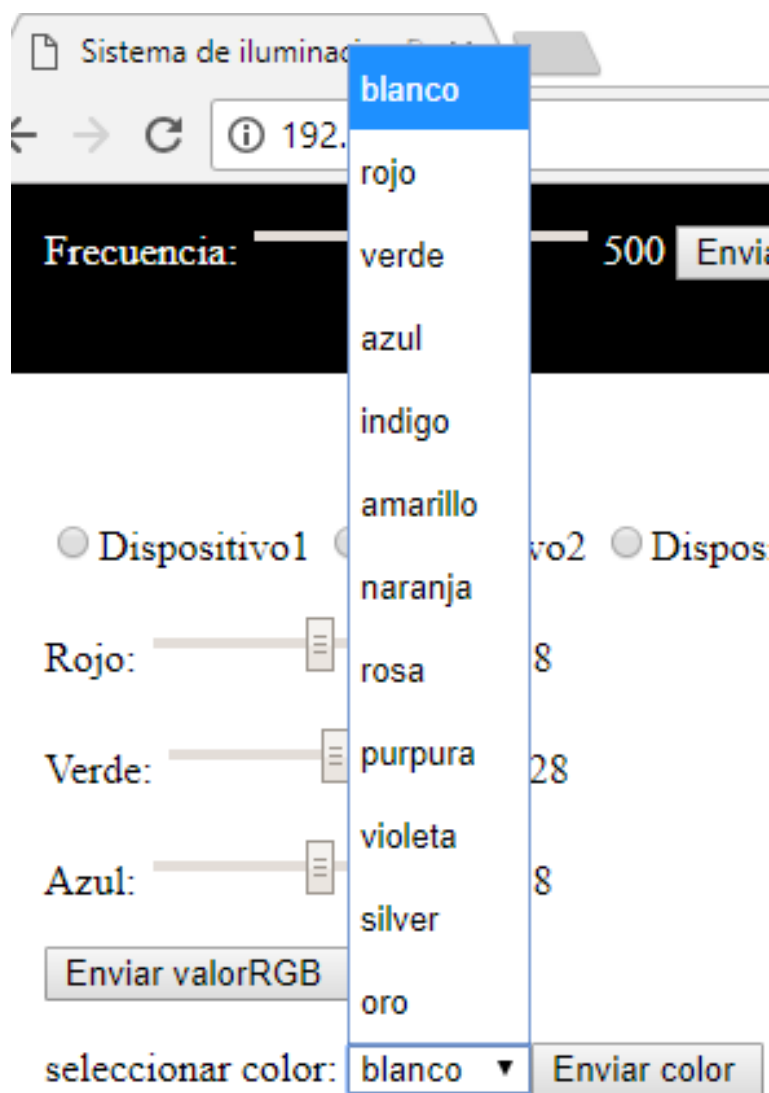


Figura 5.2-4 Lista de Selección de color. [93]

Ademas de presentar colores solidos en el módulo, WS2812 permite realizar efectos de iluminación se incluye una lista de algunos efectos realizados por WS2812 para aprovechar las características del control de iluminación. Una vez seleccionado el efecto, se envia la información presionando el botón Eviar Efecto ver Figura 5.2-5. Por ultimo está el botón OFF RGB que permite apagar el dispositivo.

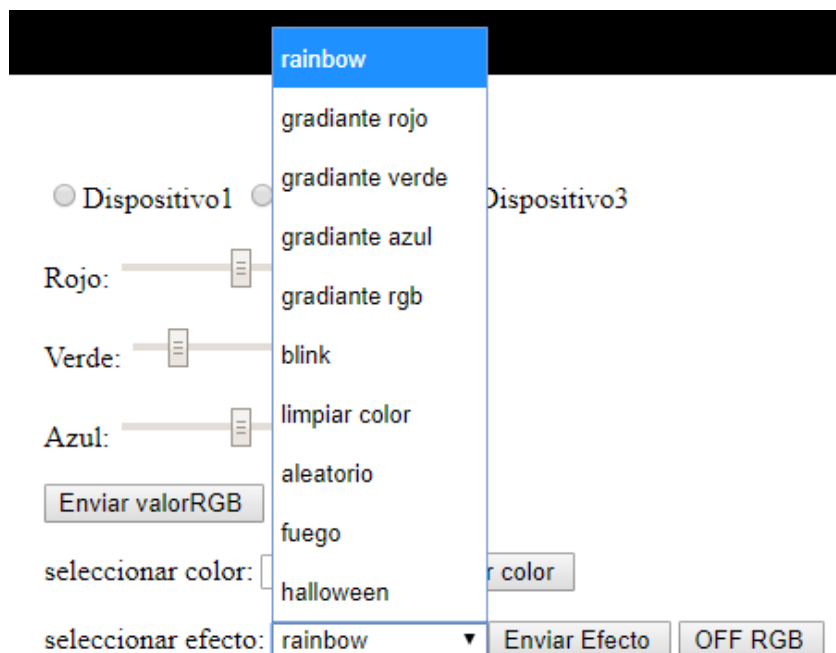


Figura 5.2-5 Lista de selección de efectos. [93]

Control de Movimiento:

Al igual que en las secciones anteriores, lo principal es seleccionar que dispositivo se quiere controlar. Sólo existen dos deslizzadores para este control, la posición vertical y la posición horizontal. Ambos funcionan de la misma manera la importancia aquí es la orientación en la que se coloquen los servomotores. Los deslizzadores solo permiten moverse en tres posiciones izquierda, centro y derecha. Esto se respeta para el movimiento horizontal, pero para el movimiento vertical, izquierda se refiere abajo, centro sigue siendo el centro y derecha será arriba esto ya dependerá también como se oriente este servomotor. Ya posicionando el deslizzador enviamos la el dato presionando el botón, dependiendo si se quiere modificar la posición vertical u horizontal ver Figura 5.2-6.



Figura 5.2-6 Control de Movimiento. [93]

5.3. Observaciones y conclusiones

Una de las observaciones de este sistema es sobre un problema con el navegador Safari, este navegador es más común en los dispositivos Apple la página si es visible el problema surge al enviar algún dato, al presionar cualquier botón la página se queda colgada, son mejoras que deben hacerse para que el sistema pueda ser utilizado en cualquier dispositivo y cualquier navegador.

El control de LED es el que permite control para focos o lámparas de AC utilizados comúnmente en el hogar esto se logra conectado un circuito de potencia entre ambos dispositivos y se obtendrán los mismos resultados.

Cuando accedes a una red Wi-Fi, el punto de acceso te asigna una IP aleatoria por el tiempo que estés conectado a esa red, esto no sirve para este sistema por lo que cada módulo NodeMCU conectado a la red debe tener una IP específica para solucionar este inconveniente se asignó una IP específica para el dispositivo que es usado de interfaz y así evitar confusión al momento de trabajar con el sistema.

Esto es un punto de mejora ya que si el usuario final quiere agregar más dispositivos a su red, tendrá que acudir con el diseñador del sistema o entender cómo funciona el sistema desde la programación para agregar un dispositivo nuevo a la red manualmente.

A pesar de lo anterior los resultados son satisfactorios para un sistema de iluminación en un área local de conexión Wi-Fi y permitiendo realizar el control de cualquier actuador desde cualquier punto del área local por medio de un dispositivo en este caso enfocado más a realizarlo desde un Smartphone el cual es un dispositivo comúnmente utilizado por la mayoría de seres humanos sin embargo existen varios dispositivos con las mismas características necesarias para ser usados como interfaz.

6. Referencias

- [1] E. HARPER, «BOOKS,» LIMUSA, 2000. [En línea]. Available: <https://books.google.com.mx/books?id=3MJ7B0q6Ehkc&printsec=frontcover&dq=isbn:9681860500&hl=es-419&sa=X&ved=0ahUKEwiW1uza07PaAhUF9YMKHfwcACMQ6AEIKDAA#v=onepage&q&f=false>. [Último acceso: Enero 2018].
- [2] J. A. G. Alejos, «issuu,» 26 noviembre 2015. [En línea]. Available: <https://issuu.com/joseangelgranda/docs/luminotecnia.pptx>. [Último acceso: enero 2018].
- [3] M. Lara, «LosNueveLibros.com Artículos de Historia y mitología,» prehistoria, 6 junio 2015. [En línea]. Available: <http://losnuevelibros.com/descubrimiento-del-fuego-en-la-prehistoria/>. [Último acceso: Enero 2018].
- [4] Blogger, «Blogspot,» 1 diciembre 2012. [En línea]. Available: <http://atenea1b.blogspot.mx/2012/12/homo-erctus-el-descubridor-del-fuego.html>. [Último acceso: diciembre 2017].
- [5] teknikailuminacion, «teknicalightingconsulting,» 6 Marzo 2013. [En línea]. Available: <https://teknikailuminacion.wordpress.com/2013/03/06/del-mito-de-la-antorcha-al-candil-y-la-vela/>. [Último acceso: Enero 2018].
- [6] avancesegipto, «wordpress,» mayo 2016. [En línea]. Available: <https://avancesegipto.wordpress.com/lampara-de-aceite/>. [Último acceso: diciembre 2017].
- [7] h. samarema, «ASHE,» 6 enero 2011. [En línea]. Available: <http://www.ashe.com.ve/foro/viewtopic.php?t=20270>. [Último acceso: diciembre 2017].
- [8] «FRAGA ILUMINACIÓN,» [En línea]. Available: <https://www.fragailuminacion.com.ar/publicaciones/historia-de-la-iluminacion/>. [Último acceso: enero 2017].
- [9] «significado de velas,» [En línea]. Available: <https://significadodelasvelas.com/significado-de-las-velas-cuando-se-consumen/>. [Último acceso: diciembre 2017].
- [10] J. M. SANCHIS, «museosm,» [En línea]. Available: <http://www.museosm.com/exposiciones/lamparastempestad.pdf>. [Último acceso: diciembre 2017].
- [11] teknikailuminación, «teknicalighting,» 16 abril 2013. [En línea]. Available: <https://teknikailuminacion.wordpress.com/2013/04/16/de-la-candela-a-la-lampara-de-gas/>. [Último acceso: diciembre 2017].
- [12] «todocoleccion,» 4 marzo 2012. [En línea]. Available: <https://www.todocoleccion.net/antiguedades/antigua-lampara-gas-modernista-ano-1900~x34552994>. [Último acceso: diciembre 2017].

- [13] A. J. Leon, «aiu.edu,» ATLANTIC INTERNATIONAL UNIVERSITY, HAWAII, invierno 2007. [En línea]. Available: <https://www.aiu.edu/applications/DocumentLibraryManager/upload/Lighting.pdf>. [Último acceso: diciembre 2017].
- [14] stockvault, «freepik,» 2013. [En línea]. Available: https://www.freepik.es/foto-gratis/lampara-incandescente_602575.htm. [Último acceso: diciembre 2017].
- [15] «wordpress,» 25 mayo 2008. [En línea]. Available: <https://energias.wordpress.com/>. [Último acceso: diciembre 2017].
- [16] «eic.cat,» OSRAM, [En línea]. Available: <http://www.eic.cat/gfe/docs/2781.pdf>. [Último acceso: diciembre 2017].
- [17] «uv.es,» Universidad de València, [En línea]. Available: <https://www.uv.es/~hertz/hertz/Docencia/teoria/Historia.pdf>. [Último acceso: diciembre 2017].
- [18] «tecnologiadecomunicacion,» [En línea]. Available: <https://sites.google.com/site/latecnologiadecomunicacion/comunicacion-analambrica-e-inalambrica>. [Último acceso: diciembre 2017].
- [19] «La Web de Anilandro,» [En línea]. Available: <https://sites.google.com/site/anilandro/00400-c3-chispas-cohesores?tmpl=%2Fsystem%2Fapp%2Ftemplates%2Fprint%2F&showPrintDialog=1>. [Último acceso: diciembre 2017].
- [20] «wanadoo.es,» 4 marzo 2006. [En línea]. Available: <http://perso.wanadoo.es/ecatob/radioelectricidad.htm>. [Último acceso: diciembre 2017].
- [21] «EA8EX,» [En línea]. Available: <http://www.ea8ex.com/evolucion-de-la-radio.html>. [Último acceso: diciembre 2017].
- [22] M. d. científico, «Manual del científico,» [En línea]. Available: <http://manualdelcientifico.blogspot.mx/2015/04/25-de-abril-guillermo-marconi-y-su.html>. [Último acceso: diciembre 2017].
- [23] «casaradiotv,» [En línea]. Available: <http://www.casaradiotv.com/es/ducretet.htm>. [Último acceso: abril 2018].
- [24] J. Garreta, «pleasenetworks,» 13 enero 2017. [En línea]. Available: <http://pleasenetworks.com/blog/post/13/historia-y-evolucion-de-las-redes-wifi-de-tecnologia-inalambrica-a-aplicacion-ligera-para-retailers>. [Último acceso: diciembre 2017].
- [25] J. P. Blázquez, «exabyteinformatica,» Universidad Oberta de Catalunya, [En línea]. Available: [https://www.exabyteinformatica.com/uoc/Informatica/Tecnologia_y_desarrollo_en_dispositivos_moviles/Tecnologia_y_desarrollo_en_dispositivos_moviles_\(Modulo_1\).pdf](https://www.exabyteinformatica.com/uoc/Informatica/Tecnologia_y_desarrollo_en_dispositivos_moviles/Tecnologia_y_desarrollo_en_dispositivos_moviles_(Modulo_1).pdf). [Último acceso: diciembre 2017].
- [26] «LINIO,» [En línea]. Available: <https://www.linio.com.mx/p/cabeza-robotica-para-iluminacion-profesional-rgb-modos-audio-tematica-y-secuencial-lccxwn>. [Último acceso: diciembre 2017].
- [27] D. Canchola, *Dispositivos de iluminacion*, Morelia , 2018.
- [28] e. LED, «efecto LED,» 12 mayo 2017. [En línea]. Available: <http://www.efectoled.com/blog/sistemas-de-control-de-la-iluminacion/>. [Último acceso: enero 2018].

- [29] V. Rodriguez, «SliderPlayer,» [En línea]. Available: <http://slideplayer.es/slide/13954249/>. [Último acceso: enero 2018].
- [30] J. J. Y. Torres, «bibing,» [En línea]. Available: <http://bibing.us.es/proyectos/abreproy/11138/fichero/memoria%252FCap%C3%ADulo+3.pdf>. [Último acceso: enero 2018].
- [31] The Adbus Salam International Centre for Theoretical Physics, «eslared.org,» [En línea]. Available: http://www.eslared.org.ve/walc2012/material/track1/05-Introduccion_a_las_redes_WiFi-es-v2.3-notes.pdf. [Último acceso: enero 2018].
- [32] «Wordpress,» [En línea]. Available: <https://cdsmante5.wordpress.com/especificacion-ieee-802-11/>. [Último acceso: enero 2018].
- [33] «hivelocity,» Hivelocity, [En línea]. Available: <https://www.hivelocity.co.jp/blog/23006/>. [Último acceso: abril 2018].
- [34] «PROGRAMMING,» 14 noviembre 2017. [En línea]. Available: <http://programming.sstne.com/2017/11/14/microcontroller/#respond>. [Último acceso: enero 2018].
- [35] mikroelektronika, «mikroelektronika,» [En línea]. Available: <https://www.mikroe.com/rs/ebooks/microcontroladores-pic-programacion-en-basic/introduccion-al-mundo-de-los-microcontroladores>. [Último acceso: enero 2018].
- [36] C. Ubach Granados, «upcommons,» abril 2017. [En línea]. Available: https://upcommons.upc.edu/bitstream/handle/2117/105042/Memoria_TFG_Carles_Ubach.pdf?sequence=1&isAllowed=y. [Último acceso: febrero 2018].
- [37] espressif, «espressif.com,» [En línea]. Available: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf. [Último acceso: febrero 2018].
- [38] A. Carrasco, 24 noviembre 2013. [En línea]. Available: <http://gpsk15.blogspot.mx/>. [Último acceso: febrero 2018].
- [39] L. d. V. Hernandez, «programarfácil,» [En línea]. Available: <https://programarfácil.com/podcast/como-configurar-esp01-wifi-esp8266/>. [Último acceso: febrero 2018].
- [40] «ELECTRONILAB,» [En línea]. Available: <https://electronilab.co/tienda/esp8266-esp-02-modulo-transceptor-wifi-serial/>. [Último acceso: febrero 2018].
- [41] «PotentialLabs,» [En línea]. Available: <https://potentiallabs.com/cart/buy-esp-03-online-india>. [Último acceso: febrero 2018].
- [42] smarplotcom, «smarpl,» 03 20 2015. [En línea]. Available: <http://smarpl.com/content/esp8266-esp-04-tips-and-tricks>. [Último acceso: febrero 2018].
- [43] «lelong,» [En línea]. Available: <https://www.lelong.com.my/esp-05-wifi-module-enewground-190149755-2019-03-Sale-P.htm>. [Último acceso: febrero 2018].
- [44] Bobby, «hackster,» 2016. [En línea]. Available: <https://www.hackster.io/iboboc/esp-07-or-esp-12-with-nodemcu-going-into-deep-sleep-a0560e>. [Último acceso: febrero 2018].

- [45] Naguissa, «foroelectro,» 17 noviembre 2016. [En línea]. Available: <https://www.foroelectro.net/electronica-digital-microcontroladores-f8/pinout-y-conexion-normal-y-modo-flash-para-modelos-t130.html>. [Último acceso: febrero 2018].
- [46] Naguissa, «foroelectro,» 30 noviembre 2016. [En línea]. Available: <https://www.foroelectro.net/electronica-digital-microcontroladores-f8/pinout-y-conexion-normal-y-modo-flash-para-modelos-t130-s10.html#p280>. [Último acceso: febrero 2018].
- [47] «wa0uwh.blogspot,» 28 agosto 2015. [En línea]. Available: <http://wa0uwh.blogspot.mx/2015/08/esp8266-esp-13-to-dip-adaptor.html>. [Último acceso: febrero 2018].
- [48] «nettigo,» [En línea]. Available: <https://nettigo.eu/products/esp-8266-12-wifi-module-with-9-gpio--2>. [Último acceso: febrero 2018].
- [49] J. A. García Álvarez, «asifunciona,» [En línea]. Available: http://www.asifunciona.com/fisica/ke_led/ke_led_2.htm. [Último acceso: marzo 2018].
- [50] «hamilton.mforos,» 04 febrero 2017. [En línea]. Available: <http://hamilton.mforos.com/2101807/12871749-electronica/>. [Último acceso: marzo 2018].
- [51] A. G. González, «panamahitek,» 2 diciembre 2016. [En línea]. Available: <http://panamahitek.com/ques-y-como-funciona-un-servomotor/>. [Último acceso: marzo 2018].
- [52] «areatecnologia,» [En línea]. Available: <http://www.areatecnologia.com/electricidad/servomotor.html>. [Último acceso: marzo 2018].
- [53] «INGENIERIA ELECTRICA,» 15 diciembre 2009. [En línea]. Available: <http://ing-electrica-indu7.blogspot.mx/>. [Último acceso: marzo 2018].
- [54] «deskontrol,» [En línea]. Available: <https://www.deskontrol.net/es/leds/289-ws2812b-diodo-led-digital-inteligente.html>. [Último acceso: marzo 2018].
- [55] L. Llamas, «luisllamas.es,» 7 febrero 2016. [En línea]. Available: <https://www.luisllamas.es/arduino-led-rgb-ws2812b/>. [Último acceso: enero 2018].
- [56] «DRLEDRGB,» [En línea]. Available: <http://drledrgb.com/content/ws2812b-tira-led-pixel-digital-60-ledm-ip68>. [Último acceso: enero 2018].
- [57] «DRLEDRGB,» [En línea]. Available: <http://drledrgb.com/content/diferencias-entre-tiras-de-led-ws2812b-y-ws2811>. [Último acceso: enero 2018].
- [58] R. V. Sharma, «RED HAT DEVELOPER PROGRAM,» 17 octubre 2017. [En línea]. Available: <https://developers.redhat.com/blog/2017/10/17/what-is-a-socket/>. [Último acceso: enero 2018].
- [59] D. Marató, «tlm.unavarra,» julio 2004. [En línea]. Available: https://www.tlm.unavarra.es/~daniel/docencia/rc_itig/rc_itig04_05/slides/clase9y10-SocketsTCP.pdf. [Último acceso: enero 2018].
- [60] gabriel, «MDN web docs,» 20 noviembre 2017. [En línea]. Available: <https://developer.mozilla.org/es/docs/Web/HTML>. [Último acceso: enero 2018].

- [61] R. Alvarez, «desarrolloweb,» 29 septiembre 2016. [En línea]. Available: <https://desarrolloweb.com/articulos/534.php>. [Último acceso: enero 2018].
- [62] «tutorialesprogramacionya,» [En línea]. Available: <http://www.tutorialesprogramacionya.com/htmla/temarios/descripcion.php?inicio=0&cod=69&punto=2>. [Último acceso: enero 2018].
- [63] R. Alvarez, «desarrolloweb,» 29 septiembre 2016. [En línea]. Available: <https://desarrolloweb.com/articulos/535.php>. [Último acceso: enero 2018].
- [64] «tutorialesprogramacionya,» [En línea]. Available: <http://www.tutorialesprogramacionya.com/htmla/temarios/descripcion.php?cod=92&punto=10&inicio=>. [Último acceso: enero 2018].
- [65] L. d. V. Hernandez, «Programarfacil,» [En línea]. Available: <https://programarfacil.com/podcast/nodemcu-tutorial-paso-a-paso/>. [Último acceso: febrero 2018].
- [66] Carlos, «rogerbit,» 24 mayo 2017. [En línea]. Available: <http://rogerbit.com/wprb/2017/05/como-hacer-un-jammer-wifi-con-esp8266/>. [Último acceso: marzo 2018].
- [67] «AliExpress,» [En línea]. Available: <https://es.aliexpress.com/item/CH340G-NodeMcu-V3-Lua-Wireless-Internet-Of-Things-Development-Board-3-3V-Network-WIFI-Connector-Module/32796455118.html>. [Último acceso: Marzo 2018].
- [68] «electronilab,» [En línea]. Available: <https://electronilab.co/tienda/micro-servo-metalico-mg90s-towerpro/>. [Último acceso: marzo 2018].
- [69] «engineering,» [En línea]. Available: <https://engineering.tamu.edu/media/4247823/ds-servo-mg90s.pdf>. [Último acceso: Marzo 2018].
- [70] «wikipedia,» 12 diciembre 2017. [En línea]. Available: https://es.wikipedia.org/wiki/Servomotor_de_modelismo. [Último acceso: marzo 2018].
- [71] «lapantech,» [En línea]. Available: <http://lapantech.com/FC-102-Rainbow-RGB-LED-lighting-arduino-fotografi>. [Último acceso: marzo 2018].
- [72] D. Canchola, *Diagrama a bloques del prototipo de iluminación*, Morelia, 2018.
- [73] «nodemcu,» [En línea]. Available: <http://nodemcu.readthedocs.io/en/dev/en/build/>. [Último acceso: marzo 2018].
- [74] D. Canchola, *Firmware Programmer*, Morelia, 2018.
- [75] «nodmcu,» MkDocs, [En línea]. Available: <https://nodemcu-build.com/>. [Último acceso: Marzo 2018].
- [76] D. Canchola, *Lua Uploader*, Morelia, 2018.
- [77] «NodeMCU Documentation,» MkDocs, [En línea]. Available: <http://nodemcu.readthedocs.io/en/dev/en/modules/wifi/>. [Último acceso: Noviembre 2017].
- [78] D. Canchola, *Conexion a la red*, Morelia, 2017.

- [79] Carlos y Enrique, «roboticaaleixandre4a,» 03 junio 2012. [En línea]. Available: <http://roboticaaleixandre4a.blogspot.mx/2012/06/7-actuadores-que-son-y-tipos.html>. [Último acceso: abril 2018].
- [80] D. Canchola, *Diagrama de conexión de un LED*, Morelia, 2018.
- [81] D. Canchola, *Encendido y apagado de un LED*, Morelia, 2018.
- [82] D. Canchola, *Diagrama de conexión de un servomotor y NodeMCU*, Morelia, 2018.
- [83] D. Canchola, *Posicionamiento de un servomotor*, Morelia, 2018.
- [84] D. Canchola, *Diagrama de conexión para rainbow*, Morelia, 2018.
- [85] «NodeMCU Documentation,» MkDocs, [En línea]. Available: <http://nodemcu.readthedocs.io/en/dev/en/modules/ws2812-effects/>. [Último acceso: diciembre 2017].
- [86] Danny, «github,» 14 Noviembre 2015. [En línea]. Available: https://github.com/orionrobots/esp8266_video_series/blob/master/ws2812/ws2812_panel.lua. [Último acceso: diciembre 2017].
- [87] D. Canchola, *Rainbow encendido en color oro*, Morelia, 2018.
- [88] D. Canchola, *Script en HTML*, Morelia, 2018.
- [89] D. Canchola, *Navegador con botones*, Morelia, 2018.
- [90] D. Canchola, *Script para crear un cliente y conectarse con un servidor*, Morelia, 2018.
- [91] D. Canchola, *Sistema de Iluminación*, Morelia, 2018.
- [92] D. Canchola, *Pruebas de implementación*, Morelia, 2018.
- [93] D. Canchola, *Página web*, Morelia, 2018.
- [94] M. D. B. Tzancof, «citeseerx,» 6 diciembre 2011. [En línea]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.889.8790&rep=rep1&type=pdf>. [Último acceso: enero 2018].
- [95] M. Ubl y E. Kitamura, «html5rocks,» 20 octubre 2010. [En línea]. Available: <https://www.html5rocks.com/es/tutorials/websockets/basics/>. [Último acceso: enero 2018].
- [96] «nodemcu,» MkDocs, [En línea]. Available: <http://nodemcu.readthedocs.io/en/dev/en/modules/wifi/#wifiapgetconfig>. [Último acceso: Marzo 2018].
- [97] D. Canchola, *PWM en LED*, Morelia, 2018.