

UACM

Universidad Autónoma
de la Ciudad de México

Nada humano me es ajeno

Colegio de Ciencia y Tecnología

**"Diseño y construcción de un sistema IoT
enfocado al control de luminarias utilizando
módulos NODEMCU"**

TRABAJO RECEPCIONAL

QUE PARA OBTENER EL TÍTULO DE:
**LICENCIADO EN INGENIERÍA EN SISTEMAS ELECTRÓNICOS
Y DE TELECOMUNICACIONES**

Presenta:

Alfredo García de la Rosa

Director:

M. en C. Joel Yazbek Buendía Gómez

30 de junio de 2021

SISTEMA BIBLIOTECARIO DE INFORMACIÓN Y DOCUMENTACIÓN



UNIVERSIDAD AUTÓNOMA DE LA CIUDAD DE MÉXICO COORDINACIÓN ACADÉMICA

RESTRICCIONES DE USO PARA LAS TESIS DIGITALES

DERECHOS RESERVADOS[©]

La presente obra y cada uno de sus elementos está protegido por la Ley Federal del Derecho de Autor; por la Ley de la Universidad Autónoma de la Ciudad de México, así como lo dispuesto por el Estatuto General Orgánico de la Universidad Autónoma de la Ciudad de México; del mismo modo por lo establecido en el Acuerdo por el cual se aprueba la Norma mediante la que se Modifican, Adicionan y Derogan Diversas Disposiciones del Estatuto Orgánico de la Universidad de la Ciudad de México, aprobado por el Consejo de Gobierno el 29 de enero de 2002, con el objeto de definir las atribuciones de las diferentes unidades que forman la estructura de la Universidad Autónoma de la Ciudad de México como organismo público autónomo y lo establecido en el Reglamento de Titulación de la Universidad Autónoma de la Ciudad de México.

Por lo que el uso de su contenido, así como cada una de las partes que lo integran y que están bajo la tutela de la Ley Federal de Derecho de Autor, obliga a quien haga uso de la presente obra a considerar que solo lo realizará si es para fines educativos, académicos, de investigación o informativos y se compromete a citar esta fuente, así como a su autor ó autores. Por lo tanto, queda prohibida su reproducción total o parcial y cualquier uso diferente a los ya mencionados, los cuales serán reclamados por el titular de los derechos y sancionados conforme a la legislación aplicable.

Resumen

Actualmente vivimos en una era en constante cambio en donde las tecnologías de la Información y comunicaciones cobra cada vez mayor fuerza, dotando a las personas de una mejor comunicación y acceso a la información de manera más rápida segura y eficiente sin importar la región del mundo donde nos encontremos(Colmena, 2017).

No obstante, el Internet de las cosas es una tendencia en el que muchos países están adoptando con mayor frecuencia, logrando recabar datos por múltiples y pequeños sensores, para el análisis y procesamiento de información que están vinculadas a la automatización de los hogares o espacios de trabajo.

Debido a esta creciente área se plantea desarrollar un sistema domótico con un enfoque orientado al Internet de las Cosas, en inglés Internet of Things que permita monitorear un espacio doméstico confinado (sala, comedor, etc), por medio de diferentes sensores (temperatura, presencia e intensidad luminosa), con un par de módulos programados (NodeMCU) que pueden establecer una comunicación para controlar una luminaria de manera manual o automática y al mismo tiempo poder recopilar datos.

Por ende, lo que se obtiene de este trabajo es un prototipo que se puede instalar en nuestro hogar de manera fácil y rápida, específicamente en un socket de una luminaria convencional, con una bombilla ahorradora de tecnología LED.

Contenido

Lista de Figuras	IV
Lista de Tablas	I
1 Planteamiento del problema	3
1.1 Introducción	3
1.2 Presentación del problema	3
1.3 Justificación	4
1.4 Objetivo general	5
1.5 Objetivos específicos	5
1.6 Metodología	6
1.6.1 Diseño e implementación electrónica	6
1.6.2 Integración del sistema	6
2 Conceptos básicos	7
2.1 Definición de IoT	7
2.2 Red 5G Y IoT.	7
2.3 Definición de un sistema embebido	8
2.4 Open source	9
2.5 Arduino	9
2.6 Módulos WiFi ESP8266	10
2.6.1 Reloj principal	11
2.7 Microcontrolador	11
2.7.1 Procesador	11
2.7.2 Arquitectura	12
2.7.3 Unidad de Procesamiento Central	13
2.7.4 SoC ESP8266	13
2.7.5 Puertos de entrada y salida	14
2.7.6 Pines digitales	14
2.7.7 Pin analógico	14
2.7.8 Interface I2C	14
2.7.9 Modulación de ancho de pulso (PWM)	14
2.7.10 Memoria flash	14
2.7.11 Comandos AT	15

2.7.12	Firmware AT	15
2.7.13	ESP-01	17
2.8	ESP201	18
2.8.1	Versiones de Node MCU	19
2.8.2	ESP8266 Node MCU	19
2.8.3	Energía y alimentación	20
2.9	Sensores	20
2.9.1	Definición de Sensor	21
2.9.2	Clasificación de sensores	21
2.9.3	Sensor PIR	21
2.9.4	Principios de funcionamiento	22
2.9.5	Sensor LDR	25
2.9.6	Módulo de Sensor DHT11	26
2.10	Tecnología WiFi	27
2.10.1	Características de la tecnología WiFi	28
3	Estado del arte	31
3.0.1	Technological Proposal for Controlling a Residential Lighting System	31
3.0.2	Sistema de monitorización domiciliaria	31
3.0.3	Diseño e implementación de un prototipo de control de iluminación LED con aplicación móvil compatible con Arduino	32
3.0.4	Control remoto del encendido y apagado y monitorización del consumo de energía eléctrica de un sistema de iluminación	32
3.0.5	Análisis de resultados	33
4	Análisis y diseño	35
4.0.1	Diseño	36
5	Desarrollo del proyecto	41
5.1	Descarga e instalación del entorno de programación de Arduino: IDE	41
5.2	Instalación del gestor de archivos del ESP8266 para el IDE de Arduino	41
5.2.1	Configuración del URL de soporte del procesador ESP8266	41
5.2.2	Instalación y configuración del gestor de archivos	42
5.3	Instalación del controlador CH340 para Arduino en Windows 10	43
5.3.1	Descarga del controlador CH340	43
5.4	Instalación del controlador Prolific PL2303	44
5.4.1	Descarga del controlador para el PL2303HX del convertidor USB to Serial	44
5.5	Pruebas de funcionamiento utilizando ESP-01	44
5.5.1	Conexión del ESP-01	44
5.6	Selección de la versión del módulo Generic ESP8266 Module	45
5.6.1	Blinky LED	46
5.6.2	Servidor web para encender y apagar un LED	48

5.7	Pruebas de Funcionamiento utilizando el ESP201	50
5.7.1	Resistencias pull up y pull down	50
5.7.2	Resistencia pull down	51
5.7.3	Resistencia pull up	51
5.8	Encendido y apagado de un LED con push button	51
5.8.1	Encendido de un LED con Sensor PIR	51
5.8.2	Obtención de datos de temperatura y humedad con sensor DHT11 . .	52
5.8.3	Funcionamiento de Relay con ESP-201	53
5.9	Node MCU	54
5.10	Ejemplos con Node MCU	55
5.10.1	Servidor web modo Acces Point con procesador ESP8266 (Node MCU)	55
5.10.2	Protocolo ESP-NOW para establecer una comunicación inalámbrica entre módulos Node MCU-ESP8266	57
5.10.3	Medición del nivel de luz con fotoresistencia LDR usando Node MCU	59
5.11	Diagramas de flujo para prototipo	59
5.12	Circuito final para prototipo	62
5.12.1	Diseño del case para albergar el sistema electrónico	63
6	Conclusiones y Recomendaciones para Trabajo Futuro	65
	Bibliografía	67
7	Anexo A: Código para las aplicaciones	75
7.1	Hola Mundo	75
7.2	Servidor web para encender y apagar un LED con ESP-01	75
7.3	Encendido de un LED con sensor PIR	79
7.4	Uso de relay con ESP201 para encender una bombilla	81
7.5	Servidor web modo Acces Point con procesador ESP8266 (Node MCU) . . .	82
7.6	Obtención de direcciones MAC	85
7.7	Comunicación ESPNOW/Maestro	85
7.8	Comunicación ESPNOW/Esclavo	87
7.9	Programa final apagador	88
7.10	Programa final Luminaria	90

Lista de Figuras

Figura 2.1	Logo Open Source	9
Figura 2.2	Modelos de placas Arduino.	10

Figura 2.3	Procesador ESP8266EX.	12
Figura 2.4	Arquitectura Harvard.	12
Figura 2.5	Interior de un EPS8266 con sus respectivos periféricos.	13
Figura 2.6	Módulo ESP01.	17
Figura 2.7	Versión ESP8266-12 y ESP8266-12E.	19
Figura 2.8	Módulo de placa de desarrollo de red WIFI ESP8266 ESP-12E.	19
Figura 2.9	Sensor PIR.	21
Figura 2.10	Funcionamiento del sensor HC-SR501	23
Figura 2.11	Ajuste de los resistores variables del sensor	24
Figura 2.12	Rango de detección del sensor PIR	24
Figura 2.13	Sensor LDR	25
Figura 2.14	Módulo sensor de temperatura y humedad DHT11.	27
Figura 2.15	Bandas ISM.	30
Figura 4.1	Socket Sensor De Movimiento Regulable.	35
Figura 4.2	Socket WiFi inteligente con temporizador y control remoto a través de un celular.	36
Figura 4.3	Sistema de adquisición de datos.	37
Figura 4.4	Circuito Optoacoplador TRIAC y MOC3021.	37
Figura 4.5	Captura de la instalación del prototipo en una sala de un espacio doméstico.	38
Figura 4.6	Representación de las horas del Sol.	38
Figura 5.1	Configuración del URL de soporte del procesador ESP8266.	42
Figura 5.2	Instalación del gestor de tarjetas.	42
Figura 5.3	Instalación de tarjetas para el procesador ESP8266	43
Figura 5.4	Diagrama de conexión para programar el firmware de ESP-01.	45
Figura 5.5	Selección de la placa ESP-01.	46
Figura 5.6	Subiendo el primer programa contenido en el IDE de Arduino: Blink	47
Figura 5.7	Compilando programa de ejemplo contenido en el IDE de Arduino: Blink.	48
Figura 5.8	Encendido y apagado de un LED desde una página web.	49
Figura 5.9	Diagrama de conexión para grabar el firmware de ESP201.	50
Figura 5.10	Configuración de resistencias Pull Down y Pull Up.	50
Figura 5.11	Encendido y apagado de un led con resistencia configurada en modo pull down.	51
Figura 5.12	Circuito Sensor de movimiento	52
Figura 5.13	Medición de temperatura y humedad con ESP201 y sensor DHT11.	53
Figura 5.14	Circuito Relay para encender Bombilla.	54
Figura 5.15	Circuito ESP201 con Relay	54
Figura 5.16	Encendido y apagado de un LED mediante un servidor web.	56
Figura 5.17	Red WiFi visible llamada ProyectoIoT.	57
Figura 5.18	Circuito Node MCU Maestro.	58
Figura 5.19	Nivel de luz con sensor LDR.	59

Figura 5.20	Diagrama de flujo apagador.	60
Figura 5.21	Diagrama de flujo para luminaria.	61
Figura 5.22	Diagrama de flujo para sensor de presencia.	62
Figura 5.23	Diagrama de conexión para circuito final.	63
Figura 5.24	Primer diseño en impresora 3D.	63
Figura 5.25	Segundo diseño con circuito electrónico preparado para ser utilizado.	64

Listas de Tablas

Tabla 1.1	Comparación de luminarias(Ramírez y otros, 2019).	4
Tabla 2.1	Lista de comandos AT (Naylamp Mecatronics, 2017).	16
Tabla 2.2	Especificaciones técnicas del módulo ESP-01.	18
Tabla 2.3	Especificaciones técnicas del módulo Node MCU ESP-12E.	20
Tabla 2.4	Especificaciones técnicas del sensor HC-SR501S	25
Tabla 2.5	Especificaciones técnicas del sensor LDR	26
Tabla 2.6	Especificaciones técnicas del sensor de temperatura y humedad DHT11	27
Tabla 3.1	Comparación de las características relacionadas con la elaboración de este trabajo.	33
Tabla 5.1	Modo de operación para el módulo ESP-01.	45

Notación

IOT Internet of Things
SOC System on a Chip
MCU Microcontroller Unit
E/S Entrada/Salida
UART: Universal Asynchronous Receiver-Transmitter
CPU Central Processing Unit
GPIO General Purpose Input Output
PCB Printed Circuit Board
I²C Inter Integrated Circuit
ADC Analog to Digital Conversion
PWM Pulse Width Modulation
SDK Software Development Kit
LED Light Emitting Diode
ITU-T International Telecommunication Union
IP Internet Protocol
TCP Transmision Control Protocol
IDE Integrated Development Environment
TTL Transistor Transistor Logic
API Aplication Programming Interface
MIT Massachusetts Institute of Technology
WLAN Wireless Local Area Network
WPA Wi-Fi Protected Access
SSID Server Set Identifier
AP Access Point
USB Universal Serial Bus
SMD Surface Mounted Device

Planteamiento del problema

1.1. Introducción

Actualmente vivimos en una sociedad en donde la comunicación entre usuarios ya "no es suficiente", a medida que la tecnología ha ido evolucionando, las personas tienen una mejor interacción con el entorno que los rodea y esto es debido al aumento en la variedad de dispositivos que están conectados a Internet.

Las nuevas tecnologías de redes WSN (Wireless Sensor Network) involucran redes inalámbricas en conjunto con los sensores, en donde su objetivo principal es obtener información del entorno a través de Internet y la interconexión de sensores u objetos digitales cotidianos, en una red de comunicación y actuación que ha proliferado, tanto en entornos urbanos como ambientales, dando paso a lo que se conoce como IoT (Botia y otros, 2016).

La posibilidad de un entorno automatizado en los hogares, aunado a tecnologías de IoT, permite a los usuarios interactuar y tener un mejor control de los dispositivos que los rodean, para responder ante las necesidades de las personas de una manera más rápida, segura y eficiente.

Debido a esta creciente tendencia que implica el uso de IoT, actualmente existen diversas aplicaciones que permiten a los usuarios interactuar con esta tecnología, dando una nueva visión en el ámbito de la automatización de sistemas a través del uso de microcontroladores, sensores y actuadores que pueden conectarse a Internet, para realizar tareas específicas, como el control de luminarias para ahorrar en el consumo de energía eléctrica, que es una de las razones por la cual fue realizado este trabajo.

1.2. Presentación del problema

Tras un periodo de más de 100 años, desde que Thomas Alva Edison tuvo éxito en iluminar una bombilla incandescente con electricidad, han sido desarrollados nuevos tipos de luminarias que brindan luz de gran calidad pero con poca eficiencia en el consumo de energía.

La reducción del consumo eléctrico en los hogares, ha sido una de las prioridades de las personas y el desarrollo de las nuevas tecnologías. La creación de lámparas ahorradoras como las fluorescentes o las que están fabricadas con tecnología LED, ha ido en aumento

y reemplazando cada vez más a las casi extintas lámparas incandescentes, que generan un mayor consumo de energía eléctrica en los hogares.

Con base en un estudio realizado por Ramírez Montáñez y otros en el que llevan a cabo una comparación entre el costo y consumo de tres tipos de luminarias: incandescentes (1), ahorradoras (2), y LED (3) en un periodo de 8 horas, considerando 10 focos para obtener el cálculo de cada tipo de luminaria, tomando como referencia las tarifas establecidas por la Comisión Federal de Electricidad en el año 2018, obtuvieron los siguientes resultados del cuadro 1.1(Ramírez y otros, 2019).

	1	2		3	
	Philips	Marca Mavrik	Marca ETCO	Philco	GE Dimeable
Precio de compra de 10 focos	\$350	\$400	\$480	\$1,000	\$1,500
Precio individual	\$35	\$40	\$48	\$1,000	\$1,500
Consumo individual (W)	100	25	24	15	15
Iluminación individual (W)	100	100	95	100	100
Consumo de energía bimestral (KW)	\$600	\$150	\$144	\$90	\$90
Primeros 75 KW	\$59.48	\$59.48	\$29.48	\$59.48	\$59.48
Siguientes 65 KW	\$62.14	\$62.14	\$62.14	\$62.14	\$62.14
Resto	\$1268	\$28	\$11.2	\$ -	\$ -
Precio Total Bimestral	\$1,410	\$149.8	\$132.8	\$121.6	\$121.6
Precio por día	\$23.51	\$2.49	\$2.21	\$2.03	\$2.03

Cuadro 1.1: Comparación de luminarias(Ramírez y otros, 2019).

Con esta información, el autor demuestra que la relación costo-beneficio generada por la luminaria incandescente, supera el precio del consumo de cualquiera de los otros tipos. Por lo tanto, a largo plazo considerando el precio de las luminarias LED seguido por el ahorrador de 24W, son la mejor opción para ahorrar en costo y consumo de energía eléctrica.

1.3. Justificación

La mayoría de las luminarias en espacios residenciales son controladas manualmente, utilizar sistemas automatizados aunados a tecnologías IoT, permite a los usuarios tener un mayor control de los dispositivos, disminuir el consumo de energía eléctrica, ahorrar costos y hacernos la vida más confortable.

En México, por ejemplo, Hugo Pérez Rebolledo da algunas alternativas para reducir el consumo de energía eléctrica residencial ya que los “electrodomésticos de mayor consumo en los hogares son la iluminación y los refrigeradores, con un porcentaje promedio del 70 % del consumo total”. Entre las alternativas se encuentran: el uso de tecnologías que están

ligadas a las redes “inteligentes” que involucran el uso de sensores, redes de comunicación y procesamiento de datos, sistemas de actuación y control(Rebolledo, 2011).

Por otro lado, en un estudio publicado por la Asociación de Tecnología de Consumo (CTA / Consumer Technology Association), el uso de la automatización residencial a través de IoT tiene el potencial para ahorrar energía eléctrica hasta en un 10 %. Dicho ahorro puede ser posible adoptando productos automatizados que nos indiquen los cambios de temperatura, el control de iluminación, etc.(Sánchez, 2016).

Al desarrollar un sistema automatizado con un enfoque orientado a IoT, usando componentes de bajo costo y software de código abierto se puede implementar esta tecnología en cualquier espacio residencial que cuente con un socket de una luminaria convencional, para reducir el consumo de energía eléctrica.

De esta manera, en este trabajo se propone controlar el encendido y apagado de una luminaria por medio de dos módulos ESP8266 (Node MCU), que utilizan tecnología inalámbrica y están enfocados al IoT. Además del control de las luminarias de manera manual, se agregan sensores de presencia e intensidad luminosa para aprovechar la luz del día y contrarrestar el consumo de energía eléctrica en los hogares. Así, es posible que el sistema responda ante los cambios de intensidad luminosa, encendiendo y apagando una luminaria de manera automática, o controlarlo con un apagador del cual dispone el sistema. Por otro lado, también lo que nos permite este sistema es recopilar variables de temperatura y humedad, que podrían ser utilizadas en un futuro para controlar por ejemplo un sistema de aire acondicionado, e ir agregando conectividad inalámbrica a cada uno de los dispositivos electrodomésticos que tenemos en nuestros hogares, para tener una mayor gestión sobre cada uno de estos a través de la interacción de sensores, la automatización y las redes inalámbricas.

Si logramos ahorrar energía eléctrica con la ayuda de nuevas tecnologías tan sencillas como lámparas de tecnología LED y el control de nuestros dispositivos en nuestros hogares, estaremos disminuyendo la emisión de gases contaminantes hacia la atmósfera (CO₂).

1.4. Objetivo general

Diseñar e implementar un sistema con tecnología IoT que permita recopilar datos a través de diferentes sensores como temperatura, presencia e intensidad luminosa, para controlar de manera automática una luminaria en un espacio doméstico, y disminuir el consumo de energía eléctrica.

1.5. Objetivos específicos

- Crear un programa utilizando el IDE de Arduino para controlar los módulos Wifi y poder llevar a cabo la transmisión de los datos.
 - Diseñar el prototipo de una caja que contendrá el sistema electrónico, utilizando un software denominado Solidworks.
-

- Utilizar una impresora 3D (Anycubic) para imprimir el prototipo de la caja que contendrá el módulo programable y los sensores.

1.6. Metodología

Para el desarrollo del presente proyecto se implementó una investigación en la que se emplearon conocimientos de tipo electrónico e informático que permitieron desarrollar un prototipo con tecnología IoT, para dar una solución en el ahorro de energía eléctrica de una luminaria en un espacio doméstico, iluminando solo cuando es necesario, a través del monitoreo y condiciones del entorno.

El tipo de investigación consultada fue la siguiente:

- Bibliográfica, ya que se obtuvo información de libros, fuentes de Internet, artículos, revistas, tesis relacionadas al control de luminarias para reducir el consumo de energía eléctrica y temas referentes a tecnología IoT.
- Experimental, porque se llevaron a cabo diferentes pruebas para el correcto funcionamiento del sistema de iluminación y poder obtener un ahorro energético con la puesta en marcha del prototipo.

El proyecto aborda las siguientes faces operativas (diseño e implementación electrónica, e integración del sistema), que culminan en un prototipo viable para los usuarios y el medio ambiente, que se suma al campo del conocimiento e información científica sobre la automatización y uso de tecnología IoT.

1.6.1. Diseño e implementación electrónica

El diseño y desarrollo del sistema electrónico se basa en los siguientes componentes:

- Hardware, componentes electrónicos y accesorios (tarjetas electrónicas, circuitos impresos o PCB, etc.).

Firmware, que es un código de software que contiene las instrucciones que se instalan en las unidades de hardware (microprocesadores, microcontroladores o sistemas embebidos)(Carballar, 2010).

1.6.2. Integración del sistema

Se describe la implementación entre cada uno de los componentes que conforman el sistema electrónico, las pruebas realizadas a lo largo del proyecto, los resultados obtenidos y recomendaciones para trabajo a futuro.

Capítulo 2

Conceptos básicos

2.1. Definición de IoT

En la actualidad contamos con varias definiciones para IoT, una de ellas es la propuesta por el organismo especializado en Telecomunicaciones de las Naciones Unidas - UIT (Unión Internacional de Telecomunicaciones), quien señala lo siguiente en el ámbito de las Tecnologías de la Información y Comunicaciones:

“IoT es una infraestructura global para la sociedad de la información, que permite servicios avanzados interconectando cosas (físicas y virtuales) basadas en las existentes y en evolución de las tecnologías interoperables (TICs)” (UIT - Telecommunication Standardization Sector, 2015).

Una definición también válida puede ser la siguiente: IoT es una red compuesta por diferentes objetos físicos que hacen uso de la red para su funcionamiento.

En 1999 el concepto de IoT se le acuñó a Kevin Ashton al darlo a conocer por primera vez al público en una presentación realizada para Procter y Gamble (Chan y otros, 2016). Desde entonces, surge un nuevo paradigma para las redes de comunicación, ya que estas darán un mayor fortalecimiento para el despliegue y desarrollo de los servicios de las redes inalámbricas.

Hoy en día la importancia del Internet de las Cosas, radica en la necesidad de tener control sobre la mayor parte de las cosas a las que se les pueda adaptar una conexión a internet, donde el usuario pueda gestionar u observar los dispositivos.

Como resultado del rápido crecimiento que ha tenido IoT, los sectores privados e instituciones que involucran proyectos de investigación, han motivado a las empresas para que la producción en serie de estos dispositivos (módulos, chips, sensores, actuadores, etc.) que involucran conexión a Internet aumente, comenzando a ser vendidos y fabricados a gran escala (Sotelo y otros, 2015).

2.2. Red 5G Y IoT.

La 5ta generación de telefonía móvil o “IMT-2020”, término que utiliza la UIT para referirse a las redes 5G será una clave primordial para el despliegue actual del IoT.

Si bien sabemos el desarrollo que impulsó el uso de las tecnologías de comunicaciones fue la comunicación entre personas, a lo largo del tiempo esta idea ha ido evolucionando, ya que con base a un estudio publicado por la empresa Huawei para el año 2025 se estima que habrá alrededor de 100,000 millones de dispositivos: sensores, smartphones, vehículos, electrodomésticos, etc. de los cuales sólo el 10 % de las interconexiones será sólo entre personas, el resto equivaldrá entre máquinas y dispositivos de manera independiente.

Por lo tanto, el reto que enfrenta hoy en día las redes 5G aunado al IoT son los siguientes:

- Implementar un uso de ancho de banda que haga frente a los grandes volúmenes de información que generarán todos los dispositivos que interactuen de manera independiente y en conjunto con las personas.
- Soporte para todas las comunicaciones masivas de tipo máquina para IoT, que requerirán de un bajo consumo de energía y bajas tasas de transmisión de datos.
- Comunicaciones confiables que requerirán de una baja latencia para todas las aplicaciones que generará la interconexión de estos sistemas de comunicación.
- Cuestiones de seguridad, ya que al tratar con grandes volúmenes de información se imponen más desafíos sin dejar a un lado que la 5G es la primera generación de redes móviles pensada para la comunicación M2M.

Por lo tanto, el foco rojo que enfrenta la 5G es poder generar el mejor estándar para poder soportar los miles de millones de dispositivos que estarán generando grandes volúmenes de información, reducir la latencia, aumentando la fiabilidad entre conexiones, mejorar la eficiencia del espectro y afrontar el reto al hacer uso de bandas de alta frecuencia.

Finalmente, el gran esfuerzo que se requerirá para implementar el estándar 5G estará enfocado a soportar miles de millones de dispositivos inalámbricos y fusionarse con las comunicaciones de todas partes.

En materia de regulación del espectro radio eléctrico, los fabricantes de tecnologías móviles, módems, etc. tendrán que apegarse a las normas requeridas por el Global Certification Forum (GCF) para que puedan ser lanzados al mercado (Ávila y otros, 2020).

2.3. Definición de un sistema embebido

Los sistemas embebidos son definidos como un tipo de sistema electrónico, cuya función principal no es computacional, pero es controlado por un computador programable. Su objetivo es realizar tareas específicas en tiempo real, mediante el uso de un microcontrolador (Pérez, 2019).

Si se hace una comparación de un sistema embebido con una computadora, se dice que el segundo se distingue, porque este puede realizar diversas tareas dependiendo de las instrucciones definidas por los usuarios y el primero se limita a realizar ciertas tareas en específico.

2.4. Open source

El término código abierto es un concepto que describe el código fuente compartido junto con el desarrollo colaborativo detrás del software, como lo son los sistemas operativos Linux y Free BSD, Servidor Web Apache, Perl, etc., en el que los programadores crean, modifican, y comparten código fuente sin tarifas de licencia.

Algunas de las licencias que se han derivado del código abierto son: la licencia pública GNU (GPL), la licencia BSD con los derivados de Berkeley Unix, la licencia pública de Mozilla, etc. (O' Reilly, 1999), (McDonald y otros, 2003).



Figura 2.1: Logo Open Source

2.5. Arduino

Arduino es una plataforma Open Source Hardware, tanto su diseño como su distribución, respaldada por una gran cantidad de personas que hacen uso de sus programas o contribuyen a la elaboración de nuevos proyectos o mejorando los ya existentes sin haber adquirido ninguna licencia. Está enfocada especialmente para cualquier persona que tiene conocimientos básicos en programación, electrónica y le interese contribuir o crear programas con nuevas ideas o proyectos. Estos dispositivos están fabricados con una placa electrónica de hardware y software libre flexibles y fáciles de usar.

Arduino incorpora un microcontrolador reprogramable que utiliza el Arduino Programming Language (basado en Wiring) y el Arduino Development Environment (basado en Processing)(Herrador, 2009).

La placa Arduino dispone de una serie de pines hembra que a través de su configuración pueden comportarse como pines de E/S (entrada/salida), que son los encargados de que exista una comunicación con el exterior. En los pines de entrada se pueden conectar diferentes tipos de sensores (temperatura, humedad, movimiento), que recopilan información digital adquirida por el entorno. En los pines de salida se pueden conectar actuadores (motor, bombilla, altavoz), que son los encargados de llevar a cabo las instrucciones dictadas por el microcontrolador para interactuar con el medio físico (Artero, 2013).

Algunas de las ventajas que ofrece Arduino son las siguientes:

- Tiene un lenguaje simple basado en C/C++.

- El IDE que utiliza es de código abierto y uno de los más fáciles de utilizar, dando como resultado un gran interés a los usuarios que se adentran por primera vez en este proyecto.
- En cuanto a los desarrolladores, a estos les proporciona un mejor control, disponibilidad, modificación y creación de múltiples programas y librerías sin dejar a un lado los incontables sitios web, foros, información de consulta y ayuda para la elaboración de proyectos.

Actualmente existen diferentes diseños hardware de la placa Arduino, la elección dependerá del tipo de proyecto a realizar, ya que cada una cuenta con características diferentes (tamaño físico, número de pines hembra, modelo de microcontrolador, tamaño de memoria, etc.). De esta manera, al hablar de una placa Arduino tendríamos que especificar el tipo de modelo al cual nos estamos refiriendo. En la figura 2.2 se muestran algunos modelos de las placas Arduino.



Figura 2.2: Modelos de placas Arduino.

2.6. Módulos WiFi ESP8266

La plataforma de conectividad inalámbrica que caracteriza a los módulos ESP8266 consiste en un chip integrado de bajo costo que en esencia utiliza el estándar IEEE 802.11 (WiFi) y es compatible con el protocolo TCP/IP. Otro factor importante al trabajar con estos módulos son las limitaciones de tamaño de cada dispositivo y el bajo consumo de energía. Los fabricantes pertenecen a una empresa de origen China llamada Espressif.

Características principales:

- El ESP8266 integra un procesador RISC Tensilica L106 de 32 bits, de bajo consumo de energía y alcanza una velocidad máxima de reloj de 160 MHz.
- Módulo WiFi de 2.4 GHz.
- RAM de unos 64 kB.
- 1 entrada analógica de 10-bit (ADC).
- Pines de entrada y salida GPIO (de propósito general).
- Soporte con buses digitales SPI , I²C, entre otros.
- I²S interfaces con DMA (pines de intercambio con GPIO).
- UART en los pines dedicados, además de una transmisión de sólo UART se puede habilitar en la línea GPIO2.

2.6.1. Reloj principal

Para que un microcontrolador funcione este requiere de un circuito oscilador de alta frecuencia que genera una onda cuadrada que sincroniza los pulsos de reloj para llevar a cabo las operaciones del sistema, por lo que el programa, los contadores y las instrucciones ejecutadas por la CPU funcionen a una velocidad adecuada, en función de la frecuencia utilizada. En el caso del ESP8266 este alcanza una velocidad máxima de reloj de 160 MHz(Rebollo, s.f.).

2.7. Microcontrolador

Un microcontrolador es una pequeña computadora en un chip, de capacidad limitada que contiene un circuito integrado programable, capaz de realizar tareas puntuales que han sido grabadas en su memoria interna a través de un usuario(Prado, 2017). Una de las ventajas que ofrece un microcontrolador es la integración de un sistema electrónico que contiene la Unidad de Procesamiento Central, la memoria RAM, memoria ROM, puertos de entrada, salidas y otros periféricos en un reducido tamaño.

2.7.1. Procesador

Las placas o kits de desarrollo ESP8266 llevan incorporado un chip llamado SoC (System on a Chip) que está compuesto por un microcontrolador Tensilica L106, que es un procesador de 32 bits que requiere de un bajo consumo de energía. La velocidad de reloj para la CPU es de 80 MHz y logra alcanzar una velocidad de hasta 160 MHz.



Figura 2.3: Procesador ESP8266EX.

Al funcionar con 32 bits significa que puede realizar operaciones con números de ese tamaño (de 0 a 4.294.967.295 o de -2.147.483.648 a 2.147.483.647). Sin embargo, los microcontroladores más comunes como el ATmega328P de Arduino Uno trabajan con 8 bits.

Por lo tanto, el trabajo del microcontrolador es gestionar todas las entradas, salidas y cálculos necesarios para hacer funcionar el programa que hayamos cargado.

Además de la funcionalidad WiFi que ofrecen las placas ESP8266, es poder programarlos, trabajar con sensores externos y otros dispositivos a través de sus pines GPIO y el kit de desarrollo de software (SDK).

2.7.2. Arquitectura

El ESP8266 trabaja con una arquitectura Harvard que le permite a la CPU leer una instrucción (64 KB) y realizar al mismo tiempo el acceso a la memoria de datos (96KB) debido al canal de buses que se conectan a cada una de las memorias de la CPU.



Figura 2.4: Arquitectura Harvard.

De esta manera la arquitectura Harvard es como se impone en los computadores, debido a que la instrucción obtiene acceso a los datos de manera independiente y no compite por una única vía de memoria (Rentería y otros, 2017).

2.7.3. Unidad de Procesamiento Central

La Unidad de Procesamiento Central (CPU) es el componente más importante del microcontrolador, y es quien se encarga de ejecutar cada una de las instrucciones y controlar que dicha ejecución se realice de manera correcta.

La estructura interna de un procesador está compuesta por un conjunto de registros que almacenan datos y son procesados por una Unidad Aritmético-Lógica (ALU) que se encarga de realizar operaciones con ellos. Otro componente denominado unidad de control es quien coordina cada una de estas tareas.

2.7.4. SoC ESP8266

Un SoC es un sistema electrónico que está fabricado dentro de una pastilla de silicio que permite incluir múltiples circuitos integrados en un sólo dispositivo semiconductor (Han y otros, 2012).

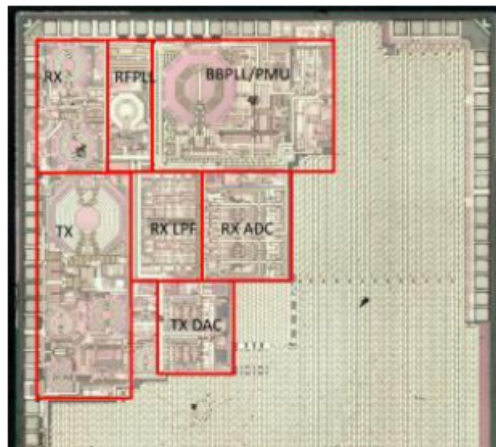


Figura 2.5: Interior de un EPS8266 con sus respectivos periféricos.

Algunas de las razones que ha motivado la fabricación de estos dispositivos son las funciones que se van agregando a esta plataforma y la miniaturización de las placas para introducir una mayor cantidad de microprocesadores dando como resultado el incremento de la velocidad de operación en estos, por ejemplo cuando surgieron los primeros celulares estos eran muy grandes, pesados y sólo permitían establecer llamadas y enviar mensajes. En comparación con un teléfono celular tradicional, este desarrolla múltiples funciones; consumen menor potencia, son más rápidos en la ejecución de cada una de las instrucciones que realizan, etc.

Los componentes que integra un SoC son la memoria del controlador, un núcleo para el procesador, funciones que son determinadas por un diseñador para el espacio destinado en memoria, interfaces de bus, drivers E/S, soporte de red, etc. (Sistemas Embebidos, FPGA + MCU, 2019).

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.

2.7.5. Puertos de entrada y salida

Los puertos de entrada y salida (E/S) de un microcontrolador son el punto de interconexión que permite al sistema agregar dispositivos externos para recibir señales que intercambian información con el exterior a través de sus múltiples interfaces(Chiquillo, 2011).

2.7.6. Pines digitales

Como veremos más adelante, las versiones de los módulos ESP8266 contienen pines digitales y analógicos. En el caso de los pines digitales estos son utilizados para conectar dispositivos externos como sensores, actuadores y transmitir un programa a través de la comunicación entre el módulo y el PC por medio de un puerto serie.

2.7.7. Pin analógico

Algunos módulos ESP8266 tienen integrado un pin analógico que admite un rango de valores de 0 a 3,3V con una resolución de 10-bit. Esto implica que dentro del código tendremos un valor entre 0 y 1023 que se mapea con el voltaje entre 0 y 3,3V.

2.7.8. Interface I2C

El ESP8266 tiene una interface I2C, que se utiliza por medio de la programación de software, esta por ejemplo se puede conecta con otros microcontroladores y otros equipos periféricos como sensores.

La funcionalidad de la interface I2C se puede realizar mediante programación de software, y la frecuencia máxima de reloj con la que puede operar es de 100 kHz.

2.7.9. Modulación de ancho de pulso (PWM)

El ESP8266 tiene una interface de salida PWM que puede ser implementada mediante la programación de software. El rango de frecuencia PWM es ajustable y puede ir de 100 Hz a 1 kHz.

2.7.10. Memoria flash

el ESP8266 no tiene memoria flash y por lo tanto necesita de una memoria externa en la que almacenar los programas y datos. Para conectarse a ella, necesita utilizar pines del propio ESP8266.

2.7.11. Comandos AT

Una de las opciones que ofrece el ESP8266 es poder agregar instrucciones a través del firmware AT por medio de un puerto serie.

Los comandos AT (Hayes) son un conjunto de instrucciones que rápidamente se convirtieron en un estándar abierto para poder configurar y parametrizar módems.

2.7.12. Firmmmware AT

En el cuadro 2.1 se muestra la lista de algunos de los comandos principales para el firmware del ESP8266:

Comando	Respuesta	Función
AT-Probar iniciación correcta		
AT	OK	Prueba si el módulo responde correctamente.
AT+RST - Reinicia el módulo		
AT+RST	OK	Resetéa el módulo
AT+CWMODE - Modo Wifi		
AT+CWMODE=?	+CWMODE:(1-3) OK	Lista los modos validos
AT+CWMODE?	+CWMODE:modo OK	Pregunta en que modo AP esta actualmente el módulo
AT+CWMODE= modo	OK	Establece el módulo en el modo dado 1 = Modo estación (cliente) 2 = Modo AP (huésped) 3 = Modo AP + Estación (modo dual)
AT+CWLAP - Lista APs disponibles		
AT+CWLAP	AT+CWLAP:ecn, ssid,rssi,mac OK	Lista los Aceso Points disponibles para conectarse. ecn: codificación, puede ser: 0 = Abierto 1 = WEP 2 = WPA PSK 3 = WPA2 PSK 4 = WPA WPA2 PSK ssid: String que contiene el SSID del AP rssi: Fuerza de la señal mac: String que contiene la dirección MAC
AT+CWLAP=ssid, mac,ch	+CWLAP:ecn,ssid, rssi,mac OK	Busca Aceso Points disponibles para conectarse con las condiciones especificadas
AT+CWJAP - Unirse a un Access Point		
AT+CWJAP?	+ CWJAP:ssid OK	Imprime el SSID al que el módulo esta conectado
AT+CWJAP=ssid, pwd	OK	El módulo se conecta a la red con el nombre ssid indicado y la contraseña pwd suministrada
AT+CWQAP - Desconectarse de una Access Point		
AT+CWQAP	OK	Se desconecta de la red que esta actualmente conectado
AT+CWSAP - Configurar el softAP del módulo		

Cuadro 2.1: Lista de comandos AT (Naylamp Mecatronics, 2017).

Como se muestra en el cuadro anterior, sólo se ejemplifican algunos de los comandos básicos que son más comunes al trabajar con el firmware AT, para tener más información sobre estos se puede consultar la fuente en el apartado referencias.

Al momento de trabajar con comandos AT por defecto se utiliza un baud rate de 9600.

Después de haber ejecutado cada comando, el ESP8266 espera los caracteres especiales de nueva línea <CR><LF> para poder llevar a cabo cada instrucción.

2.7.13. ESP-01

El primer chip lanzado al mercado se dio a conocer alrededor del mes de agosto de 2014, con la versión que integra el módulo ESP-01 que se muestra en la figura 2.6. El objetivo principal de este módulo y lo que ofrecieron las siguientes versiones fue dotar a cualquier microcontrolador el acceso a una red inalámbrica WiFi.

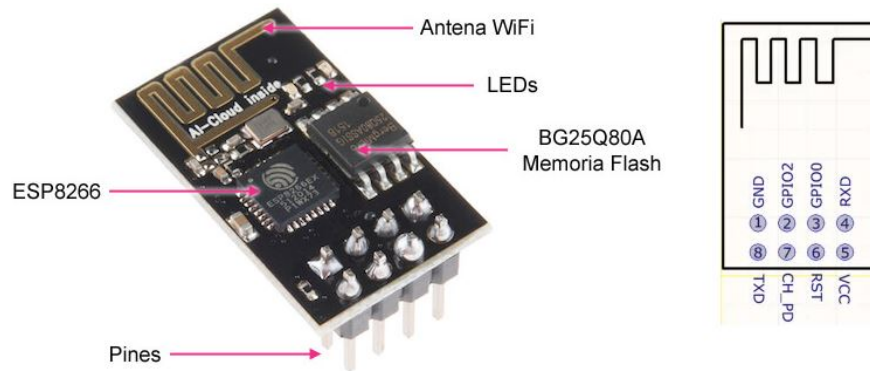


Figura 2.6: Módulo ESP01.

En sus inicios como cualquier tecnología que sale a la luz, la información con la que contaba la comunidad de desarrolladores era muy restringida y sólo se disponía de ella en el idioma chino; rápidamente por el éxito que tuvieron estos chips y la gran comunidad que había detrás de ellos, la documentación no tardó en traducirse, dando origen a la creación de los primeros firmwares para el ESP.

El módulo ESP-01 consta de un procesador RISC de 32 bits y este microcontrolador cuenta con tecnología WiFi (802.11 b / g / n) para poder conectarse a otra red inalámbrica.

En la actualidad contamos con varias versiones de estos módulos, y cada uno de ellos dispone de entradas y salidas digitales como cualquier microcontrolador, así como un convertidor analógico digital (en inglés, Analog to Digital Conversion, abreviado ADC), (¿Qué es ESP8266?, 2017), (Laborda, 2016), (Llamas, 2015).

Para este trabajo fue importante disponer de un ADC para trabajar en conjunto con un sensor de intensidad luminosa y presencia para controlar el encendido y apagado de la luminaria, sólo cuando fuera necesario a ciertas horas del día. De no contar el chip con el ADC, este tendría que haberse implementado en otro circuito de manera independiente.

Opciones de programación

Desde que surgió la primera placa ESP-01 para poder cargar las instrucciones al microcontrolador se han utilizado comandos AT a través de un firmware. Aunque existen otras opciones más sofisticadas como NodeLua, Python, BASIC, etc., en este proyecto se utilizó el IDE de Arduino.

Entradas y salidas ESP-01

El módulo ESP-01 contiene un microcontrolador ESP8266 y un chip de memoria flash. Hay dos LED: uno rojo que indica que la energía está conectada al módulo y uno azul que indica el flujo de datos. La antena WiFi se distingue fácilmente ya que se encuentra en la parte superior del módulo y tiene una forma de F invertida.

Los ocho pines con los que cuenta el módulo en la parte inferior indican la función que realiza cada uno, como se puede apreciar en la figura 2.6.

A continuación, en el cuadro 2.1 se muestran las especificaciones para el ESP-01.

Parámetros de WiFi	Protocolos WiFi	802.11 a/b/g/n
	Rango de Frecuencias	2.4 GHz a 2.5 GHz
Parámetros de Hardware	Bus periférico	UART/HSPI/I2C/Control Remoto
		GPIO/PWM
	Voltaje de operación	3 a 3.6V
	Corriente de operación	80mA
	Rango de operación de temperatura	-40° a 120°
Parámetros de software	WiFi Mode	station/softAP /softAP+station
	Seguridad	WPA/WPA2
	Encriptación	WEP/TKIP/AES
	Actualización de Firmware	UART Download / OTA (vía Network) / Download and write firmware vía host
	Protocolos de red	IPv4 TCP/HTTP/UDP/FTP

Cuadro 2.2: Especificaciones técnicas del módulo ESP-01.

2.8. ESP201

El módulo WiFi ESP8266 ESP-201 Serial, es un dispositivo tranceptor de Radiofrecuencia activado por Wi-Fi, que brinda una solución para todos los dispositivos que requieran conexión a Internet. Este producto al igual que el ESP-01 incluye una antena en formato

SMD (Surface Mounted Device) dentro del mismo diseño del PCB (Módulo Wifi Esp8266 Esp 201 Serial, 2018).

El uso de antenas en formato SMD permite acoplarlas en la parte superior de la placa de circuito impreso, para mejorar la geometría de radiación y ocupar menos espacio cuando se crea el diseño (Antenas metálicas para circuito impreso en formato SMD, 2014).

2.8.1. Versiones de Node MCU

La primera versión que surgió para el Node MCU se denominó devkit V0.9, y disponía de un ESP12 con una memoria flash de 4MB (externa que se conecta por SPI). El ESP12 es una versión similar al ESP12E y en comparación con su sucesor este se diferencia ya que carece de una hilera de pines, lo cual implica un menor número de GPIO disponibles (Mairena T, 2018).

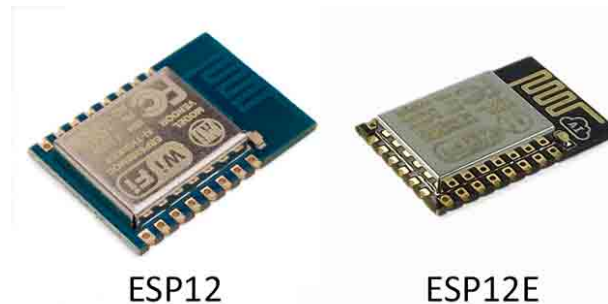


Figura 2.7: Versión ESP8266-12 y ESP8266-12E.

2.8.2. ESP8266 Node MCU

Aunque existen diversas versiones de módulos ESP8266, en este trabajo se optó por el Node MCU de la figura 2.8. Este módulo fue desarrollado por Arduino Company y es un MCU que está configurado en base a Arduino Uno (Aziz, 2018).



Figura 2.8: Módulo de placa de desarrollo de red WIFI ESP8266 ESP-12E.

El módulo Node MCU Integra un chip ESP8266EX (ESP-12E) WiFi, aplicable a la tecnología futura de IoT (Internet of Things). Este módulo fue desarrollado para utilizar el lenguaje de programación Lua, sin embargo, también se puede programar con el IDE de Arduino, comandos AT, entre otros. Está equipado con una serie de patillas GPIOs disponibles para la agregación de futuros sensores o actuadores, ADC, UART, PWM, memoria, conectorización que ofrece para energizar la placa y los canales para la transmisión de los datos a través de sus distintas versiones de los estándares 802.11/b/g/n y demás características que se muestran en el cuadro 2.3.

Parámetros de WiFi	Protocolos WiFi	802.11 a/b/g/n
	Rango de Frecuencia	2.4 GHz a 2.5GHz
Parámetros de Hardware	Bus periférico	UART/HSPI/I2C/Control Remoto GPIO/PWM
	Voltaje de operación	3 a 3.6V
	Corriente de operación	12mA
	Rango de operación de temperatura	-40° a 120°
Parámetros de software	WiFi Mode	station/softAP/softAP+station
	Seguridad	WPA/WPA2
	Encriptación	WEP/TKIP/AES
	Actualización de Firmware	UART Download/ OTA via Network/ Download and write firmware via host
	Protocolos de red	IPv4 TCP/HTTP/UDP/FTP

Cuadro 2.3: Especificaciones técnicas del módulo Node MCU ESP-12E.

2.8.3. Energía y alimentación

El NODEMCU utilizado requiere de una alimentación externa de 5 a 10V de DC a través de un cable micro-USB. Internamente, la placa otorga a los dispositivos conectados un voltaje de 3.3V. En cuanto a corriente suministrada por los pines de propósito general (GPIO), estos entregan una corriente de 6 a 12mA como intensidad máxima para no dañar el procesador(Placa NodeMCU 1.0 V2, 2016).

2.9. Sensores

Los sensores juegan un papel importante en el desarrollo de proyectos, ya que nos permiten conocer diferentes magnitudes físicas o químicas que resultan de interés para nuestras aplicaciones.

2.9.1. Definición de Sensor

Los sensores son dispositivos electrónicos que interactúan con el entorno, proporcionando información a través de ciertas variables físicas medidas, como temperatura, humedad, presencia, luminosidad, etc., en valores numéricos que pueden ser tratados para generar órdenes o activar procesos (Serna y otros, 2010).

2.9.2. Clasificación de sensores

De acuerdo a cada sensor estos se pueden clasificar en función de los valores de salida en:

- Analógicos
- Digitales

Dependiendo del tipo de sensor, estos son conectados a una entrada digital o analógica. A continuación se describen las características de los sensores utilizados en este proyecto.

2.9.3. Sensor PIR

Un sensor infrarrojo pasivo o sensor PIR por sus siglas en inglés, es un dispositivo que requiere de baja potencia, es fácil de usar y de bajo costo. Está compuesto por un sistema electrónico que se encarga de medir la radiación infrarroja radiada por las personas y animales que se encuentran situados en su campo de visión. En la figura 2.9 se muestra el sensor PIR.

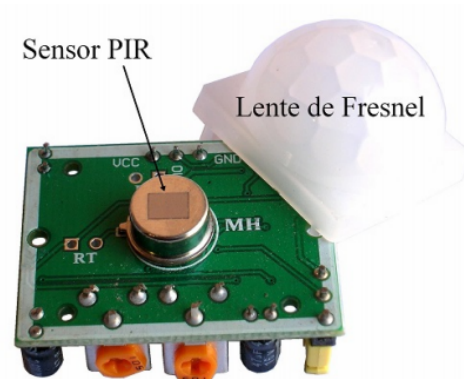


Figura 2.9: Sensor PIR.

Con frecuencia los sensores PIR son utilizados en muchos proyectos para espacios residenciales o negocios para dar solución al control de iluminación, porque ayudan a reducir el consumo de energía eléctrica, son eficientes y garantizan el control de la luz en el momento que requieran los usuarios. Su bajo costo y la eficiencia que brindan reducen el consumo y los costos que conlleva el uso de energía eléctrica (Control de iluminación, 2012).

Por lo tanto, las aplicaciones que se les dan a estos sensores son para utilizarlos como alarmas que generan un pulso de activación en su salida o controlando las luminarias de un espacio residencial, edificio, etc.

2.9.4. Principios de funcionamiento

Radiación infrarroja

Todos los seres vivos e incluso los objetos que se encuentran por encima del cero absoluto emiten radiación infrarroja (calor) de bajo nivel, debido a la temperatura en que se encuentren. Entre mayor temperatura, más radiación será emitida.

Bajo este principio es como son desarrollados los sensores PIR que responden ante determinadas fuentes de energía, tales como el calor del cuerpo humano o animales.

Lente de Fresnel

El diseño del sensor PIR contiene un encapsulado semiesférico fabricado de polietileno denominado lente de Fresnel, cuyo objetivo es detectar la radiación infrarroja que emite el cuerpo humano o animales que se encuentran en una longitud de onda aproximada de 8 a 14 micrones. Con un ángulo de apertura de 110° , permite una mejor precisión y simultáneamente protege al detector de falsas alarmas que resultan principalmente de la exposición a la luz solar u otras fuentes artificiales que pudieran estar cerca del dispositivo (Detector combinado: PIR Digital y detector de rotura de vidrio, s.f). El lente de Fresnel se ilustra en la figura 2.9.

Sensor PIR infrarrojo

Los dispositivos PIR están compuestos de un sensor electrónico denominado HC-SR501 que tiene la capacidad de captar la radiación infrarroja y convertirla en una señal eléctrica. La fabricación de este sensor se divide en dos elementos detectores independientes, los cuales son el sensor HC-SR501 y un circuito integrado BISS0001. Cuando el sensor logra captar una radiación infrarroja este a su vez genera una señal eléctrica y posteriormente ingresa al circuito integrado, que entre sus elementos internos más importantes con los que está fabricado se encuentran amplificadores operacionales que ayudan a generar una señal eléctrica a la salida del sensor. En la figura 2.10 se ilustra el funcionamiento del sensor HC-SR501.

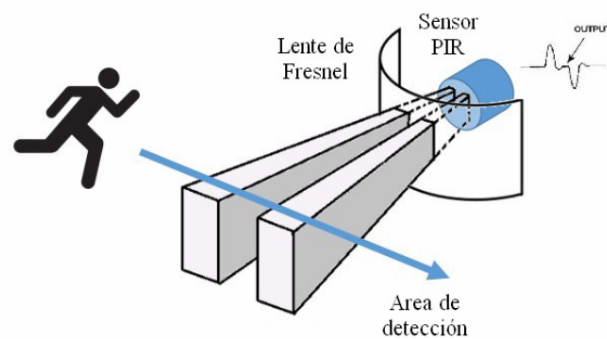


Figura 2.10: Funcionamiento del sensor HC-SR501

Ajustes y configuración del sensor

De acuerdo a las necesidades del usuario, el sensor PIR dispone de dos resistencias variables que pueden ser manipuladas para variar tanto la sensibilidad como la distancia de detección del sensor. El sentido de giro de los resistores se muestra en la figura 2.11, de esta manera el usuario puede controlar la sensibilidad y la distancia de detección (Sensor infrarrojo de movimiento PIR HC-SR501, 2017), (Sensor detector de movimiento (PIR), s.f.), (Vincent, 2017).

para fines prácticos en este trabajo se estableció que el sensor PIR captará la radiación infrarroja de las personas que se encuentren a una longitud menor o igual que 7m. y que se encuentren en un cono de detección de movimiento de 110° .

La programación asignada al microcontrolador permite que el sistema mantenga encendida la luminaria cuando se detecta radiación infrarroja, adicionalmente se considera que el tiempo de encendido de la luminaria dependerá de la última lectura que haya detectado el sensor y este se apagará después de 5 minutos una vez que ya no detecte la presencia de personas o animales.

El cono de apertura y el rango de detección se muestran en la figura 2.12.

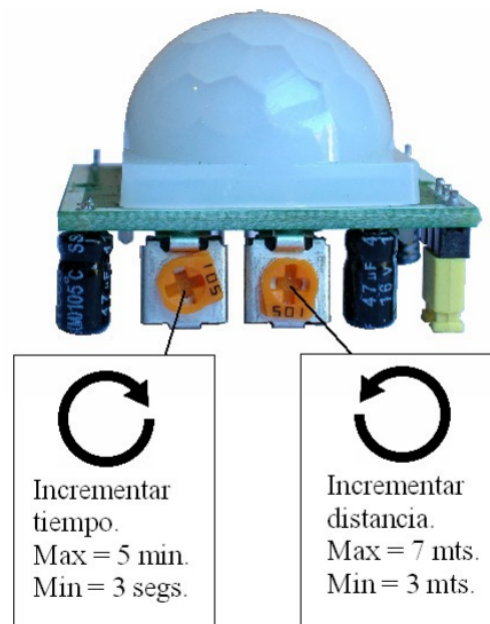


Figura 2.11: Ajuste de los resistores variables del sensor

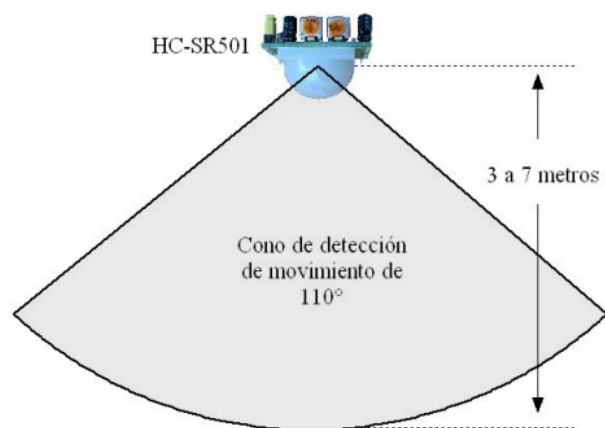


Figura 2.12: Rango de detección del sensor PIR

Las especificaciones del sensor se muestran en el cuadro 2.3:

Especificaciones técnicas del sensor HC-SR501	
Voltaje	5-12 VDC
Rango de distancia	3-7 metros
Ángulo de detección	Cono de 110 ^o
Vo	3.3V
Consumo promedio	<1mA
PIR	LH1778
Microcontrolador	BISS0001
Dimensiones:	3.2 x 2.4 x 1.8 cms

Cuadro 2.4: Especificaciones técnicas del sensor HC-SR501S

2.9.5. Sensor LDR

Para que exista un adecuado funcionamiento del sensor PIR, el dispositivo tiene que trabajar en conjunto con una fotoresistencia, esto con el propósito de no activar en el pin de salida una señal que encienda y apague la luminaria, por lo menos a ciertas horas del día que es cuando existe energía solar suficiente, como para que no esté encendida una luminaria. Al llevar a cabo esta instrucción estaremos disminuyendo el consumo de energía eléctrica.

Dependiendo de la energía solar que exista en el día el sensor de intensidad luminosa variará su resistencia eléctrica interna.

Por lo tanto, la variación que provoca la intensidad luminosa al sensor, se utiliza para ordenar al programa cuando el sensor PIR tiene que activar una señal que encienda la luminaria.

En consecuencia, un sensor LDR es un dispositivo electrónico que en función de los cambios de intensidad lumínica recibidos, su resistencia interna cambiará. En la figura 2.13 se muestra el sensor LDR utilizado.

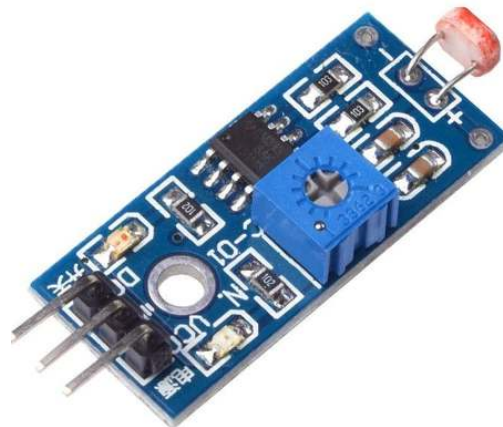


Figura 2.13: Sensor LDR

A través de estos cambios se puede medir el nivel de luz recibida en la entrada analógica

del microcontrolador.

Para que el resistor pueda captar los diferentes cambios de intensidad lumínica que inciden directamente en el componente, la fotocelda o fotoresistencia utiliza un material semiconductor fabricado con una mezcla orgánica de dos componentes denominados sulfuro de cadmio CdS. Por lo tanto, la cantidad de CdS que contiene es suficiente para que la fotocelda reaccione disminuyendo su resistencia eléctrica con mayor luz y viceversa. Algunas de las características principales del sensor LDR se pueden observar en el cuadro 2.5(Enrique, 2014).

Especificaciones técnicas del sensor LDR	
Resistencia (con luz)	1k Ohm
Resistencia (oscuridad)	10k Ohm
Material fotosensible	CdS (Sulfato de Sodio)
Sensibilidad ajustable con potenciómetro	
Voltaje	3.3-5 VDC

Cuadro 2.5: Especificaciones técnicas del sensor LDR

Típicamente, la función que lleva a cabo el semiconductor es absorber la mayor cantidad de fotones que inciden en la fotocelda. Este fenómeno de absorción, provoca que exista una excitación de los electrones logrando que estos pasen a la banda de conducción disminuyendo la resistencia eléctrica del componente.

Al quedar expuesto el sensor LDR a los cambios de intensidad luminosa bajo luz brillante, la resistencia eléctrica disminuye en un rango aproximado de 50 - 100 Ohm. En el caso contrario, cuando el sensor se encuentre en total o parcial oscuridad, se obtendrá una resistencia máxima de 1M Ohm, dependiendo de las condiciones del clima.

Por lo tanto, es importante hacer uso de un sensor LDR para tener un mejor control sobre un sistema, en el que se requiera trabajar sobre ciertas horas del día para aprovechar el nivel de luz, tanto en interiores como exteriores y reaccionar por ejemplo, encendiendo una luminaria, subiendo una persiana, o implementarlo en un seguidor solar(Llamas, 2015).

2.9.6. Módulo de Sensor DHT11

Uno de los objetivos principales de este trabajo es recopilar datos del entorno en donde se utilice el prototipo, por ejemplo, temperatura, presencia e intensidad luminosa. La adquisición de datos se lleva a cabo por uno de los módulos que ha sido llamado Luminaria, posteriormente los datos son enviados y almacenados en otro módulo de nombre Apagador.

Uno de los datos que es enviado al módulo Node MCU Apagador es el de temperatura. Esta función la lleva a cabo el módulo Node MCU Luminaria, que dispone de un sensor de temperatura y humedad.

El sensor de temperatura y humedad se denota por DHT11, para la medición de temperatura el sensor utiliza un componente de tipo resistivo y para la humedad requiere de un transductor interno del tipo NTC (Aziz, 2018).

Como se puede apreciar en la figura 2.14 el diseño del módulo tiene una señal de salida tipo digital, es de tamaño compacto y requiere de un bajo consumo de voltaje (3.3-5 VDC) para su funcionamiento (KIT-SEN-ACT: Kit de sensores y actuadores compatibles con Arduino, 2017). Para obtener más detalles del módulo DHT11, en el cuadro 2.6 se muestran las especificaciones técnicas.



Figura 2.14: Módulo sensor de temperatura y humedad DHT11.

Especificaciones técnicas del sensor DHT11	
Tiempo de sensado	2 segundos
Rango de medición de temperatura	-40° a 80°C
Voltaje	3.3-5 VDC
Rango de Medición	Humedad 20-90 % RH, Temperatura 0 ~50 °C
Precisión	Humedad + -5 % RH, Temperatura + -2 °C.
Resolución	Humedad 1 % RH, Temperatura 1 °C.

Cuadro 2.6: Especificaciones técnicas del sensor de temperatura y humedad DHT11

Los datos de temperatura y humedad que registra el módulo Node MCU (Luminaria), han sido configurados para que se muestren cada 10 segundos cuando son enviados al módulo Node MCU (Apagador), cada que se enciende o apaga de manera manual la luminaria.

La importancia de la recopilación de datos hoy en día es muy importante ya que nos conducen a tomar mejores decisiones que impactan en el entorno de las personas a través de la interacción entre dispositivo M2M y personas.

2.10. Tecnología WiFi

Al hablar sobre tecnología WiFi, se hace referencia a la WiFi Alliance, el cual es un organismo sin ánimo de lucro que garantiza y certifica la interoperabilidad de productos

inalámbricos, entre un amplio grupo de fabricantes, a través de las redes de área local en función de los estándares de la familia IEEE 802.11 (Castro, 2005).

Las redes inalámbricas de área local (WLAN - Wireless Local Area Network), permiten la interconexión de dispositivos electrónicos a través de ondas electromagnéticas, utilizando como medio de transmisión el aire. Las ondas de radio que genera esta tecnología se utilizan para establecer un enlace de comunicación entre diferentes sistemas de trabajo que contienen un adaptador inalámbrico, como: teléfonos inteligentes (smartphone), televisores inteligentes (smart tv), relojes inteligentes (smart watch), computadores, etc.

2.10.1. Características de la tecnología WiFi

A lo largo del tiempo desde que fue diseñada la primera red inalámbrica (WLAN) bajo el estándar IEEE 802.11 en el año 1991, que llevaba por nombre WaveLAN, esta llegó a alcanzar una tasa de transmisión de hasta 2 Mbps. Desde entonces, hemos sido testigos a lo largo del tiempo de la evolución y creación de esta norma (IEEE 802.11), la cual se basa en tecnología WiFi.

A continuación, se describen algunas de las versiones y características más importantes que han sido adquiridas para el surgimiento de este estándar (IEEE 802.11):

- IEEE 802.11b: se dio a conocer en el año 1999, opera en la banda de frecuencia de 2.4 GHz y a menudo era común referirse a esta norma como IEEE 802.11 HR (High Rate, Alta velocidad), porque alcanzaba y en la actualidad, una velocidad de 11 Mbps.
 - IEEE 802.11a: Es lanzada para ser operada en el mismo año que su antecesor IEEE 802.11b, se caracteriza porque opera en la banda de frecuencia de 5 GHz y permite una velocidad de hasta 54 Mbps, aunque como comenta Carballar en su libro: *WiFi lo que se necesita conocer*, con versiones propietarias como por ejemplo Netgear, la cual es una empresa especializada en la producción y venta de sistemas de redes, logró alcanzar velocidades de 72 y 108 Mbps respectivamente.
 - IEEE 802.11g: para el año 2003 se va ganando una mayor tasa de transmisión y el cambio es evidente, ya que este estándar dispone de una velocidad de hasta 54 Mbps, y opera en la banda de 2.4 GHz, aunque como menciona Carballar, la compañía US Robotics responsable del desarrollo de hardware para el área de comunicaciones, lograron alcanzar velocidades para sus productos de hasta 100 Mbps.
 - IEEE 802.11n: con la evolución y agregación de nuevas características, la versión de este estándar logra imponerse a sus antecesores, debido a que en el año 2009 se logra una mayor transferencia de datos. IEEE 802.11n ofrece una velocidad que se encuentra entre los 54 y 300 Mbps, puede trabajar en la banda de 2.4 GHz u opcionalmente en la de 5 GHz; por si fuera poco ofrece características de compatibilidad con las versiones anteriores (a, b y g).
-

- IEEE 802.11ac: lanzado en el año 2013, trabaja en la banda de 5GHz; este estándar sobrepasa considerablemente los límites de velocidad, ya que permite una tasa de transferencia que va desde los 450 Mbps hasta los 1.3 Gbps y es compatible con las versiones a y n. (Falcón, 2010)(Colmena y otros, 2017).
- IEEE 802.11ad: también conocido como "WiGig", quien ha dado un paso muy importante para los futuros consumidores que requieren transmitir grandes cantidades de información, a través de un medio de transmisión como lo es el aire, podría tener un gran éxito, pero existe poco conocimiento sobre la implementación de estas redes. En teoría, llegarían a ofrecer una velocidad de hasta 7 Gbps; rendimiento muy alto en distancias muy cortas; serían capaz de operar en las bandas de frecuencia de 2.4 GHz, 5 GHz y 60 GHz. (K. Nguyen y otros, 2017) (A. Masiukiewicz, 2013).

El desarrollo de esta serie de normas nos llevaría a pensar que existe una incompatibilidad entre cada una de las versiones expuestas, pero la mayoría es compatible con la versión IEEE 802.11b, por lo tanto, si un estándar tiene una transferencia de datos mayor como por ejemplo IEEE 802.11n, los dispositivos podrán interconectarse a una velocidad máxima de al menos 11 Mbps. En el caso de que 2 equipos terminales requieran de una mayor velocidad, ambos dispositivos tendrían que utilizar la misma versión, como por ejemplo el estándar IEEE 802.11ac, que ofrece una velocidad de hasta 1.3 Gbps(Carballar, 2010).

Como se ha mencionado, la frecuencia en que trabaja IEEE 802.11 es de 2.4 GHz, 5 GHz y 60 GHz, las cuales forman parte de las bandas de radio industriales, científicas y médicas

(ISM - Industrial, Scientific and Medical) que han sido reservadas internacionalmente y que no se habían destinado originalmente al uso de las telecomunicaciones(Valle, s.f.).

ISM se caracteriza porque pertenece a un rango de frecuencias de uso no comercial y cumplen con ciertas especificaciones para su funcionamiento, como es el bajo consumo de potencia transmitida. En la figura 2.15 se muestran las bandas ISM que pertenecen a la región 2 de la ITU en América, así como las utilizadas en la banda de frecuencia de 2.4 GHz (WiFi, Bluetooth o Zigbee) y 5 GHz para WiFi(Redes LPWAN, 2018).

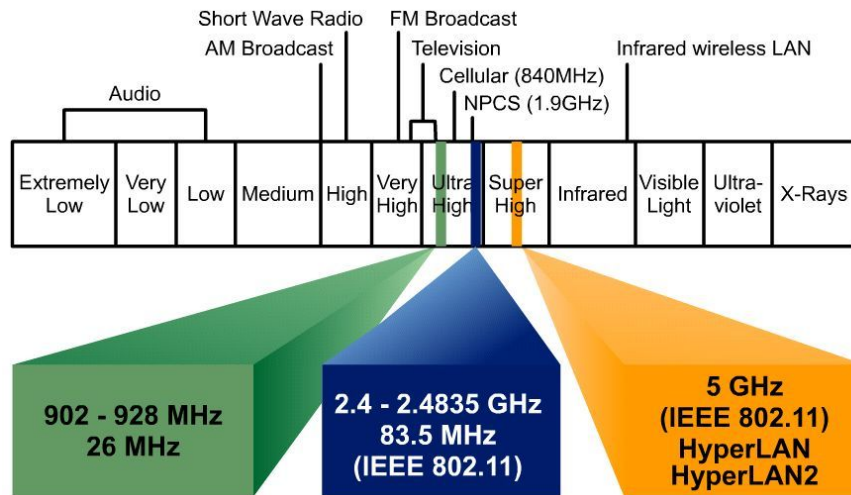


Figura 2.15: Bandas ISM.

En cuanto a alcance, las versiones de los estándares 802.11b y 802.11g ofrecen una distancia de hasta 100 m en un área libre de obstáculos, de tal manera que WiFi se ha convertido en una opción que puede reemplazar las redes cableadas en una red de área local.

La seguridad de los datos que ofrecen las versiones pertenecientes a las redes inalámbricas han tenido mejoras conforme han ido evolucionando. En sus inicios el estándar IEEE 802.11 utilizaba un cifrado WEP que en la actualidad ha quedado obsoleto porque sus mecanismos de seguridad eran vulnerables para la encriptación de los datos (Bernier y otros, 2005), como reemplazo surgió WPA el cual utiliza el protocolo TKIP, reforzando la seguridad que brindaba WEP, pero también presentó vulnerabilidades y finalmente fue reemplazado por WPA2 que ofrece un cifrado de tipo AES (Advanced Encryption Standard) que promete una mayor seguridad en una red de área local.

En cuanto a almacenamiento, dependiendo del diseño del microcontrolador, estos llegan a tener una capacidad en ROM de 512 bytes hasta 8 k bytes y en RAM de 20 a 512 bytes.

Para el caso de la memoria ROM, podemos encontrar diferentes versiones que son utilizadas en los microcontroladores.

Capítulo 3

Estado del arte

En este apartado se realizó un estudio relacionado con sistemas de monitoreo y automatización que utilizan tecnología IoT para el control de luminarias y conocer la similitud que estas guardan con este trabajo.

3.0.1. Technological Proposal for Controlling a Residential Lighting System

Una de las propuestas para controlar un sistema de iluminación residencial es el que proponen Lenin Coca y Cesar Hernández, (2018).

El objetivo de los autores es aprovechar el avance tecnológico que ha tenido la automatización y componentes electrónicos, para motivar a las personas el uso racional de la energía mediante el control de la iluminación en los hogares con tecnología que involucra el Internet de las Cosas.

Esta aplicación fue posible mediante la integración de sensores que son compatibles con el IDE de Arduino: sensor de movimiento PIR HC-SR501 y sensor de intensidad luminosa BH-1750 que es controlado a través de una placa NodeMCU. El esquema de comunicación se logra con la placa vía Wi-Fi a un teléfono móvil que utiliza una aplicación denominada App Inventor, basada en el sistema operativo Android. De esta manera, el usuario a través de la aplicación puede gestionarla programando el encendido y apagado de la luminaria o controlarla de manera manual o automática.

3.0.2. Sistema de monitorización domiciliaria

Uno de los temas más importantes en la materia de monitorización de espacios domésticos es el que propone Aitor Carricondo (2017).

El objetivo de este proyecto fue investigar los diversos sistemas que pueden implementarse para automatizar espacios domésticos, oficinas y edificios con procesos que involucran la domótica y están enfocados a tecnología IoT.

En este trabajo el autor Aicor Carricondo, desarrolla un sistema domótico en una maqueta simulando un espacio doméstico para tener un mejor control, confort y seguridad del entorno. Para ello utiliza una tarjeta NodeMCU que permite controlar diferentes sensores (intensidad

luminosa, temperatura, proximidad) y actuadores, como un servomotor. Gracias a la tecnología WiFi de la cual dispone esta tarjeta y los modos de operación con la que se puede trabajar (Acces Point y Station), el autor configura la tarjeta de tal manera que pueda ser utilizada como un Acces Point, e instala un servidor web para que otros dispositivos puedan acceder a el y gestionar la automatización de los lugares donde fue implementado el sistema. De esta manera, los espacios que se tienen para controlar son: la entrada principal de la casa que cuenta con un interruptor magnético en la puerta de acceso, para saber si está abierta o cerrada, un sensor de distancia en la puerta del garaje, para saber si hay algún vehículo cerca de la misma, control de luminaria para disminuir el consumo de energía eléctrica y control de temperatura.

Por lo que respecta a este trabajo, se basa en Software y Hardware Libre, ya que se hace uso del IDE de Arduino y NodeMCU.

3.0.3. Diseño e implementación de un prototipo de control de iluminación LED con aplicación móvil compatible con Arduino

Con Paez Montero y Sánchez Huertas (2017) se presenta una aplicación IoT relacionado con el control de iluminación LED. Los autores diseñaron e implemetaron un sistema de control de iluminación para luminarias LED de 12 Watts, el prototipo hace uso de un sensor de luminosidad (BH1750) y una aplicación móvil diseñada en la plataforma App Inventor que es compatible con Arduino. Esta última sirve como interfaz para permitir al usuario interactuar con el sistema de control a través de una comunicación inalámbrica hacia una placa NodeMCU.

Finalmente, los resultados que se obtienen del sistema son: regular la intensidad luminosa de la luminaria en un espacio confinado (sala, comedor, dormitorio), por medio de un sensor de luz que determina la intensidad luminosa necesaria para aplicar a la luminaria LED (modo manual) a través de la aplicación y por otro lado, el usuario tiene la opción de utilizar el dispositivo Android para controlar el encendido y apagado de la luminaria (modo automático).

3.0.4. Control remoto del encendido y apagado y monitorización del consumo de energía eléctrica de un sistema de iluminación

En (Tolozá Villalobos R.C, 2016), un proyecto que controla la iluminación (remota o localmente) se describe. Los autores diseñan el control de un sistema de iluminación y el monitoreo de las variables eléctricas de manera remota.

Los autores pretenden fomentar el uso racional de energía eléctrica debido a que en ocasiones el descuido de las luminarias provoca que estas queden encendidas cuando no es requerido. De esta manera para abordar dicha problemática, hacen uso de hardware y software libre, dispositivos para la medición eléctrica con protocolo de comunicación industrial modbus, el cual permite la extracción de datos y el uso de una plataforma para internet de las cosas para la gestión de los datos, control remoto a través de un teléfono inteligente que utiliza una

aplicación llamada Blynk que puede conectarse a una red inalámbrica, un sensor de proximidad para optimizar el sistema y el uso de un relé de estado sólido con el que se controla un circuito de corriente alterna con un circuito de corriente continua.

Finalmente los resultados obtenidos por los autores permitieron controlar a través de un teléfono móvil de manera remota una luminaria LED y el monitoreo periódico de las variables eléctricas consumidas por la luminaria

3.0.5. Análisis de resultados

Llevando a cabo una comparación de los trabajos investigados previamente, se obtuvo el cuadro 3.1 en el que se muestran las características relacionadas con la elaboración de este proyecto, obteniendo la siguiente información.

	3.1	3.2	3.3	3.4
Presencia	Si	Si	Si	Si
Temperatura	Si	No	No	No
WiFi	Si	No	No	Si
Intensidad luminosa	No	No	No	No
Aplicación para Android	Si	Si	No	Si

Cuadro 3.1: Comparación de las características relacionadas con la elaboración de este trabajo.

Lo que podemos observar es que somos una solución más que utilizan otros autores para disminuir el consumo de energía eléctrica en los hogares ya que en la mayoría de los casos utilizamos los mismos dispositivos. Por otro lado, lo que se obtuvo de este proyecto fue la transferencia de los datos a través de la capa 2 del modelo ISO/OSI para controlar las luminarias, otro aspecto muy importante es que se realizó el case para albergar el sistema electrónico y este puede replicarse para ampliar la red de dispositivos que pueden ser controlados en un espacio doméstico. Por lo tanto, cumplimos con una solución más que llevan a cabo otros autores que utilizan dispositivos con tecnología enfocada a IoT.

Capítulo 4

Análisis y diseño

El avance que ha tenido el desarrollo de sistemas enfocados al control de luminarias de manera automatizada en los hogares nos permite seleccionar el que mejor se adecuó a nuestras necesidades.

El objetivo de estas es disminuir el consumo de energía eléctrica, esto a través de lamparas ahorradoras con tecnología LED o sistemas como los presentados en las figuras 4.1 y 4.2.

De esta manera, en el mercado podemos encontrar algunos diseños que cuentan con algunas características similares a este trabajo, por ejemplo: sensor de presencia para el encendido de una luminaria de manera automática, tecnología IoT para el control de las luminarias a través de un Smartphone, el case para ser instalado de manera fácil y rápida, etc., esto sin dejar a un lado el enfoque de un dispositivo con tecnología IoT y el ahorro de energía eléctrica que estos dispositivos pueden ocasionar.

En la figura 4.1 se muestra un socket que contiene un sensor PIR que funciona de manera automática para encender y apagar una luminaria al regular las resistencias variables que contiene.



Figura 4.1: Socket Sensor De Movimiento Regulable.

La segunda figura 4.2 muestra el case de un socket que integra tecnología WiFi y el uso de un temporizador para poder controlar una luminaria a través de un Smartphone con conexión a Internet de manera remota.



Figura 4.2: Socket WiFi inteligente con temporizador y control remoto a través de un celular.

De esta manera, podemos concluir que somos una solución más al momento de pensar en tecnologías que nos permiten controlar una luminaria de manera manual o automática sin dejar a un lado el consumo de energía eléctrica que estos dispositivos puedan ocasionar.

4.0.1. Diseño

El diseño del sistema se llevó a cabo con la integración de un Sistema de Adquisición de Datos (SAD), un Sistema de Potencia (SP), un Sistema de Alimentación (SA) y finalmente un Sistema de Procesamiento (SPRO), dando como resultado la fabricación de un case que contiene la integración del sistema para controlar una luminaria y recopilar datos a través de diferentes sensores: temperatura, presencia e intensidad luminosa.

En la figura 4.3 se muestra el diagrama de conexión para el circuito final que está compuesto por los subsistemas que han sido mencionados.

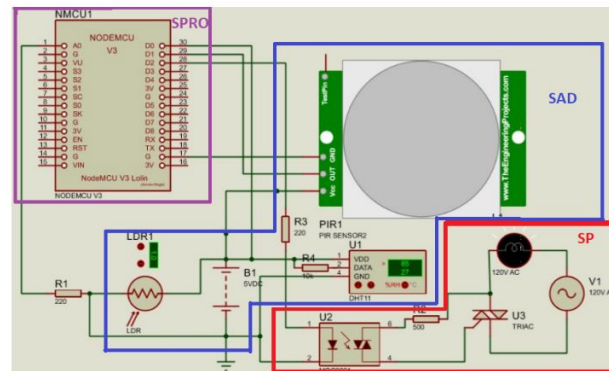


Figura 4.3: Sistema de adquisición de datos.

Para el Sistema de Potencia por ejemplo, podemos observar un opto acoplador (MOC 3021), que sirve como un interruptor para aislar dispositivos que utilizan corrientes y voltajes muy pequeños como es el caso del Node MCU.

Como se puede apreciar en el diagrama de conexión de la figura 4.3 los circuitos que quedaron aislados son los que pertenecen a la placa Node MCU y el de corriente alterna. También se utiliza un TRIAC que sirve como interruptor para conmutar la corriente alterna y pueda prender o apagar la luminaria.

Si se analiza el sistema de potencia (figura 4.4), su implementación controla un circuito de corriente alterna con un circuito de corriente continua de bajo voltaje cuando se aplica una señal digital (5V de DC) que proviene de la placa Node MCU, de lo contrario sin la implementación del sistema de potencia las pruebas para el encendido y apagado de la luminaria tendrían que haber sido reemplazadas por un LED.

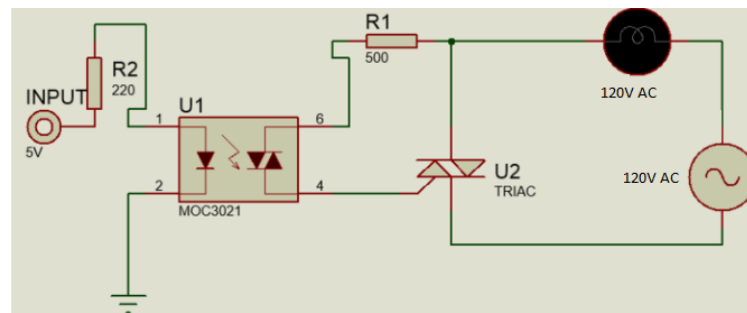


Figura 4.4: Circuito Optoacoplador TRIAC y MOC3021.

Finalmente, se dispone de 2 módulos Node MCU ESP8666 que se comunican utilizando un protocolo de comunicación inalámbrico llamado ESPNOW. Para probar su funcionamiento se conectaron los dispositivos a un socket de una sala y apagador de un espacio doméstico para realizar el intercambio de información y el control de una luminaria de manera automatizada. En la figura 4.5 se muestra la instalación del prototipo.



Figura 4.5: Captura de la instalación del prototipo en una sala de un espacio doméstico.

Pruebas realizadas en el prototipo:

- La primera prueba fue conectar el sistema denominado Luminaria al socket para que trabajara sin utilizar la placa Node MCU Apagador y este comenzará a operar cuando estuviera conectado.

Los resultados obtenidos hacen que la luminaria encienda de manera automática cuando la energía solar se encuentra antes y después de la hora azul como lo muestra la figura 4.6, que es cuando el sol todavía está oculto y se encuentra a -4° (Hora Dorada), y cuándo se está acercando al horizonte a -6 grados del centro geométrico del sol anunciando su salida (Crepúsculo Náutico)(Moreno, 2017).

Si se toma como referencia la hora azul, la iluminación en la que el sistema enciende una luminaria se encuentra en un rango de 150 - 300 lux que es la recomendada para un cuarto de estar(Niveles de iluminación, s.f.).

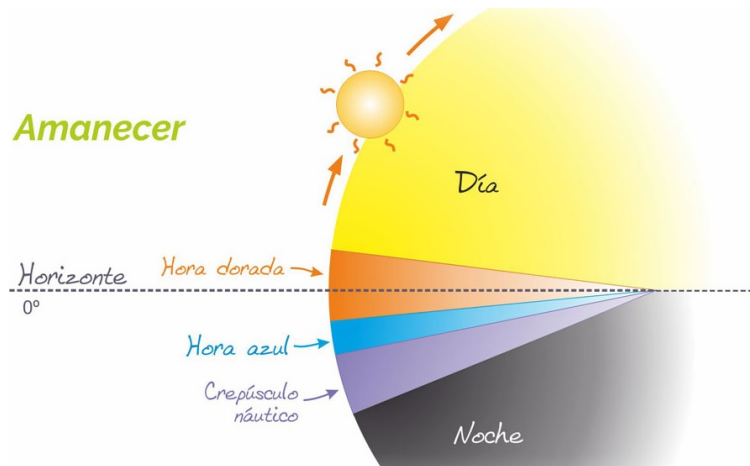


Figura 4.6: Representación de las horas del Sol.

- La segunda prueba fue conectar la placa apagador para poder controlar la luminaria, lo que se obtiene del sistema es apagar y encender la luminaria sin importar que exista suficiente energía solar gracias al uso de interrupciones programadas en la placa, ya que su función principal es interrumpir una tarea que se esté ejecutando para atender otra en cuestión, y posteriormente regresar en donde se quedó la ejecución del programa sin que este se vea afectado.

De esta manera, con las pruebas realizadas se comprobó el correcto funcionamiento del prototipo construido, a continuación en el capítulo desarrollo se explican la serie de pasos que se llevaron a cabo hasta llegar a la fase final del prototipo.

Desarrollo del proyecto

En este capítulo se explica cómo se llevó a cabo el desarrollo para obtener el prototipo final, la conexión de cada uno de los dispositivos utilizados y la programación para cada uno de los módulos ESP8266 con los que se realizaron pruebas(ESP-01, ESP-201, ESP-Node MCU).

Los pasos que se llevaron a cabo para el desarrollo del sistema fueron los siguientes:

5.1. Descarga e instalación del entorno de programación de Arduino: IDE

Para crear un entorno en el que se pudiera trabajar con los módulos ESP8266, en este trabajo se utilizó el software de código abierto denominado Arduino que se descargó desde la página www.arduino.cc. En el sitio oficial se eligió la versión más reciente 1.8.10 (Mayo de 2019) para windows 10.

A través de este software se pudieron crear los programas en conjunto con las bibliotecas para los módulos utilizados, probar los ejemplos, guardar, compilar y subir los programas en cada una de las placas.

5.2. Instalación del gestor de archivos del ESP8266 para el IDE de Arduino

5.2.1. Configuración del URL de soporte del procesador ESP8266

Una vez instalado el Software de Arduino, al abrir el IDE se agregó la URL que sirve para dar soporte al gestor de tarjetas que integra el procesador ESP8266.

A continuación, como se muestra en la figura 5.1 abrimos el IDE y seleccionamos: Archivo >Preferencias y copiamos el siguiente enlace en el Gestor de URLs adicionales de Tarjetas: http://arduino.esp8266.com/stable/package_esp8266com_index.json y terminamos dando OK(Dani, 2018).

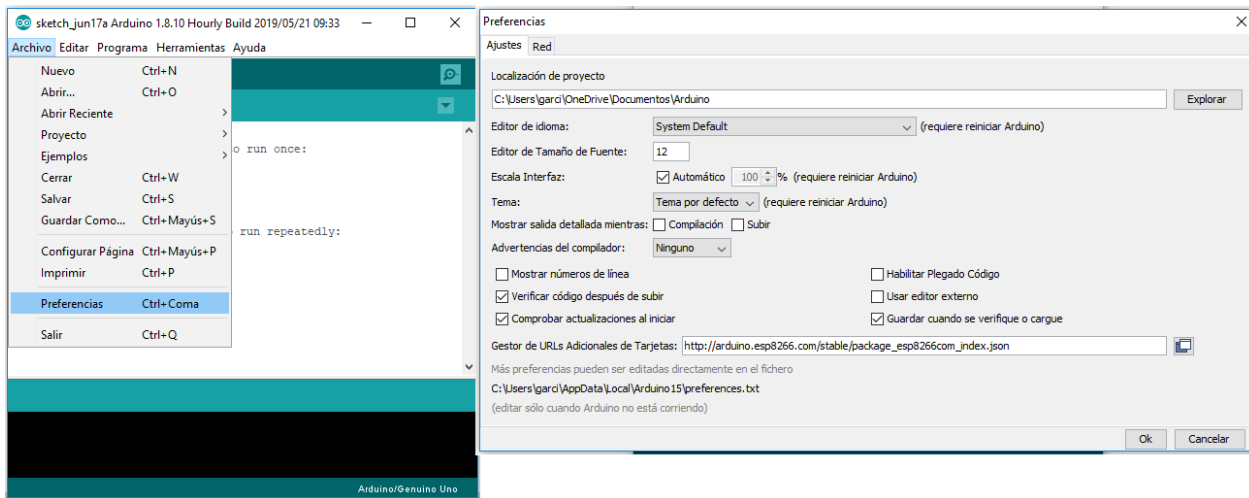


Figura 5.1: Configuración del URL de soporte del procesador ESP8266.

5.2.2. Instalación y configuración del gestor de archivos

Una vez configurado el URL se instala el gestor de tarjetas basadas en el procesador ESP8266. Para instalar el gestor de tarjetas desde el IDE seleccionamos la siguiente ruta: Herramientas >Placas >Gestor de tarjetas. El gestor de tarjetas que se utiliza es el esp8266 by ESP8266 Community. Seleccionamos la última versión, pulsamos Instalar y Cerrar (figura 5.2).

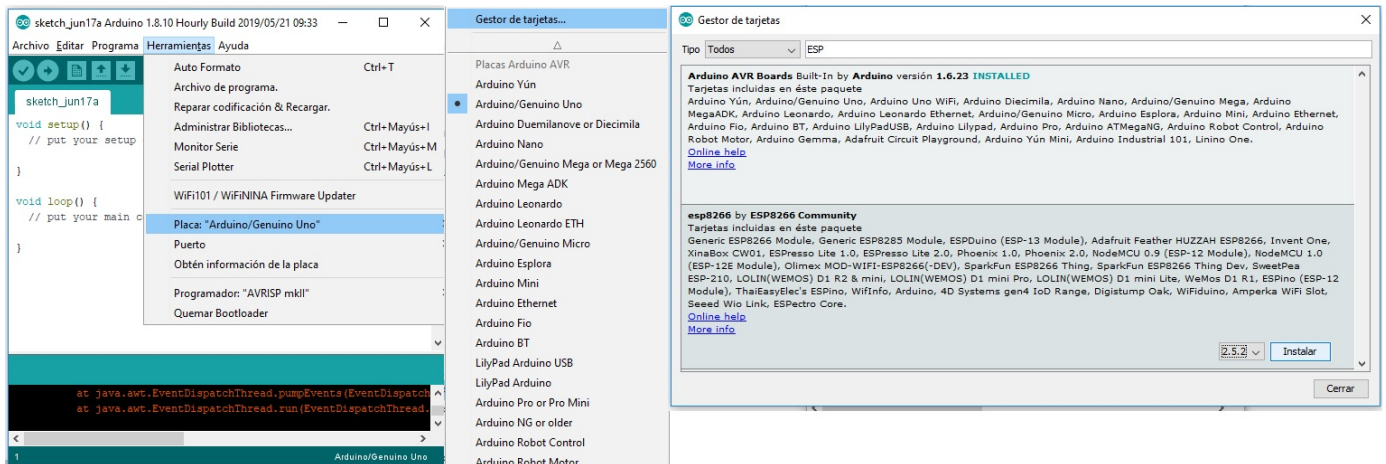


Figura 5.2: Instalación del gestor de tarjetas.

Para verificar que las versiones de las tarjetas han sido instaladas correctamente nos dirigimos nuevamente a la siguiente ruta: Herramientas >Placas >Gestor de tarjetas y veremos que han sido agregadas exitosamente para poder empezar a trabajar con el procesador ESP8266 (figura 5.3).

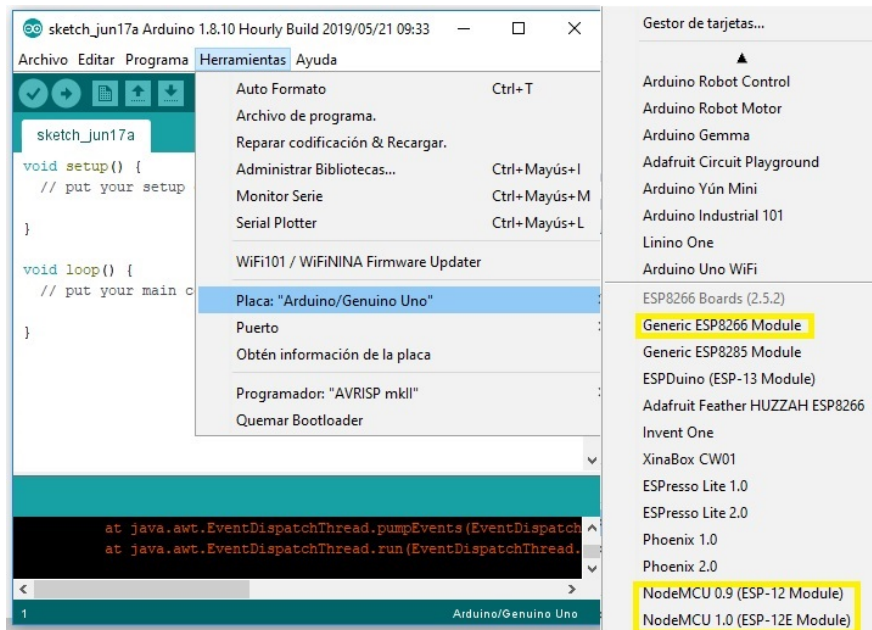


Figura 5.3: Instalación de tarjetas para el procesador ESP8266

Por último seleccionamos la versión del módulo que vamos a utilizar. La elección la haremos en función de los criterios que establezca el fabricante, por ejemplo:

- Para la versión ESP-01 seleccionamos Generic ESP8266 Module.
- Para la versión 1 seleccionamos NodeMCU 09 (ESP-12 Module).
- Para las versiones 2 y 3 seleccionamos NodeMCU 1.0 (ESP-12E Module).

Una vez que ha quedado preparado el entorno de programación, las versiones de los módulos compatibles con Arduino, específicamente la ESP01 requerirá para ser utilizada un convertidor serial con salida TTL para que pueda comunicarse con la computadora y puedan ser grabados los programas a través de un puerto serie, sin embargo, dicho convertidor también necesitará de un controlador CH340, del cual se hablará más adelante (Admin, 2015).

5.3. Instalación del controlador CH340 para Arduino en Windows 10

5.3.1. Descarga del controlador CH340

La descarga de este microcontrolador se realizó porque una de las primeras pruebas que se realizaron fueron con el módulo ESP-01, ya que este dispositivo requiere de un puerto serial USB con salida TTL para poder grabar las instrucciones a través de una computadora.

Desde la página del fabricante se descargó el driver CH340 que se encuentra disponible en el siguiente enlace:

<http://www.wch.cn/product/CH340.html>

tanto para Windows como para otros sistemas operativos, además de otras especificaciones que proporciona el fabricante sobre el chip.

5.4. Instalación del controlador Prolific PL2303

Otro aspecto importante antes de poder grabar un programa en el módulo ESP-01, es instalar el controlador para el chip, de lo contrario al conectar el dispositivo a la computadora no hubiese sido reconocido.

5.4.1. Descarga del controlador para el PL2303HX del convertidor USB to Serial

El controlador se obtuvo de la página oficial del fabricante Prolific que se encuentra disponible en el siguiente enlace:

<https://prolificusa.com/product/pl2303hx-rev-d-usb-serial-uart-bridge-controller/>

5.5. Pruebas de funcionamiento utilizando ESP-01

Al comenzar el desarrollo de este proyecto el primer contacto que se tuvo con las versiones de los módulos ESP8266 fue con la primera versión ESP-01. Este módulo al igual que los descritos más adelante (ESP-201 y Node MCU), nos brindan una conexión inalámbrica para que puedan conectarse a una red WLAN. Estos módulos pueden trabajar en 2 modos: como estación Wifi (Wifi Station) o como Punto de Acceso (Access Point), al trabajar como estación los módulo se conectan a una red Wifi presente en nuestro hogar, no obstante, el modo Access Point se usa si queremos crear una red propia en el chip y así conectarnos directamente (Módulo WiFi Serial ESP-01 ESP8266, 2019).

5.5.1. Conexión del ESP-01

El diagrama de conexión para el módulo ESP-01 requirió de un convertidor USB a UART con salida a TTL que utiliza un chip PL2303HX para poder trabajar con el. Este módulo logra convertir un puerto de comunicación USB a un puerto serial, utilizando niveles de voltaje TTL (0-5V) que son en su mayoría compatibles con las tarjetas microcontroladoras como Arduino, Pic, Raspberry Pi, etc.

Este dispositivo Posee leds que indican la transmisión y recepción de datos, salidas reguladas de voltaje de 3.3V y 5V(71). En la figura 5.4 se muestra el diagrama de conexión entre el ESP-01 y el módulo PL2303.

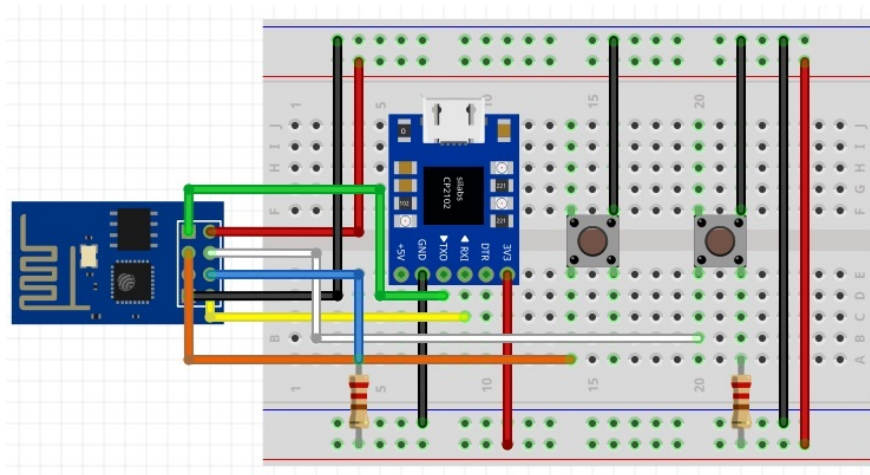


Figura 5.4: Diagrama de conexión para programar el firmware de ESP-01.

Para poder grabar un programa en el módulo ESP-01 los pines de propósito general (GPIO0 y GPIO2) necesitan tener una resistencia pull-up conectada y seleccionar el modo de operación del dispositivo con base en la siguiente tabla:

GPIO0	GPIO2	Modo	Comentarios
HIGH	HIGH	Flash	Arrancar desde SPI / Serial Peripheral Interface. (en funcionamiento normal).
LOW	HIGH	UART	Programar a través de UART Tx/Rx

Cuadro 5.1: Modo de operación para el módulo ESP-01.

5.6. Selección de la versión del módulo Generic ESP8266 Module

Teniendo en cuenta la información anterior, es posible utilizar el módulo ESP-01, pero antes verificamos que la conexión entre dispositivos y la instalación de los controladores haya sido de manera correcta dirigiéndonos a la siguiente ruta: Herramientas >Gestor de tarjetas>Generic ESP8266 module.

De ser este el caso, estaremos observando las versiones de los módulos dependiendo de la versión que necesitamos utilizar como se muestra en la figura 5.5. De esta manera podemos continuar sin ningún problema para poder grabar nuestro primer programa en el módulo ESP-01.

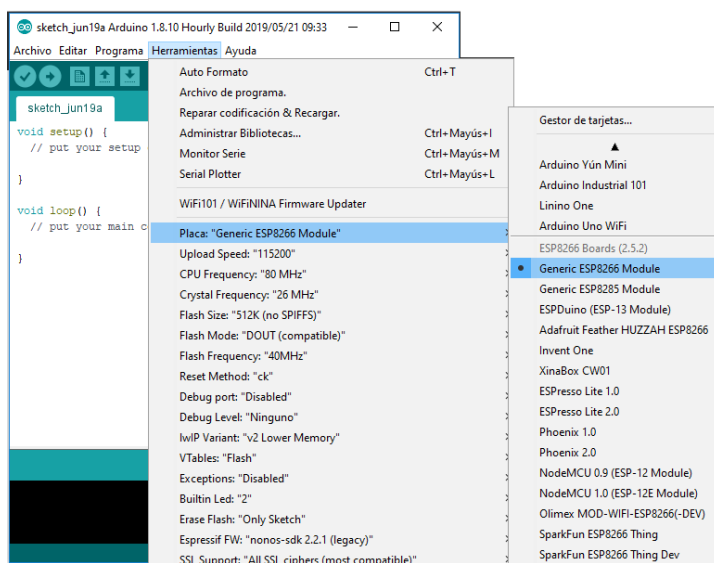


Figura 5.5: Selección de la placa ESP-01.

5.6.1. Blinky LED

El primer programa que se utilizó para comenzar a trabajar con la placa fue el tradicional Hola Mundo!, que en Arduino consiste en hacer parpadear un LED. Para este propósito se tomó un programa de ejemplo que viene contenido en el IDE de Arduino, al igual que muchas bibliotecas que son de gran utilidad y están disponibles en el propio IDE.

Como primer paso para poder trabajar con cada una de las versiones ESP8266 que serán mencionadas más adelante, es necesario seleccionar en el IDE de Arduino el puerto donde se tiene conectada la placa, por ejemplo para esta tarjeta: ESP-01 >Herramientas >Puerto. El procedimiento que se utiliza para que las demás tarjetas puedan ser reconocidas es el mismo que se lleva a cabo en este ejemplo.

Una vez seleccionado el puerto vamos a cargar el ejemplo del programah que hace parpadear el LED: Archivo >Ejemplos >01.Basics >Blink (Figura5.6).

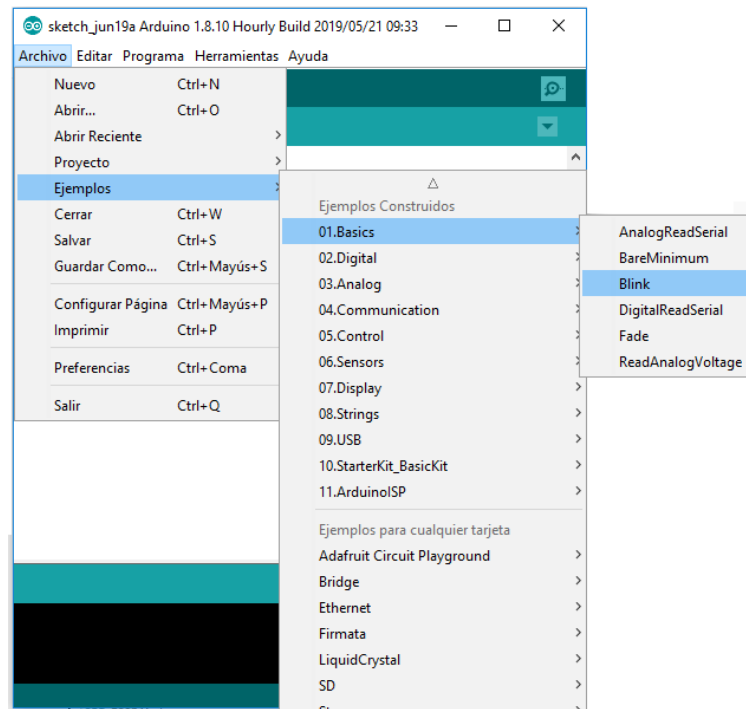


Figura 5.6: Subiendo el primer programa contenido en el IDE de Arduino: Blink

Después de elegir el programa de ejemplo en automático se despliega una nueva ventana que contiene el código del programa para hacer parpadear el LED que viene integrado en la placa ESP-01 (Figura 5.7). Por último, pulsamos subir para verificar y compilar el programa al módulo como se muestra en la figura 5.7.

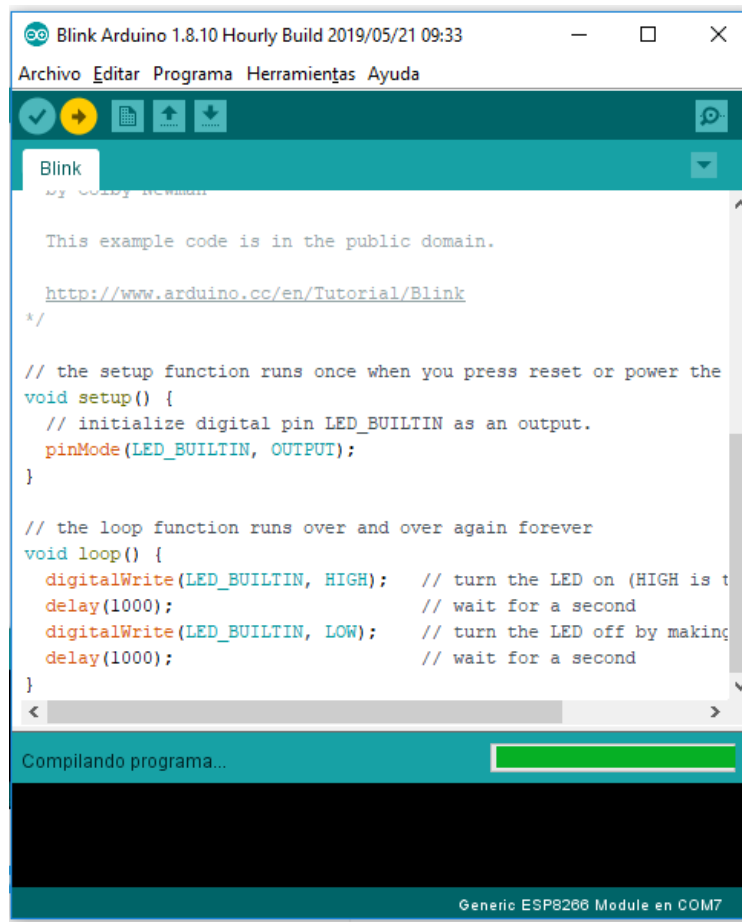


Figura 5.7: Compilando programa de ejemplo contenido en el IDE de Arduino: Blink.

El resultado obtenido funciona y se verá reflejado en el LED del módulo que va cambiando el estado del pin entre LOW (apagado) y HIGH (encendido) con un retardo de 1000 milisegundos.

Continuando con las pruebas en el módulo ESP-01 fueron desarrolladas las siguientes:

5.6.2. Servidor web para encender y apagar un LED

Una de las ventajas que ofrece el módulo ESP-01 es poder configurarlo como un servidor web que nos permite encender y apagar un LED. De esta manera, el cliente será nuestra computadora o cualquier dispositivo como un smartphone o tablet que contenga un web browser. En la página web aparecerán los botones de encendido y apagado que se encargarán de realizar las peticiones al servidor. Los pasos para la configuración fueron los siguientes:

- Configurar el módulo.

Para grabar el chip ESP8266 se toma como referencia el cuadro 5.1 para poder cargar un nuevo programa.

- Cargar el programa, el cual se encuentra disponible en el siguiente enlace:

https://github.com/jaimelaborda/Planta-Twittera/blob/master/ejemplos/esp8266_led_server/es

La biblioteca que utiliza el programa es la ESP8266WiFi.h que nos permitió controlar el módulo ESP8266.

- Configuración de las funciones del programa.

Las funciones que han sido configuradas en el programa para conectarnos a la red WiFi del ESP-01 son las siguientes:

```
const char* ssid = "ProyectoIoT";  
const char* password = "ProyectoIoT";
```

Una vez que ha sido cargado el programa al chip esp8266 nos dirigimos al monitor serie de Arduino para obtener la dirección IP que va a ser la que nos permita conectarnos al servidor web por medio del protocolo http.

Posteriormente se Ingresa al navegador web de nuestro dispositivo en cuestión, escribimos la dirección que se ha obtenido en el monitor serie de Arduino y obtendremos como resultado una página en el que se muestra un menú con los botones para controlar el encendido y apagado de nuestro LED como se muestra en la figura 5.8.



Figura 5.8: Encendido y apagado de un LED desde una página web.

Como podemos darnos cuenta sin antes adentrarnos a fondo en la familia de módulos ESP8266, el ESP-01 es una excelente opción para trabajar con proyectos enfocados a IoT, como la prueba realizada anteriormente para poder controlar un LED desde una página web.

El único problema con el que nos enfrentamos al trabajar con este módulo ESP-01 fueron los pines GPIO, ya que sólo cuenta con 2, de este modo nos limitó la implementación del

prototipo, porque en nuestro caso se requirieron de más pines para poder conectar diferentes dispositivos.

5.7. Pruebas de Funcionamiento utilizando el ESP201

Para poder hacer uso del ESP-201 se tiene que utilizar el siguiente diagrama de conexión de la figura 5.9 para poder grabar la memoria y cargar los programas que a continuación se explican.

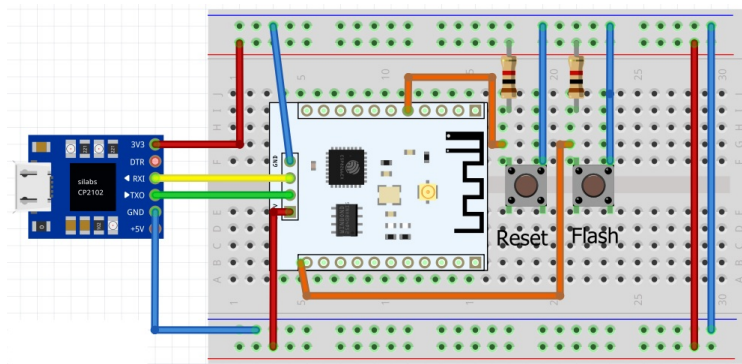


Figura 5.9: Diagrama de conexión para grabar el firmware de ESP201.

5.7.1. Resistencias pull up y pull down

Para encender un LED se utilizó un circuito con un push button configurado en modo pull down. Esta configuración al igual que la pull up utilizan valores lógicos (LOW, HIGH), en un pin o entrada de un circuito cuando se encuentra en estado de reposo. En la figura 5.10 se muestran las configuraciones para cada una de las resistencias pull up y pull down con una resistencia de 10 k Ohm que se utilizan para evitar un corto circuito y eliminar la influencia de factores externos como lo es el ruido eléctrico.

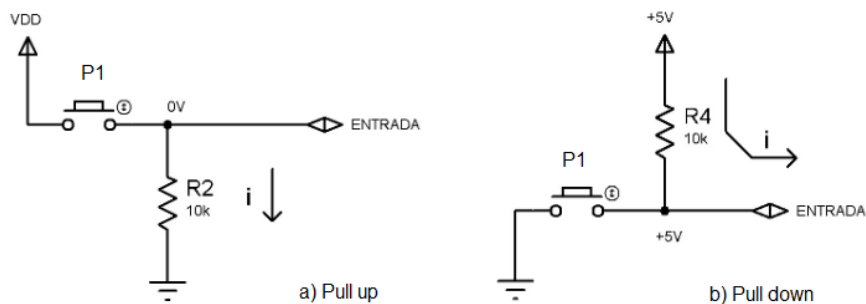


Figura 5.10: Configuración de resistencias Pull Down y Pull Up.

5.7.2. Resistencia pull down

En la configuración pull down, cuando el circuito está en reposo como se muestra en la figura 5.10, el voltaje en la resistencia es prácticamente 0V (LOW), en cambio si se pulsa P1, dejará pasar la corriente y se obtendrá un voltaje de 5V (HIGH).

5.7.3. Resistencia pull up

En la configuración pull up, cuando el circuito está en reposo (P1 sin pulsar) la caída de tensión es de 5V (HIGH) y cuando el pulsador es presionado, la corriente circula por la resistencia, luego por el pulsador y de esta manera se obtiene en la entrada del ESP201 un voltaje de 0V (Resistencias Pull-Up y Pull-Down, s.f.).

5.8. Encendido y apagado de un LED con push button

Para encender un LED con una resistencia pull up es necesario eliminar el efecto rebote que crea la construcción de este mismo componente al ser utilizado. Por lo tanto, es necesario crear un programa que nos permita agregar un retardo de 40 ms cada que es presionado el pulsador para disminuir tal efecto. El programa utilizado y al que se puede recurrir se encuentra disponible en el apartado anexo 7.4, que ayuda a leer el estado del pulsador cuando es presionado.

En la figura 5.11 se muestra el diagrama de conexión para encender un LED con un pulsador utilizando la configuración pull down.

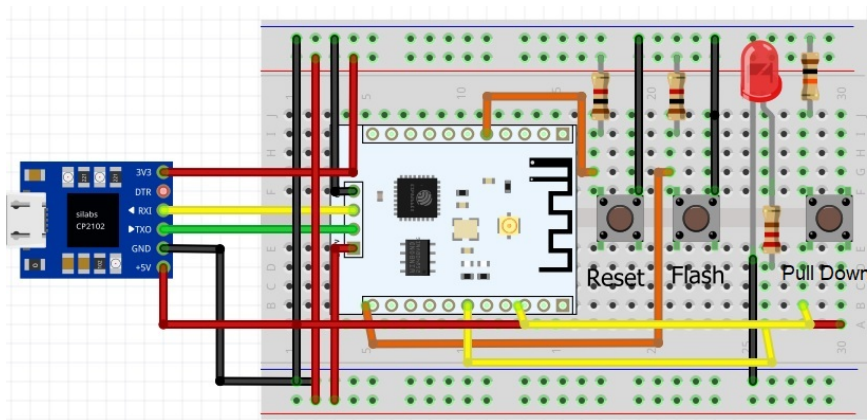


Figura 5.11: Encendido y apagado de un led con resistencia configurada en modo pull down.

5.8.1. Encendido de un LED con Sensor PIR

El diagrama del circuito que se presenta en la figura 5.12 permite detectar cambios de de temperatura emanados por los cuerpos u objetos que se encuentran por arriba del cero

absoluto, específicamente aquellos que pertenecen a personas o animales.

De esta manera lo que se logra con el sensor infrarojo es detectar un cambio de temperatura generando una señal digital en la salida del dispositivo y encender un LED. El circuito utiliza un voltaje de alimentación de 5V DC que es tomado de uno de los pines disponibles del convertidor serial USB a TTL ya que brinda dos voltajes, tanto de 5v y 3.3.v DC.

En cuanto al programa este está creado para que el LED quede encendido por un periodo de 5 minutos y finalmente se apaga cuando no detecte la temperatura de personas o animales que se encuentren en el cono de detección.

En el IDE de arduino en la ruta Herramientas >Monitor serie se puede visualizar un mensaje que indica cuando está activo el sensor. De esta manera la aplicación es utilizada posteriormente reemplazando el LED por una lámpara fabricada de la misma tecnología en conjunto con un circuito de potencia como se hizo en este proyecto.

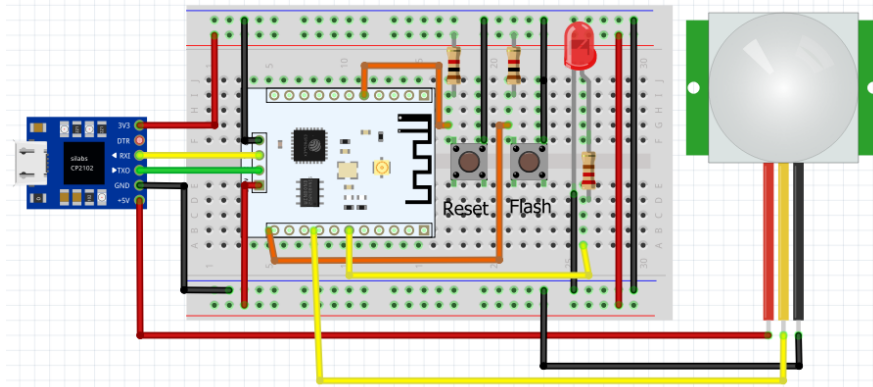


Figura 5.12: Circuito Sensor de movimiento

5.8.2. Obtención de datos de temperatura y humedad con sensor DHT11

Existen en el mercado diferentes tipos de sensores que nos ayudan a medir la temperatura, humedad o ambos valores al mismo tiempo como lo hace el sensor DHT11 que se ilustra en la figura 5.13.

Se alimenta con un voltaje de 5V y el chip integrado que contiene logra hacer la conversión de temperatura y humedad, dando como resultado un valor de medida digital sin necesidad de utilizar un pin analógico (Macho, 2018).

La biblioteca que fue utilizada en este dispositivo se llama DHT11 y se encuentra disponible en el siguiente enlace:

<https://www.prometec.net/sensores-dht11/>

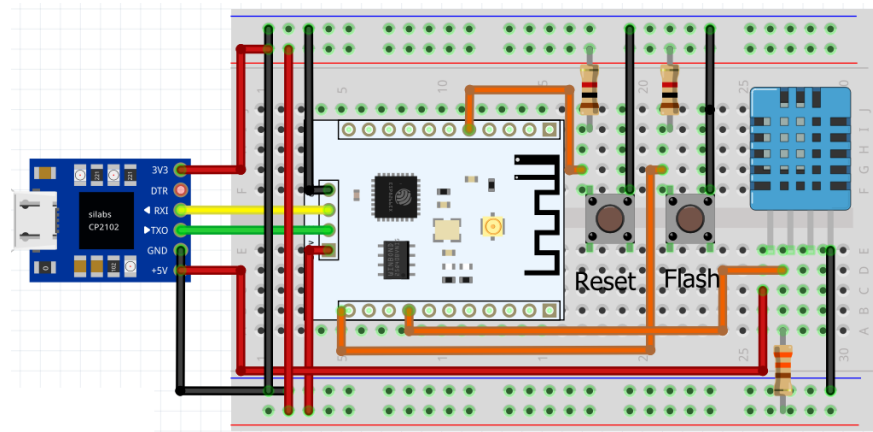


Figura 5.13: Medición de temperatura y humedad con ESP201 y sensor DHT11.

Una vez importada la biblioteca podemos observar que tenemos disponible la función para poder operar el sensor DHT11 y sólo hay que definir el pin del sensor donde ha sido conectado el dispositivo.

5.8.3. Funcionamiento de Relay con ESP-201

Un componente de gran utilidad que se puede utilizar para controlar circuitos de voltaje y corriente superior al que admite Arduino, como el control de luminarias se puede lograr con un relay o un transistor.

El funcionamiento del relay se basa en un fenómeno electromagnético de corriente eléctrica que fluye a través de una bobina y produce un campo magnético que logra magnetizar un núcleo de hierro de ferrita. El núcleo al quedar inducido provoca un movimiento que fuerza a un contacto a unirse, provocando un flujo de corriente eléctrica para encender una luminaria, bomba de agua, etc. Cuando la fuente de polarización del relay se ve interrumpida los contactos se vuelven a separar. En la figura 5.14 se muestra el diagrama de conexión para encender una bombilla eléctrica utilizando un relay.

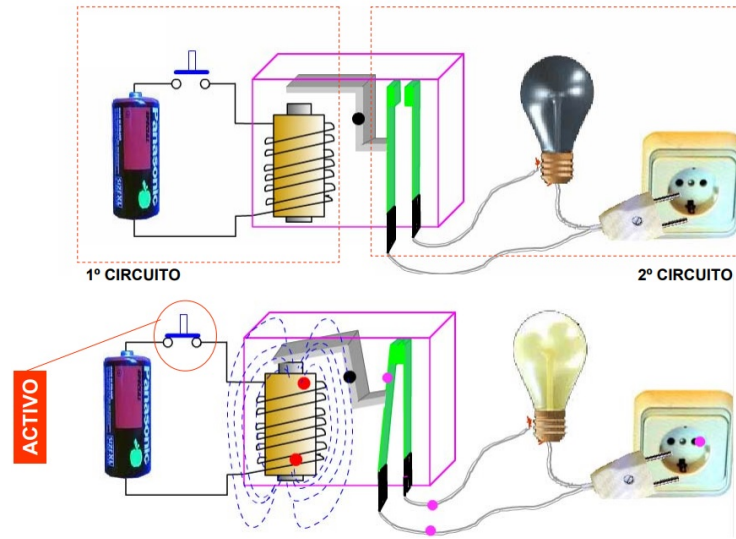


Figura 5.14: Circuito Relay para encender Bombilla.

El relé que fue utilizado se alimenta de una fuente de voltaje de 5V DC, la cual fue tomada de uno de los pines del convertidor Serial USB a TTL. Para poder encender la bombilla se utilizó el programa de ejemplo Hola Mundo, dando un tiempo de encendido y apagado a la bombilla de 10 segundos, que fue tomada de una señal digital proveniente del ESP-201 (GPIO2).

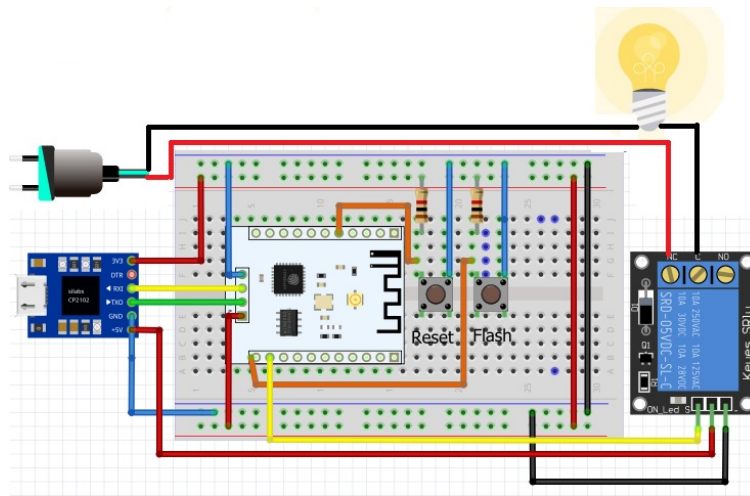


Figura 5.15: Circuito ESP201 con Relay

5.9. Node MCU

La placa con procesador ESP8266 (Node MCU) puede ser configurada por software de tres maneras distintas: Wireless Local Area Network – WLAN, punto de acceso Enabled

Access Point – SoftAP o simplemente AP.

De las tres configuraciones mencionadas se optó primero por el punto de acceso WiFi, donde el Node MCU difunde una red WLAN con su identificador SSID (Service Set Identifier) y su contraseña, para que los dispositivos clientes puedan conectarse a esta red.

5.10. Ejemplos con Node MCU

5.10.1. Servidor web modo Acces Point con procesador ESP8266 (Node MCU)

Este servidor es utilizado para controlar el encendido y apagado de un LED a través de los dispositivos clientes que se conecten, como PC's, Smart Phones, Tablets, etc. La red que utiliza este servidor para permitir la conexión a cada uno de los dispositivos es una red privada de clase C:

192.168.4.0

y puede ser cambiada utilizando la instrucción **WiFi.softAPConfig(local_ip, gateway, subnet)**. Los tres parámetros que la componen se definen a continuación:

- **local_ip**: dirección IP del Access Point.
- **gateway**: es la puerta de enlace.
- **subnet**: dirección IP de la subred.

Otra instrucción muy importante que utiliza la biblioteca ESP8266WiFi.h para trabajar en modo AP es la siguiente:

WiFi.softAP(ssid, password, channel, hidden)

A continuación se definen cada uno de los parámetros que componen la instrucción:

- **SSID**: es el nombre de la red que puede ser configurado y tiene un rango de entre 8 y 63 caracteres.
 - **Password**: es un parámetro opcional que puede ser o no configurado para que los clientes puedan acceder a la red. Al no incluir una contraseña al Access Point la red queda abierta para los clientes. El tipo de seguridad que utiliza para la encriptación de los datos se basa en un algoritmo de cifrado WPA2-PSK que debe tener un mínimo de 8 caracteres.
 - **Channel**: es un parámetro opcional, que es utilizado para establecer el canal de comunicación WiFi. Los canales disponibles están conformados del 1 al 13, y por defecto el canal 1 es el que viene configurado.
-

- Hidden: es también un parámetro opcional, que permite ver u ocultar a los usuarios el identificador de la red. Por ejemplo 1 permite ver la red y 0 la oculta(Daniel, 2016).

En la figura 5.16 se puede observar el diagrama de conexión para el Node MCU.

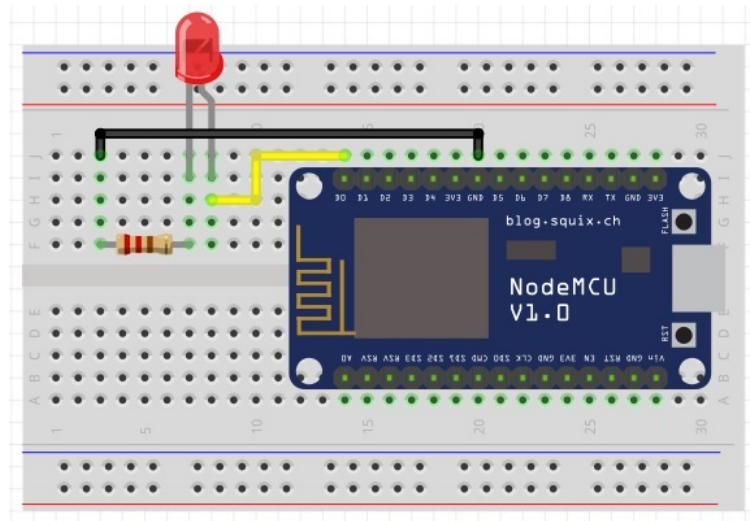


Figura 5.16: Encendido y apagado de un LED mediante un servidor web.

Para poder encender y apagar el LED los dispositivos clientes tendrán que buscar las redes visibles e ingresar a la que lleva por nombre **ProyectoIoT** con la misma contraseña como hace referencia la figura 5.17, y desde un web browser introducir la dirección IP:

<http://192.168.4.1>



Figura 5.17: Red WiFi visible llamada ProyectoIoT.

Una vez ingresada la dirección IP en el navegador se desplegará una página web que contiene las pestañas para poder encender y apagar un LED mediante el control de un servidor web.

El programa utilizado para esta aplicación se encuentra disponible en el apartado anexo 7.7(Daniel, 2016).

5.10.2. Protocolo ESP-NOW para establecer una comunicación inalámbrica entre módulos Node MCU-ESP8266

Cada uno de los dispositivos Wifi ESP8266 cuenta con dos direcciones MAC que posibilitan las conexiones vía WiFi que corresponden a cada uno de los modos que pueden ser configurados en cada placa: Access Point y Station (AP MAC o STA MAC). La configuración depende de las necesidades del proyecto, por ejemplo para este caso se tiene que conocer la dirección MAC (modo Station) de uno de los dispositivos, el cual debe ser configurado en modo esclavo para poder recibir los datos y encender un LED.

En la sección anexos (6.8 - 6.10) se encuentran disponibles los programas que permiten obtener las direcciones MAC de los dispositivos y el emparejamiento de los mismos mediante una comunicación maestro y esclavo.

El uso de direcciones MAC en esta aplicación se lleva a cabo para hacer una comunicación a nivel de capa 2 (del modelo ISO/OSI), por medio de un protocolo privado perteneciente a la empresa Espresif denominado ESP-NOW.

El dispositivo maestro es aquel que toma lecturas de voltaje que deja pasar un potenciómetro que está conectado directamente a la entrada analógica de la placa. La función de dicho dispositivo es establecer a través del protocolo ESP-NOW un enlace de comunicación que envía los datos a otro dispositivo que está configurado en modo esclavo. Por último, el circuito esclavo es quien recibe las lecturas de voltaje que registra el potenciómetro en un tiempo dado en milisegundos, y en función de cada una de estas medidas dependerá la intensidad de iluminación del LED, el cual está directamente conectado a una salida PWM de la placa.

Para la transmisión de la información se hace uso de la siguiente función:

```
esp_now_send(uint8_t *da, uint8_t *data, uint8_t *len)
```

A continuación, se describen los parámetros que utiliza:

- `uint8_t *da`: array de la dirección MAC del par al que se envía el paquete de datos. Si la dirección es NULL los datos se envían a todas las direcciones de la Tabla de Comunicación.
- `uint8_t *data`: array con el paquete de datos a enviar.
- `uint8_t len`: longitud del array del paquete de datos.

Por lo tanto, en esta aplicación lo que se logra es enviar un paquete entre dos dispositivos Node MCU que están configurados en modo maestro y esclavo para encender un LED con la ayuda de un potenciómetro.

El programa que se utilizó para esta aplicación está disponible en el apartado anexo (7.9 y 7.10)(Daniel, 2016).

El diagrama de conexión para el circuito maestro se muestra a continuación y para el circuito esclavo, se retoma el que se utilizó en la figura 5.16.

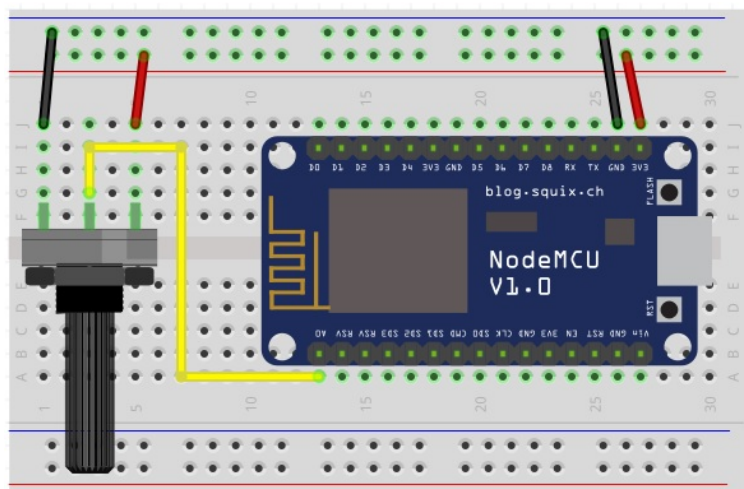


Figura 5.18: Circuito Node MCU Maestro.

Algo importante de mencionar al hacer uso del protocolo ESP-NOW, es verificar en el gestor de tarjetas para procesadores ESP, que se encuentra disponible en el IDE de Arduino si se tiene instalada la versión 2.3.0, si es el caso, es necesario editar el fichero:

`Library/Arduino15/packages/esp8266/hardware/esp8266/2.1.0/platform.txt`

y buscar *compiler.c.elf.libs* agregando al final de la línea *-lespnow*. De no hacerlo se obtendrá un mensaje de error: *espnow library not found* (79).

5.10.3. Medición del nivel de luz con fotoresistencia LDR usando Node MCU

Las resistencias LDR tiene la propiedad de cambiar su resistencia eléctrica en función de la cantidad de energía solar que perciben. Así, cuanto más iluminada esté la estancia en la que se halle dicho elemento, menos resistencia presentará, y lo contrario sucederá cuando el lugar esté menos iluminado.

El circuito implementado está compuesto por un LED una resistencia LDR y un Node MCU que enciende el LED cuando la intensidad de luz baja de un nivel determinado. El diagrama del circuito se muestra en la figura 5.19.

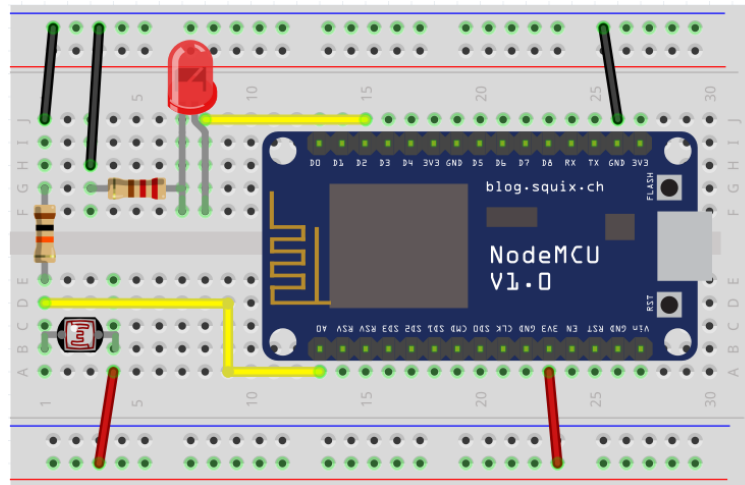


Figura 5.19: Nivel de luz con sensor LDR.

5.11. Diagramas de flujo para prototipo

Una vez realizadas las pruebas anteriores con cada uno de los dispositivos y tarjetas ESP8266, se muestran a continuación los diagramas de flujo que se utilizaron para cumplir con el objetivo de este trabajo: Diseñar y construir un sistema IoT enfocado al control de luminarias utilizando módulos Node MCU. Los diagramas de flujo se relacionan con los dispositivos que han sido probados, uniéndolos para que formen parte de un mismo programa

y finalmente grabarlos en los módulos que están configurados como maestro - esclavo y viceversa, para que puedan comunicarse ejecutando cada una de las instrucciones declaradas en los módulos.

Una particularidad de los tres diagramas de flujo que se ilustran a continuación, es que comparten las mismas variables que se utilizan en ambos módulos, porque éstas contienen la misma información para el funcionamiento del prototipo.

El diagrama de flujo de la figura 5.20 se emplea para el Node MCU que lleva por nombre Apagador. En el se muestra una estructura de decisión do - while (haz - mientras) llamada Push Button, utilizando como referencia el numero 1 para indicar que ha sido presionado el nombre de la estructura y envía un dato (mensaje), que enciende una luminaria, y al mismo tiempo el Node MCU Apagador recibe un valor de temperatura con el módulo que está emparejado de nombre Luminaria.

De esta manera, se puede apreciar como la comunicación entre los módulos está configurada para que ambos se comporten como maestro y esclavo al mismo tiempo, porque ambos se estarán comunicando y actualizando las variables que están definidas en cada uno de los programas.

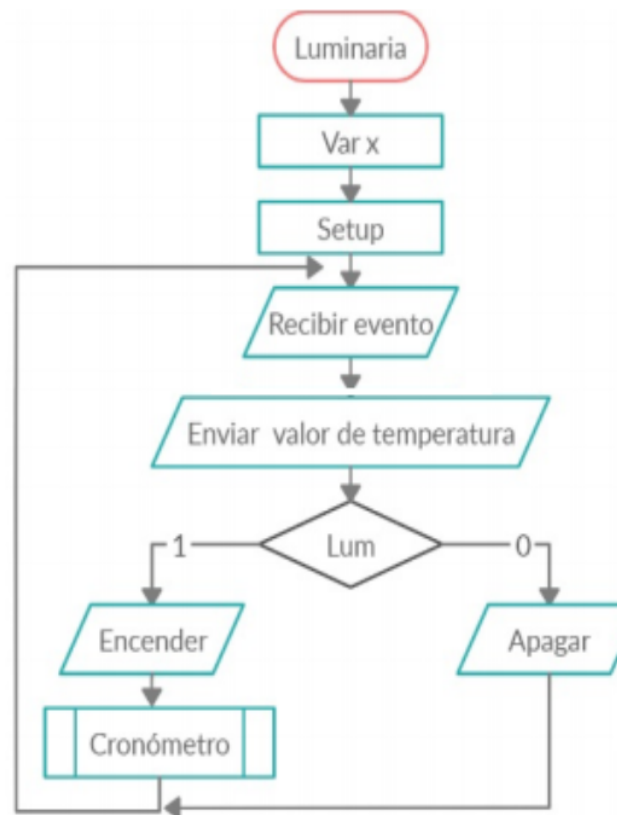


Figura 5.20: Diagrama de flujo apagador.

Para el diagrama de flujo de la figura 5.21, fue necesario hacer uso de interrupciones en el programa para que el sistema electrónico funcionara de manera óptima, logrando que res-

ponda cuando una persona presione un pulsador (Push Button) que forma parte del sistema Node MCU (Apagador), y si el usuario en algún momento dejara encendida la luminaria por un tiempo máximo de 5 minutos el Node MCU (Luminaria) ejecute una instrucción que apaga la bombilla de manera automática.

Otro aspecto importante del sistema que controla la luminaria, es que está pensado para encenderse de manera automática cuando detecte presencia con la ayuda de un sensor PIR y en conjunto con una fotoresistencia sólo realiza la tarea definida cuando no haya suficiente energía solar, sin necesidad de acercarse al lugar donde se encuentra el apagador que contiene el Push Button. En el caso de haber suficiente energía solar y el sistema detecte presencia no se encenderá la luminaria; en caso de querer encender de manera manual y exista suficiente energía solar el sistema lo permitirá y cuando no detecte presencia apagará la luminaria de manera automática.

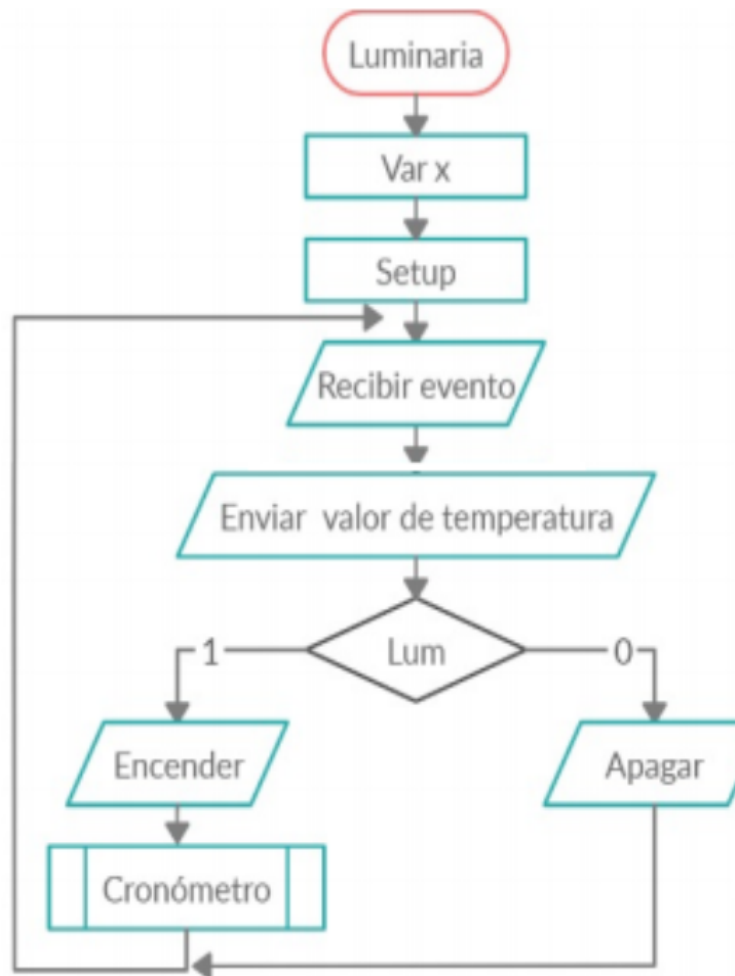


Figura 5.21: Diagrama de flujo para luminaria.

En la figura 5.22 se muestra el diagrama de flujo que pertenece a la interrupción denominada sensor de presencia en la que las estructuras de decisión if else para SP (Sensor

de Presencia) y los números que utilizan significan: 0 - No existe presencia, y 1 - Detecta presencia. Para el caso de SL (Sensor de Luz) se indica:

- 0 - No hay suficiente energía solar
- 1 - Hay suficiente energía solar

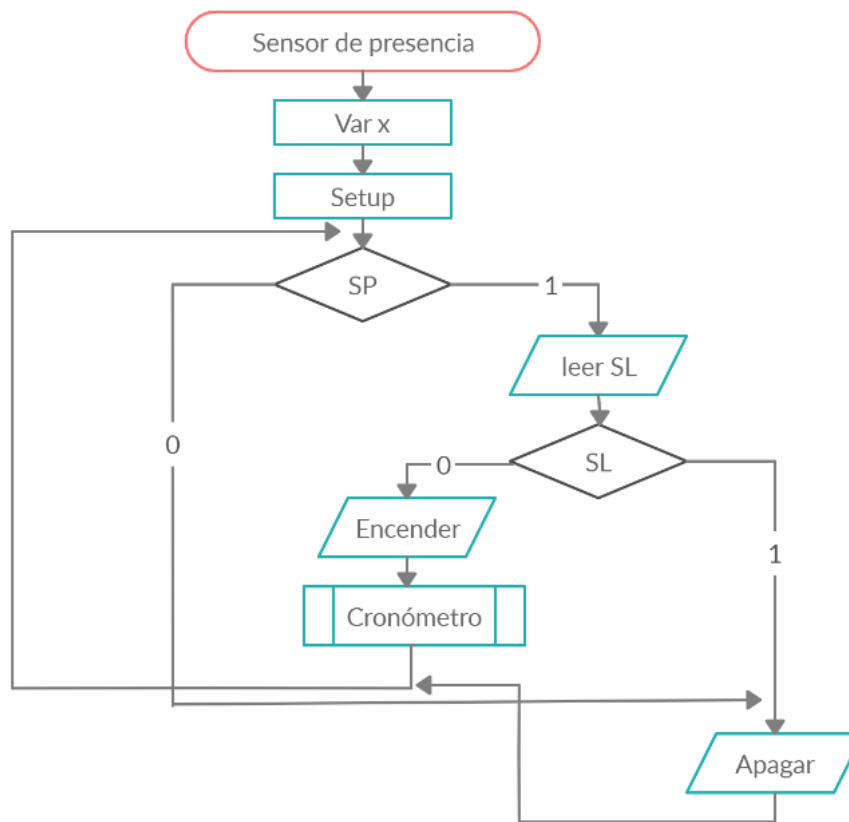


Figura 5.22: Diagrama de flujo para sensor de presencia.

5.12. Circuito final para prototipo

Después de haber realizado la integración de todos los componentes, el circuito quedó como se muestra en la figura 5.23, que detalla la interacción de todos los dispositivos.

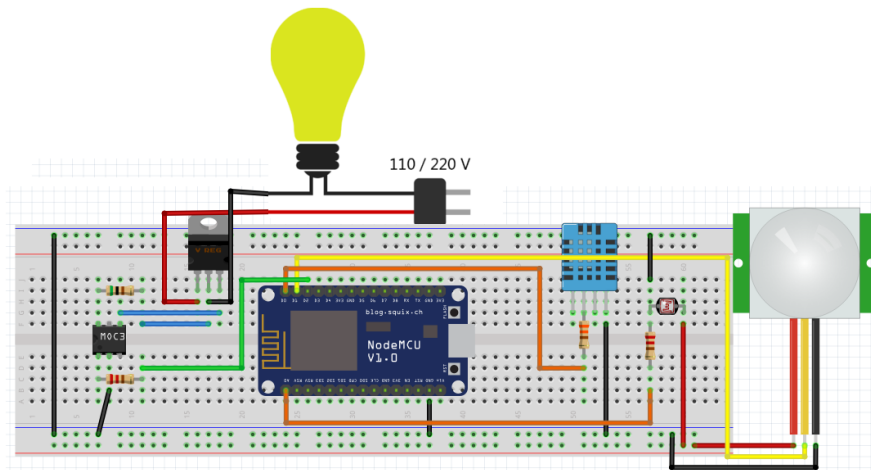


Figura 5.23: Diagrama de conexión para circuito final.

5.12.1. Diseño del case para albergar el sistema electrónico

Para realizar el diseño que contendría el circuito se utilizó una versión de prueba gratuita de SolidWorks en línea llamada MySolidWorks que fue muy útil e indispensable para crear los diseños de almacenamiento del sistema. Para albergar el sistema electrónico se imprimieron los siguientes prototipos de manera física utilizando una impresora 3D llamada Anycubic, obteniendo como resultado los diseños de las figuras 5.24 y 5.25

Algunos de los diseños realizados durante el desarrollo del trabajo fueron los siguientes:



Figura 5.24: Primer diseño en impresora 3D.

Como se puede observar se obtuvo una mejora significativa en cada una de las impresiones físicas. Finalmente quedó definida la siguiente impresión.

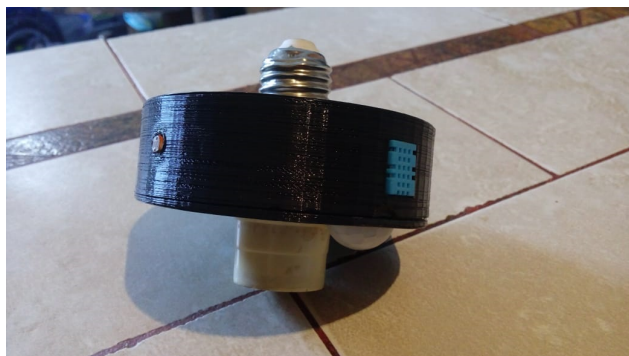


Figura 5.25: Segundo diseño con circuito electrónico preparado para ser utilizado.

|

Conclusiones y Recomendaciones para Trabajo Futuro

A lo largo del proyecto se logró integrar un par de dispositivos para IoT con módulos WiFi Node MCU que pueden gestionar la comunicación entre sensores en un sistema sencillo y funcional.

El prototipo logra realizar la comunicación emparejándose con otro módulo antes mencionado, recopilando datos que pueden ser visualizados a través del monitor serie de cada una de las tarjetas cuando ambos dispositivos se comunican. El sistema puede ser controlado a través de un Push Button como un apagador que tenemos en casa, propiamente el sistema puede trabajar de manera automatizada en ciertas horas del día y la noche cuando las condiciones del lugar donde se encuentre el prototipo lo requieran.

Algo muy importante y gratificante de este sistema es la sencillez que tiene para ser utilizado en un espacio doméstico, como habitación, comedor, cocina etc., porque sólo es cuestión de retirar la bombilla de LED del socket, conectar el prototipo y nuevamente la bombilla. En automático tendremos el prototipo operando y podemos dejarlo que funcione de manera automática o controlarlo con el propio apagador.

El haber trabajado con Arduino fue una experiencia muy placentera al ir integrando diferentes sensores al circuito obteniendo cada vez mejores resultados y nuevas variables del entorno (temperatura, presencia, intensidad luminosa, etc.) en donde se utilice el prototipo.

La utilización de una impresora 3D también fue algo muy enriquecedor y me despertó el interés por adquirir una propia. La facilidad que esta conlleva para ser utilizada hizo que fuera más atractiva para realizar réplicas de diseños en 3D a partir de un diseño hecho por computadora, como por ejemplo Solid Works, el cual fue utilizado en este trabajo para el case que contiene al sistema electrónico.

Aunque sabía que existían Solid Works y Auto CAD para el diseño de dibujos asistidos por computadora fue para mí un conocimiento nuevo que tuve que aprender, y decidí optar por Solid Works para crear el diseño del prototipo, porque una de las capacidades adicionales que se distingue de Auto Cad es que emula las propiedades físicas de un objeto, por ende, es muy utilizado en sectores como la arquitectura y el diseño industrial. Aunque no se haya emulado el prototipo por falta de experiencia en la herramienta, en un futuro cercano una de las prioridades sería aprender a emular las propiedades de los objetos y someterlos a pruebas para ver como reaccionan ante las condiciones del clima como la temperatura por el contacto

directo que tiene el sistema con las luminarias, y el tipo de material a utilizar.

Como recomendaciones y trabajo a futuro se proponen:

- Mostrar la información que generan los sensores en un panel de control desde una página web, donde el usuario pueda controlar directamente las luminarias, y ver la temperatura y humedad actual.
- Mejorar la implementación del prototipo para que sea más resistente mediante la emulación de piezas mecánicas en Solid Works.
- Ampliar el control de luminarias para monitorear un espacio doméstico a través de un proceso que lleva a cabo un protocolo de transporte de mensajería denominado MQTT por medio de la publicación y suscripción de mensajes generados por los sensores.

De esta manera cada uno de los sensores podrán mandar información sobre un topic a un MQTTBroker (Servidor que gestiona la arquitectura cliente-servidor), a otro dispositivo que podrá suscribirse a este topic, de modo que cada vez que el primer dispositivo envíe información, ésta llegue al segundo dispositivo y así consecutivamente con todos los demás sensores que contenga el sistema.

Bibliografía

- [1] Colmena, A. (Abril de 2017). *Diseño e implementación de un sistema de control de iluminación basado en dispositivos móviles inteligentes y módulos empotrados de bajo coste* (Tesis de Licenciatura). Universidad Politécnica de Cartagena, España. Recuperado de <http://repositorio.upct.es/bitstream/handle/10317/5956/pfc6395.pdf?sequence=1&isAllowed=y>
- [2] Botia, D. J., Patiño, R., Ospina, E., y Aristizabal, J., *Implementación de un Sistema Domótico Basado en una Plataforma de Internet de las Cosas*. Memorias de la Décima Quinta Conferencia Iberoamericana en Sistemas, Cibernética e Informática (CISCI 2016).
- [3] Montañez, Olivares y otros. (2019). *Diseño y Desarrollo de una Luminaria Inteligente con Tecnología IoT*. Sinergia Mecatrónica. Recuperado de: <http://www.mecamex.net/Libros/2019-Libro-SinergiaMecatronica.pdf>
- [4] Rebolledo, H. P., (2011). *Alternativas para la reducción del consumo residencial en tarifas DAC*. Boletín IIE.
- [5] Sánchez, M. J., (Mayo de 2016) La domótica en el hogar puede ahorrar costes y hasta un 10 % de energía, según un estudio americano. Smartlighting. Recuperado de <https://smart-lighting.es/la-domotica-hogar-puede-ahorrar-costes-10-energia-segun-estudio-americano/>
- [6] Carballar, J., Falcón, C. (2010). *WI-FI. Lo que se necesita conocer*. RC Libros.
- [7] Evans, D. (Abril de 2011). Cisco Online. *Internet of Things la próxima evolución de internet lo está cambiando todo*. Recuperado de <http://iie.fing.edu.uy/institucional/biblioteca/presentaciones/CitasIEEE-2011.pdf>
- [8] Chaverra, D.(Junio de2016). *Automatización e Internet de las Cosas pueden ahorrar un 10 por ciento de energía*. ACR Latinoamérica. Recuperado de <https://www.acrlatinoamerica.com/201606076493/noticias/empresas/automatizacion-e-internet-de-las-cosas-pueden-ahorrar-un-10-de-energia.html>

-
- [9] *Diferencias entre bombillas LED y lámparas fluorescentes: Cuadros comparativos e infografías* (Julio de 2016) Ciencia, Consejos, Cuadros Comparativos, Ecológicos, Energía, Generales, Tecnología. Recuperado de <http://cuadroscomparativos.com/diferencias-entre-bombillas-led-y-lamparas-fluorescentes-cuadros-comparativos-e-infografias/>
- [10] *¿Qué es SolidWorks?* (Octubre de 2018). adr Soluciones eLearning. Recuperado de https://www.adrformacion.com/knowledge/ingenieria-y-proyectos/_que_es_solidworks_.html
- [11] Morales, D., Álvaro, E., y Picazzo, P. (Junio de 2015). *Actitudes y creencias del consumo de energía eléctrica en los hogares: caso aplicado al área metropolitana de Monterrey*. Revista Electrónica de medio ambiente. UCM. Recuperado de <https://revistas.ucm.es/index.php/MARE/article/download/49593/46142>
- [12] Comisión Estatal de Energía de Baja California. *Guía práctica ahorro energía eléctrica en el hogar*. EnergíaBC. Recuperado de <http://www.coees.sonora.gob.mx/images/descargas/Eficiencia-Energetica/Guia-HOGAR.pdf>
- [13] *ITU Telecommunication Standardization Sector* (Junio de 2015). Overview of the Internet of things. Recuperado de <http://hanle.itu.int/11.1002/1000/1159-en?locatt=format:pdf&auth>
- [14] Chan, O. A., Sandoval, J., y Peña. J. J. (Diciembre de 2016). *Internet de las cosas para controlar el encendido y apagado de aires acondicionados y luminarias*. Pistas educativas. Vol. 38. No. 122. pp. 101-102.
- [15] Sotelo, S., Olivo, M. y Rodríguez, J. (Enero de 2015). *Desarrollo de Aplicación Domótica con Comunicación Inalámbrica Bluetooth*. La Mecatrónica en México. Vol. 4. No. 1. Recuperado de <http://www.mecamex.net/revistas/LMEM/revistas/LMM-V04-N01-04.pdf> Págs.: 29-32.
- [16] Ávila y otros. (Marzo de 2020). *Retos de la 5G en el despliegue del IOT*. Informática, XVIII convención y feria Internacional 2020. Recuperado de: <http://www.informaticahabana.cu/sites/default/files/ponencia-2020/TEL09.pdf>
- [17] Pérez, D. (2009). *Sistemas embebidos y sistemas operativos embebidos*. Lecturas en ciencias de la computación. Universidad Central de Venezuela.
- [18] Han, Woojong y Athreya, Madhu y Shoemaker, Ken y Mandhani, Arvind y Wagh, Mahesh y Thakkar, Ticky. (Octubre de 2012). *Administración de energía para un sistema en un chip (SoC)*. Google Patents. Patente de EE. UU. 8,286,014.
-

-
- [19] *Sistemas Embebidos: Innovando hacia los Sistemas Inteligentes*. (2016). SemanticWebBuilder. Recuperado de http://www.semanticwebbuilder.org.mx/es_mx/swb/Sistemas_Embebidos_Innovando_hacia_los_Sistemas_Inteligentes
- [20] *Sistemas Embebidos, FPGA + MCU* (2019). GENERA TECNOLOGÍAS. Recuperado de https://www.generatecnologias.es/sistemas_embebidos_fpga.html
- [21] O'Reilly, Tim. (1999). *Lecciones del desarrollo de software de código abierto*. Comunicaciones de la ACM. Vol. 42. Núm. 4. Págs. 32-37. Ed. ACM Nueva York, NY, EE. UU.
- [22] McDonald, Clement J y otros. (2003). *Software de código abierto en informática médica: por qué, cómo y qué*. Revista internacional de informática médica. Vol. 69. Núm- 2-3. Págs. 175-178. Ed. Elsevier.
- [23] Herrador, R. (2009). *Guía de usuario de Arduino*. Universidad de Córdoba. Vol. 13.
- [24] Artero, O. (2013). *Arduino. Curso práctico de formación*. Ed. RC libros.
- [25] Aziz, D. (2018). *Webserver Based Smart Monitoring System Using ESP8266 Node MCU Module*. International Journal of Scientific and Engineering Research. Vol 9. Págs. 801.
- [26] Serna, A. Ros y otros. (2010). *Guía practica de sensores*. Ed. Creaciones Copyright SL.
- [27] *What is open source?*. (2012). Opensource.com, Recuperado en Enero 2019 de <https://opensource.com/resources/what-open-source>
- [28] *¿Qué es arduino?*. (s.f.). ARDUINO.cl, Recuperado en Abril 2019 de arduino.cl/que-es-arduino/
- [29] *¿Qué es GitHub? Una Guía para principiantes sobre GitHub*. (Agosto de 2018). Kinsta. Recuperado en Junio 2019 de <https://kinsta.com/es/base-de-conocimiento/que-es-github/>
- [30] Saenz, M. (s.f.). *Curso básico de arduino*. Recuperado en Marzo 2019 de <https://www.wittronics.com.mx/wp-content/uploads/2018/08/Curso-de-Arduino-Basico-1ra-Edicion-Saenz-Flores-Misael.pdf>
- [31] Llamas, L. (Junio de 2018). *NODEMCU, la popular placa de desarrollo con ESP8266*. Ingeniería Informática y diseño. Recuperado de <https://www.luisllamas.es/esp8266-nodemcu/>
-

-
- [32] Aziz, D. (Junio de 2018). *Webserver Based Smart Monitoring System Using ESP8266 Node MCU Module*. International Journal of Scientific and Engineering Research. Vol. 9. págs 801.
- [33] *¿Qué es ESP8266?*. (Septiembre de 2017). Aprendiendo Arduino. Recuperado de <https://aprendiendoarduino.wordpress.com/2017/09/12/que-es-esp8266/>
- [34] Laborda, J. (Octubre de 2016). *Introducción al NODEMCU y ESP8266*. Jaimelaborda/Planta-twitera. Recuperado de <https://github.com/jaimelaborda/Planta-Twittera/wiki/1.-Introducci%C3%B3n-al-ESP8266-y-NodeMCU>
- [35] Mairena, T. (Diciembre de 2018). *Diseño e instalación de un sistema de monitoreo y control de los equipos de producción en la empresa KAIZEN S.A. utilizando un sistema Andon*. Universidad Nacional de Ingeniería, Facultad de Electrónica y Computación, Departamento de Electrónica. Recuperado de: <https://core.ac.uk/download/pdf/250146299.pdf>
- [36] *Placa NodeMCU 1.0 V2*. (Agosto de 2016). NODEMAKERS. Recuperado de https://i2.wp.com/www.esploradores.com/wp-content/uploads/2016/08/PINOUT-NodeMCU_1.0-V2-2.jpg
- [37] *Control de iluminación*. (2012). bticino. Recuperado en Junio 2019 de <http://acosa.com.mx/catalogos/Soluciones%20tecnologicas/Bticino/control-de-iluminacin.pdf>
- [38] *Detector combinado: PIR Digital y detector de rotura de vidrio*. (s.f.). ALASTOR. Recuperado en Junio 2019 de http://www.alastornet.com.ar/_files/manuales/698_PATROL-801.pdf
- [39] *Sensores Arduino*. (Abril de 2018). Aprendiendo Arduino. Recuperado en Abril 2019 de <https://aprendiendoarduino.wordpress.com/2018/04/14/sensores-arduino-3/>
- [40] *Sensor infrarrojo de movimiento PIR HC-SR501*. (2017). Punto flotante. Recuperado en abril 2019 de <https://puntoflotante.net/MANUAL-DEL-USUARIO-SENSOR-DE-MOVIMIENTO-PIR-HC-SR501.pdf>
- [41] *Sensor detector de movimiento (PIR)*. (s. f.). MASTER Innovación en Electrónica. Recuperado en Abril 2019 de <http://www.master.com.mx/img/fichas/AR-PIR.pdf>
- [42] Vinent, Aarón. (Julio de 2017). *Diseño e implementación de una cámara trampa de bajo coste* (Tesis de Licenciatura) Universidad Politécnica de Cataluña, España. Recuperado en Abril 2019 de <https://upcommons.upc.edu/bitstream/handle/2117/114319/memoria.pdf?sequence=1&isAllowed=y>
-

- [43] Enrique. (Octubre de 2014). *Cómo utilizar LDR con Arduino - Fotorresistencia*. EDUCACHIP. Recuperado en Mayo 2019 de <http://www.educachip.com/utilizar-ldr-arduino/>
- [44] Llamas, L. (Marzo de 2015). *Medir Nivel de Luz con Arduino y Fotorresistencia LDR (GL55)*. Ingeniería Informática y Diseño. Recuperado de <https://www.luisllamas.es/medir-nivel-luz-con-arduino-y-fotorresistencia-ldr/>
- [45] *KIT-SEN-ACT: Kit De Sensores Y Actuadores Compatibles Con Arduino* (Noviembre de 2017). AG Electrónica. Recuperado de <http://www.agspecinfo.com/pdfs/K/KITSENAT.PDF>
- [46] Castro, R. (2005). *Avanzando en la seguridad de las redes WIFI*. Boletín de RedIRIS. Vol. 73 págs. 23-33.
- [47] Falcón, JA. (s.f.). *WiFi lo que se necesita Conocer*. Madrid España: Grupo RC. Obtenido de Telecomunicación.
- Valle, L. (2010). *Los servicios inalámbricos*. Recuperado de: <https://www.palermo.edu/ingenieria/downloads/CyT6/6CyT%2012.pdf>
- [48] Colmena, M., Adrián y otros. (2017). Diseño e implementación de un sistema de control de iluminación basado en dispositivos móviles inteligentes y módulos empotrados de bajo coste. Recuperado de <https://repositorio.upct.es/handle/10317/5956>
- [49] Nguyen, K. Ishizu y otros. (2017). 2017 IEEE International Conference on Communications Workshops. *Empirical investigation of IEEE 802.11ad network*. Págs. 192-197.
- [50] Masiukiewicz, A. (2013). @articleM2013, author=.^A. Masiukiewicz", *802.11ad : 60 GHz WLAN implementation*. Elektronika : konstrukcje, technologie, zastosowania. Vol. 54. Págs. 72-73.
- [51] Carballar, F. (2010). *WiFi lo que se necesita Conocer*. Madrid España: Grupo RC. Obtenido de Telecomunicación.
- [52] Bernier, A., Vega, V., y Martínez, D. (2005). *Seguridad WiFi* Redes y Sistemas de Radio. Recuperado de <http://trajano.us.es/fornes/RSR/2005/SeguridadWIFI/Trabajo%20WIFI.pdf>
- [53] *Redes LPWAN*. (Mayo de 2018). Aprendiendo Arduino. Recuperado de <https://aprendiendoarduino.wordpress.com/tag/banda-ism/>
- [54] Aguayo, P. (2004). *Introducción al microcontrolador*. Recuperado de: <https://d1wqtxts1xzle7.cloudfront.net/39407044/micro.pdf?1445749965=&response-content-disposition=inline%3B+filename%3DMicro.pdf&Expires=1593921268&Signature=WuQRBG2vu8j9LMuTO2G5Di62O8CvbkdB6EAYvPUXltOhNJHVjMBOjIPyNI5uOssCpSEmdfDFbYqZFPwbSlp4657V-d0SuVfPrV7NeCLAIQu6ENcvaUqm-EbhuevLWUCaqmjUQUqARe tH Jfnp3kPWYEiK4ColDmVRsAtZdWxaw47zw1oPgEX SmQ>
-

3XGDrM 80fE4yndcVSK8W6XSbl1IbEz2Gm-fJmMEWjB6gaqsKvjYftXMM EKkzV-PoanqTkAp XTLoNjQ-IxTj0YOINcuAp9fu7dNqK4HV67NAlucpzl1kWKpMWHusvdg-vw0FEoNjDQVRKeeLFWgkU gqg__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

- [55] Lenin, C. y Hernández, C. (2018). *Technological Proposal for Controlling a Residential Lighting System*. International Journal of Applied Engineering Research. Vol. 13. Recuperado de https://www.ripublication.com/ijaer18/ijaerv13n11_118.pdf
- [56] Aitor, M. (Septiembre de 2017). *Desarrollo de un sistema de monitorización domiciliaria basado en la plataforma NodeMCU V3*, Tesis de Licenciatura. Universidad Politécnica de Valencia, España. Recuperado de <https://riunet.upv.es/bitstream/handle/10251/92938/CARRICONDO%20-%20Desarrollo%20de%20un%20sistema%20de%20monitorizaci%3%b3n%20domiciliaria%20basado%20en%20la%20plataforma%20Node....pdf?sequence=1&isAllowed=y>
- [57] Páez, M., y Sánchez, S. (2017). *Diseño e implementación de un prototipo de control de iluminación LED con aplicación móvil compatible con Arduino*. (Tesis de Licenciatura). Universidad Distrital Francisco José de Caldas. Recuperado de: <https://repository.udistrital.edu.co/bitstream/handle/11349/6060/PaezMonteroMiguelFelipe2017.pdf;jsessionid=07C21DE82862B4A1B277B7E8477210B5?sequence=1>
- [58] Soto, O. (2011). *Comparación de la eficiencia volumétrica entre redes inalámbricas WiFi y WiMAX*. (Tesis de Licenciatura). Universidad Nacional Autónoma de México, Ciudad de México. Recuperado de <http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/164/A6.pdf?sequence=6>
- [59] Prado, E. (Octubre de 2017). *Sistema de gestión y control de acceso basado en IoT y smartphones*. (Tesis de licenciatura). Universidad Politécnica de Cataluña, Barcelona. Recuperado de https://upcommons.upc.edu/bitstream/handle/2117/108916/Memoria_Eduardo_de_Prado.pdf
- [60] Chiquillo, F. (Junio de 2011). *Automatización Industrial*. SlideShare. Recuperado de <https://es.slideshare.net/tecautind/puertos-de-entradasalida>
- [61] NAYLAMP MECHATRONICS. (Marzo de 2017). *TUTORIAL ESP8266 PARTE I*. Recuperado de: https://naylampmechatronics.com/blog/21_tutorial-esp8266-parte-i.html
- [62] Rebollo, P. (s. f.) *El procesador*. Universidad Politécnica de Valencia, España. Recuperado de https://riunet.upv.es/bitstream/handle/10251/10673/El_procesador.pdf
- [63] Rentería y otros. (Septiembre de 2017). *Módulo ESP8266 y sus aplicaciones en el internet de las cosas*. Revista de Ingeniería Eléctrica. Vol.1 No.2, 24-36. Recuperado de: https://www.ecorfan.org/republicofperu/research_journals/Revista_de_Ingenieria_Electri
-

- ca/vol1num2/ECORFAN_Revista_de_Ingenier%C3%ADa_El%C3%A9ctrica_VI_N2_3.pdf
- [64] Valle, L.(Noviembre de 2018). *Guía de introducción a MQTT con ESP8266 y Raspberry Pi*. Programarfácil. Recuperado de <https://programarfácil.com/esp8266/mqtt-esp8266-raspberry-pi/>
- [65] Dani. (Agosto de 2016) *Instalación del entorno de programación de Arduino: IDE*. ESPloradores. Recuperado de <http://www.esploradores.com/instalar-el-ide-de-arduino/>
- [66] *Simuladores Arduino* (Marzo de 2015) Aprendiendo Arduino. Recuperado de <https://aprendiendoarduino.wordpress.com/category/fritzing/>
- [67] Daniel. (Diciembre de 2018). *Instalación del gestor del ESP8266 para el IDE de Arduino*. Recuperado de <https://www.esploradores.com/instalacion-del-gestor-del-microprocesadores8266-en-el-ide-de-arduino/>
- [68] Admin. (Julio de 2015) *Driver CH340 para arduino chinos o genericos*. GeekFactory. Recuperado de <https://www.geekfactory.mx/tutoriales/tutoriales-arduino/driver-ch340-para-arduino-chinos-o-genericos/>
- [69] Balderas, A. (Febrero de 2018). *¿Cómo instalar driver CH340 para Arduinos genéricos?* Electrocrea. Recuperado de <https://electrocrea.com/blogs/tutoriales/como-instalar-driver-ch340-para-arduinios-genericos>
- [70] *Módulo WiFi Serial ESP-01 ESP8266*. (Junio de 2019). Naylamp Mechatronics Recuperado de <https://naylampmechatronics.com/inalambrico/48-modulo-wifi-serial-esp-01-esp8266.html>
- [71] *Módulo PL2303 Conversor USB a TTL*. (Junio de 2019). Naylamp Mechatronics. Recuperado de <https://naylampmechatronics.com/conversores-ttl/40-modulo-pl2303-convertidor-usb-a-ttl.html>
- [72] *Problema Driver Prolific PL2303*. (Mayo de 2017). OBDii ELM327 Recuperado de <https://obd2-elm327.es/solucion-problema-driver-prolific-pl2303>
- [73] *Estructura Sketch Arduino*. (Junio de 2017). Aprendiendo Arduino. Recuperado de <https://aprendiendoarduino.wordpress.com/2017/06/19/estructura-sketch-arduino/>
- [74] *Módulo Wifi Esp8266 Esp201Serial*. (Septiembre de 2018). Geek Factory. Recuperado de <https://www.geekfactory.mx/tienda/radiofrecuencia/modulo-wifi-esp8266-esp-201serial/>
- [75] *Antenas metálicas para circuito impreso en formato SMD*. (Mayo de 2014). mundoELECTRONICO.com Recuperado de <http://www.mundo-electronico.com/?p=364761>
- [76] *Resistencias Pull-Up y Pull-Down* (s.f.). Naylamp Mechatronics Recuperado de https://naylampmechatronics.com/blog/39_Resistencias-Pull-Up-y-Pull-Down.html
-

- [77] Macho, J. C. (Mayo de 2018). *Sensores de temperatura DHT11*. Prometec Recuperado de <https://www.prometec.net/sensores-dht11/>
- [78] Daniel. (Septiembre de 2016). *Servidor WEB modo ACCESS POINT con procesador ESP- Encendido y apagado de un LED* ESPloradores. Recuperado de <http://www.esploradores.com/access-point-servidor-web-nodemcu/>
- [79] Daniel. (Noviembre de 2016). *Conexión ESP-NOW* ESPloradores. Recuperado de <https://www.esploradores.com/practica-6-conexion-esp-now/>
- [80] *OPTOACOPLADOR TRIAC MOC 3021* (s.f.). Electroall. Recuperado de <https://che-charls-electroall.webnode.es/optoacoplador-triac-moc-3021/>
- [81] Moreno. B. (Jnio de 2017). *¿Conoces la hora dorada y azul? Aprovecha las horas mágicas en tus fotografías* The imagen. Recuperado de <https://www.theimagen.com/conoces-la-hora-dorada-azul-aprovecha-las-horas-magicas-tus-fotografias/>
- [82] *Niveles de Iluminación recomendados* (s.f.) Recuperado de https://www.noao.edu/education/QLTkit/es/Safety_Activity_Poster/LightLevels_outdoor+indoor_es.pdf
-

Anexo A: Código para las aplicaciones

7.1. Hola Mundo

```
void setup() {  
  // initialize digital pin LED_BUILTIN as an output.  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the  
    voltage level)  
  delay(1000); // wait for a second  
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the  
    voltage LOW  
  delay(1000); // wait for a second  
}
```

7.2. Servidor web para encender y apagar un LED con ESP-01

```
\#include <ESP8266WiFi.h> //Se Incluye la libreria WiFi  
  
//Credenciales del WiFi
```

```
const char* ssid = ‘‘ProyectoIoT’’;
const char* password = ‘‘ProyectoIoT’’;
int ledPin = D0; // D0 is the LEDBUILDIN (Active LOW)
WiFiServer server(80);
void setup() {
  Serial.begin(115200); //Inicio del puerto serie
  delay(10);
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);
  // Se Conecta con la red WiFi en modo estacion (STA)
  Serial.println();
  Serial.println();
  Serial.print(‘‘Conectando a ’’);
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(‘‘. ’’);
  }
  Serial.println(‘‘ ’’);
  Serial.println(‘‘WiFi conectado’’);
  // Inicializo el servidor
  server.begin();
  Serial.println(‘‘Servidor iniciado’’);
  // Se muestra la direccion IP local asignada. URL del servidor
```

```
Serial.print('‘Usa esta URL para conectar al servidor: ’’);  
Serial.print('‘http://’’);  
Serial.print(WiFi.localIP());  
Serial.println('‘/’’);  
}  
void loop() {  
  // Comprueba si hay un cliente disponible (una petici n)  
  WiFiClient client = server.available();  
  if (!client) {  
    return; // En caso de no haber un cliente, no hago nada  
  }  
  // Espera hasta que el cliente realice una petici n  
  Serial.println('‘ Nuevo cliente!’’);  
  while(!client.available()){  
    delay(1);  
  }  
  // Lee la primera linea de la petici n del cliente  
  String request = client.readStringUntil('r');  
  Serial.println(request); //Imprime la petici n  
  client.flush(); //Limpia el buffer  
  // Interpreta lo que ha recibido  
  int value = LOW;  
  if (request.indexOf('‘/LED=ON’’) != -1) {  
    digitalWrite(ledPin, HIGH);  
    value = HIGH;  
  }  
}
```

```
if (request.indexOf('/LED=OFF') != -1) {
  digitalWrite(ledPin, LOW);
  value = LOW;
}

// Se coloca ledPin al valor que ha solicitado el cliente en la
// petici n
// Devuelve la respuesta al cliente

client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println(""); // do not forget this one
// A partir de aqu creo la pagina en raw HTML
client.println("<!DOCTYPE HTML>");
client.println("<html>");
client.print("El LED esta: ");
if(value == HIGH) {
  client.print("ON");
} else
{
  client.print("OF");
}

client.println("<br><br>");
client.println("<a href='/LED=ON'><button>Encender </button></a>");
// Los botones con enlace
client.println("<a href='/LED=OFF'><button>Apagar </button></a><br />");
client.println("</html>");

delay(1);

Serial.println("Cliente desconectado"); // Se desconecta del cliente
```

```
Serial.println('');  
}
```

7.3. Encendido de un LED con sensor PIR

```
const int LEDPin = 7;    //Pin para el LED  
const int PIRPin = 6;    //Pin de entrada  
(for PIR sensor)  
int pirState = LOW;      //En un inicio no hay movimiento  
int val = 0;             //Estado del pin  
void setup()  
{  
  pinMode(LEDPin, OUTPUT);  
  pinMode(PIRPin, INPUT);  
  Serial.begin(9600);  
}  
void loop()  
{  
  val = digitalRead(PIRPin);  
  if (val == HIGH)  
  {  
    digitalWrite(LEDPin, HIGH); //LED encendido  
    if (pirState == LOW) //Si previamente estaba apagado  
    {  
      Serial.println('Sensor activado');  
      pirState = HIGH;  
    }  
  }  
}
```

```
}  
}  
  
else //Si esta desactivado  
{  
  
digitalWrite(LEDpin, LOW); // LED apagado  
  
if (pirState == HIGH) //Si previamente  
estaba encendido  
{  
  
Serial.println("Sensor parado");  
  
pirState = LOW;  
  
}  
  
}  
  
}  
section{Encendido de un LED con push button}  
  
int pulsador=0; //almacena el estado del bot n  
int estado=0; //0=LED apagado  
1=led encendido  
int pulsadorAnt=0; //almacena el estado anterior del boton  
  
/** Programa **/  
  
void setup() {  
    pinMode(2, OUTPUT); //Se declara el pin 2 como salida  
    pinMode(4, INPUT); //Se declara el pin 4 como entrada  
}  
  
void loop() {  
    pulsador = digitalRead(4); //Lee si el bot n esta pulsado  
  
if((pulsador==HIGH)&&(pulsadorAnt==LOW)){ //Si el bot n es pulsado y  
    antes no lo estaba  
estado=1-estado;  
    delay(40); //Pausa de 40 ms}pulsadorAnt=pulsador; //  
    Actualiza el nuevo estado del bot n  
  
if(estado==1) { //si el estado es 1 se enciende el LED  
    digitalWrite(2, HIGH);  
}  
}
```

```

else{ //Si el estado es 0 se apaga el LED
    digitalWrite(2, LOW);
}
}

\section{Obtención de datos de temperatura y humedad con sensor DHT11}

\begin{lstlisting}
#include <DHT11.h>

int pin=2;
DHT11 dht11(pin);

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  int err;
  float temp, hum;
  if((err = dht11.read(hum, temp)) == 0) // Si devuelve 0 es que ha
    le do bien
  {
    Serial.print("Temperatura: ");
    Serial.print(temp);
    Serial.print("Humedad: ");
    Serial.print(hum);
    Serial.println();
  }
  else
  {
    Serial.println();
    Serial.print("Error Num :");
    Serial.print(err);
    Serial.println();
  }
  delay(1000); //El valor de temperaturutra y humedad se lee una vez por
    segundo
}

```

7.4. Uso de relay con ESP201 para encender una bombilla

```

int LedPin = 9;

void setup() {
  pinMode(LedPin, OUTPUT);
}

```

```

void loop() {
digitalWrite(LedPin, HIGH); //Enciende el LED
delay(1000); //Espera medio segundo
digitalWrite(LedPin, LOW); //Apaga el LED perteneciente al pin n mero
        6
delay(1000); //Espera medio segundo
}

```

7.5. Servidor web modo Acces Point con procesador ESP8266 (Node MCU)

```

#include <ESP8266WiFi.h> //Incluye la libreria ESP8266WiFi

const char ssid[] = "ProyectoIoT"; //Se define el SSID de nuestro
    servidor WiFi y nombre de red
const char password[] = "ProtectoIoT"; //Se define la contraseña
    del servidor
WiFiServer server(80); //Se Define el puerto de comunicaciones

int PinLED = 2; //Definición del pin de salida
    - GPIO2 / D4
int estado = LOW; //Definición de la variable que
    va a tomar el estado del LED

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

pinMode(PinLED, OUTPUT); //Se inicializa el GPIO2 como salida
digitalWrite(PinLED, LOW); //Se deja inicialmente el GPIO2 apagado

server.begin(); //Se inicializa el servidor
WiFi.mode(WIFI_AP);
WiFi.softAP(ssid, password); //Red con clave, en el canal 1 y
    visible
//WiFi.softAP(ssid, password,3,1); //Red con clave, en el canal 3 y
    visible
//WiFi.softAP(ssid); //Red abierta

Serial.println();

Serial.print("Dirección IP Access Point - por defecto: "); //
    Imprime la dirección IP
Serial.println(WiFi.softAPIP());
Serial.print("Dirección MAC Access Point: "); //Imprime la
    dirección MAC
Serial.println(WiFi.softAPmacAddress());

```

```

//IPAddress local_ip(192, 168, 1, 1);           //Modifica la direcci n
//IP
//IPAddress gateway(192, 168, 1, 1);
//IPAddress subnet(255, 255, 255, 0);
//WiFi.softAPConfig(local_ip, gateway, subnet);
//Serial.println();
//Serial.print("Access Point - Nueva direccion IP: ");
//Serial.println(WiFi.softAPIP());
}

void loop()
{
// Comprueba si el cliente ha conectado
WiFiClient client = server.available();
if (!client) {
return;
}

//Espera hasta que el cliente env a alguna petici n
Serial.println("nuevo cliente");
while(!client.available()){
delay(1);
}

//Imprime el n mero de clientes conectados
Serial.printf("Clientes conectados al Access Point: %dn", WiFi.
softAPgetStationNum());

//Lee la petici n
String peticion = client.readStringUntil('r');
Serial.println(peticion);
client.flush();

// Comprueba la petici n
if (peticion.indexOf('/LED=ON') != -1) {
estado = HIGH;
}
if (peticion.indexOf('/LED=OFF') != -1){
estado = LOW;
}

//Enciende o apaga el LED en funci n de la petici n
digitalWrite(PinLED, estado);

// Env a la pagina HTML de respuesta al cliente
client.println("HTTP/1.1 200 OK");
client.println("");

client.println("<!DOCTYPE HTML>");
client.println("<meta charset='UTF-8'>");

```

```

client.println('‘<html>’');

//Imprime el estado del led
client.print('‘<h1>El LED esta ahora: ’’);
if(estado == HIGH) {
client.print('‘ENCENDIDO </h1>’');
} else {
client.print('‘APAGADO </h1>’');
}

//Se crean enlaces para modificar el estado del LED
client.println('‘Presiona <a href=’/LED=ON’>AQU </a> para encender el
  LED<br>’');
client.println('‘Presiona <a href=’/LED=OFF’>AQU </a> para apagar el
  LED<br><br>’');

//Se crean cajas de comprobaci n (checkbox) para modificar el estado
del LED
client.println('‘<input type=’checkbox’ onClick=location.href=’/LED=ON’>
  ENCENDER </input><br>’');
client.println('‘<input type=’checkbox’ onClick=location.href=’/LED=OFF’>
  > APAGAR </input><br><br>’');

//Se crean botones para modificar el estado del LED
client.println('‘<button type=’button’ onClick=location.href=’/LED=ON’>
  ENCENDER </button>’');
client.println('‘<button type=’button’ onClick=location.href=’/LED=OFF’>
  APAGAR </button><br><br>’');

//Se crean botones para modificar el estado del LED
client.println('‘<button type=’button’ onClick=location.href=’/LED=ON’
  style=’margin:auto; background-color:green; color:#A9F5A9; padding:5
  px; border:2px solid black; width:200;’><h2> ENCENDER</h2> </button>
  ’’);
client.println('‘<button type=’button’ onClick=location.href=’/LED=OFF’
  style=’margin:auto; background-color:red; color:#F6D8CE; padding:5px
  ; border:2px solid black; width:200;’><h2> APAGAR</h2> </button><br>
  ><br>’');

//Se crea un nico bot n para modificar el estado del LED
if(estado == HIGH) {
client.print('‘<button type=’button’ onClick=location.href=’/LED=OFF’>
  APAGAR </button><br><br>’');
} else {
client.print('‘<button type=’button’ onClick=location.href=’/LED=ON’>
  ENCENDER </button><br><br>’');
}
client.println('‘</html>’');
delay(1);
Serial.println('‘Petici n finalizada’'); // Se finaliza la
petici n al cliente. Se inicializa la espera de una nueva petici n.

```

```

//Desconexión de los clientes
//WiFi.softAPdisconnect();
}

```

7.6. Obtención de direcciones MAC

```

#include <ESP8266WiFi.h>

void setup() {
  Serial.begin(115200); Serial.println();Serial.println();
  Serial.print('AP MAC: '); Serial.println(WiFi.softAPmacAddress());
  Serial.print('STA MAC: '); Serial.println(WiFi.macAddress());
}

void loop() {}

```

7.7. Comunicación ESPNOW/Maestro

```

#include <ESP8266WiFi.h>
extern 'C' {
#include <espnw.h>
}

//***ESTRUCTURA DE LOS DATOS TRANSMITIDOS MAESTRO/ESCLAVO***//
//Se de establecer la misma estructura en el par esclavo
struct ESTRUCTURA_DATOS {
  uint16_t potenciometro;
  uint32_t tiempo;
};

void setup() {

  //***INICIALIZACION DEL PUERTO SERIE***//
  Serial.begin(115200); Serial.println();Serial.println();

  //***INICIALIZACION DEL PROTOCOLO ESP-NOW***//
  if (esp_now_init()!=0) {
    Serial.println('*** ESP_Now init failed');
    ESP.restart();
    delay(1);
  }

  //***DATOS DE LAS MAC (Access Point y Station) del ESP***//

```

```

Serial.print("Access Point MAC de este ESP: "); Serial.println(WiFi.
    softAPmacAddress());
Serial.print("Station MAC de este ESP: "); Serial.println(WiFi.
    macAddress());

/**DECLARACION DEL PAPEL DEL DISPOSITIVO ESP EN LA COMUNICACION**//
//0=OCIOSO, 1=MAESTRO, 2=ESCLAVO y 3=MAESTRO+ESCLAVO
esp_now_set_self_role(1);

/**EMPAREJAMIENTO CON EL ESCLAVO**//
// Direcci n MAC del ESP con el que se empareja (esclavo)
// Se debe introducir la STA MAC correspondiente
uint8_t mac_addr[6] = {0x5C, 0xCF, 0x7F, 0x16, 0x50, 0xB9};    uint8_t
    role=2;
uint8_t channel=3;
uint8_t key[0]={};    //no hay clave
//uint8_t key[16] = \{0,255,1,1,1,1,1,1,1,1,1,1,1,1,1,1\};
uint8_t key_len=sizeof(key);
Serial.print("Tama o de *key: "); Serial.println(key_len);
esp_now_add_peer(mac_addr,role,channel,key,key_len);
}

void loop() {

/**DATOS A ENVIAR**//
ESTRUCTURA_DATOS ED;
ED.potenciometro = analogRead(A0);
Serial.print("Dato potenciometro: "); Serial.print(ED.potenciometro);
delay(20);
ED.tiempo = millis();
Serial.print(". Dato tiempo: "); Serial.print(ED.tiempo);

/**ENV O DE LOS DATOS**//uint8_t *da=NULL; //NULL env a los
datos a todos los ESP con los que esta emparejado
uint8_t da[6] = {0x5C, 0xCF, 0x7F, 0x16, 0x50, 0xB9\};
uint8_t data[sizeof(ED)]; memcpy(data, \&ED, sizeof(ED));
uint8_t len = sizeof(data);
esp_now_send(da, data, len);

delay(1); //Si se pierden datos en la recepci n se debe subir este
    valor

/**VERIFICACION DE LA RECEPCI N CORRECTA DE LOS DATOS POR EL ESCLAVO
    **//
esp_now_register_send_cb([](uint8_t* mac, uint8_t status) {
char MACesclavo[6];
sprintf(MACesclavo, "%02X:%02X:%02X:%02X:%02X:%02X",mac[0],mac[1],
    mac[2],mac[3],mac[4],mac[5]);
Serial.print(". Enviado a ESP MAC: "); Serial.print(MACesclavo);
Serial.print(". Recepci on (0=OK - 1=ERROR): "); Serial.println(status)
;

```

```

    });
}

```

7.8. Comunicación ESPNOW/Esclavo

```

#include <ESP8266WiFi.h>
extern 'C' {
#include <espnw.h>
}

/**ESTRUCTURA DE LOS DATOS TRANSMITIDOS MAESTRO/ESCLAVO**//
//Se de establecer IGUAL en el par maestro
struct ESTRUCTURA_DATOS {uint16_t potenciometro = 0;
uint32_t tiempo = 0;
};

/**PIN de conexi n del LED a regular con el potenciometro del ESP
MAESTRO**//
int PinLED = 5; //Pin D1

void setup() {

/**INICIALIZACI N DEL PUERTO SERIE**//
Serial.begin(115200); Serial.println();

/**INICIALIZACI N DEL PROTOCOLO ESP-NOW**//
if (esp_now_init()!=0) {
Serial.println('Protocolo ESP-NOW no inicializado...');
ESP.restart();
delay(1);
}

/**DATOS DE LAS MAC (Access Point y Station) del ESP**//
Serial.print('AP MAC: '); Serial.println(WiFi.softAPmacAddress());
Serial.print('STA MAC: '); Serial.println(WiFi.macAddress());

/**DECLARACI N DEL PAPEL DEL DISPOSITIVO ESP EN LA COMUNICACI N**//
//0=OCIOSO, 1=MAESTRO, 2=ESCLAVO y 3=MAESTRO+ESCLAVO
esp_now_set_self_role(2);

/**DECLARACI N del PinLED como SALIDA**//
pinMode(PinLED, OUTPUT);
}

void loop() {

/**RECEPCI N DE LA COMUNICACI N ESP-NOW**//
esp_now_register_recv_cb([](uint8_t *mac, uint8_t *data, uint8_t len) {

```

```

char MACMaestro[6];
sprintf(MACMaestro, "%02X:%02X:%02X:%02X:%02X:%02X", mac[0], mac[1],
        mac[2], mac[3], mac[4], mac[5]);
Serial.print("Recepcion desde ESP MAC:      "); Serial.print(MACMaestro);

ESTRUCTURA_DATOS ED;
memcpy(&ED, data, sizeof(ED));

Serial.print("Dato potenciometro: "); Serial.print(ED.potenciometro)
;
Serial.print("Dato tiempo: "); Serial.println(ED.tiempo);

analogWrite(PinLED, ED.potenciometro);
});
}

```

7.9. Programa final apagador

```

#include <DHT11.h>
#include <ESP8266WiFi.h>
extern 'C' {
#include <espnw.h>
}

/**ESTRUCTURA DE LOS DATOS TRANSMITIDOS APAGADOR: LUM1 Y LUM2**/
int pul1=0;
int edo1=0;
int recepcion1=0;

int pul2=0;
int edo2=0;
int recepcion2=0; char MAC_Lum2[30]; //ESP1 LUM2

int pin=16;
DHT11 dht11(pin);
int err;
float temp, hum;
int cuentaintentos=0;

void setup() {
wifi_set_opmode(STATION_MODE);
//Declaracion de los pines de los pulsadores como entrada
pinMode(5, INPUT); //D1: GPIO5 Push boton Lum1
/**INICIALIZACION DEL PUERTO SERIE**/
Serial.begin(115200);
Serial.println();
//INICIALIZACION DEL PROTOCOLO ESP-NOW

```

```

if (esp_now_init()!=0) {
Serial.println('*** ESP_Now init failed');
ESP.restart();
delay(1);
}//Fin de if-Inicializaci n del protocolo ESP-NOW MAESTRO

/***/DATOS DE LAS MAC (Access Point y Station) del ESP***/
Serial.print('Access Point MAC de este ESP: ');
Serial.println(WiFi.softAPmacAddress());
Serial.print('Station MAC de este ESP: ');
Serial.println(WiFi.macAddress());

//DECLARACI N DEL PAPEL DEL DISPOSITIVO ESP EN LA COMUNICACI N
//0=OCIOSO, 1=MAESTRO, 2=ESCLAVO y 3=MAESTRO+ESCLAVO
esp_now_set_self_role(3);
//EMPAREJAMIENTO CON LUM1 Y LUM2
// DIRECCI N MAC DE LAS LUMINARIAS CON LAS QUE SE EMPAREJA
// Se debe introducir la STA MAC correspondiente
uint8_t mac_Lum1[6] = {0x84,0xF3,0xEB,0xB6,0xF8,0x45}; //LUM1
uint8_t role=3;
uint8_t channel=3; //Canal Lum1

uint8_t key[1]={}; //no hay clave

uint8_t key_len=sizeof(key);

void loop() {
pull=digitalRead(5); // D1 GPIO5 Push boton
if((pull==HIGH)&&(pul2==LOW)){
{
edo1=1-edo1;
delay(100); //Se crea una pausa de 40 ms
//do lo utilizo mientras esta presionado el bot n para el retardo
do{
pull=digitalRead(5);
}while(pull==1);
Serial.println('Entra al if del pull presiona boton');

//ENV O DE LOS DATOS A Lum 1 (Node MCU)
//uint8_t *da=NULL; //NULL env a los datos a todos los ESP con los
que esta emparejado
uint8_t da1[6] = \{0x84,0xF3,0xEB,0xB6,0xF8,0x45}; //MAC
uint8_t data1[sizeof(edo1)];
memcpy(data1, &edo1, sizeof(edo1));
uint8_t len1 = sizeof(data1);
cuentaintentos=0;
//Este do lo utilizo para reenviar el dato hasta que tenemos una
recepcci n de datos exitosa
do{

```

```

esp_now_send(dai, data1, len1);
delay(10); //Si se pierden datos en la recepci n se debe subir este
          valor
cuentaintentos++;
//VERIFICACI N DE LA RECEPCI N CORRECTA DE LOS DATOS LUM1 (Node MCU)
//Serial.print('Se envia el dato y estos son los intentos : ');
//Serial.println(cuentaintentos);

esp_now_register_send_cb([](uint8_t* mac, uint8_t status) {
char MAC_Lum1[20];
recepcion1=status;

//Serial.print("Registro de recepci n estado: ");

});

}while(recepcion1==1);

} // Fin del if

//Esta secci n tiene un comportamiento como esclavo
//RECEPCI N DE LOS DATOS DE TEMPERATURA Y HUMEDAD DE Lum1
esp_now_register_rcv_cb([](uint8_t *mac, uint8_t *data2, uint8_t len2){
char MAC_Lum1[30];
sprintf(MAC_Lum1, "%02X:%02X:%02X:%02X:%02X:%02X", mac[0], mac[1],
        mac[2], mac[3], mac[4], mac[5]);
Serial.print('Recepcion desde ESP MAC: ');
Serial.print(MAC_Lum1);
memcpy(&(temp), data2, sizeof(temp));
Serial.print('Dato temperatura: ');
Serial.println(temp);

//          }

}); // Fin/Recepci n de la comunicaci n LUM1
} // Fin del loop

```

7.10. Programa final Luminaria

```

#include <DHT11.h>
#include <ESP8266WiFi.h>
extern 'C' {
#include <espnw.h>
}
//Se deben establecer las mismas variables en el par maestro
int EdoLuminaria;
int evento;

```

```

int pul1=0;
int edo1=0;
int recepcion1=0;

char MAC_Lum1[30]; //Node MCU LUM1
int PinLED =4; //D2: Pin GPIO4
int pin=16; //Sensor Temp. y Hum.
DHT11 dht11(pin);
int err;
float temp, hum;

int SensP;
int SensL;
int SensorPIR=5; //D1: pin GPIO5
int PinLDR = A0;
int ValorLDR;

unsigned long start = 0;
bool InicioDisparo = false;

void setup() {wifi_set_opmode(STATION_MODE);
//INICIALIZACION DEL PUERTO SERIE
Serial.begin(115200);
Serial.println();
Serial.println();

EdoLuminaria=0;

//INICIALIZACION DEL PROTOCOLO ESP-NOW
if (esp_now_init()!=0) {Serial.println("Protocolo ESP-NOW no
inicializado...");
ESP.restart();
delay(1);
}
//DATOS DE LAS MAC (Access Point y Station) del ESP
Serial.print("AP MAC: ");
Serial.println(WiFi.softAPmacAddress());
Serial.print("STA MAC: ");Serial.println(WiFi.macAddress());

/**DECLARACION DEL PAPEL DEL DISPOSITIVO ESP EN LA COMUNICACION**//
//0=OCIOSO, 1=MAESTRO, 2=ESCLAVO y 3=MAESTRO+ESCLAVO
esp_now_set_self_role(3);

//EMPAREJAMIENTO CON EL Node MCU APAGADOR
// Se debe introducir la STA MAC correspondiente
uint8_t mac_Apagador[6] = {0x5C, 0xCF, 0x7F, 0xD3, 0x7A, 0xF8}; //MAC
APAGADOR
uint8_t role=3;
uint8_t channel=6;
uint8_t key[1]={}; //no hay clave
//uint8_t key[16] = \{0,255,1,1,1,1,1,1,1,1,1,1,1,1,1,1\};

```

```

uint8_t key_len=sizeof(key);
Serial.print("Tama o de *key:  '");
Serial.println(key_len);
esp_now_add_peer(mac_Apagador,role,channel,key,key_len);

//DECLARACION DE LOS PINES: ENTRADA Y SALIDA
pinMode(PinLED, OUTPUT);
digitalWrite(PinLED, LOW);
pinMode(pin,OUTPUT);
digitalWrite(pin,LOW);pinMode(SensorPIR, INPUT_PULLUP);
digitalWrite(SensorPIR,HIGH);

attachInterrupt(digitalPinToInterrupt(SensorPIR), SensorLuzPresencia,
  FALLING);

evento=0;

} //Fin de setup

void loop() {
  ***RECEPCION DE LA COMUNICACION ESP-NOW APAGADOR***//
  esp_now_register_rcv_cb([](uint8_t *mac, uint8_t *data1, uint8_t len1){
  evento=1;
  EdoLuminaria=1-EdoLuminaria;
  char MAC_Apagador[30];
  memcpy(&edo1, data1, sizeof(edo1));
  Serial.print(EdoLuminaria);
  Serial.println(" . Se ha presionado un boton:  '");
  });

  if (evento==1){
  temp=21;
  ***ENV O DE LOS DATOS***//
  //uint8_t *da=NULL;  //NULL env a los datos a todos los ESP con los
  que esta emparejado
  uint8_t da2[6] = {0x5C, 0xCF, 0x7F, 0xD3, 0x7A, 0xF8};//
  uint8_t data2[sizeof(temp)];
  memcpy(data2, &(temp), sizeof(temp));
  uint8_t len2 = sizeof(data2);
  //cuentaintentos=0;
  do{
  esp_now_send(da2, data2, len2);
  delay(10); //Si se pierden datos en la recepcion se debe subir este
  valor
  //cuentaintentos++;
  //VERIFICACION DE LA RECEPCION CORRECTA DE LOS DATOS Apagador***//
  esp_now_register_send_cb([](uint8_t* mac, uint8_t status) {
  char MAC_Apagador[30];
  recepcion1=status;

  });
}

```

```
}while(recepcion1==1);
evento=0;
}

if(EdoLuminaria==1) {
digitalWrite(PinLED,HIGH);
}

else{
digitalWrite(PinLED,LOW);
}

}

void SensorLuzPresencia(){
delay (10);
SensP=digitalRead(SensorPIR);
//SensP=HIGH;
ValorLDR = analogRead(PinLDR);
Serial.print(SensP);
Serial.print(' ');
Serial.print(ValorLDR);
Serial.print(' ');
Serial.println(start);

if (ValorLDR < 180) SensL = 1; else SensL = 0;

if (SensP == HIGH && SensL == LOW){
InicioDisparo = true;
digitalWrite(PinLED, HIGH);
EdoLuminaria=1;
start = millis();// cargo Start con el valor de millis()
}

if (InicioDisparo && (millis()-start > 20000UL)) {
digitalWrite(PinLED, LOW);
EdoLuminaria=0;
InicioDisparo = false;
}
}
```