

UNIVERSIDAD MICHOACANA DE SAN NICOLÁS DE HIDALGO



**FACULTAD DE
INGENIERÍA ELÉCTRICA**



TESIS

**DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE DOMÓTICA VÍA
WIFI A TRAVÉS DE UNA PÁGINA WEB.**

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

Presenta:

Pedro Palacios Núñez

Asesor de Tesis

M.C. Félix Jiménez Pérez

Morelia, Michoacán

Diciembre 2016

Resumen

Este documento presenta un trabajo de control a distancia de una casa, mejor conocido como domótica, de tal manera que dispositivos eléctricos, electrónicos y electromecánicos de la misma se puedan controlar para realizar una tarea específica en un determinado instante.

El sistema está compuesto principalmente por dos partes:

- Servidor web, donde se aloja la página web
- Módulo de control, compuesto por un módulo wifi, el microcontrolador y un relevador de estado sólido.

Operación básica: Cada dispositivo a controlar está conectado a un módulo de control que conecta o desconecta de la corriente eléctrica al dispositivo. El módulo controlador, que realiza las tareas asignadas por medio de la página web, se comunica vía wifi al servidor conectado a la red local de la casa. Cualquier dispositivo electrónico conectado a la red podrá interactuar con la página web.

Palabras clave: Construcción, sistema, domótica, wifi, página web.

Abstract

This document presents a system of remote control of a house, better known as Home Automation, in such a way that electrical, electronic and electromechanical devices of the house can be controlled to carry out specific tasks at predetermined moment.

The system is composed mainly of two parts:

- Web server, where the web page is hosted.
- Control Module, consisting of a wifi module, a microcontroller and a solid-state relay.

Basic Operation: Each device to be controlled is connected to a control module that connects or disconnects the electric current to the device. The controller module, which performs tasks assigned via the web page, communicates via wifi to the server connected to the local network of the house. Any electronic device connected to the network can interact with the web page.

Contenido

Resumen	i
Abstract	ii
Contenido	iii
Lista de figuras	v
Lista de tablas.....	vi
Capítulo 1 Introducción.....	1
1.1 Antecedentes	1
1.2 Objetivo	3
1.2.1 Objetivo general	3
1.2.2 Objetivo particular	3
1.3 Justificación.....	3
1.4 Metodología	4
1.5 Contenido de la tesis.....	4
Capítulo 2 Fundamentos teóricos.....	5
2.1 Domótica	5
2.1.1 X-10.....	5
2.1.2 KNX/EIB	5
2.1.3 LonWorks	6
2.1.4 BUSing.....	7
2.1.5 ZigBee	7
2.1.6 Z-wave	7
2.1.7 Otros	7
2.2 Sistema de comunicación y enlace.....	9
2.2.1 Servidor web.....	9
2.2.2 Cliente web	12
2.2.3 Socket	13
2.3 Módulo controlador.....	15
2.3.1 Microcontrolador	16
2.3.2 Módulo wifi.....	16
2.3.3 Relevador de Estado Sólido	16
Capítulo 3 Implementación del Hardware	18
3.1 Sistema de comunicación y enlace.....	18
3.1.1 Servidor	20
3.1.2 Router y Modem	21
3.2 Modulo controlador.....	24
3.2.1 Fuente.....	25
3.2.2 Módulo wifi.....	25

3.2.3 Microcontrolador	28
3.2.4 SSR.....	31
Capítulo 4 Implementación del Software	32
4.1 Software Sistema de comunicación y enlace.....	32
4.1.1 Servidor	32
4.1.2 Software Pagina web	33
4.2 Software Módulo de control	34
4.2.1 Software Módulo wifi	35
4.2.2 Control del SSR.....	36
4.2.3 Software del Microcontrolador.....	37
Capítulo 5 Pruebas y resultados.....	43
5.1 Sistema de comunicación y enlace.....	43
5.2 Módulo de control.....	45
5.2.1 Módulo wifi.....	45
5.2.2 Microcontrolador	50
5.2.3 Relevador SSR.....	52
5.2.4 Circuito impreso del módulo de control.....	52
5.3 Sistema completo.....	53
Capítulo 6 Conclusiones y trabajos futuros.....	54
6.1 Conclusiones generales.....	54
6.2 Conclusiones personales	55
6.3 Trabajos futuros.....	56
Anexo A Código fuente del microcontrolador	56
A.1 Establecer configuración	56
A.2 Configuración del relevador.....	59
A.3 Configuración del timer.....	60
A.4 Configuración del módulo wifi.....	61
Anexo B Código fuente de la página web	65
Bibliografía	68

Lista de figuras

Figura 1. 1 Imagen relacionada con la domótica[1].....	1
Figura 1. 2 Imagen con diferentes dispositivos que se pueden automatizar[4].....	2
Figura 2. 1 Estructura básica de un servidor[11].	10
Figura 2. 2 Representación de clientes en una arquitectura web[17]	12
Figura 2. 3 Conexión de sockets[20].....	14
Figura 2. 4 Conexiones de computadores con dirección IP[21].....	15
Figura 3. 1 Hardware del sistema a bloques	18
Figura 3. 2 Arquitectura web tipo A.	19
Figura 3. 3 Arquitectura web tipo B.	19
Figura 3. 4 Raspberry Pi 3 modelo B.....	20
Figura 3. 5 Diagrama esquemático de la conexión del servidor.....	21
Figura 3. 6 Imagen Modem	21
Figura 3. 7 Imagen de conexión del Modem con la Raspberry	22
Figura 3. 8 Diagrama a bloques de la conexión del Modem	22
Figura 3. 9 Imagen Router.....	23
Figura 3. 10 Conexión Router con Raspberry	23
Figura 3. 11 Diagrama a bloques de la conexión del Router.....	23
Figura 3. 12 Conexión modulo controlador	24
Figura 3. 13 Diagrama a bloques del módulo de control.....	25
Figura 3. 14 Diagrama a bloques de la conexión de la fuente.....	25
Figura 3. 15 Imagen del módulo wifi	26
Figura 3. 16 Ubicación de los pines del módulo wifi [27].	27
Figura 3. 17 Conexión esquemática del módulo wifi	28
Figura 3. 18 Imagen del microcontrolador.....	28
Figura 3. 19 Ubicación de pines del microcontrolador [24].....	29
Figura 3. 20 Conexión esquemática del microcontrolador.	30
Figura 3. 21 Conexión esquemática del SSR.....	31
Figura 4. 1 Diagrama de flujo de la Pagina web.....	33
Figura 4. 2 Imagen de la página web.....	34
Figura 4. 3 Diagrama de flujo del módulo de control.....	35
Figura 4. 4 Diagrama de flujo de comandos AT	35
Figura 4. 5 Diagrama de flujo del software del microcontrolador.....	37
Figura 4. 6 Diagrama de flujo de la operación de las interrupciones.....	41
Figura 5. 1 Visualización de la interface de la página web.....	43
Figura 5. 2 Respuesta del botón	44
Figura 5. 3 Respuesta del botón en la aplicación Simple Socket Tester.....	44
Figura 5. 4 Respuesta del botón para el socket.....	45
Figura 5. 5 Conexión del módulo wifi al módulo convertidor de puerto serial a USB	46

Figura 5. 6 Interface de CoolTerm.....	46
Figura 5. 7 Prueba y respuesta del módulo wifi en la interface.	47
Figura 5. 8 Prueba AT+RST del módulo wifi en la interface CoolTerm.....	47
Figura 5. 9 Pruebas de comandos del módulo wifi en la interface CoolTerm.....	48
Figura 5. 10 Programa Simple Socket Tester enviando “Luz 1 encendida”	49
Figura 5. 11 Respuesta del módulo wifi en la interface CoolTerm	50
Figura 5. 12 Conexión del microcontrolador y el módulo wifi.	51
Figura 5. 13 Conexión y envió de dato con el programa Simple Socket Tester.....	51
Figura 5. 14 Funcionamiento correcto del microcontrolador.....	52
Figura 5. 15 Funcionamiento conjunto del microcontrolador y el relevador SSR	52
Figura 5. 16 Funcionamiento del circuito impreso del módulo de control.....	53
Figura 5. 17 Comprobación del funcionamiento del sistema.....	53

Lista de tablas

Tabla 2.1 Comparativa para protocolos más comunes[8]	8
Tabla 3.1 Características de microcontroladores MSP430G2553IN20 [28].....	30
Tabla 4.1 Comandos AT del módulo wifi	36

Capítulo 1 Introducción

En esta tesis se presenta el desarrollo de un sistema de domótica. Primeramente, se define la domótica y sus avances hasta el momento, además de los sistemas más conocidos en la actualidad. Posteriormente, se dan a conocer los elementos con los que se desarrolla el sistema de domótica y se explica su funcionamiento en conjunto.

La domótica es el conjunto de tecnologías aplicadas a la automatización y control inteligente de la vivienda que integra la tecnología en los sistemas de gestión energética, seguridad, bienestar y monitoreo, así como de comunicación entre el usuario y el sistema [1]. Tal como se muestra en la figura 1.1.



Figura 1. 1 Imagen relacionada con la domótica[1]

El sistema presentado en este trabajo se puede considerar un sistema de domótica porque emplea tecnologías de automatización y control, además de la comunicación entre el usuario y el mismo.

1.1 Antecedentes

La domótica inicia a comienzos de la década del '70, al aparecer los primeros dispositivos de automatización en edificios a base de prueba piloto; pero fue en la década del '80 cuando los sistemas integrados se utilizaron a nivel comercial para luego desarrollarse en el aspecto doméstico de las casas urbanas. En ese momento la domótica consigue integrar dos sistemas (el eléctrico y el electrónico) en pos de la comunicación integral de los dispositivos del hogar.

El desarrollo de la tecnología informática permite la expansión del sistema, sobre todo en países de vanguardia como Estados Unidos, Alemania y Japón [2].

1.2 Objetivo

1.2.1 Objetivo general

Otorgar a las personas un sistema para el hogar con el cual puedan controlar a los dispositivos que utilicen corriente eléctrica mediante una página web, sin restricción de plataforma, de simple instalación y uso intuitivo.

1.2.2 Objetivo particular

Construir un sistema que pueda controlar a los dispositivos que utilicen la corriente eléctrica de una casa vía wifi, a través de la construcción de una arquitectura web.

Instalar un servidor web.

Diseñar y programar una Pagina web.

Programar un microcontrolador que controle un relevador de estado sólido y un módulo wifi.

Diseñar un protocolo de comunicación entre el servidor web, el módulo de control y el cliente.

1.3 Justificación

El aprendizaje y la utilización de nuevos conocimientos nos hacen más competentes y capaces de resolver problemas puesto que la evolución de la tecnología es constante. Además, este trabajo pretende contribuir con el desarrollo de la Facultad de Ingeniería Eléctrica.

En ocasiones también el dispositivo que se desea controlar, por ejemplo una lámpara, se encuentra distante de la parte de la casa donde el usuario esta, por lo que el presente sistema permita controlar ese dispositivo de forma remota, ahorra tiempo y la fatiga derivada del desplazamiento.

De la misma forma cuando se está fuera de casa y se desea controlar algún dispositivo del hogar, el presente sistema permitirá hacerlo.

Los sistemas actuales de domótica también suelen tener una instalación compleja, ya sea cableada o inalámbrica, por lo que la mayoría de las veces se requiere un técnico especializado. El sistema propuesto permite empezar a utilizarlo con la simple colocación y configuración del mismo de manera sencilla.

Actualmente existen una gran cantidad sistemas de control y automatización que están hechos para plataformas específicas como Android, IOS, Windows etc. Este sistema posibilita que cualquier dispositivo con un navegador web interactúe con la página de manera intuitiva.

1.4 Metodología

Investigación bibliográfica de la Domótica y los sistemas que se utilizan actualmente.

Instalar un servidor.

Consulta de libros e Internet para el desarrollo de la página web.

Revisión de la hoja de datos del microcontrolador y del módulo wifi.

Desarrollo del software y hardware del módulo de control.

Implementación de pruebas, corrección y comprobación del funcionamiento.

1.5 Contenido de la tesis

En el Capítulo 1 se da una breve introducción a este trabajo, lo que es la domótica, como fue evolucionando, los principales sistemas de domótica que se utilizan, así mismo se describe el objetivo y se explica la importancia y como se desarrollará este trabajo.

En el Capítulo 2 se informa sobre los elementos básicos que se usan para la comunicación en el sistema propuesto, algunos de los principales sistemas que existen hoy en día y cómo funcionan, también se dan a conocer los conceptos de elementos con los que se estructura y desarrolla.

El Capítulo 3 explica el desarrollo del hardware del sistema, los elementos de los cuales se compone, las características de estos elementos y como son sus conexiones.

En el Capítulo 4 se describe el desarrollo del software del sistema, de qué modo se instala el servidor para este sistema, cómo es la programación de la página web y la del software del microcontrolador.

En el Capítulo 5 se presentan las pruebas que se realizaron a los elementos del sistema y los resultados que se obtuvieron.

En el Capítulo 6 se dan a conocer las conclusiones específicas y generales y posibles trabajos futuros.

Capítulo 2 Fundamentos teóricos

En este capítulo se darán a conocer los principales sistemas que se han usado en la domótica desde sus inicios hasta hoy en día, además de las partes principales con los que se elabora el sistema de domótica implementado:

- Sistema de comunicación y el enlace
- Módulo de control.

2.1 Domótica

El término Domótica proviene de la unión de las palabras *domus* (que significa casa en latín) y *tica* (de automática, palabra en griego, ‘que funciona por si sola’) [3], [6].

Algunos de los sistemas de domótica más conocidos en su ámbito son X-10, KNX/EIB, Long Works, BUSing, ZigBee y Z-wave, los cuales se describen en la siguiente sección.

2.1.1 X-10

Por méritos propios, el sistema X-10 puede ser considerado el precursor de los sistemas domóticos actuales. Fue desarrollado entre los años 1976 y 1978 por la empresa escocesa Pico Electronics y, pese a las décadas transcurridas desde su creación, todavía es uno de los sistemas más populares en el mercado [7].

Utiliza la línea eléctrica (220 V ó 110 V) para transmitir señales de control entre los equipos de automatización del hogar en forma digital. Las señales de control del X-10 se basan en la transmisión de ráfagas de pulsos de radio frecuencia (120KHz) que representan información digital. Estos pulsos se sincronizan en el cruce por cero de la señal de red (50 ó 60 Hz). Su principal ventaja es que utiliza el cableado de red eléctrica para la comunicación, lo que libera de la necesidad de instalar cableado para ello. Sin embargo, es un sistema lento y es muy sensible a ruidos[6].

2.1.2 KNX/EIB

En 1997, mediante sus respectivas asociaciones, los tres sistemas de automatización europeos más arraigados (Batibus, EHS y EIB) comenzaron un proceso de convergencia hacia la Asociación Konnex, que finalmente cuaja en la Asociación KNX y su norma homónima.

El estándar KNX, basado en la tecnología EIB (Bus de Instalación Europeo), es aun compatible con los productos EIB previamente instalados, pero aumentó la funcionalidad de estos gracias a los modos de configuración de Batibus y EHS (European Home System) Sistema de Casa Europeo, añadiendo, además, nuevos medios físicos al estándar [7].

El Bus de Instalación Europeo (EIB o EIBus) es un sistema de domótica basado en un Bus de datos. Originalmente conocido por Instabus[6].

2.1.3 LonWorks

El sistema LON (Local Operative Network) Red Local Operativa, fue introducido por Echelon en 1990 y ya en 1996 era accesible para cualquier fabricante que estuviese interesado en adquirirlo. Su éxito fue tal que fue adoptado como estándar por el ANSI (American National Standards Institute) Instituto Nacional de Estándares, en el año de 1999 (estándar ANSI-EIA 709.1-A-1999), dando lugar a la creación del organismo independiente LonMark, con competencia para decidir si un producto es compatible con LonWorks. Este organismo decide también sobre actualizaciones del estándar demandadas por las necesidades de la industria.

Mike Markkula y Ken Oshmann decidieron basar la nueva tecnología en un chip microcontrolador al que bautizarían como Neutrón, y de cuya fabricación se encargarían las compañías Toshiba y Motorola. Más adelante, la compañía Cypress relegó a Motorola y se consolidó como segundo proveedor del chip. Más tarde un hito importante en la evolución de la tecnología fue la introducción de los servicios de red de LonWorks, que consistían en una plataforma de desarrollo de aplicaciones abiertas unida a un conjunto de herramientas de integración [7].

En la actualidad existen más de 100 millones de dispositivos instalados en el mundo. Una ventaja importante respecto a otras tecnologías de control es que implementa todos los niveles del modelo de referencia ODI (Overseas Development Institute) Instituto de Desarrollo de Ultramar. De este modo, servicios tales como el reenvío automático tras una pérdida de trama o la autenticación del emisor de la trama están totalmente implementados [6].

2.1.4 BUSing

BUSing fue creado por Ingenium, una empresa fundada en el año 1998 en la ciudad de Oviedo (España). Esta empresa fue creada con el objetivo de introducir en el mercado un protocolo de comunicaciones accesible para sistemas domóticos que siguiese las bases de protocolos distribuidos ya consolidados como KNX o LonWorks. De este modo surgió el protocolo BUSing, que se caracteriza por ser un sistema libre de regalías, abierto a otros fabricantes y cuyos campos principales de aplicación son la domótica e inmótica. La capacidad de control de BUSing abarca, en efecto, opciones tan diversas como: control de iluminación, gestión de toldos y persianas, gestión de alarmas técnicas, control de temperatura o riego, gestión de sistemas de seguridad y sistemas audiovisuales[7].

2.1.5 ZigBee

ZigBee es el nombre de la especificación de un conjunto de protocolos de alto nivel de comunicación inalámbrica para su utilización con radiodifusión digital de bajo consumo, basado en el estándar IEE 802.15.4 de redes inalámbricas de área personal (Wireless Personal Area Network, WPAN). Su objetivo son las aplicaciones que requieren comunicaciones seguras con baja tasa de envío de datos y maximización de vida útil de sus baterías [6].

2.1.6 Z-wave

Z-wave es un protocolo inalámbrico por ondas de radio destinado para la comunicación entre dispositivos domésticos eléctricos. Sirve para crear una red única con los equipos eléctricos como: luz, termostatos, alarmas, ordenadores, teléfonos, aire acondicionado y el manejo de las ventanas eléctricas y persianas. A la red se pueden conectar todos los dispositivos eléctricos que tengan el módulo del sistema Z-Wave. Hoy en día en todo el mundo al sistema Z-Wave están asociados más de 200 fabricantes especializados en dispositivos para el uso doméstico. En el mercado ya están disponibles muchos productos con esta tecnología[6].

2.1.7 Otros

Implantada desde hace más de treinta años, la domótica ha progresado a gran escala desde que se desarrollaron las redes informáticas de comunicación, ya sea por sistema cableado o vía wifi.

Con la domótica aplicada a la automatización hogareña se mejora en seguridad, confort y ahorro energético, aspectos muy observados por los poseedores de estos sistemas. La llegada de Internet a gran velocidad provocó un giro favorable para su desarrollo [2].

En la vivienda también se suelen utilizar sistemas de comunicación estándar, como por ejemplo ADSL (Asymmetric Digital Subscriber Line) Línea de Suscripción Digital Asimétrica, Ethernet, infrarrojos, bluetooth, GPRS (General Packet Radio Service) Paquete General de Servicio de Radio, etc, sobre todo para enlazar con sistemas estándar como pueden ser los servicios de web, monitorización y controlar la vivienda desde un teléfono móvil, etc.[6].

En la tabla 2.1 se muestra una comparativa de los protocolos más comunes, respecto a sus medios, velocidades, y distancias de transmisión.

Tabla 2.1 Comparativa para protocolos más comunes[8].

Tecnología	Medio de transmisión	Velocidad de transmisión	Máxima distancia a la que puede transmitir
Ethernet	Par trenzado sin blindaje	10 Mbit/s - 1 Gbit/s	100m
	Fibra óptica	1 Gbit/s - 10 Gbit/s	2Km - 15Km
wifi	Radio frecuencia	11 Mbit/s – 24.8 Mbit/s	30m - 100m
Bluetooth	Radio frecuencia	1 Mbit/s - 10 Mbit/s	10m - 100m
IRDA	Infrarrojo	14 Mbit/s - 1Gbit/s	2m
C-BUS	Par trenzado/ Cable eléctrico/ Radio frecuencia/ Infrarrojo/ Ethernet/ wifi	1200 bit/s - 9600 bit/s	1000m
LonWorks	Par trenzado/ Cable eléctrico/ Radio frecuencia/ Cable coaxial	1.7 Kbit/s - 1.28 Mbit/s	1500m - 2700m
X10	Cable eléctrico	50 bit/s - 60 bit/s	
KNX-EIB	Par trenzado/ Cable eléctrico/ Radio frecuencia/ Infrarrojo/ Ethernet	1200 bit/s - 9600 bit/s	300m - 1000m
ZigBee	Radio frecuencia	20 Kbit/s - 250Kbit/s	10m - 75m
Zwave	Radio frecuencia	9.6 Kbit/s - 40Kbit/s	1m - 75m

2.2 Sistema de comunicación y enlace

Una de las partes principales del sistema de domótica implementado es la de comunicación y enlace, la cual se encarga de comunicar al usuario con el administrador de datos y el módulo de control a través de una red de conexiones.

Existen diferentes formas de hacer una red de conexiones para enlazar y comunicar al administrador de datos con el módulo de control y el usuario. Una de ellas es a través de una arquitectura web.

La arquitectura web abarca toda la tecnología utilizada para poner en marcha un servidor que permita a un usuario determinado visualizar contenidos a través de Internet [9].

Un Router puede hacer una red de conexiones para este sistema formando una arquitectura web que no requieren de internet.

En este sentido la arquitectura web es la que permite al usuario (es decir, el cliente web del sistema) interactuar con el sistema de domótica a través del navegador web, haciendo peticiones a un administrador de datos (considerado el servidor web del sistema), a través de un intermediario que puede ser un Router o Internet.

2.2.1 Servidor web

Un servidor web es un programa que funciona en un ordenador de gran capacidad cuya función principal es facilitar información solicitada por otros ordenadores en forma de documentos con una estructura específica. El servidor sólo trabaja cuando se le solicita información, utilizando un protocolo denominado HTTP. Esta técnica es muy eficaz ya que el cliente solo necesita una pequeña parte de los recursos del servidor [10]. En la figura 2.1 se muestra una estructura básica de un servidor web al cual acceden computadoras por medio de internet mediante HTTP.

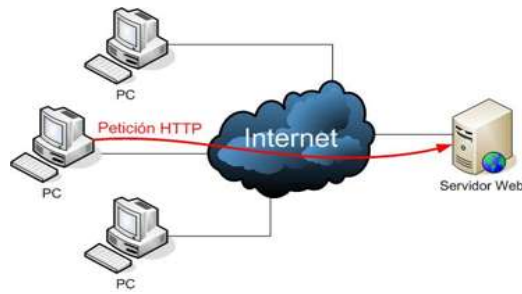


Figura 2. 1 Estructura básica de un cliente/servidor [11].

Para propósitos de domótica, en los que se tiene poco manejo de información y un número pequeño de clientes, basta con un ordenador sencillo que funcione como servidor para el manejo de datos de la casa. El ordenador para este proyecto es una Raspberry Pi.

La Raspberry Pi es un ordenador de bajo costo del tamaño de una tarjeta de crédito, el cual se conecta a una pantalla, utiliza un teclado y un ratón estándar. Es capaz de hacer todo lo que se espera de una computadora de escritorio [12].

El servidor web es el componente de software que hay que instalar y configurar para que el servidor sea operativo. Existen varios programas que permiten a un ordenador actuar como servidor. Siendo uno de los más importantes el Apache creado originalmente para Linux, pero también existen versiones para Windows. Es multiplataforma, gratuito y de código abierto [9].

El servidor web puede llegar a manejar mucha información, por lo que así mismo requiere en la mayoría de los casos un software para gestionar la información de una base de datos como MySQL o MariaDB, aunque hay muchos otros.

Una base de datos (cuya abreviatura es DB) es una entidad en la cual se pueden almacenar datos de manera estructurada, con la menor redundancia posible. Diferentes programas y distintos usuarios deben poder utilizar estos datos. Por lo tanto, el concepto de base de datos generalmente está relacionado con el de red, ya que se debe poder compartir esta información. De allí el término base.

"Sistema de información" es el término general utilizado para la estructura global que incluye todos los mecanismos para compartir datos [13].

Otro software que se requiere en el servidor web es un programa que realice la interface entre el servidor y el cliente (página web); como Java o PHP.

Página web toda página web tiene detrás un código fuente que la define. Cuando se accede a una página web, aunque en el monitor se puedan visualizar documentos con imágenes y texto de diferentes tamaños, colores y formatos, debe tenerse presente que detrás de eso hay un documento de texto que incorpora una serie de instrucciones o comandos que son los que realmente generan la página que se está visualizando. Este documento fuente está definido en un lenguaje especial[14].

Un lenguaje de programación web es aquel que se ejecuta en el servidor. Un programa creado con uno de los lenguajes como PHP puede acceder a recursos (una base de datos, por ejemplo) antes de crear la página que recibirá el cliente como respuesta de la petición.

PHP son las siglas de HiperText Preprocessor Procesador de Híper Texto (originalmente Personal Homes Pages). Este popular lenguaje se presenta en forma de scripts embebidos dentro de las paginas. Es abierto y gratuito y relativamente fácil de usar con las notaciones clásicas de programación, soportando actualmente orientación a objetos. Un código en PHP generalmente es interpretado, aunque también se puede compilar. Entre los sitios web desarrollados con PHP se encuentra Facebook[9].

Se denomina "**LAMP**" a un grupo de software de código libre que se instala normalmente en conjunto para habilitar un servidor para alojar sitios y aplicaciones web dinámicas. Este término en realidad es un acrónimo que representa un sistema operativo Linux con un servidor Apache, el sitio de datos es almacenado en base de datos MySQL y el contenido dinámico es procesado con PHP[15]. Para este sistema de domótica se instaló LAMP en la Raspberry Pi.

Un **Router** es un dispositivo de interconexión de redes informáticas que permite asegurar el enrutamiento de paquetes entre redes o determinar la ruta que debe tomar el paquete de datos.

HTTP (HyperText Transfer Protocol): Protocolo de transferencia de hipertexto: Es el mecanismo más extendido utilizado por los servidores y los navegadores para comunicarse.

Este protocolo regula las solicitudes de los navegadores y la transferencia de ficheros HTML por parte de los servidores

URL(Universal Resources Locator): Localizador Uniforme de Recursos: dirección que localiza una información en Internet.[10].

Cuando un usuario accede a una URL, el cliente web (navegador) consulta al servidor de nombre de dominio, el cual le indica la dirección del equipo deseado[16].

2.2.2 Cliente web

El usuario, considerado cliente web, podrá hacer peticiones al servidor a través de cualquier dispositivo electrónico que tenga un navegador web.

Un cliente web, también llamado navegador, es un programa que se ejecuta en el equipo del usuario y que permite a este solicitar documentos a un servidor. Para ello debe ser capaz de comunicarse con un servidor y comprender los diversos lenguajes y formatos en los que se maneja la información en la WWW[10].

Los clientes web son ejecutados en cualquier dispositivo electrónico como computadoras, celulares, tablets, etc., que tenga acceso a internet. Un esquema común se muestra en la figura 2.2, en el que se representa la comunicación de los clientes y el servidor a través de internet en una arquitectura web.

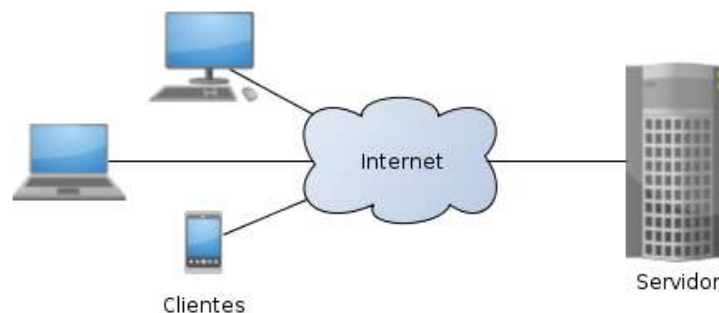


Figura 2. 2 Representación de clientes en una arquitectura web[17]

Frente a los lenguajes del lado del servidor, están los lenguajes de programación del lado del cliente. Estos son independientes del servidor y se descargan e interpretan a través del

cliente que accede a ellos[9]. Por lo que cada dispositivo que se conecte e interactúe a través del navegador web con el servidor web del sistema de domótica, será un cliente. Este cliente requiere de ciertos elementos para poder interpretar la información que se recibe y envía al servidor. Estos elementos son HTML, JavaScript y CSS.

HTML es un lenguaje puramente descriptivo que permite definir las páginas web. Las páginas creadas únicamente con HTML son básicamente estáticas, es decir, siempre muestran la misma información y no ofrecen ningún grado de interactividad con el usuario[14].

JavaScript es un lenguaje interpretado basado en guiones que son integrados directamente en el código HTML. El código es transferido al cliente para que este lo interprete al cargar la página[14].

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas.

Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc [18].

2.2.3 Socket

Para poder enviar y recibir información entre el servidor y el módulo de control es necesario el uso de Sockets. Los Sockets o Zócalos son entidades que nos permiten establecer una conexión entre programas locales o con diferentes computadoras de una red, como se muestra en la figura 2.3.

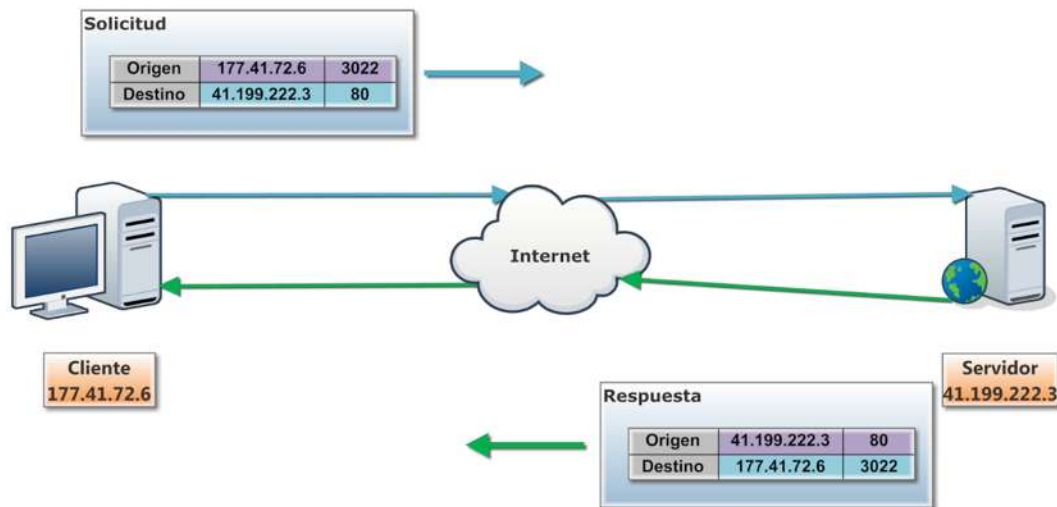


Figura 2.3 Conexión de sockets[20].

Otra manera de entenderlos es como una interfaz de programación de aplicaciones API (Application Programming Interface) Interfaz de Programación de Aplicaciones, que permite al sistema operativo administrar los mensajes.

En Internet los Sockets brindan el mecanismo para la entrega de datos, sus tres características básicas son:

- La IP donde se conecta
- El puerto que se conecta
- El protocolo de comunicación (comúnmente TCP)[19].

IP: (Internet Protocol) Principal protocolo de Internet. Su tarea es entregar paquetes desde una máquina origen a una máquina destino determinada por la dirección IP contenida en los paquetes. En la figura 2.4 se muestra una imagen con computadoras conectadas con diferentes direcciones IP.

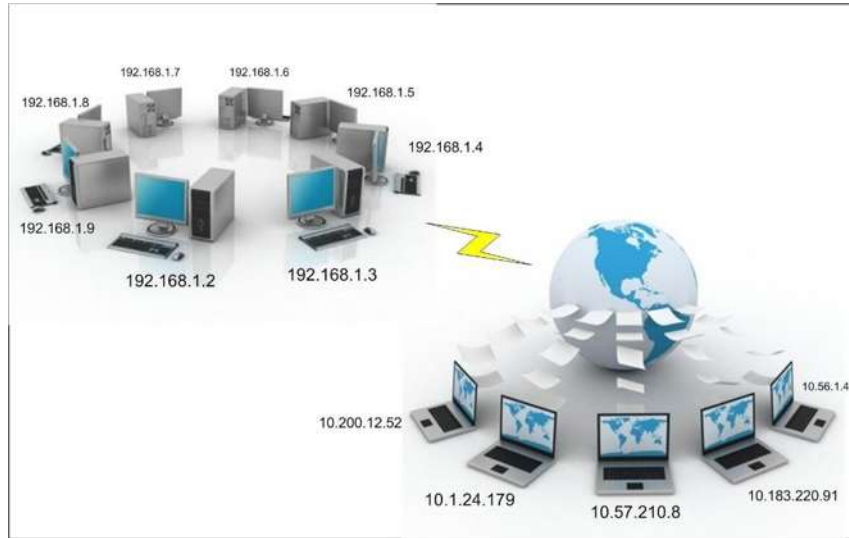


Figura 2. 4 Conexiones de computadores con dirección IP[21]

UDP: protocolo no orientado a conexión. Cada datagrama contiene la información necesaria para llegar a su destino, pero no existe información de entrega ni control de errores, Además los paquetes pueden no llegar en el mismo orden en que son enviados.

TCP: protocolo orientado a conexión. Los paquetes son entregados en orden, existiendo control de errores. [9].

2.3 Módulo controlador

El módulo de control es el encargado de recibir las peticiones del usuario a través del servidor, para conectar o desconectar de la corriente eléctrica a los dispositivos que se conecten a través de él. Se compone de tres partes principales.

- Un administrador de procesos, el cual se encarga de procesar, enviar y recibir la información a otros elementos del módulo de control.
- Elemento de transmisión inalámbrica de datos que se encarga de recibir y enviar información al servidor web y, además, de comunicarse con el administrador de procesos.
- Sistema de potencia, encargado de conectar o desconectar de la corriente eléctrica los dispositivos a través de las ordenes que le mande el administrador de procesos.

Uno de los administradores de procesos más conocido es el microcontrolador.

2.3.1 Microcontrolador

Un microcontrolador (abreviado μC , UC o MCU) es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica. Un microcontrolador incluye en su interior las tres principales unidades funcionales de una computadora: unidad central de procesamiento, memoria y periféricos de entrada/salida [22] [23].

Existen diferentes fabricantes de microcontroladores como Arduino, Microchip, Texas Instrument etc.

2.3.2 Módulo wifi

Existen diferentes elementos de transmisión inalámbrica que utilizan diferentes protocolos de transmisión como Radio Frecuencia, Bluetooth y Wi-Fi, para este sistema de domótica se utiliza uno ellos, el módulo wifi ya que una de las principales ventajas es que puede comunicarse con arquitecturas web.

El módulo wifi es el ESP8266. Espressif Systems' Smart Connectivity Platform (ESCP) es un conjunto de alto rendimiento, alta integración SOC (system-on-chip) Sistema en un circuito integrado, inalámbrica, diseñado para espacio y energía limitados para desarrolladores de plataformas móviles. Proporciona capacidad sin igual para incorporar capacidades wifi dentro de otros sistemas o para funcionar de manera independiente.

El ESP8266EX ofrece una solución completa y autónoma para las redes wifi; puede ser utilizado como sede de la aplicación o para descargar funciones de red wifi de otro procesador de aplicaciones. Alternativamente, actúa como un adaptador wifi. El acceso inalámbrico a Internet puede ser añadido a cualquier microcontrolador basando su diseño de una conectividad sencilla (SPI / SDIO o interfaz / UART I2C). [24].

2.3.3 Relevador de Estado Sólido.

Para el sistema de potencia se ocupa un interruptor eléctrico para que conecte o desconecte la corriente eléctrica del dispositivo a controlar y además una entrada de voltaje que active este interruptor por medio del microcontrolador; para ello son muy convenientes los relevadores de estado sólido.

Un relevador de estado sólido SSR (Solid State Relay) es un switch eléctrico que conmuta entre estados abierto-cerrado. Los relevadores de estado sólido cambian de estado cuando

un voltaje externo es aplicado en una junta, ya sea un voltaje positivo o negativo según el modelo de SSR. A diferencia de los contactores o relevadores electromecánicos los relevadores de estado sólido no cuentan con partes móviles o platinos de contacto, su funcionamiento se basa completamente en electrónica de semiconductores [25].

Capítulo 3 Implementación del Hardware

En esta sección se da a conocer el hardware del sistema y se muestran las conexiones inalámbricas y alámbricas del mismo como se muestra en la figura 3.1. Además, se describe cada parte del sistema correspondiente al Hardware.

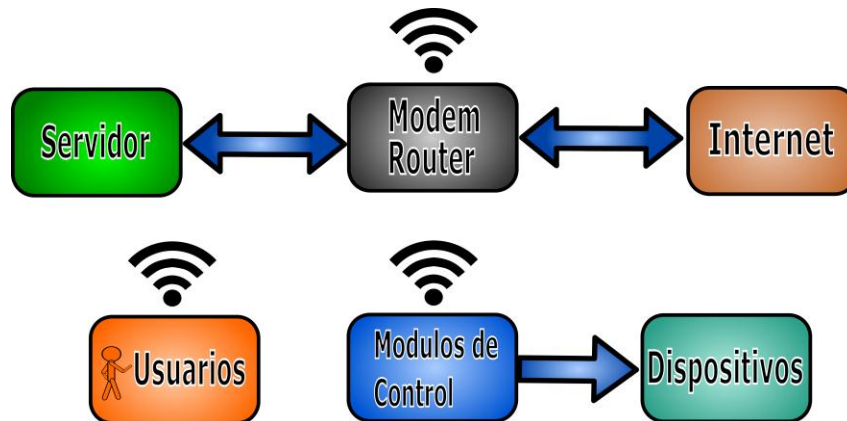


Figura 3. 1 Hardware del sistema a bloques

En la figura 3.1 los componentes de hardware que están conectados de manera alámbrica se señalan con flechas azules que indican el flujo de la información. Los componentes conectados de manera inalámbrica se señalan con el símbolo de wifi en la parte superior del componente del sistema.

Para la descripción del hardware el sistema se divide en tres partes, el módulo de control, el sistema de comunicación y enlace y el cliente web.

3.1 Sistema de comunicación y enlace

El sistema de comunicación y enlace es el encargado de procesar, recibir, transmitir y enlazar la información a través de la arquitectura web del sistema. Una parte importante de este sistema es el cliente web, el cual puede ser cualquier dispositivo electrónico con navegador web, para poder visualizar e interactuar con el sistema. El cliente web se conecta de manera alámbrica o inalámbrica con el sistema de comunicación y enlace. Para este sistema se pueden utilizar dos configuraciones de arquitectura web, con o sin acceso a internet.

En la primera configuración tipo A los clientes web se conectan al servidor web a través de internet, es decir, el administrador de datos del sistema de domótica se encuentra conectado

a internet dentro de la casa y los usuarios pueden acceder a él en cualquier parte donde se tenga acceso a internet, ya sea dentro o fuera de la casa. Esta arquitectura se muestra en la figura 3.2.

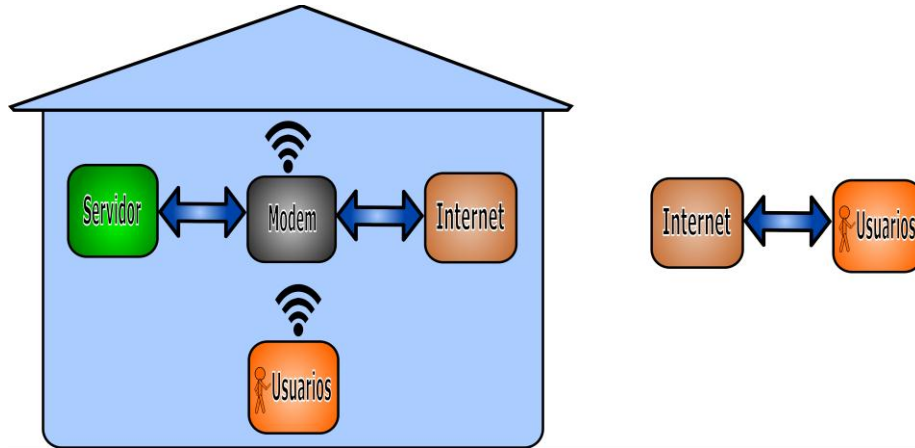


Figura 3. 2 Arquitectura web tipo A.

En la segunda configuración tipo B los clientes web se conectan al servidor web a través de un Router, es decir el administrador de datos del sistema de domótica se encuentra conectado al Router dentro de la casa y los usuarios solo pueden acceder a él estando dentro de la red local, sin importar si se tiene acceso a internet. Esta arquitectura se muestra en la figura 3.3.

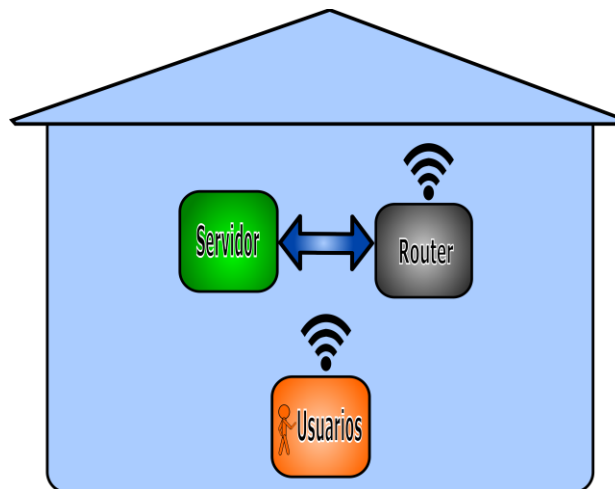


Figura 3. 3 Arquitectura web tipo B.

3.1.1 Servidor

Un servidor, físicamente, es una computadora que esté conectada a una red local de la casa. Para este sistema basta con una computadora de capacidades básicas. Una Raspberry Pi es una computadora de capacidades suficientes para utilizarla como servidor. La que se utiliza en este sistema es una Raspberry Pi 3 Model B, como se muestra en la figura 3.4, la cual tiene las características siguientes:

- Procesador ARMv8 a 1.2GHz de 64 bits cuatro núcleos
- 1GB de RAM
- 4 puertos USB
- Puerto HDMI completo
- Interface serial de cámara CSI (Serial Camera Interface)
- Ranura para tarjeta Micro SD para almacenar de datos y del sistema operativo.
- Puerto de Ethernet
- Procesador de gráficos VideoCore IV 3D [26]



Figura 3. 4 Raspberry Pi 3 modelo B

La Raspberry funge como el servidor y está conectada al Router o Modem para comunicarse con los demás componentes del sistema de manera alámbrica a través del puerto de Ethernet. Esta transmite y recibe información, como se muestra en la figura 3.5

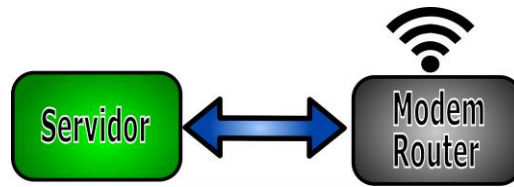


Figura 3. 5 Diagrama esquemático de la conexión del servidor

3.1.2 Router y Modem

El Modem es el que provee internet a la casa, está conectado a internet de manera alámbrica a través de fibra óptica, cable coaxial o cable de teléfono, etc, dependiendo del proveedor de servicios de internet. Un ejemplo de Modem se muestra en la figura 3.6 y en la imagen 3.7 la conexión del Modem con la Raspberry.



Figura 3. 6 Imagen Modem

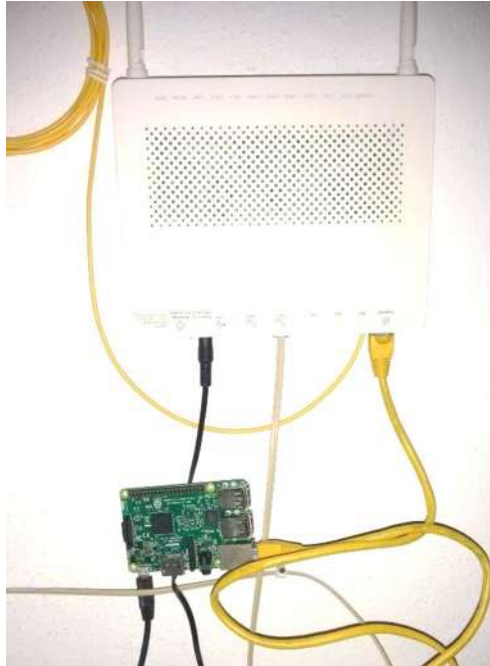


Figura 3. 7 Imagen de conexión del Modem con la Raspberry

El Modem también es el intermediario entre todos los componentes del sistema, enviando y recibiendo información. Está conectado a internet y al servidor de manera alámbrica y de manera inalámbrica con los clientes web (usuarios) y el módulo de control, como se muestra en la figura 3.8.

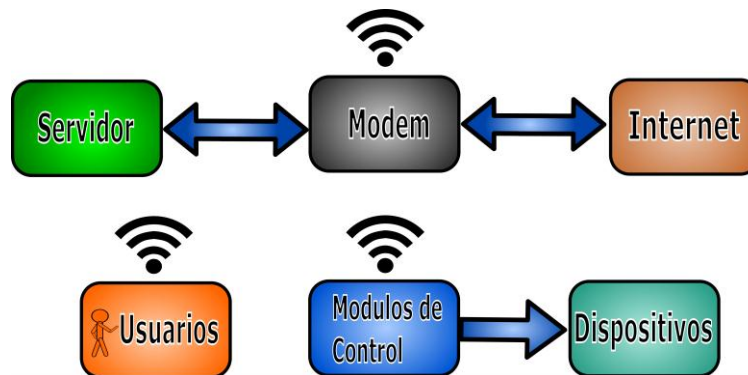


Figura 3. 8 Diagrama a bloques de la conexión del Modem

El Router permite realizar la conexión entre el servidor, los clientes web y el módulo de control sin necesidad de internet, pero éste tiene la restricción de que no se puede acceder al

sistema de domótica fuera de la red local. La imagen de un Router se muestra en la figura 3.9 y en la imagen 3.10 la conexión del Router con la Raspberry.



Figura 3. 9 Imagen Router



Figura 3. 10 Conexión Router con Raspberry

El Router está conectado al servidor de manera alámbrica y de manera inalámbrica con los clientes web y el módulo de control, como se ve en la figura 3.11.

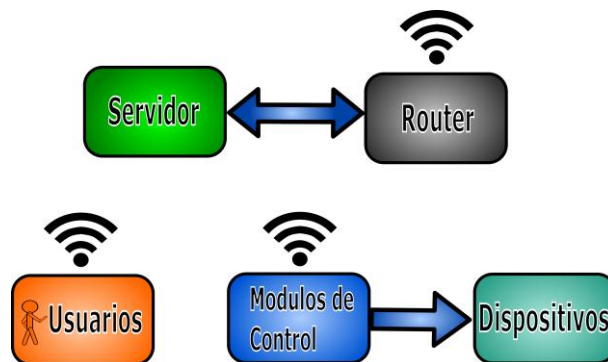


Figura 3. 11 Diagrama a bloques de la conexión del Router

3.2 Modulo controlador

El módulo de control se encarga del manejar a de los dispositivos de la casa. Está conectado de manera alámbrica, por un lado, a la corriente eléctrica de la casa, por otro, al dispositivo que se controla, y de manera inalámbrica al Router o Modem para la transferencia y recepción información, como se observa en la figura 3.12

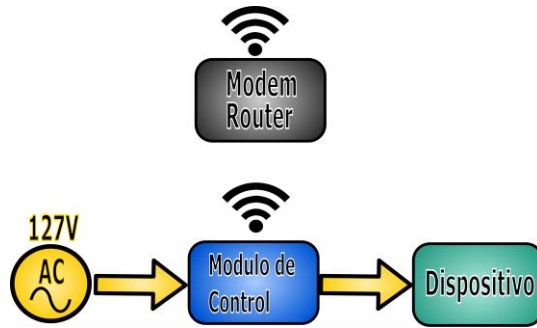


Figura 3.12 Conexión modulo controlador

El módulo de control internamente está compuesto por 4 partes principales:

- Fuente de alimentación.
- Módulo wifi
- Microcontrolador.
- SSR

La corriente alterna (líneas amarillas) se deriva a dos partes del módulo, una para la fuente de alimentación (bloque morado) y otra para el dispositivo a controlar, un polo por el SSR (bloque verde). La fuente proporciona alimentación regulada (líneas naranjas) al módulo wifi y al Microcontrolador. El módulo wifi (bloque azul) se comunica de manera inalámbrica con el router y de forma alámbrica microcontrolador (bloque rojo). El microcontrolador se comunica de igual forma con el módulo wifi enviando y recibiendo información (líneas azules) y activando o desactivando al SSR (bloque verde). El SSR cierra o abre uno de los polos de la corriente alterna. Esta configuración se presenta en la figura 3.13

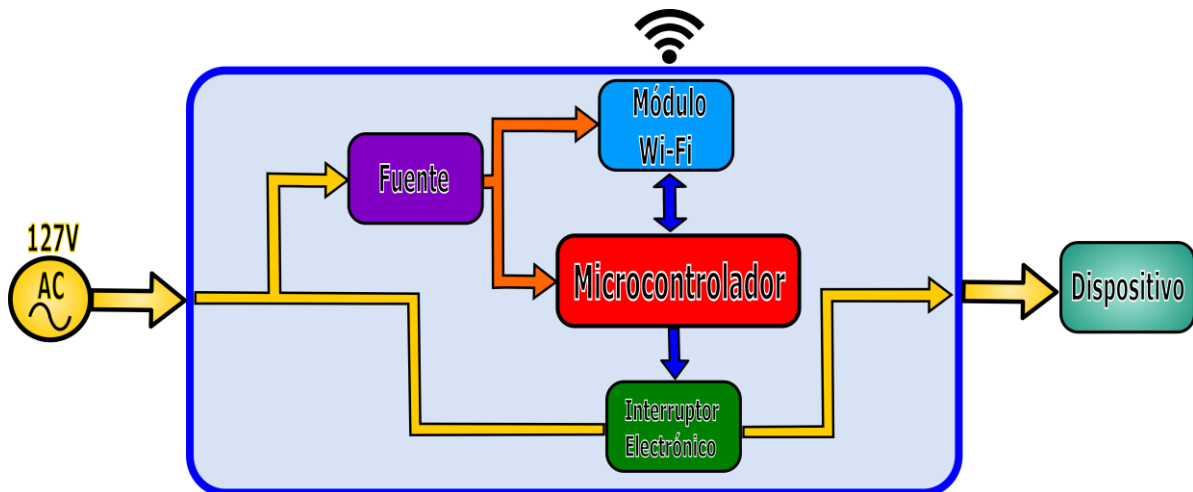


Figura 3. 13 Diagrama a bloques del módulo de control

3.2.1 Fuente

La fuente de alimentación siempre está activa, toma los 127V de corriente alterna de la casa (líneas amarillas) y los transforma a 3.3V (línea naranja) y su tierra (líneas negras) para alimentar al Microcontrolador y al módulo wifi, como se muestra en la figura 3.14.

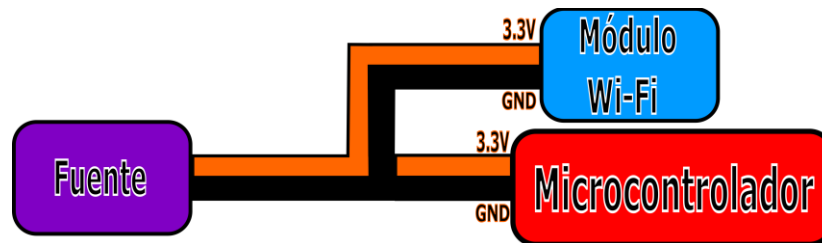


Figura 3. 14 Diagrama a bloques de la conexión de la fuente

Esta fuente fue construida utilizando un cargador de celular que toma la corriente alterna de 127Volts y la transforma a 5Volts de corriente directa a 1 Ampere, utilizando un regulador AMS1117 que regulan los 5Volts a 3.3Volts de corriente directa.

3.2.2 Módulo wifi

El módulo wifi que se utiliza es un ESP8266. Tiene una parte en específico dentro del mismo llamada UART (Universal Asynchronous Receiver-Transmitter), la cual es utilizada

para comunicarse con el microcontrolador de manera alámbrica. En la figura 3.15 se muestra una imagen del módulo wifi.

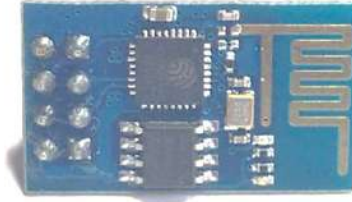


Figura 3. 15 Imagen del módulo wifi

El módulo wifi tiene 8 pines, los cuales son:

- VCC
- GND
- UTXD
- URXD
- RST
- CH_PD
- GPIO2
- GPIO0

La ubicación de los pines en el módulo wifi se observan en la figura 3.16

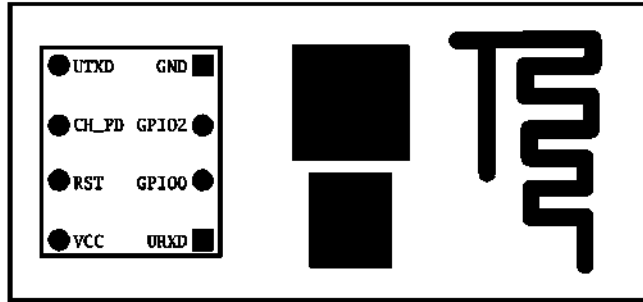


Figura 3. 16 Ubicación de los pines del módulo wifi[27].

En su conexión se utilizan tres cables para comunicación serial con el microcontrolador, uno para transmisión (UTXD) que va a RX del microcontrolador, otro para recepción (URXD) que va a TX del microcontrolador, y otro para tierra (GND) que va GND del microcontrolador; otros dos cables para la alimentación de la fuente de 3.3V (VCC) y tierra (GND) y un último para su habilitación (CH_PD) que va a (VCC). Los pines restantes no se conectan. También se comunica al Router o Modem de manera inalámbrica. Esta conexión se muestra esquemáticamente en la figura 3.17.

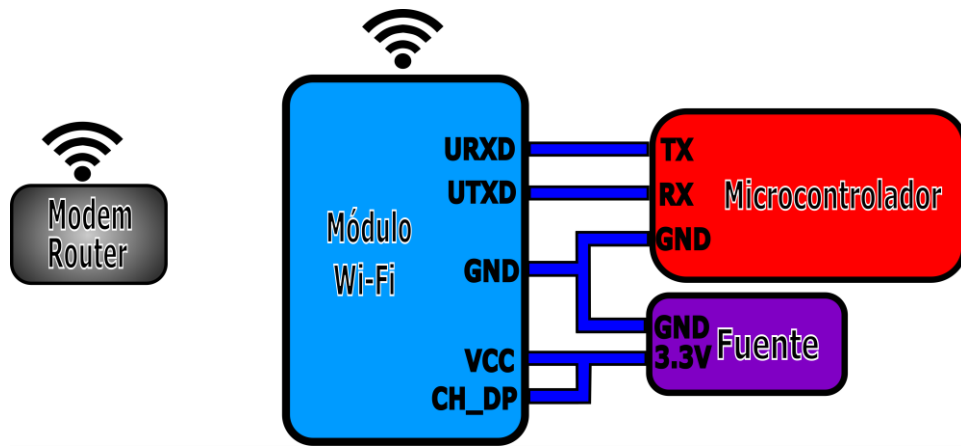


Figura 3. 17 Conexión esquemática del módulo wifi

3.2.3 Microcontrolador

El microcontrolador que se utiliza es un MSP430G2553 de Texas Instruments. En la figura 3.18 se ve una imagen del mismo.



Figura 3. 18 Imagen del microcontrolador

La ubicación de los pines del microcontrolador se muestra en la figura 3.19

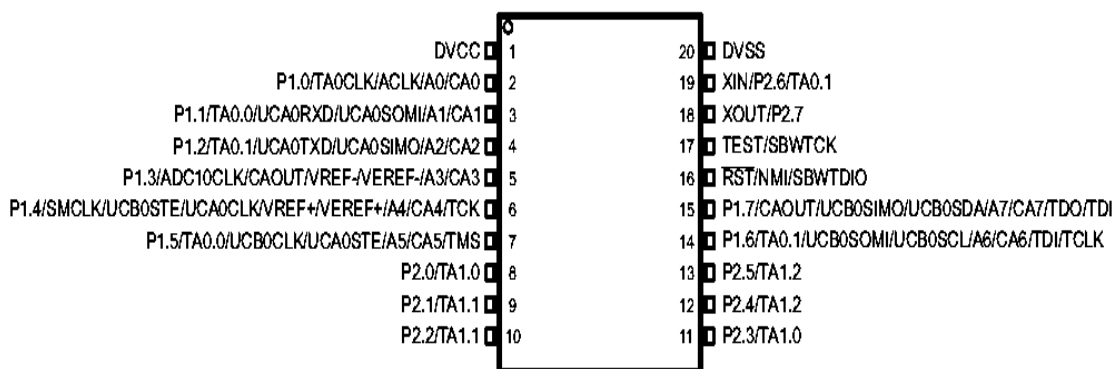


Figura 3. 19 Ubicación de pines del microcontrolador [24].

El microcontrolador cuenta con las siguientes características:

- Bajo rango de voltaje de alimentación: 1.8 V to 3.6 V
- Consumo de poder ultra bajo
 - Modo activo: 230 μ A a 1 MHz, 2.2 V
 - Modo de espera (Standby): 0.5 μ A
 - Modo apagado (retención de RAM): 0.1 μ A
- Cinco modos de ahorro de energía
- Reactivación ultra rápida de modo de espera (Standby) en menos de 1 μ s
- Arquitectura 16-Bit RISC (Reduced Instruction Set Computer) Computador con Conjunto de Instrucciones Reducidas, Tiempo de 62.5-ns en un ciclo de instrucción.
- Configuraciones básicas de los módulos de reloj.
 - Frecuencias internas de hasta 16 MHz con cuatro modos para la calibración de la frecuencia.
 - Oscilador interno de ultra bajo consumo de baja frecuencia (LF)
 - 32-kHz Cristal
 - Fuente externa de reloj digital
- Dos Contadores (timer) de 16-Bits con tres registros de Captura/Comparación
- Interface Universal de Comunicación Serial (USCI), UART Transmisor Receptor Universal mejorado, soporta la detección automática de la velocidad de transmisión.
 - IrDA Codificador y Decodificador.
 - Protocolo I²C™(Inter-Integrated Circuit)

- La programación serial sobre la tableta no necesita voltaje externo, cuenta con protección del código programado por un fusible de seguridad
- Opciones de empaquetado
 - TSSOP: 20 Pines, 28 Pines
 - PDIP: 20 Pines
 - QFN: 32 Pines

En la tabla 3.1 se muestran las características específicas del microcontrolador MSP430F5529IN20 que se usa para este proyecto.

Tabla 3.1 Características de microcontroladores MSP430G2553IN20 [28]

BSL	EEM	Flash (KB)	RAM (B)	Contador_A	COMP_A+	ADC10	USCI_A0, USCI_B0	Reloj	I/O	Tipo de empaquetado
1	1	16	512	2x TA3	8	8	1	LF, DCO, VLO	16	20-PDIP

El microcontrolador tiene dentro del mismo un dispositivo UART llamado UCA0 que se utiliza para comunicarse con el módulo wifi conectándose con los tres mismos cables RX, TX y GND. Además, se conecta con el SSR para mandar una salida digital por medio de una resistencia de 220Ω al puerto P1.0 del microcontrolador y un cable a tierra. También ocupa de alimentación de la fuente utilizando 3.3V y GND. Estas conexiones se observan en la figura 3.20.

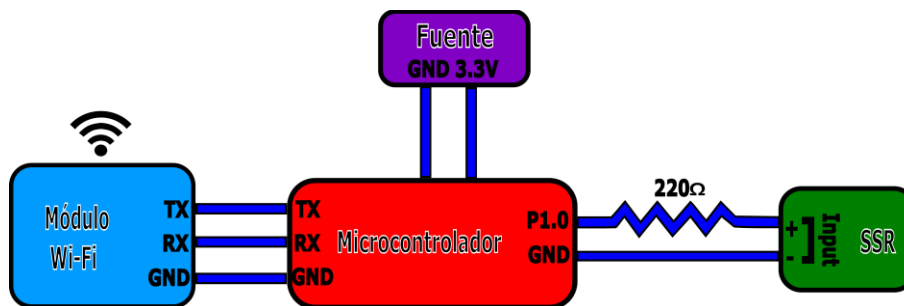


Figura 3. 20 Conexión esquemática del microcontrolador.

El valor de la resistencia de 220Ω se obtiene a partir de la ley de Ohm, teniendo en cuenta que la salida digital del microcontrolador es de $3.3V$ y que la entrada del relevador para su activación ocupa de entre 15 y $20mA$. Para calcular el valor de la resistencia a partir de estos datos se tiene la siguiente ecuación.

$$R = \frac{V}{I} \qquad R = \frac{3.3V}{15mA} \qquad \text{Resistencia} = 220\Omega$$

3.2.4 SSR

Para realizar la conexión del SSR se conecta la entrada del relevador con el microcontrolador mediante una conexión del (input +) con el pin P1.0 a través de una resistencia, además se une el (input -) al GND.

Para la salida (output), una de sus terminales se conecta a uno de los polos de la corriente alterna de la casa y la otra terminal se conecta directamente al dispositivo a controlar.

Uno de los polos restantes de la corriente alterna pasará directamente al dispositivo a controlar. Esta conexión se muestra de manera esquemática en la figura 3.21.

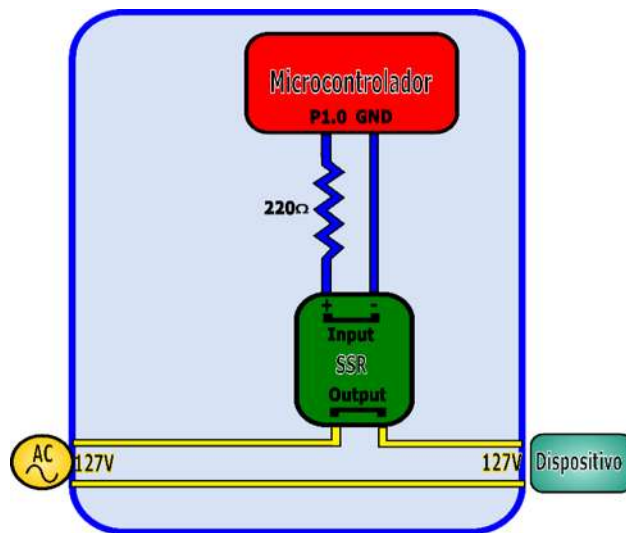


Figura 3. 21 Conexión esquemática del SSR

El SSR conectará o desconectará de la corriente eléctrica a los dispositivos que se conecten al módulo de control, cerrando o abriendo uno de los polos del circuito que proviene de la corriente alterna, el mismo módulo se conecta directamente a la corriente alterna de la casa.

Capítulo 4 Implementación del Software

En este capítulo se ofrece una descripción de los componentes del software como son sus interconexiones, diagramas de flujo y el pseudocódigo de cada uno. El sistema tiene dos partes esenciales de software:

- Sistema de comunicación y enlace.
- Módulo de control

4.1 Software Sistema de comunicación y enlace

Este sistema se encarga de comunicar y enlazar al módulo de control con el cliente web. El cliente web se enlaza al servidor para transmitir y recibir instrucciones al módulo de control a través del servidor.

El software del sistema de comunicación y enlace se divide en dos partes, la instalación del servidor y la programación de la página web.

4.1.1 Servidor

Para la instalación del servidor en la Raspberry Pi se realizaron los siguientes pasos:

- Instalación del sistema operativo Raspbian Jessie with Pixel en la Raspberry Pi.
- Instalación de Apache, MySQL y PHP en Raspberry Pi.
- Programación de la página web.

La instalación del sistema operativo se puede realizar de forma sencilla entrando a la página web oficial de Raspberry en downloads y escogiendo raspbian, o bien ingresando a en la siguiente liga <https://www.raspberrypi.org/downloads/raspbian/>, dentro habrá una guía de instalación y los elementos que se ocupan.

De igual forma para el servidor se puede ingresar desde la página web oficial y buscando LAMP con el buscador de la página web, aparecerán diferentes apartados, se escoge el que muestre como instalar LAMP, o también se puede ingresar a la siguiente liga <https://www.raspberrypi.org/learning/lamp-web-server-with-wordpress/worksheet/> en esta misma página web enseña cómo diseñar una página web sencilla.

4.1.2 Software Pagina web

El software de la página web se diseñó en PHP, instalado en el servidor, por lo que la edición y la programación se hace dentro del mismo, por medio de la terminal del sistema operativo.

Para que la persona pueda visualizar el contenido de la página web, el dispositivo que utilice debe tener un navegador web (cliente web) que sirve como el intérprete entre la persona y la página web.

En la página web, cuando se presiona un botón que escogió el usuario, perteneciente a un dispositivo a controlar específico, se realiza el encendido o apagado del dispositivo en cuestión. Esta información se envía al servidor y es procesada por el mismo. Una vez procesada, el servidor se comunica con los módulos de control para enviar el dato que realiza el control del dispositivo. La operación de la página web se muestra en el diagrama de flujo de la figura 4.1

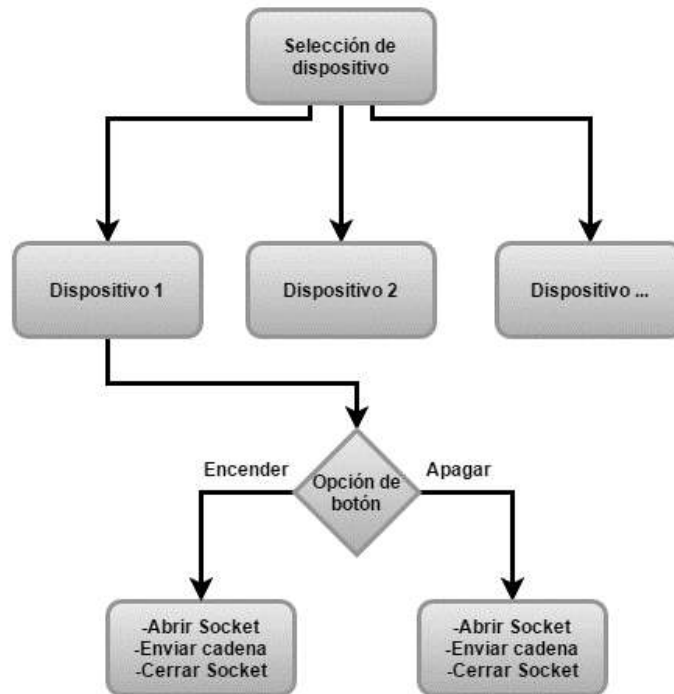


Figura 4. 1 Diagrama de flujo de la Pagina web.

En la figura 4.2 se presenta un ejemplo de la interface de la página web, en la cual se tienen dos dispositivos, una lámpara de la entrada principal y otra lámpara del patio trasero, cada dispositivo cuenta con dos botones, una opción para encendido y otra para apagado además de dos rectángulos para el monitoreo de su estado.



Figura 4. 2 Imagen de la página web

Selección del dispositivo a controlar En la ventana del navegador web se muestran los diferentes dispositivos a controlar, cada uno tiene dos opciones disponibles, activar o desactivar; al seleccionar alguno de los botones también se está escogiendo el dispositivo a controlar.

Opción del botón. Una vez que el usuario ha seleccionado el botón este realiza la petición, ya se encender o apagar el dispositivo seleccionado. Enseguida, suceden una serie de instrucciones:

Encender/Apagar. La opción seleccionada del botón del dispositivo crea una conexión con socket, que contiene la dirección IP y el puerto, escogiendo así el módulo de control al que se enviará la información, para que se encienda o apague el dispositivo; una vez enviada la cadena de caracteres se cierra el socket y queda a la espera de la nueva interacción del usuario.

4.2 Software Módulo de control

El módulo de control contiene un microcontrolador que se encarga de recibir, enviar y procesar la información que llega a través del módulo wifi, para activar o desactivar el SSR. El diagrama de flujo de la figura 4.3 representa con flechas la recepción y transmisión de la información entre estos tres componentes del módulo de control.

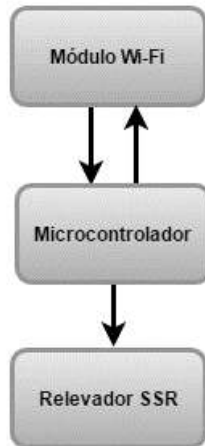


Figura 4. 3 Diagrama de flujo del módulo de control

4.2.1 Software Módulo wifi

El microcontrolador transfiere y recibe datos a través del protocolo serial (RS232); el módulo wifi a su vez, se comunica con el microcontrolador por medio de este mismo protocolo. Esta comunicación se realiza por el puerto serial del módulo wifi y del microcontrolador.

La comunicación con el módulo wifi y el microcontrolador se realiza a través de comandos AT que son una serie de instrucciones para configurar, enviar y recibir información, estos comandos los envía el microcontrolador y el módulo wifi le manda su respuesta del comando específico que recibió. Lo anterior se representa en el diagrama de flujo de la figura 4.4.

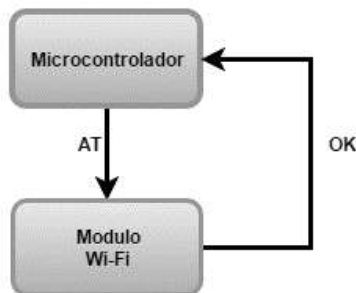


Figura 4. 4 Diagrama de flujo de comandos AT

El diagrama de flujo de la figura 4.4 representa la comunicación entre el microcontrolador y el wifi: el microcontrolador envía el comando “AT” y el módulo wifi le responde un “OK”.

Existen diferentes comandos para la comunicación con el módulo wifi, en la tabla 4.1 se enlista una serie de comandos, su respuesta y la descripción de lo que hace el módulo con el comando.

Tabla 4.1 Comandos AT del módulo wifi

Comando	Respuesta	Descripción
AT	OK	Prueba de Inicio
AT+RST	OK	Reiniciar módulo
AT+CWMODE	OK	Modo del wifi (Punto de acceso/ Estación/ Punto de acceso + Estación)
AT+CIPMUX	OK	Conexiones (Única/Múltiples)
AT+CIPSERVER	OK	Modo Servidor
AT+CWSAP	OK	Configurar Red (nombre, contraseña, conexiones, seguridad)
AT+CIPAP	OK	Establecer dirección IP

4.2.2 Control del SSR

Para la comunicación con el SSR, el microcontrolador solo tiene que enviar una salida lógica por el puerto P1.0 a la entrada (Input) del relevador para activarlo o desactivarlo. Se presentan dos estados posibles de su funcionamiento:

- Cuando la salida del microcontrolador es un uno lógico, la salida (output) del relevador cierra el circuito.
- Cuando la salida del microcontrolador es un cero lógico, la salida (output) del relevador abre el circuito.

4.2.3 Software del Microcontrolador

El software del microcontrolador se compone por la función principal, donde se configuran los elementos que se van a utilizar, las variables y, además, para este microcontrolador en específico, se requiere de interrupciones que también se configuran. La estructura del software del microcontrolador se exhibe en el diagrama de flujo de la figura 4.5.

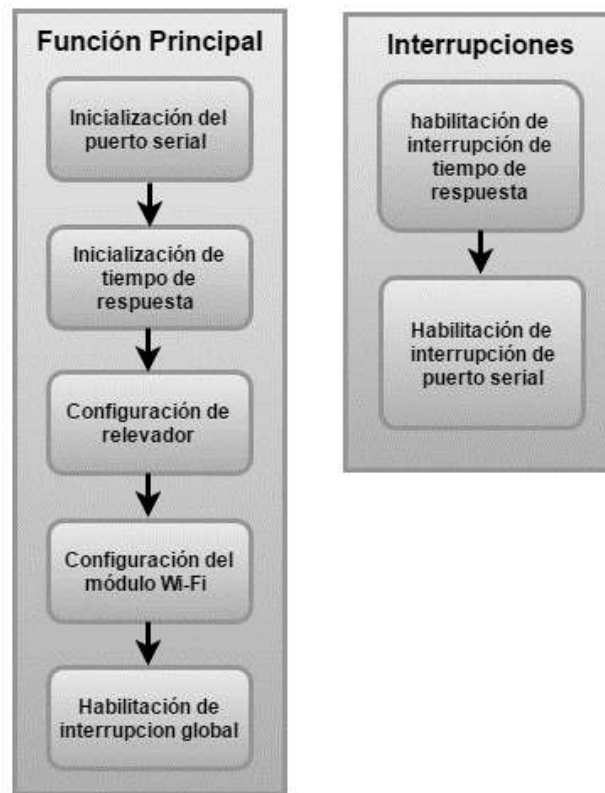


Figura 4. 5 Diagrama de flujo del software del microcontrolador

El diagrama de flujo de la figura 4.5 representa las dos partes esenciales del microcontrolador: la función principal y las interrupciones. Éstas son la base del software del microcontrolador, la función principal se encarga de la configuración e inicialización, mientras que las interrupciones lo hacen de las acciones y procesos que debe realizar el microcontrolador.

Dentro del software del microcontrolador se utilizan partes en específico para la realización de este proyecto, como son:

- El puerto serial que se usa para comunicación, mediante la transferencia y recepción de datos en serie, a través de un protocolo llamado RS232, el cual utiliza cadenas de datos de 1byte a una velocidad especificada en bits por segundo (bps). Este protocolo es empleado por el microcontrolador y el módulo wifi para comunicarse.
- El timer es un temporizador contador que se encarga de iniciar un conteo y terminar en un tiempo definido exacto, este conteo puede ser ascendente, descendente o continuo. Existen varias formas de configurarlo y usarlo. Este timer es utilizado por el microcontrolador para establecer un tiempo de respuesta de espera entre recepción y transmisión de datos.

Función principal

Esta función es la base del microcontrolador. Se encarga de la inicialización y la configuración de los elementos que utilizará el microcontrolador para comunicarse con los dispositivos externos, como el puerto serial y el tiempo de respuesta, los cuales ocupan inicializarse mientras que el módulo wifi y el relevador son configurados. En la siguiente sección se enlista la inicialización y configuración de cada bloque de la función principal que se observa en el diagrama de flujo de la figura 4.5.

Inicialización del puerto serial: Esta sección del código ocupa tres partes para su inicialización:

1. La habilitación del puerto serial -puesto que es un parte del microcontrolador que se utilizara-.
2. La configuración de la velocidad -en este caso es a 9600bps, ya que es la velocidad con la cual se puede comunicar con el módulo wifi-.
3. Por último, la habilitación de las interrupciones del puerto serial, para que se activen solamente cuando se requiera comunicar con el puerto serial y cuando no, esté en modo de bajo consumo.

El pseudocódigo se muestra a continuación:

Inicialización del puerto serial.
Habilitar puerto serial.
Configurar velocidad (9600bps).
Habilitar interrupciones de puerto serial.

Inicialización de tiempo de respuesta: En este punto el código ocupa tres partes para su inicialización:

1. La habilitación del timer -el cual es el encargado de esperar un tiempo definido de espera para la recepción y transmisión de datos-.
2. Definir el tiempo base mínimo que va a esperar y de ser necesario se incrementa.
3. La habilitación de las interrupciones del timer para que se activen solamente cuando termine de transmitir o enviar una cadena de datos.

El pseudocódigo se muestra a continuación:

Inicialización de tiempo de respuesta.
Habilitar de timer
Definir tiempo base(0.1s)
Habilitar interrupciones timer

Configuración del relevador: En esta parte del código se configuran los pines digitales que se utilizan para activar o desactivar los relevadores, definiendo inicialmente los pines del microcontrolador P1.0 o los pines que se ocupen, como salidas digitales. Se les iniciará con un estado digital para que estén apagados cuando se encienda el microcontrolador. El pseudocódigo se muestra a continuación:

Configuración del relevador.
Definir pines digitales salida
Inicializar el estado de los pines digitales

Configuración de módulo wifi: En esta parte del código se configura e inicializan parámetros del módulo wifi de modo que esté listo para enviar y recibir datos con el servidor. Se establece que funcione como estación – este modo permite enviar y recibir datos-, se le asigna una dirección IP al módulo, se configuran los parámetros de red como el nombre, la contraseña y la seguridad, se permite que acepte múltiples conexiones y esté en modo servidor, el cual se conectará a un socket con su IP y puerto específico. El pseudocódigo se muestra a continuación:

Configuración de Módulo wifi

- Confirmación de respuesta del modulo
- Reinicio del módulo
- Configuración en modo estación
- Asignar dirección IP del modulo
- Configurar red seguridad, nombre y contraseña
- Establecer múltiples conexiones
- Configurar modo servidor y puerto

Habilitación de interrupciones globales: En esta sección se habilitan las interrupciones generales del microcontrolador, es decir el microcontrolador se configura para que trabaje con interrupciones y sólo utilice las que se designen y, por tanto, entre en modo de bajo consumo.

El modo de bajo consumo hace referencia a que el microcontrolador consume poca energía, operando solamente con los módulos necesarios, saldrá de tal estado en caso de que se active alguna interrupción, atendiéndola, y regresando al modo de bajo consumo.

Interrupciones

Hay tres interrupciones principales que utiliza el microcontrolador: éstas son la interrupción del timer, que es el tiempo de respuesta, y las del puerto serial que son dos, la transmisión y la recepción. En la figura 4.6 se muestra el diagrama de flujo de la operación de estas interrupciones.

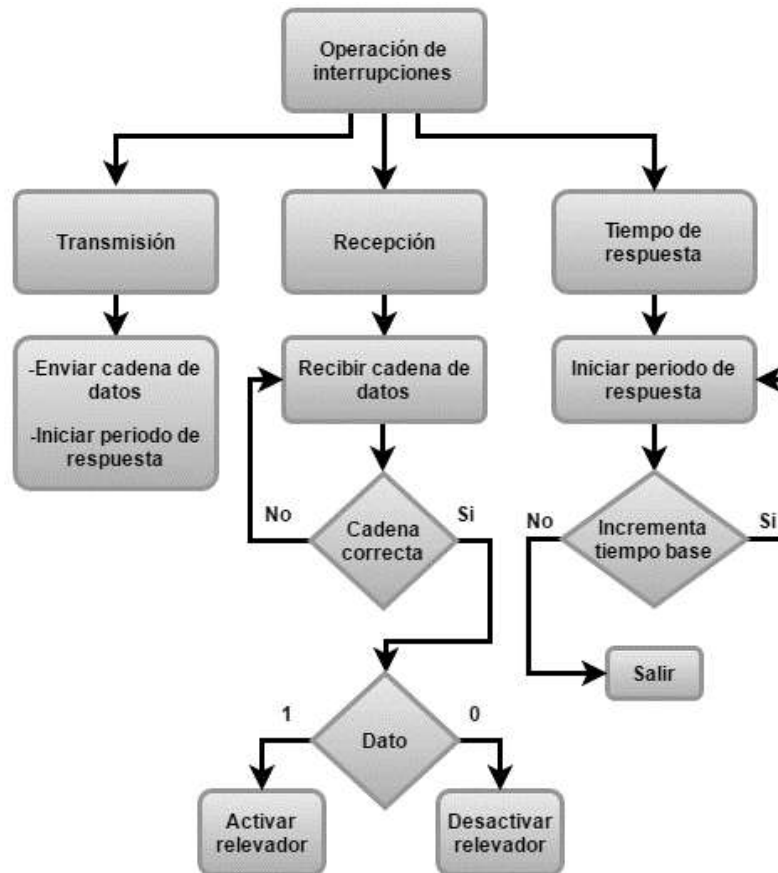


Figura 4. 6 Diagrama de flujo de la operación de las interrupciones.

El microcontrolador, cuando no está realizando ninguna acción, está en modo de bajo consumo. Estas tres interrupciones se activan solamente cuando se llama a la interrupción, realizando la acción que hay en la interrupción y regresando al modo de bajo consumo. Es posible que desde una interrupción se puede mandar llamar a otra. A continuación, se describe la operación de estas tres interrupciones:

Transmisión: La interrupción de transmisión se activa cuando se envía una cadena de caracteres hasta el término de fin de la misma, transfiriendo un dato al módulo wifi. Cuando concluye la interrupción de la transmisión del puerto serial se inicia un periodo de respuesta con la ayuda de la interrupción del tiempo de respuesta, para aguardar un tiempo a que se terminen de transmitir los datos y el módulo wifi finalice de recibir y procesar los datos.

Recepción: La interrupción de recepción se activa cuando recibe una cadena de caracteres hasta el término de fin de la misma, cuando esto ocurre se verifica si una cadena es válida, es decir, si se recibió la información correcta o cadenas desconocidas o erróneas. Si esto último pasa se regresará a esperar para recibir nuevamente. Si la cadena es correcta, se analizará el dato recibido, ya sea cero o uno, y se realizará la acción para activar o desactivar al relevador mandando una salida lógica al puerto digital del microcontrolador.

Tiempo de respuesta: La interrupción del tiempo de respuesta se activa cuando se requiere esperar un tiempo. Se define un periodo de tiempo que dura un número determinado de veces el tiempo base del contador (timer). Una vez definido el periodo de tiempo, el tiempo de respuesta revisará si este se cumple, si no sucede así se incrementará y realizará nuevamente el conteo del tiempo base, pero si se cumple durante este periodo de respuesta no realizará más incrementos del tiempo base y saldrá de la interrupción.

Capítulo 5 Pruebas y resultados

En este capítulo se realizarán y explicarán las pruebas del funcionamiento básico de cada uno de los dispositivos que componen el sistema de domótica. Además, se expondrán los resultados obtenidos en las dos partes esenciales del sistema:

- El módulo de control
- El sistema de comunicación y enlace.

5.1 Sistema de comunicación y enlace

En esta parte del sistema se efectuarán las pruebas a la página web y su interacción con el cliente web (navegador web). La primera prueba es el funcionamiento de la página. Para comprobar que está funcionando simplemente se tecldea la dirección IP del servidor o también el URL y las posibles subcarpetas que se tengan; por ejemplo, escribiendo en el navegador web `http://localhost/Rasp/index.php`. En la figura 5.1 se muestra la interface de la página.



Figura 5. 1 Visualización de la interface de la página web.

Si se logra visualizar la página web entonces el servidor se encuentra operando correctamente.

La siguiente prueba es la operación correcta de los botones de cada dispositivo. Para comprobar que funcione de manera correcta, se dará clic a uno de los botones y este mostrará un mensaje en la parte de arriba, y además el estado en cuadro de texto delante del dispositivo, indicando la acción realizada con el botón. La figura 5.2 muestra una imagen de la respuesta de botón.

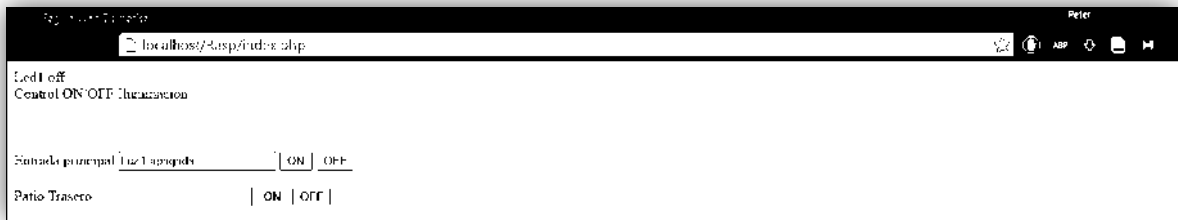


Figura 5. 2 Respuesta del botón

El botón manda la respuesta “Led1 off” en la parte de arriba y “Luz 1 apagada” en cuadro de texto de la página, de esta manera comprobamos el correcto funcionamiento de los botones.

La siguiente prueba es la respuesta de los sockets, para ello se utiliza un programa para dispositivos Android disponible en la PlayStore llamado “Simple Socket Tester”, el cual permite enviar y recibir información a través de un socket. En el programa Simple Socket Tester se establece como servidor por defecto y una configuración predeterminada con una dirección IP local 192.168.100.2 con el puerto 1300 y además el programa muestra la información recibida de la acción del botón, la conexión y envío de la información al socket, como se observa en la figura 5.3.



Figura 5. 3 Respuesta del botón en la aplicación Simple Socket Tester

Al dar clic en el botón de encendido de la entrada principal se actualizará el estado y mandará la información a través del socket conectándose el servidor que contiene la página como cliente. La figura 5.4 muestra una imagen de página al presionar el botón.



Figura 5. 4 Respuesta del botón para el socket

Con lo que se puede comprobar que se actualizó el estado de la página correctamente.

Por este medio se comprueba que se envió la información correctamente y por tanto los sockets de la página funcionan.

Con estas pruebas se verifica el funcionamiento de la interface de la página web y los sockets, así mismo, se comprueba el correcto funcionamiento del servidor web ya que es donde se comunica y enlaza la información y sin éste no funcionaría la página web. De esta forma se comprueba así el correcto funcionamiento del sistema de comunicación y enlace.

5.2 Módulo de control

En esta sección se realizan pruebas a las tres partes esenciales del módulo de control compuesto principalmente por:

- El módulo wifi
- El microcontrolador
- El relevador.

A continuación, se dan a conocer los resultados.

5.2.1 Módulo wifi

En esta parte se realizan las pruebas correspondientes al módulo wifi, mediante los comandos AT mencionados en el capítulo anterior del módulo de control.

La primera prueba que se realiza al módulo wifi es el envío de los comandos AT para su configuración, estas pruebas se realizan con el módulo wifi conectado a otro módulo convertidor de puerto serial a USB. En la figura 5.5 muestra esta conexión.

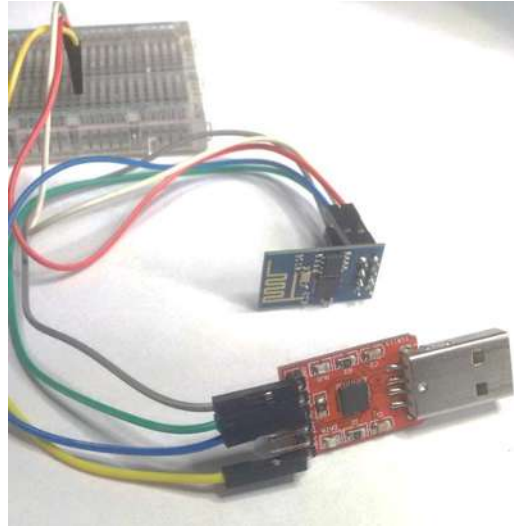


Figura 5. 5 Conexión del módulo wifi al módulo convertidor de puerto serial a USB

Se utiliza una interface para visualizar la recepción y transmisión de los datos, esta se llama CoolTerm, aunque existen muchas otras interfaces que cumplen el mismo propósito de mostrar los datos, se escogió esta porque es la que el autor conoce más. La interface de CoolTerm se muestra en la figura 5.6.

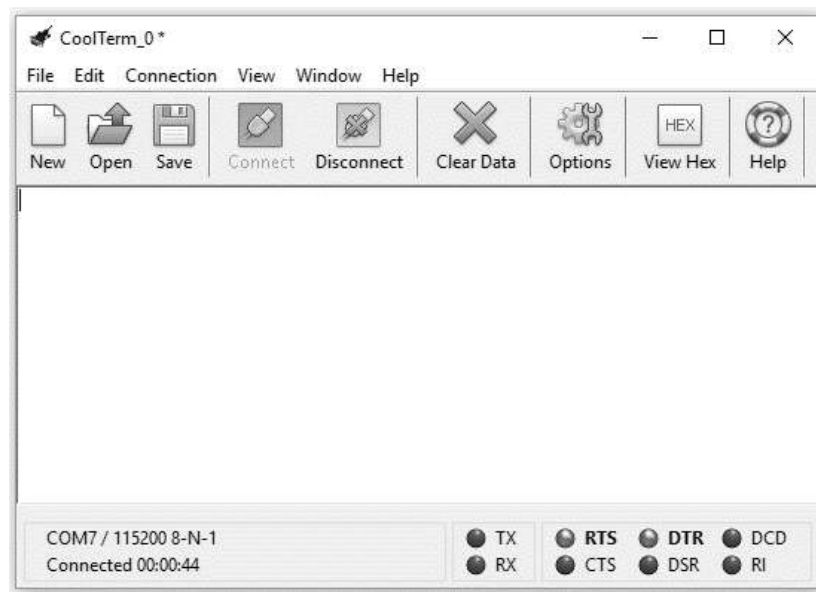


Figura 5. 6 Interface de CoolTerm

La primera prueba que se realiza es la respuesta del módulo ante el comando AT, se escribe el comando “AT” y si el dato llegó bien el módulo responderá con “OK”, comprobando así que funciona correctamente. La figura 5.7 muestra la prueba del módulo en la interface.

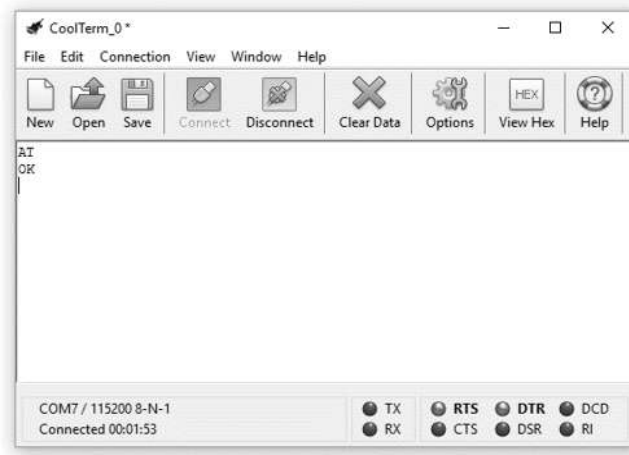


Figura 5. 7 Prueba y respuesta del módulo wifi en la interface.

La siguiente prueba es la configuración básica de este mismo módulo wifi mediante la misma interface CoolTerm, a través de los comandos AT, se manda el comando “AT+RST” y se obtiene la respuesta “OK” y una serie de datos que se muestran en la figura 5.8.

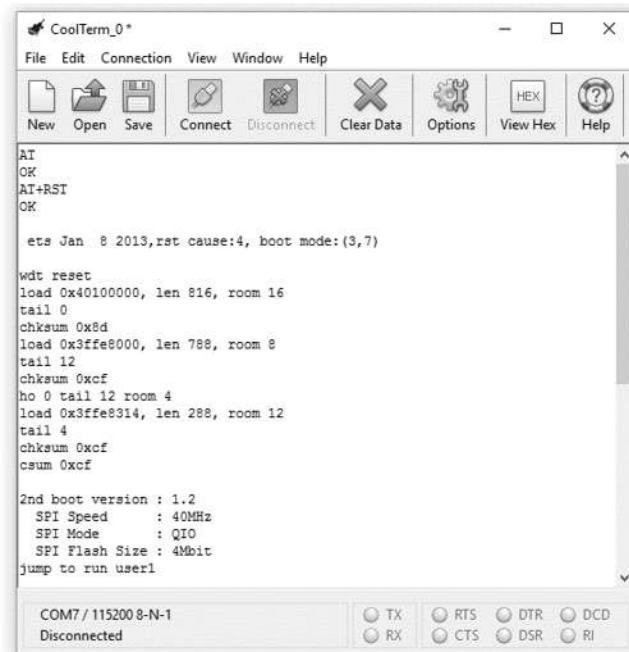


Figura 5. 8 Prueba AT+RST del módulo wifi en la interface CoolTerm

Después se manda los comandos:

```
AT+CWMODE
```

```
AT+CIPMUX=1
```

```
AT+CIPSERVER=1,13000
```

```
AT+CIPSTA="192.168.100.15"
```

```
AT+CWAUTOCONN=1
```

```
AT+CWJAP="Black Cyber","35LP64036"
```

La prueba de estos comandos y su respuesta se observa en la figura 5.9

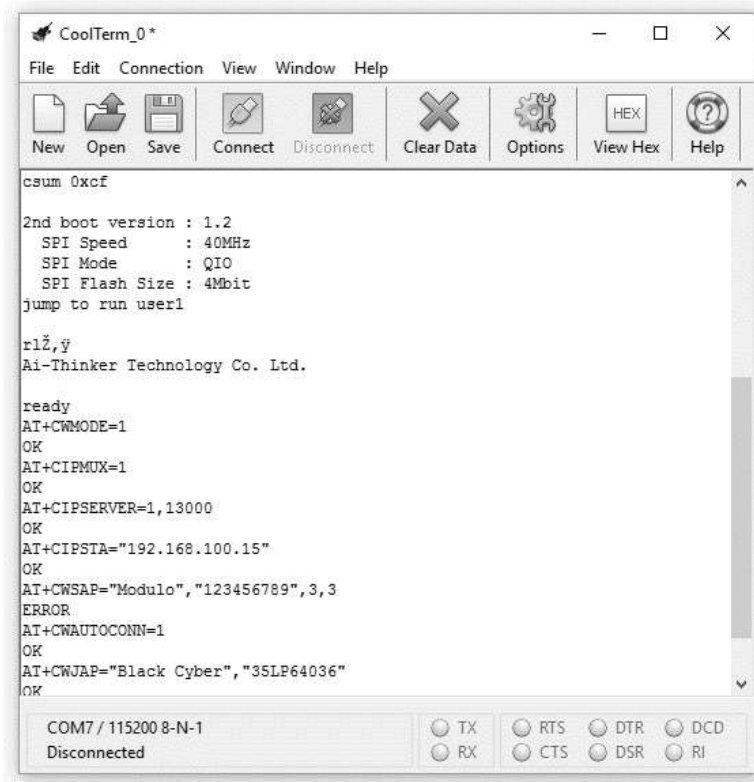


Figura 5. 9 Pruebas de comandos del módulo wifi en la interface CoolTerm

De este modo, mediante las respuestas que se muestran en el capítulo anterior en el software del módulo wifi específicamente en el apartado 4.2.3, se comprueba que se ha configurado correctamente, es decir, que recibe y envía información a través de los comandos.

La última prueba para comprobar el correcto funcionamiento es el envío de información, de igual forma, por medio del programa Android Simple Socket Tester configurado como cliente para conectarse al servidor del módulo wifi, y además se manda la cadena de datos “Luz 1 encendida” para comprobar si recibe de manera correcta la información. En la figura 5.10 se muestra una imagen del programa Simple Socket Tester como cliente “Luz 1 encendida” enviando la información.



Figura 5. 10 Programa Simple Socket Tester enviando “Luz 1 encendida”

Si la información se envió correctamente aparecerá al final de la interface CoolTerm, como se visualiza en la figura 5.11.

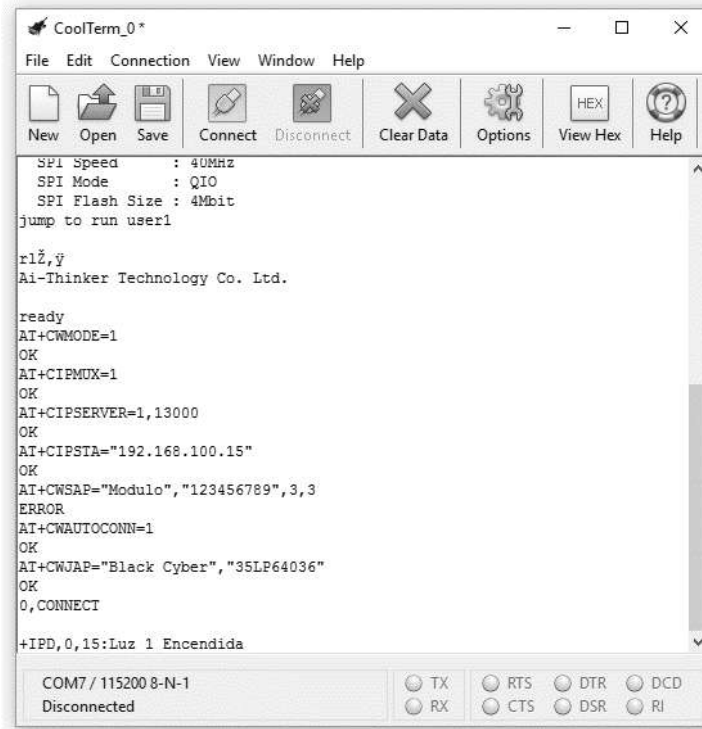


Figura 5. 11 Respuesta del módulo wifi en la interface CoolTerm

5.2.2 Microcontrolador

En esta sección se presentan las pruebas relacionadas al microcontrolador, para comprobar que se realizó bien la programación del microcontrolador y la comunicación por medio de la recepción y transmisión de los datos el con módulo wifi. La conexión del microcontrolador y el módulo wifi se muestra en la figura 5.12. El esquema de conexiones se encuentra en la sección 3.2.2. En las salidas digitales P1.0 y P4.7 se colocan LEDs para el monitoreo.

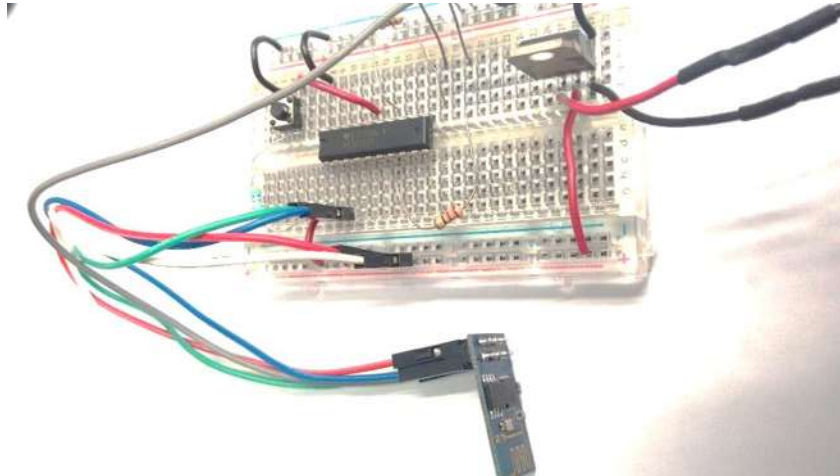


Figura 5. 12 Conexión del microcontrolador y el módulo wifi.

Para verificar que se programa de manera correcta el microcontrolador, se comprueba mediante el mismo programa Android Simple Socket Tester que se conecte y pueda enviar un dato para encender el LED del pin P1.0. En la figura 5.13 se muestra la conexión con el programa Simple Socket Tester, y él envió del dato “01” dentro del recuadro. Este dato fue programa anteriormente en el microcontrolador.

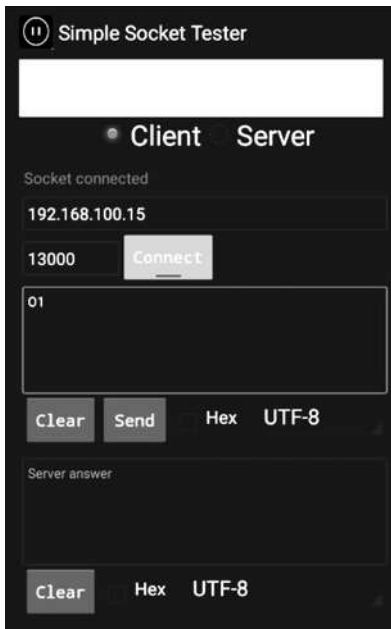


Figura 5. 13 Conexión y envió de dato con el programa Simple Socket Tester.

Para comprobar que se envió bien el dato se observa el LED del pin P1.0 y se corrobora que efectivamente esté encendido y se comprueba así que funciona correctamente la

programación y funcionamiento del microcontrolador. En la figura 5.14 Se muestra el microcontrolador funcionando adecuadamente con el módulo wifi.

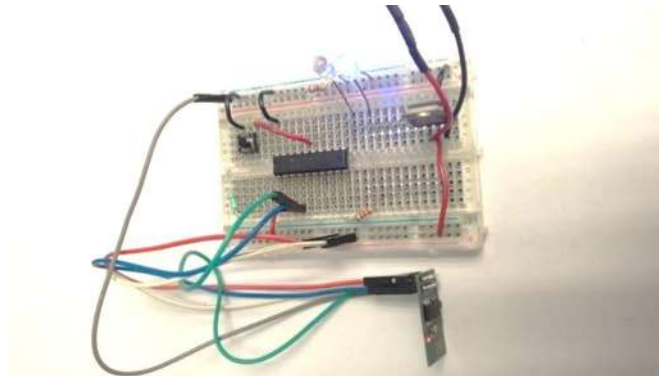


Figura 5. 14 Funcionamiento correcto del microcontrolador

5.2.3 Relevador SSR

Para corroborar el funcionamiento del relevador se conecta a la salida digital del microcontrolador P1.0 a la entrada digital del relevador. Posteriormente se activa P1.0 a través del programa Android Simple Socket Tester con el dato “01”, encendiendo el dispositivo de prueba que es una lámpara. La figura 5.15 muestra la conexión y el funcionamiento del microcontrolador en conjunto con el relevador.



Figura 5. 15 Funcionamiento conjunto del microcontrolador y el relevador SSR

5.2.4 Circuito impreso del módulo de control

En esta sección se verifica el correcto funcionamiento del circuito impreso realizando pruebas de encendido y apagado de las salidas digitales, estas mismas monitoreándolas con

LEDs en el circuito impreso. La figura 5.16 muestra el funcionamiento del circuito impreso del módulo de control.

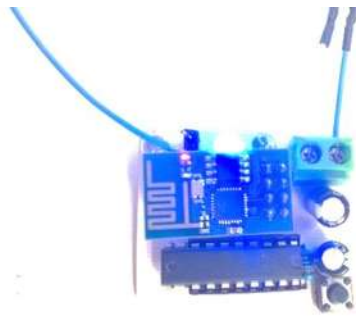


Figura 5. 16 Funcionamiento del circuito impreso del módulo de control

5.3 Sistema completo

Una vez realizadas las pruebas de las partes del módulo de control, y el sistema de comunicación y enlace, se incorporan las partes de todo el sistema para probarlas conjuntamente. Con el botón a través de la página web se envía la opción seleccionada para que el módulo de control active el dispositivo a controlar, que en este caso es una lámpara. En la figura 5.17 Se muestra el funcionamiento en conjunto de todo el sistema.

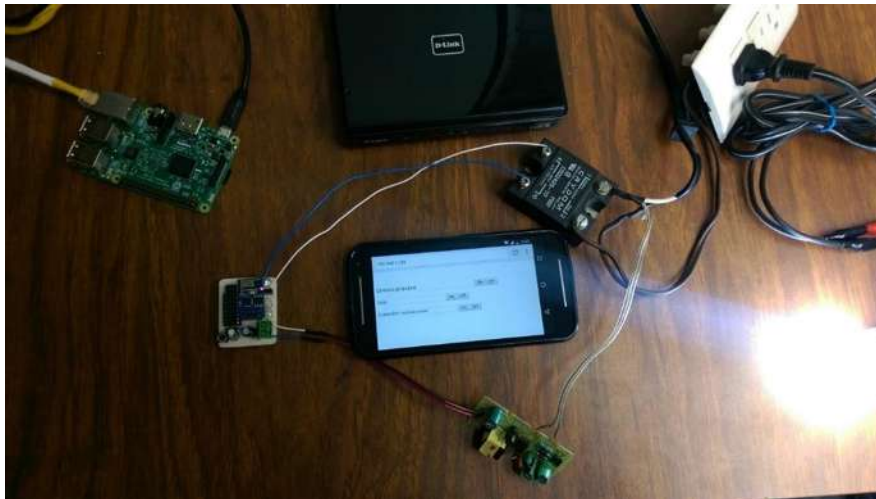


Figura 5. 17 Comprobación del funcionamiento del sistema

Capítulo 6 Conclusiones y trabajos futuros

En esta sección se dan a conocer las conclusiones generales y específicas, y los trabajos futuros. Se concluyen todos los aspectos importantes de la tesis, incluyendo las partes adecuadamente, además se consideran posibles mejoras en el futuro.

6.1 Conclusiones generales

Dentro de las tecnologías que son consideradas en la domótica como el control, la automatización y el monitoreo de una casa. Este trabajo muestra el proyecto enfocado al control de una casa.

Se logró construir un sistema con el cual se pueden controlar dispositivos que utilicen corriente eléctrica a través de una página web vía wifi.

Objetivos cumplidos:

- Instalar un servidor web
- Diseñar y programar una página web
- Programar un microcontrolador que controle un relevador de estado sólido y un módulo wifi
- Diseñar un protocolo de comunicación entre el servidor web, el módulo de control y el cliente web.

Las restricciones de este sistema son:

- El número de usuarios máximo de doscientos cincuenta y cinco.
- Los módulos de control ocupan instalación eléctrica.

- El número de dispositivos a controlar esta limitado por la página web.
- Control sin retroalimentación del estado del dispositivo.
- Se desconoce el estado del dispositivo.

Mejoras respecto a los sistemas comerciales:

- Utiliza dispositivos que se encuentran instalados dentro de la oficina y/o casa habitación.
- Concede el acceso desde una página web sin necesidad de una aplicación específica para un sistema operativo.
- No repercute en el desempeño de la red por el bajo tráfico de datos.

6.2 Conclusiones personales

El desarrollo de este proyecto presentó un gran reto, ya que además de integrar conocimientos adquiridos en la Facultad, también hubo necesidad de adquirir nuevos conocimientos, debido a que el proyecto integra múltiples elementos, siendo de naturaleza multidisciplinario.

Antes de comenzar el proyecto fue necesario el aprendizaje de redes computacionales, aprender a instalar un servidor, a diseñar una página web y el uso de sockets en redes computacionales, los cuales fueron indispensables para desarrollar el sistema de comunicación y enlace del sistema de domótica.

Uno de los inconvenientes en la implementación del proyecto fue la comunicación del módulo wifi con el microcontrolador, ya que se tuvo que realizar un protocolo que funcionara correctamente con el módulo wifi y considerar y medir los tiempos de respuesta del mismo.

6.3 Trabajos futuros

Es de mi interés seguir mejorando el desarrollo de este sistema, añadiendo más dispositivos y automatizándolos, como variación de la intensidad de luz, lectura de sensores, activación de actuadores, cámaras de vigilancia, identificación de entrada y salida de usuarios.

Por la parte del sistema de comunicación y enlace, la interface de la página web podría mejorarse para hacerla más intuitiva y confortable visualmente, añadiendo una configuración que permita el acceso desde cualquier conexión de Internet no solo dentro de la red local. También se podría agregar más seguridad a este sistema, aumentar la estructura web para incrementar el número de personas y pueda utilizarse en sistemas donde habiten muchos clientes web, como edificios y oficinas, los cuales también poseen una gran cantidad de dispositivos que controla cada usuario.

Anexo A Código fuente del microcontrolador

A.1 Establecer configuración

```
/*
 * TESIS
 * Comunicacion mediante el puerto serial para el Modulo Wifi ESP8266
 * Encendido y apagado de salidas digitales
 */
#include <msp430.h>
#include "../Relevador.h"
#include "../Timer.h"
#include "../Wifi.h"

/*****Variables y
apuntadores*****/
unsigned char tiempospuesta; //timer

unsigned char indiceTx; //indice del bufferTx
unsigned char datoTx; //Dato de 8 bits para interrup Tx
unsigned char bufferTx[50]; //Buffer para 50 caracteres

unsigned char indiceRx; //indice del bufferRx
unsigned char datoRx; //Dato de 8 bits para interrup Rx
unsigned char bufferRx[256]; //Buffer para 50 caracteres

unsigned char bandera; //Bandera de termino de recepcion
unsigned char iniciocadena; //Inicio de la cadena
unsigned char inicioI; //inicio interrupcion
unsigned char fincadena; //indice de fin de cadena

unsigned char InOut, Canal, Estado; //datos a comparar
```

```

unsigned char entradas;           //Entradas al main

/*****Comandos AT
*****/
unsigned char IPD[] = {"+IPD"}; //Comando de aceptacion

/*****Funcion
principal*****/
int main(void) {
    WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer

    digitalWrite();           //init salidas digitales
    timer();                  //init timer
    USART();                  //init puerto serial

    __enable_interrupt();
    __no_operation();

    initWifi();              //Inicializacion Wifi
    initRxBuff();           //Inicializo Recepcion

    while(1){
        if(bandera == 1){
            //Si termino cadena
            if(compara(IPD, &bufferRx[iniciocadena]) == 1){
                //Si el inicio es +IPD
                entradas++;
                //Entradas del IPD

                InOut=bufferRx[iniciocadena+9];
                //Entrada o salida
                Canal=bufferRx[iniciocadena+10];
                //Canal
                Estado=bufferRx[iniciocadena+11];
                //Estado

                if(InOut == '0'){
                    switch(Canal){
                        case '0':
                            if(Estado == '1')
                                P1OUT |= BIT0;
                            //P1.0 Prendemos led
                            if(Estado == '0')
                                P1OUT &= ~BIT0;
                            //P1.0 Apagamos led
                            break;
                        case '1':
                            if(Estado == '1')
                                P1OUT |= BIT3;
                            //P1.0 Prendemos led
                            if(Estado == '0')
                                P1OUT &= ~BIT3;
                            //P1.0 Apagamos led
                    }
                }
            }
        }
    }
}

```



```

        break;
    case '2':
        if(Estado == '1')
            P1OUT |= BIT4;
        //P1.0 Prendemos led
        if(Estado == '0')
            P1OUT &= ~BIT4;
        //P1.0 Apagamos led
        break;
    case '3':
        if(Estado == '1')
            P1OUT |= BIT5;
        //P1.0 Prendemos led
        if(Estado == '0')
            P1OUT &= ~BIT5;
        //P1.0 Apagamos led
        break;
    }
}

```

```

if(InOut == 'I'){
    switch(Canal){
        case '0':
            if(Estado == '1')
                P1OUT |= BIT0;
            //P1.0 Prendemos led
            if(Estado == '0')
                P1OUT &= ~BIT0;
            //P1.0 Apagamos led
            break;
        case '1':
            if(Estado == '1')
                P1OUT |= BIT3;
            //P1.0 Prendemos led
            if(Estado == '0')
                P1OUT &= ~BIT3;
            //P1.0 Apagamos led
            break;
        case '2':
            if(Estado == '1')
                P1OUT |= BIT4;
            //P1.0 Prendemos led
            if(Estado == '0')
                P1OUT &= ~BIT4;
            //P1.0 Apagamos led
            break;
        case '3':
            if(Estado == '1')
                P1OUT |= BIT5;
            //P1.0 Prendemos led
            if(Estado == '0')

```

```

        P1OUT &= ~BIT5;
        //P1.0 Apagamos led
        break;
    }
}
}
bandera=0; //Inicilizo bandera
}
}

//__bis_SR_register(GIE);
//__no_operation();
}

```

A.2 Configuración del relevador

```

/*
 * Relevador.h
 *
 */

#ifndef RELEVADOR_H_
#define RELEVADOR_H_

void digitalout(void);

#endif /* RELEVADOR_H_ */

/*
 * Relevador.c
 *
 */

#include <msp430.h>

/*****Configuracion salidas digitales
SSR*****/
void digitalout(void){
    //LED1
    P1DIR |= BIT0; //P1.0 Como salida
    P1OUT &= ~BIT0; //P1.0 Apagamos led
    //LED2
    P1DIR |= BIT3; //P1.3 Como salida
    P1OUT &= ~BIT3; //P1.3 Apagamos led
    //LED3
    P1DIR |= BIT4; //P1.4 Como salida

```

```

        P1OUT &= ~BIT4;    //P1.4 Apagamos led
        //LED4
        P1DIR |= BIT5;    //P1.5 Como salida
        P1OUT &= ~BIT5;    //P1.5 Apagamos led
        //LED5
        P1DIR |= BIT6;    //P1.6 Como salida
        P1OUT |= BIT6;    //P1.6 Encendemos led
    }

```

A.3 Configuración del timer

```

/*
 * Timer.h
 *
 */

#ifndef TIMER_H_
#define TIMER_H_

void timer(void);
void retardo(unsigned char);

#endif /* TIMER_H_ */

/*
 * Timer.c
 *
 */

#include <msp430.h>
#include "../Relevador.h"

extern unsigned char tiemporespuesta;

/*****Timer Configuracion
*****/
void timer(void){
    TACCTL0 = CCIE;           //Habilitacion de
interrup
    TACCR0 = 52429;          //0.05s
DC0=SMCLK=1.28MHz
    TACTL = TASSEL_2 + MC_1 + TACLRL; // SMCLK, UPmode, ClearTimer

```

```

}

/*****Funcion de retardo
*****/
void retardo(unsigned char tiempo){
    tiemporespuesta=tiempo;           //asigno a tiemporespuesta
    limite
    while(tiemporespuesta);          //mientras tiemporespuesta
    diferente de cero
    P1OUT ^= BIT6;                    //Cambia estado cuando
    termina total de tiempo
}

/*****Interrupcion
*****/
#pragma vector=TIMER0_A0_VECTOR
__interrupt void TIMER0_A0_ISR(void){
    //P1OUT ^= BIT0;                  //Cambia estado cuando
    activa timer

    if(tiemporespuesta!=0)            //Si es diferente de cero
        tiemporespuesta--;           //decrementa
}

```

A.4 Configuración del módulo wifi

```

/*
 * Wifi.h
 *
 */

#ifndef WIFI_H_
#define WIFI_H_

void USART(void);
void sendString(unsigned char *);
void initWifi(void);
void initRxBuff(void);
unsigned char compara(unsigned char *, unsigned char *);

#endif /* WIFI_H_ */

/*
 * Wifi.c
 *
 */

#include <msp430.h>
#include "../Relevador.h"
#include "../Timer.h"

/*****Variables globales
*****/
extern unsigned char bufferTx[50];
extern unsigned char indiceTx;
extern unsigned char datoTx;

```

```

extern unsigned char bufferRx[50];
extern unsigned char indiceRx;
extern unsigned char datoRx;

extern unsigned char bandera;
extern unsigned char iniciocadena;
extern unsigned char inicioI;
extern unsigned char fincadena;

/*****Comandos AT
*****/
//Inicializacion
const unsigned char AT[] = {"AT\r\n"};
//Confirmacion
const unsigned char AT_RST[] = {"AT+RST\r\n"};
//Reseteo
const unsigned char AT_CWMODE[] = {"AT+CWMODE=1\r\n"};
//Modo Station
const unsigned char AT_CIPMUX[] = {"AT+CIPMUX=1\r\n"};
//Multiples conexiones
const unsigned char AT_CIPSERVER[] = {"AT+CIPSERVER=1,13000\r\n"};
//Modo servidor, puerto 13000
//const unsigned char AT_CIPSTA[] = {"AT+CIPSTA=\"192.168.100.15\"\r\n"};
//Dirección IP del modulo casa
const unsigned char AT_CIPSTA[] = {"AT+CIPSTA=\"192.168.1.135\"\r\n"};
//Dirección IP del modulo router
const unsigned char AT_CWAUTOCONN[] = {"AT+CWAUTOCONN=1\r\n"};
//Autoconectar con esta IP
//const unsigned char AT_CWJAP[] = {"AT+CWJAP=\"Black
Cyber\", \"35LP64036\"\r\n"}; //Unirse a la red
const unsigned char AT_CWJAP[] = {"AT+CWJAP=\"Domotica
Potter\", \"0123456789\"\r\n"}; //Unirse a la red

/*****Inicializacion puerto serial
*****/
void USART(void){
    UCA0CTL1 |= UCSWRST; //Reset maquina de estados
en 1
    UCA0CTL1 |= UCSSEL_2; //Seleccionando SMCLK
    UCA0BR0 = 9; //Low byte Velocidad 115200
    baudios con SMCLK=1.048MGHz
    UCA0BR1 = 0; //High byte Velocidad 115200
    baudios con SMCLK=1.048MGHz
    UCA0MCTL |= UCBSRS_1 + UCBRF_0; // UCBRF sobre muestreo 1, UCBSRS
    submuestreo 0

    P1DIR &= ~BIT1; //Rx ENTRADA
    P1DIR |= BIT2; //Tx SALIDA
    P1SEL = BIT1 + BIT2; //Seleccionando perifericos
puerto serial
    P1SEL2 = BIT1 + BIT2;
    UCA0CTL1 &= ~UCSWRST; //Inicializando maquina de
estados
}

```

```

/*****Llenar cadena para Tx
*****/
void sendString(const unsigned char *ptrCadena){

    unsigned char i=0;                //indice del arreglo de la cadena

    while(*ptrCadena != '\0')        //Mientras la cadena sea diferente de
cero
        bufferTx[i++]=*ptrCadena++; //llena el buffer con la cadena
        bufferTx[i]=*ptrCadena;      //Se agrega el cero al buffer

    indiceTx=0;                       //inicializo indice del
buffer Tx en interrup
    IE2 |= UCA0TXIE;                  //Habilito interrupcion Tx
    IFG2 |= UCA0TXIFG;
}

```

```

/*****Enviar comandos AT Tx
*****/

```

```

void initWifi(void){
    retardo(100);
    sendString(AT);
    retardo(40);
    sendString(AT_RST);
    retardo(60);
    sendString(AT_CWMODE);
    retardo(40);
    sendString(AT_CIPMUX);
    retardo(40);
    sendString(AT_CIPSERVER);
    retardo(40);
    sendString(AT_CIPSTA);
    retardo(40);
    sendString(AT_CWAUTOCONN);
    retardo(40);
    sendString(AT_CWJAP);
    retardo(100);
}

```

```

void initRxBuff(void){
    indiceRx=0;                       //Inicializo indiceRx
    bandera=0;                        //Inicilizo bandera
    inicioI=0;
    IE2 |= UCA0RXIE;                 //Habilito interrupcion Rx
}

```

```

unsigned char compara(unsigned char *Cadena, unsigned char *ptrRecib){
    while(*Cadena != '\0'){
        if(*ptrRecib++ != *Cadena++) //Si algun dato de
Recib es diferente de IPD
    }
}

```

```

        return 0;
    //Regreso 0
    }
    return 1; //Si es
correcto Regreso 1
}

/*****Interrupcion Puerto Serial
*****/
#pragma vector=USCIAB0TX_VECTOR //Wifi Tx
__interrupt void USCI0TX_ISR(void)
{
    datoTx = bufferTx[indiceTx++]; //datoTx = lo que
tengo en el arreglo buffer
    if(datoTx == 0) //Si el
datoTx = 0 entonces
        IE2 &= ~UCA0TXIE;
//deshabilito la interrupcion
    else //Si no
        UCA0TXBUF = datoTx;
//transmito el contenido del dato
}

#pragma vector=USCIAB0RX_VECTOR //Wifi Rx
__interrupt void USCI0RX_ISR(void)
{
    datoRx = UCA0RXBUF; //datoRx = lo
que tengo en el buffer
    bufferRx[indiceRx] = datoRx; //Lleno el buffer

    if(bufferRx[indiceRx] == 0x0A){ //Si es igual a
termino de cadena
        if(bufferRx[indiceRx-1] == 0x0D){
            fincadena=indiceRx;
//Mando fin de candena
            iniciocadena=inicioI;
//iniciocadena=inicioInterrupcion
            inicioI=fincadena+1;
//inicioInterrupcion=findecadena
            bandera=1;
//Habilito bandera de buffer
        }
    }

    indiceRx++;
//Incremento indice
    if(indiceRx>255)
//Inicializo indice si se exede
        indiceRx=0;
}

```

Anexo B Código fuente de la página web

```
    } else {  
        fwrite ( $fp, "030" );  
        fclose ( $fp );  
    }  
}
```



```

?php
function led1on() {
    $fp = stream_socket_client ( "tcp://192.168.1.135:13000", $errno, $errstr, 2);
    if (! $fp) {
        echo "$errstr ($errno)<br />\n";
    } else {
        fwrite ( $fp, "O11" );
        fclose ( $fp );
    }
}
function led1off() {
    $fp = stream_socket_client ( "tcp://192.168.1.135:13000", $errno, $errstr, 2);
    if (! $fp) {
        echo "$errstr ($errno)<br />\n";
    } else {
        fwrite ( $fp, "O10" );
        fclose ( $fp );
    }
}
function led2on() {
    $fp = stream_socket_client ( "tcp://192.168.1.135:13000", $errno, $errstr, 2);
    if (! $fp) {
        echo "$errstr ($errno)<br />\n";
    } else {
        fwrite ( $fp, "O21" );
        fclose ( $fp );
    }
}
function led2off() {
    $fp = stream_socket_client ( "tcp://192.168.1.135:13000", $errno, $errstr, 2);
    if (! $fp) {
        echo "$errstr ($errno)<br />\n";
    } else {
        fwrite ( $fp, "O20" );
        fclose ( $fp );
    }
}
function led3on() {
    $fp = stream_socket_client ( "tcp://192.168.1.135:13000", $errno, $errstr, 2);
    if (! $fp) {
        echo "$errstr ($errno)<br />\n";
    }
}

```

```

switch (true) {
    case isset ( $_POST ['l1on'] ) :
        $estado1 = "Luz1 encendida";
        led1on ();
        break;
    case isset ( $_POST ['l1off'] ) :
        $estado1 = "Luz1 apagada";
        led1off ();
        break;
    case isset ( $_POST ['l2on'] ) :
        $estado2 = "Luz2 encendida";
        led2on ();
        break;
    case isset ( $_POST ['l2off'] ) :
        $estado2 = "Luz2 apagada";
        led2off ();
        break;
    case isset ( $_POST ['l3on'] ) :
        $estado3 = "Luz3 encendida";
        led3on ();
        break;
    case isset ( $_POST ['l3off'] ) :
        $estado3 = "Luz3 apagada";
        led3off ();
        break;
}
?>

<html>
<head>
<title>Pagina Web Domotica</title>
</head>
<body>
    <form method="post">
        Control ON/OFF Iluminacion
        <br /> <br /> <br /> <br />

        Entrada principal <input type="text" value="<?php echo @$estado1;?>" />
                            <input type="submit" name="l1on" value="ON" />
                            <input type="submit" name="l1off" value="OFF" />

        <br /> <br />
        Sala <input type="text" value="<?php echo @$estado2;?>" />
            <input type="submit" name="l2on" value="ON" />
            <input type="submit" name="l2off" value="OFF" />

        <br /> <br />
        Comedor <input type="text" value="<?php echo @$estado3;?>" />
                <input type="submit" name="l3on" value="ON" />
                <input type="submit" name="l3off" value="OFF" />

        <br /> <br />

    </form>
</body>
</html>

```

Bibliografía

- [1] Asociación Española de Domótica e Inmótica - CEDOM, “Qué es Domótica”. [En línea]. Disponible en: <http://www.cedom.es/sobre-domotica/que-es-domotica>. [Consultado: 01-may-2016].
- [2] Arkiplus, “Historia de la Domótica”, *Arkiplus*, 05-oct-2013. [En línea]. Disponible en: <http://www.arkiplus.com/historia-de-la-domotica>. [Consultado: 02-may-2016].
- [3] J. C. Martín, *Iniciación a la domótica (Instalaciones domóticas)*. Editex, 2011.
- [4] casamultimedia, “5 Tendencias Del Concepto Smart Home | CASA MULTIMEDIA”.
- [5] J. M. Huidobro y R. J. M. Tejedor, *Manual de domótica*. Creaciones Copyright SL, 2010.
- [6] Nicolás Montés Sánchez y Aurelio Pons Puig, *SMLhouse: prototipo de vivienda solar prefabricada*. TC Cuadernos, 2012.
- [7] J. M. M. TORREBLANCA, *Domótica para ingenieros*. Ediciones Paraninfo, S.A., 2015.
- [8] “Revista Electroindustria - Protocolos de Comunicaciones para Automatización de Edificios”. [En línea]. Disponible en: <http://www.emb.cl/electroindustria/articulo.mvc?xid=1223>. [Consultado: 10-jun-2016].
- [9] R. L. G. L. Paz, *Desarrollo de aplicaciones web en el entorno servidor. IFCD0210*. IC Editorial, 2015.
- [10] *Conocimientos Basicos de Informatica Ebook*. MAD-Eduforma.
- [11] “Introduccion a la Arquitectura Cliente-Servidor (1)”, *Miguel Angel López WEB*, 20-ene-2015.
- [12] “What is a Raspberry Pi?”, *Raspberry Pi*.
- [13] “Introducción - Bases de datos”, *CCM*. [En línea]. Disponible en: <http://es.ccm.net/contents/66-introduccion-bases-de-datos>. [Consultado: 05-may-2016].
- [14] Á. Cobo, *PHP y MySQL: Tecnología para el desarrollo de aplicaciones web*. Ediciones Díaz de Santos, 2005.
- [15] “¿Cómo instalar Linux, Apache, MySQL, PHP (LAMP) en Ubuntu 14.04?”, *DigitalOcean*. [En línea]. Disponible en: <https://www.digitalocean.com/community/tutorials/como-instalar-linux-apache-mysql-php-lamp-en-ubuntu-14-04-es>. [Consultado: 05-may-2016].

- [16] “Equipos de red - Router”, *CCM*. [En línea]. Disponible en:
<http://es.ccm.net/contents/299-equipos-de-red-router>. [Consultado: 08-jun-2016].
- [17] “3.3.2 Por su relación funcional (Igual- igual (p2p, cliente – servidor) - Sr.Linkoerk”. [En línea]. Disponible en:
<https://sites.google.com/site/srlinkoerk/home/informatica-y-computacion-ii/-332-Por-su-relacin-funcional-Igual-igual-p2p-cliente--servidor>. [Consultado: 13-jun-2016].
- [18] “Capítulo 1. Introducción (Introducción a CSS)”. [En línea]. Disponible en:
http://librosweb.es/libro/css/capitulo_1.html. [Consultado: 05-may-2016].
- [19] A. C. Bazua, *Capítulos 7 y 8 Libro C# para automatización electrónica e industrial: Ejercicios con microcontroladores PIC y C#.NET*. Aaron Castro Bazua, 2012.
- [20] P. por E. Perez, “Traducciones, howtos, Unix, Linux, Windows, redes.: Direccionamiento en los protocolos de Capa de Transporte TCP / IP (TCP y UDP): Puertos y sockets.” .
- [21] “Conceptos generales de redes de datos | Redes Telemáticas” .
- [22] “Microcontrolador - Wikitronica”. [En línea]. Disponible en:
<http://wikitronica.labc.usb.ve/index.php/Microcontrolador>. [Consultado: 06-may-2016].
- [23] “5. INTRODUCCIÓN A LOS MICRO CONTROLADORES - Electrónica Digital”. [En línea]. Disponible en: <https://sites.google.com/site/electronicadigitaluvfime/5-introduccion-a-los-microconttroadores>. [Consultado: 06-may-2016].
- [24] Espressif Systems IOT Team, “ESP8266EX Datasheet Version 4.3” .
- [25] “Relevadores de estado solido, SSR”. [En línea]. Disponible en:
<http://teii.com.mx/relevadores-estado-solido-ssr.html>. [Consultado: 06-may-2016].
- [26] “Raspberry Pi 3 Model B”, *Raspberry Pi* .
- [27] “ESP8266_wifi_Module_Quick_Start_Guide_v_1.0.4.pdf”. [En línea]. Disponible en:
http://rancidbacon.com/files/kiwicon8/ESP8266_wifi_Module_Quick_Start_Guide_v_1.0.4.pdf. [Consultado: 08-jun-2016].
- [28] “msp430g2553.pdf”. [En línea]. Disponible en:
<http://www.ti.com/lit/ds/symlink/msp430g2553.pdf>. [Consultado: 08-jun-2016].