



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES
INDUSTRIALES

**Diseño e implementación de un sistema domótico con acceso remoto
utilizando una pasarela residencial.**

Trabajo de titulación presentado para optar el grado académico de:
INGENIERO ELECTRÓNICO EN CONTROL Y REDES
INDUSTRIALES

AUTOR: EDISON JAVIER LEMA POMAQUIZA.
TUTOR: ING. EDWIN VINICIO ALTAMIRANO SANTILLAN.

Riobamba-Ecuador

2016

©2016, Edison Javier Lema Pomaquíza

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor

FACULTAD DE INFORMÁTICA Y ELECTRONICA

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES INDUSTRIALES

El Tribunal del Trabajo de Titulación certifica que: El trabajo de investigación: “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DOMÓTICO CON ACCESO REMOTO UTILIZANDO UNA PASARELA RESIDENCIAL”, de responsabilidad del señor Edison Javier Lema Pomaquiza, ha sido minuciosamente revisado por los Mienbros del Tribunal del Trabajo de Titulación, quedando autorizada su presentación.

Dr. Miguel Tasambay Ph.D.

DECANO

Ing. Alberto Arellano.

**DIRECTOR DE ESCUELA DE
INGENIERÍA ELECTRÓNICA
EN CONTRO Y REDES
INDUSTRIALES**

Ing. Edwin Altamirano.

**DIRECTOR DEL TRABAJO
DE TITULACIÓN**

Ing. Henry Vallejo.

MIEMBRO DEL TRIBUNAL

**DOCUMENTALISTA
SISBIB ESPOCH**

Yo, Edison Javier Lema Pomaquíza soy responsable de las ideas, doctrinas y resultados expuestos en esta Tesis y el patrimonio intelectual de la Tesis de Grado pertenece a la Escuela Superior Politécnica De Chimborazo

EDISON JAVIER LEMA POMAQUIZA

DEDICATORIA

A mis maestros que formaron parte de mi vida estudiantil, que supieron indicarme el camino hacia el saber y el triunfo, que día tras día han cooperado con su voluntad, impartiendo conocimientos que me han servido para alcanzar las metas propuestas al inicio de este trajinar académico, por la comprensión brindada en cada una de las aulas de clases, y que aportaron con sus conocimientos para poder cumplir una etapa de mi vida.

A mi hija y esposa quienes forma parte de todo mí ser, y es por quienes he logrado este legado, y han sido mi inspiración durante todos estos años, que cada instante las llevo conmigo.

Edison

AGRADECIMIENTO

El más sincero agradecimiento a la Escuela Superior Politécnica de Chimborazo, por darme la oportunidad de obtener una profesión y ser una ayuda para la sociedad.

A mi director de Tesis por haber brindado toda su colaboración, impartiendo las experiencias y conocimientos para alcanzar las metas propuestas.

A mis grandes amigos que formaron parte de mi vida académica, que de alguna u otra forma estuvieron conmigo en los buenos y malos momentos, que con esfuerzo y dedicación lo he logrado lo que me propuse al inicio de esta carrera.

A mi padre y a mi madre por haberme brindado toda la colaboración desde el inicio de este camino hacia el conocimiento, a mis hermanos y hermanas que siempre estuvieron conmigo en todo momento.

A mis tías, por su ejemplo de vida, que siempre me estuvieron alentando para seguir adelante, con su colaboración para alcanzar el éxito, y aprender de ellas el verdadero valor de la educación.

Edison

TABLA DE CONTENIDO

	Páginas
PORTADA.....	I
DERECHO DE AUTOR.....	II
HOJA DE CALIFICACIÓN.....	III
RESPONSABLE.....	IV
DEDICATORIA.....	V
AGRADECIMIENTO.....	VI
RESÚMEN.....	XVII
SUMARY.....	XVIII
INTRODUCCIÓN.....	1
OBJETIVO GENERAL.....	2
OBJETIVOS ESPECÍFICO.....	2
CAPÍTULO I	
1. MARCO TEÓRICO	3
1.1. Domótica	3
1.1.1. Beneficios	4
1.1.2. Funcionamiento.....	5
1.1.3. Arquitectura de un sistema domótico	5
1.1.3.1. Arquitectura centralizada	5
1.1.3.2. Arquitectura distribuida o descentralizada	6
1.1.4. Casas inteligentes	6
1.1.5. Protocolos de comunicación usados	7
1.1.5.2. Zigbee	7
1.2. Pasarela residencial.....	8
1.2.1. Características.....	9
1.3. Protocolo de comunicación Modbus	10
1.3.1. Generalidades	10
1.3.2. Comunicación maestro-esclavo	11

1.3.3.	<i>Comunicación serie rs485</i>	12
1.3.3.1.	<i>Output Coils</i>	14
1.3.3.2.	<i>Holding Registers</i>	14
1.3.4.	<i>Modos de transmisión</i>	15
1.3.4.1.	<i>Modo RTU</i>	15
1.3.4.2.	<i>Modo ASCII</i>	15
1.4.	<i>Aplicación web</i>	16
1.4.1.	<i>Introducción</i>	16
1.4.2.	<i>Protocolo http</i>	16
1.4.3.	<i>Lenguaje HTML</i>	17
1.4.3.1.	<i>Características</i>	17
1.4.4.	<i>Java script</i>	18
1.4.5.	<i>Lenguaje Php</i>	19
1.4.5.1.	<i>Características</i>	20
1.4.5.2.	<i>Principales tareas</i>	20
1.4.6.	<i>Estructura de una aplicación web</i>	21
1.4.6.1.	<i>Ventajas</i>	22
1.4.6.2.	<i>Desventajas</i>	23
1.4.6.3.	<i>Cliente</i>	23
1.4.6.4.	<i>Servidor</i>	24
1.4.7.	<i>Entornos web</i>	24
1.5.	<i>Microcontrolador</i>	25
1.5.1.	<i>Características de un microcontrolador</i>	25
1.5.2.	<i>Proceso de desarrollo</i>	26
1.5.2.1.	<i>Desarrollo de software</i>	26
1.5.2.2.	<i>Programación del microcontrolador</i>	26
1.5.2.3.	<i>Prueba y verificación</i>	27
1.5.3.	<i>Comunicación uart</i>	28
1.5.4.	<i>Integrado max485</i>	28

1.5.5.	<i>PIC18F2550</i>	30
1.5.5.1.	<i>Características</i>	31
1.6.	Raspberry pi modelo b	32
1.6.1.	<i>Especificaciones hardware</i>	33
1.7.	Altium Designer	33
1.7.1.	<i>Productos</i>	34
1.7.2.	<i>Herramientas de diseño</i>	35
1.7.3.	<i>Partes importantes para elaborar un PCB</i>	36
1.7.4.	<i>Pasos para crear un proyecto de PCB</i>	37
1.8.	MikroC pro for Pic	37
1.8.1.	<i>Características principales</i>	38
1.9.	Componentes y elementos del sistema	39
1.9.1.	<i>Sensor magnético</i>	39
1.9.2.	<i>Optoacoplador moc3021</i>	40
1.9.3.	<i>Fusibles</i>	41
1.9.4.	<i>Resistencia eléctrica</i>	41
1.9.5.	<i>Regulador LM7805</i>	42
1.9.6.	<i>Regulador LM317</i>	43
1.9.7.	<i>Sensor de corriente Acs714</i>	43
1.9.8.	<i>Triac</i>	45
1.9.9.	<i>Mosfet</i>	46
1.9.10.	<i>Trimpot</i>	48
1.9.11.	<i>Cámara Wireless</i>	49
1.9.12.	<i>Motor 12 V DC</i>	50
1.9.13.	<i>Tira de diodos Led 12V</i>	51
CAPÍTULO II		
2.	MARCO PLICATIVO	52
2.1.	Diseño del sistema domótico	52
2.1.1.	<i>Estructura del sistema</i>	52

2.1.2.	<i>Etapa de control</i>	53
2.1.3.	<i>Etapa de adquisición de datos</i>	53
2.1.4.	<i>Etapa de potencia</i>	54
2.2.	Etapa de alimentación del sistema	54
2.3.	Método de programación del microcontrolador	54
2.3.1.	<i>Programación por interrupción</i>	54
2.3.2.	<i>Programación Uart</i>	55
2.4.	Interfaz de control con la aplicación web	55
2.5.	Plano domótico de la instalación	56
2.6.	Construcción de las placas	57
2.6.1.	<i>Partes de control de las placas</i>	57
2.6.2.	<i>Etapa de comunicación</i>	58
2.6.3.	<i>Placa master</i>	59
2.6.3.1.	<i>Etapa de la placa Raspberry pi</i>	59
2.6.3.2.	<i>Etapa de alimentación</i>	60
2.6.3.3.	<i>PCB master</i>	60
2.6.4.	<i>Placa de luces AC</i>	62
2.6.4.1.	<i>Etapa de cruce por cero</i>	62
2.6.4.2.	<i>Etapa de potencia</i>	63
2.6.4.3.	<i>PCB de la placa de Luces AC</i>	64
2.6.5.	<i>Placa de luces RGB</i>	65
2.6.5.1.	<i>Etapa de alimentación</i>	65
2.6.5.2.	<i>Etapa de entradas de pulsantes</i>	66
2.6.5.3.	<i>Etapa de potencia</i>	66
2.6.5.4.	<i>PCB placa de luces RGB</i>	67
2.6.6.	<i>Placa del sistema de alarma</i>	68
2.6.6.1.	<i>Etapa de sensores magnéticos</i>	68
2.6.6.2.	<i>Etapa de potencia</i>	69
2.6.6.3.	<i>PCB Placa del sistema de alarma</i>	70

2.6.7.	<i>Placa de control de la persiana</i> -----	72
2.6.7.1.	<i>Etapa de entradas</i> -----	72
2.6.7.2.	<i>Etapa de potencia</i> -----	72
2.6.7.3.	<i>PCB de la persiana</i> -----	74
2.7.	Programación de la placa Raspberry pi -----	75
2.7.1.	<i>Instalación del sistema operativo Ubuntu en la placa Raspberry pi</i> -----	75
2.7.2.	<i>Instalación del protocolo SSH</i> -----	75
2.7.3.	<i>Instalación de la librería wiring pi</i> -----	76
2.7.4.	<i>Instalación de la base de datos</i> -----	76
2.8.	Creación de la aplicación web -----	77
2.8.1.	<i>Opción alarma</i> -----	77
2.8.2.	<i>Opción luces RGB</i> -----	78
2.8.3.	<i>Opción luces Ac</i> -----	79
2.8.3.1.	<i>Botones On/Off</i> -----	80
2.8.3.2.	<i>Botón Dimmer</i> -----	80
2.8.3.3.	<i>Opción para la cámara</i> -----	82
 CAPÍTULO III		
3.	COSTOS, PRUEBAS Y RESULTADOS	84
3.1.	Costo de materiales -----	84
3.2.	Pruebas realizadas -----	85
3.3.	Análisis estadístico -----	86
3.3.1.	<i>Confiabilidad del sistema</i> -----	86
3.3.2.	<i>Tiempo de respuesta del sistema</i> -----	87
CONCLUSIONES		92
RECOMENDACIONES		94
 BIBLIOGRAFÍA		
 ANEXOS		

ÍNDICE DE TABLAS

	Páginas
Tabla 1-1 Direcciones MODBUS	13
Tabla 1-2 Pines de conexión MAX-485	29
Tabla 2-1 Identificación del plano domótico	56
Tabla 3-1 Costos de materiales del sistema	84
Tabla 3-2 Datos para la confiabilidad del sistema	86
Tabla 3-3 Tabla de tiempos para la respuesta del sistema	87
Tabla 3-4 Retardo de tiempo desde el interior y exterior de la vivienda	91

ÍNDICE DE ILUSTRACIONES

	Páginas
Figura 1-1 Domótica	3
Figura 1-2 Casa inteligente.....	7
Figura 1-3 Módulo Zigbee.....	9
Figura 1-4 Pasarela residencial	8
Figura 1-5 Arquitectura MODBUS.....	10
Figura 1-6 Maestro y esclavos	12
Figura 1-7 Estructura Web	21
Figura 1-8 Programación de un microcontrolador	26
Figura 1-9 Proceso de programación de un PIC	27
Figura 1-10 Pruebas de un microcontrolador	27
Figura 1-11 Conexión max-485.....	29
Figura 1-12 Conexión Max-485	30
Figura 1-13 PIC18F2550.....	30
Figura 1-14 Raspberry pi B	32
Figura 1-15 Partes de Altium Designer	35
Figura 1-16 Partes de MikroC Pro	38
Figura 1-17 Sensor magnético	39
Figura 1-18 Optoacoplador.....	40
Figura 1-19 Fusible	41
Figura 1-20 Resistencia.....	42
Figura 1-21 LM7805.....	42
Figura 1-22 LM317.....	43
Figura 1-23 Sensor de corriente.....	44
Figura 1-24 Triac	45
Figura 1-25 Transistor de potencia	46
Figura 1-26 Trimpot.....	48

Figura 1-27	Cámara Wireless	49
Figura 1-28	Motor DC	50
Figura 1-29	Tira led RGB	51
Figura 2-1	Diseño del sistema	52
Figura 2-2	Diagrama de bloques del sistema.....	53
Figura 2-3	Diagrama de bloques de la adquisición de datos	53
Figura 2-4	Diagrama de bloques de la etapa de potencia	54
Figura 2-5	Interfaz de control con la aplicación Web	55
Figura 2-6	Plano domótico	56
Figura 2-7	Etapa comunicación de la placa master.....	58
Figura 2-8	Etapa comunicación de la placa master.....	58
Figura 2-9	Etapa de la placa Raspberry pi.....	59
Figura 2-10	Pines de conexión de la placa Raspberry pi.....	60
Figura 2-11	Etapa de alimentación de la placa master	60
Figura 2-12	PCB de la placa master.....	61
Figura 2-13	PCB de la placa master implementado.....	61
Figura 2-14	Circuito cruce por cero y entrada al microcontrolador.....	62
Figura 2-15	Circuito cruce por cero implementado	63
Figura 2-16	Etapa de potencia de la placa de luces AC	64
Figura 2-17	PCB de la placa de luces Ac	64
Figura 2-18	PCB de la placa de luces AC implementado	65
Figura 2-19	Etapa de control de la placa de luces RGB.....	65
Figura 2-20	Entradas de pulsantes	66
Figura 2-21	Circuito de potencia de la placa de luces RGB.....	66
Figura 2-22	PCB de la placa de luces RGB.....	67
Figura 2-23	PCB de la placa de luces RGB implementado.....	67
Figura 2-24	Luces RGB colocadas en el jardín	68
Figura 2-25	Etapa de entrada de sensores magnéticos	68
Figura 2-26	Colocación de sensores magnéticos en las ventanas y puertas	69

Figura 2-27	Etapa de potencia de la placa de alarma	69
Figura 2-28	Sirena instalada en la vivienda.....	70
Figura 2-29	PCB de la placa de la alarma	70
Figura 2-30	PCB de la placa de la alarma implementado	71
Figura 2-31	Módulo GSM para el envío de mensajes de texto al usuario.....	71
Figura 2-32	Etapa de entradas para pulsantes.....	72
Figura 2-33	Etapa de potencia de la placa de la persiana.....	73
Figura 2-34	Persiana colocada en el estudio.....	73
Figura 2-35	PCB de la de la placa de la persiana.....	74
Figura 2-36	PCB de la placa de la persiana implementado	74
Figura 2-37	Instalación del SO Ubuntu.....	75
Figura 2-38	Instalación de ssh	75
Figura 2-39	Instalación de librería Wiring pi	76
Figura 2-40	Instalación de mysql-server	76
Figura 2-41	Configuración mysql-server y phpmyadmin	77
Figura 2-42	Diseño en la Web para la alarma	78
Figura 2-43	Espacio para luces RGB.....	79
Figura 2-44	Panel de luces On/Off	80
Figura 2-45	Panel de la persiana on/off.....	80
Figura 2-46	Espacio luces Dimmer.....	82
Figura 2-47	Opción para la cámara.....	82
Figura 2-48	Cámara instalada en la vivienda	83

ÍNDICE DE ANEXOS

- Anexo A:** Placas de conmutación Ac con TRIAC BT-A12 y opto acopladores Moc3011.
- Anexo B:** Placa de luces RGB con PWM amplificado para 12V con Mosfet IRF540.
- Anexo C:** Motor utilizado para el movimiento de la persiana.
- Anexo D:** Placa master, y salida al internet mediante una Raspberry Pi modelo B.
- Anexo E:** Caja domótica donde están ubicados los módulos para el sistema domótico.
- Anexo F:** Aplicación Web con la cámara Ip con los botones de las luces Ac de la vivienda.
- Anexo G:** Código para el envío de datos desde la aplicación para la luz dimmer.
- Anexo H:** Código de la persiana con dirección 35 y envío de datos para subir y bajar.
- Anexo I:** Código para las luces RGB con PWM, se envían 3 datos de R, G, B.
- Anexo J:** Código de luces RGB, con dirección 31 y rango entre 1 y 100 para el PWM.
- Anexo K:** Código para la alarma y sensores instalados en la vivienda.
- Anexo L:** Código para las luces AC On/Off, con dirección Modbus 13.
- Anexo M:** Código para las demás luces AC, pero con dirección Modbus 10.
- Anexo N:** Código para ingresar a la aplicación con nombre de usuario y contraseña.
- Anexo Ñ:** Código para crear un nuevo usuario y pueda logarse en la aplicación.
- Anexo O:** Código de la placa master donde se declara las direcciones Modbus del sistema.
- Anexo P:** Comunicación UART con dirección Modbus esclavo 35 para la persiana.
- Anexo Q:** Comunicación UART con dirección Modbus esclavo 21 para la luz dimmer AC.
- Anexo R:** Comunicación UART con dirección Modbus esclavo 31 para las luces RGB.
- Anexo S:** Comunicación UART con dirección Modbus esclavos.
- Anexo T:** Comunicación UART con dirección Modbus 43 para la alarma del sistema.

RESUMEN

En la investigación se diseñó e implementó un sistema domótico, a través de una pasarela residencial, que permite controlar una vivienda desde una parte exterior a ella, realizado en la ciudad de Cuenca, con una aplicación Web, insertando botones que son para encender una luz, slider para hacer dimmer un foco, un panel de combinación de colores, para controlar luces RGB que definen colores(rojo, verde, azul), botones en forma de flechas que indican arriba o abajo para mover la persiana y por último un menú que es donde se encuentra la alarma y con una señal de un sensor que acciona directamente o desde la aplicación web, también se colocó la escenas predefinidas para activar dispositivos con un solo comando, cada gráfica en la aplicación tiene comunicación con placas modulares que se han implementado y mantienen comunicación mediante el protocolo Modbus, la placa master del sistema con los 5 esclavos, el controlador es un microcontrolador 18F2550 y un chip MAX485 para la interfaz serie del protocolo modbus y se han podido conectar hasta 8 esclavos en el sistema, las placas están diseñadas con el Software Altium Designer que permite el desarrollo de circuitos electrónicos profesionales, el sistema utiliza una Raspberry Pi que es un dispositivo que trabaja como un pequeño ordenador instalando un Sistema Operativo Linux, se ha instalado programas y herramientas necesarias para ejecutar el programa que se encuentra en ejecución en este dispositivo, la confiabilidad del sistema domótico está basado en la confiabilidad para el usuario el mismo que califica de la siguiente manera: el 66% de los usuarios representa a una facilidad de manejo del sistema, el 66% representa la seguridad media que brinda a los usuarios y el 100% corresponde al confort y comodidad que ha generado el sistema en la vivienda, a su vez puedo concluir que la pasarela residencial es eficiente para esta investigación, con la cual se ha logrado controlar la vivienda desde cualesquier parte con un dispositivo móvil, se puede recomendar que para el desarrollo de los modulos que controlan el sistema se diseñen con un software profesional que tenga herramientas necesarias para el desarrollo de circuitos electrónicos impresos.

Palabras claves: <COMPUTADOR PORTÁTIL> <APLICACIÓN WEB> <SISTEMA DOMÓTICO> <VIVIENDA RESIDENCIAL> <ELECTRÓNICO Y CONTROL> <SISTEMA DE CONTROL> <DISPOSITIVO RASPBERRY PI>.

SUMMARY

In research designer a demotic System was implemented through a residential Gateway that allows control Housing from an outside party to it, held in Cuenca City, a Web application, inserting buttons are to light, slider for dimming a light bulb, a panel of color scheme to control RGB lights that define colors (red, green, blue) buttons as arrows indicating up or down to move the blind and finally a menu that is where is the alarm and a signal from a sensor which operates directly or from the Web application, predefined to enable devices with a single command, each graph in the application has communication through the modbus protocol, the board master system with the 5 slaves, the controller is a microcontroller 18F2550 and MAX485 chip for serial interface Modbus protocol and were able to connect up to 8 slaves in the system, the plates are designed with the Altium Designer software that allows the development of professional electronic circuits, the system uses a Raspberry pi is a device that works with a small computer by installing a Linux operating system, installed programs and tools needed to run the program that is in running on the device, the reliability of the automation system is based on reliability for the user the same rate as follows: 66% of users represents a facility management system, 66% of management represents security media that gives users and 100% corresponds to the comfort and convenience generated by the system in housing, in turn can conclude that the residential gateway is efficient for the research, which was controlled housing from any part a mobile device, it may recommend to the development of modules that control the system are designed with a professional software that has tools necessary for development of printed circuits.

Keywords: PORTABLE COMPUTER / WEB APPLICATION / HOME AUTOMATION SYSTEM / RESIDENTIAL HOUSING / ELECTRONIC AND CONTROL / CONTROL SYSTEM / DEVICE RASPBERRY PI /

INTRODUCCIÓN

En la actualidad la electrónica, es una de las ciencias más importantes, que ha colaborado con la creación de varios dispositivos, que de alguna u otra forma ha contribuido con el desarrollo y progreso de un determinado país, haciendo de este un ambiente tecnológico, facilitando a las personas en sus actividades y disminuyendo los tiempos en las mismas.

De igual manera la electrónica es una ciencia que se puede combinar con otras ciencias, una de ellas la informática, la misma que contienen sistemas y bases de datos que pueden ser estudiadas y tener una interconexión con sistemas de control, y formar un solo sistema inteligente, que se solución a un problema dentro de la sociedad.

Es por ello que uno de los sistemas modernos en la actualidad son los domóticos, los mismos que integran tanto la parte de control desde una aplicación web, para ejecutar las acciones necesarias en una vivienda, de esta manera mejorar en confort.

Desde los inicios de la aparición de la domótica a esta también se le conoce como una casa inteligente, la misma que para su funcionamiento se necesita de un dispositivo de control único que funciona como el cerebro del sistema.

La interconexión esta efectuada mediante protocolos de comunicación, para que los distintos sistemas pueden comunicarse y analizar los datos que viajan en la red, por lo tanto existen varios protocolos con los cuales podemos contar para efectuar un proyecto de una casa inteligente.

Pero la domótica está estrechamente ligada a una pasarela residencial, la misma que tiene por funcionamiento el control de una residencia desde el espacio exterior, mediante un sistema en la nube logrando controlar los dispositivos para iluminación en una vivienda.

A su vez un sistema domótico de una vivienda necesita una seguridad adecuada, donde se usan actuadores y sensores magnéticos, como también cámaras que graban video en vivo, todos estos elementos son necesarios para lograr un sistema más robusto y completo para tener el control de los dispositivos de la vivienda. Como estamos en una época altamente tecnológica, pues de igual manera tenemos al alcance de nuestras manos, empresas que se dedican a la distribución de herramientas para poner en marcha un proyecto de electrónica. En cuanto al desarrollo de una aplicación web, es necesario tener un servidor el cual este alojado en un computador con su respectiva dirección en la Web.

OBJETIVO GENERAL

Diseñar e implementar un sistema domótico con acceso remoto utilizando una pasarela residencial.

OBJETIVOS ESPECIFICOS

- ❖ Diseñar e implementar una placa central domótica.
- ❖ Diseñar e implementar los módulos de domótica para accionar las escenas en la vivienda.
- ❖ Diseñar e implementar una aplicación Web.
- ❖ Realizar la comunicación entre aplicación, placa central y modulos (microcontroladores).
- ❖ Verificar el control y monitoreo de escenas programables desde la Web.

CAPÍTULO I

1. MARCO TEÓRICO

1.1. Domótica

La domótica se define como el conjunto de tecnologías que se dedican al control y la automatización inteligente de una vivienda, que permite una gestión eficiente del uso de la energía, que coopera con la seguridad y confort, además de comunicación entre el usuario y el sistema.



Figura 1-1 Domótica

Fuente: <http://www.cedom.es/images/web/queesdomotica.jpg>

Un sistema domótico se dedica a recoger información la misma que proviene de señales de algunos sensores o entradas, procesarla y emitir órdenes a unos actuadores o salidas. El sistema puede acceder a redes exteriores de comunicación o información. La domótica permite dar respuesta a los requerimientos que plantean estos cambios sociales y las nuevas tendencias de nuestro estilo de vida, facilitando el diseño de casas y hogares más humanos, más personales, poli funcionales y flexibles. La red de control del sistema domótico se integra con la red de energía eléctrica y se coordina con el resto de redes con las que tenga relación: telefonía, televisión, y tecnologías de la información, cumpliendo con las reglas de instalación aplicables a cada una de ellas. Las distintas redes coexisten en la instalación de una vivienda o edificio, la instalación interior eléctrica y la red de control del sistema domótico está regulada por la instrucción ITC-BT-51 Instalaciones de sistemas de automatización, gestión técnica de la energía y seguridad para viviendas y edificios (Cedom.es, 2016).

1.1.1. Beneficios

La domótica contribuye a mejorar la calidad de vida del usuario:

- Facilitando el ahorro energético: gestiona inteligentemente la iluminación, climatización, agua caliente sanitaria, el riego, los electrodomésticos, etc., aprovechando mejor los recursos naturales, utilizando las tarifas horarias de menor coste, y reduciendo así, la factura energética. Además, mediante la monitorización de consumos, se obtiene la información necesaria para modificar los hábitos y aumentar el ahorro y la eficiencia.
- Fomentando la accesibilidad: permite manejar de manera adecuada los elementos del hogar a las personas con discapacidades de la forma que más se ajuste a sus necesidades, además de ofrecer servicios de tele asistencia para aquellos que lo necesiten.
- Aportando seguridad mediante la vigilancia automática a través de cámaras que graben personas, animales y bienes, así como de incidencias y averías. Mediante controles de intrusión, cierre automático de todas las aberturas, simulación dinámica de presencia, fachadas dinámicas, cámaras de vigilancia, alarmas personales, y a través de alarmas técnicas que permiten detectar incendios, fugas de gas, inundaciones de agua, fallos del suministro eléctrico, etc.
- Convirtiendo la vivienda a un ambiente más confortable a través de la gestión de dispositivos y actividades domésticas. La domótica brinda servicios como: abrir, cerrar, apagar, encender, regular... los electrodomésticos, la climatización, ventilación, iluminación natural y artificial, persianas, toldos, puertas, cortinas, riego, suministro de agua, gas, electricidad.)
- Garantizando las comunicaciones mediante el control y supervisión remoto de la vivienda a través de su teléfono, PC, que permitirá la visualización de la información del funcionamiento de equipos e instalaciones. La instalación domótica permite la transmisión de voz y datos, incluyendo textos, imágenes, sonidos (multimedia) con redes locales (LAN) y compartiendo acceso a Internet; recursos e intercambio entre todos los dispositivos, acceso a nuevos servicios de telefonía IP, televisión digital, por cable, diagnóstico remoto, videoconferencias, tele-asistencia.

1.1.2. Funcionamiento

El funcionamiento de un sistema domótico está íntimamente ligado a los procedimientos de transmisión de información los cuales son los que permitan el diálogo entre periféricos y la unidad central.

Los terminales (electrodomésticos, puntos de luz, etc.), son equipos convencionales a los que se aporta una inteligencia o capacidad de comunicación a través de una interfaz y un protocolo. Los elementos de campo son todo el conjunto de sensores que permiten convertir una magnitud física en señal eléctrica, y los actuadores son aquellos capaces de transformar una señal eléctrica en una acción sobre algún dispositivo que se encuentra en el hogar.

Los elementos de campo o sensores envían y reciben señales a través de una red de comunicaciones (bus domótico), para comunicarse entre ellos y conjuntamente con la unidad central encargada de gestionar los intercambios de información. Estas señales de control están codificadas de una manera específica debido a los protocolos de comunicación.

Las aplicaciones de domótica hoy en día en el mercado son muy extensas. Sin embargo, las áreas en las que se han enfocado y expuesto son las relativas a la seguridad, la automatización de tareas domésticas, la gestión de la energía y las comunicaciones, así como la iluminación (marin et al., 2012).

1.1.3. Arquitectura de un sistema domótico

La estructura de un sistema domótico tiene una estrecha relación con los nodos de la red los cuales son los elementos que se encuentran dentro del hogar o residencia, por lo tanto podemos encontrar varios conceptos que describen esta estructura:

1.1.3.1. Arquitectura centralizada

En una arquitectura centralizada, un controlador centralizado, envía la información a los diferentes controladores que se encuentran, es decir la información en primera instancia debe pasar por la central y esta enviar a los distintos controladores.

1.1.3.2. Arquitectura distribuida o descentralizada

En esta arquitectura el elemento de control puede ser ubicado en cualquier parte de la vivienda, y por lo tanto este puede ejecutar las acciones que estén programadas en el mismo, sin necesidad de recibir información de otro controlador.

1.1.4. Casas inteligentes

Es una casa con un diseño arquitectónico propio y una tecnología avanzada, todo esto integrado y desarrollado en conjunto para que las personas que la habitan vivan aún más cómodamente.

Al hablar de diseño arquitectónico nos referimos a la creación de espacios y a la organización de la casa en un conjunto que permita llevar a cabo tanto actividades tradicionales del hogar como nuevas actividades basadas en tecnologías de vanguardia (Construtips.com, 2016).

Estas actividades pueden ser por ejemplo: el trabajo profesional a través de una computadora personal, el poder realizar movimientos bancarios y de compras en línea a través del Internet, o simplemente el uso video juegos frente a la televisión. La aplicación de la tecnología avanzada consiste en el uso simultáneo de la electricidad, la electrónica y la informática, aplicadas al diseño técnico de las viviendas consiste en la modificación local o remota de los parámetros de funciones cómo son:

Beneficios

- ❖ El confort ambiental óptimo, y que se logra a través del control del medio ambiente interno con la programación de horarios específicos para equipos de climatización, iluminación, etc.
- ❖ La seguridad propia y de cada uno de los integrantes de la vivienda, a través de dispositivos automáticos de control cómo lo son: alarmas para intrusión y pánico, control de fuego y humos, vigilancia interna y remota, etc.
- ❖ La comunicación con el exterior, a través del acceso a la red de Internet, la intercomunicación dentro de la vivienda, y el enlace de todos los sistemas de aplicación avanzada.
- ❖ El ahorro de la energía a través del control de la temperatura interna de los locales, el control de la iluminación y así como del control del consumo de los electrodomésticos.



Figura 1-2 Casa inteligente

Fuente: <http://www.construtips.com/images/stories/fotos/casaint3.jpg>

1.1.5. Protocolos de comunicación usados

En la actualidad existe variedad de tecnologías en el mercado dedicadas a ciertas áreas de las viviendas. Los tres sectores son: las casas ya construidas, las casas nuevas y los grandes edificios.

A continuación veremos las principales tecnologías existentes en el mercado para el desarrollo del sistema.

1.1.5.1. Zigbee

ZigBee es un estándar que se define como un conjunto de protocolos que están encaminados para el armado de redes inalámbricas de corta distancia y baja velocidad de datos. Estos trabajan en las bandas de 868 MHz, 915 MHz y 2.4 GHz y puede transferir datos con velocidades de hasta 250Kbps. Fue desarrollado por la Alianza ZigBee, que tiene a cientos de compañías desde fabricantes de semiconductores y desarrolladores de software a constructores de equipos OEMs e instaladores. Esta organización sin fines de lucro nace en el año 2002. Desarrolla un protocolo que adopta al estándar IEEE 802.15.4 para sus 2 primeras capas, es decir la capa física (PHY) y la subcapa de acceso al medio (MAC) y agrega la capa de red y de aplicación.

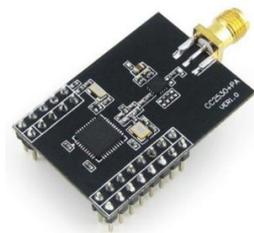


Figura 1-3 Módulo Zigbee

Fuente: <http://g01.a.alicdn.com/kf/HTB1jX5WIVXXXXbCXVXXq6xXFXXXq/ZigBee-font-b-Module-b-font-Wireless-Communication-far-Over-1500-Meters-Zigbee-Core-Board-CC2530F256.jpg>

La idea por la cual usar una conexión inalámbrica. La idea de usar una conexión inalámbrica para controlar sensores y adquirir datos tiene muchos años. En la actualidad existen diversas propiedades existen numerosas soluciones usadas en domótica pero el gran inconveniente que tienen es la incompatibilidad entre sensores, controles y equipos de procesamiento de datos que obliga a crear pasarelas (Gateway) para la interconexión dispositivos de diferentes marcas y se utiliza el estándar ZigBee fue creado con las siguientes especificaciones.

- ❖ Ultra bajo consumo que permita usar e quipos a batería.
- ❖ Bajo costo de dispositivos y de instalación y mantenimiento de ellos.
- ❖ Alcance reducido (típico menor a 50 metros).
- ❖ Optimizado para ciclo efectivo de transmisión menor a 0.1 %
- ❖ Velocidad de transmisión menor que 250 kbps. Típica: menor que 20 kbps

1.2. Pasarela residencial

Una Pasarela Residencial es un dispositivo que se encarga de conectar las redes de telecomunicaciones (datos, control, automatización) de la vivienda a una red pública de datos, como en este caso tenemos el Internet.

La Pasarela Residencial por lo general combina las funciones de un router, de un hub, de un modem con acceso a Internet y son repartidos para varios PC, de cortafuegos e incluso de servidor de aplicaciones de entretenimiento, como Vídeo/Audio bajo demanda, de comunicaciones, como Voz IP (telefonía sobre Internet) o de telecontrol como la domótica.

Es un dispositivo el mismo que permitirá la conectividad de forma total en los hogares con el mundo exterior que los rodean. El mismo que será capaz de realizar un sistema de tele control de los distintos electrodomésticos, sistemas de seguridad, de domótica, de gestión energética, equipos de electrónica de consumo como vídeos y TV, ordenadores personales y muchos más.

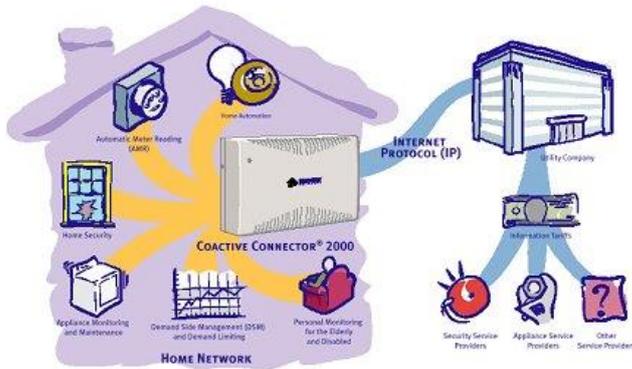


Figura 1-4 Pasarela residencial

Fuente: http://www.domicaviva.com/noticias/018-240802/coactive_conectorb.jpg

1.2.1. Características

Para que un equipo sea realmente catalogado como Pasarela Residencial, tenga cierto éxito o alcance una implantación masiva, el público tiene que sentir que realmente los e-services son útiles y aportan valor, confort y tranquilidad en su modo de vida. Para ello los expertos están de acuerdo en que las Pasarelas Residenciales deberían tener las siguientes características:

La instalación debe ser sencilla y la configuración rápida y asequible. Una vez conectada a la roseta telefónica o a alguna de las bocas del modem de cable o del router ADSL, la configuración debe ser inmediata. Igualmente, la asignación y especificación de las funciones que puede hacer cada dispositivo domótico o electrodoméstico debería ser automática (Casadomo.com, 2001).

- ❖ Tele carga de software. Es un proveedor que brinda servicios, que directamente el usuario, bajo supervisión del proveedor, deberían tener la capacidad de actualizar o tele cargar e incorporar nuevos servicios, y de igual manera poder configurarlos remotamente.
- ❖ Soporte para redes. En las Pasarelas Residenciales se deberían tener interfaces que permitan conectar las redes de datos de banda ancha (>10Mbps) que contengan tecnologías como las más esenciales como el cableado Ethernet.
- ❖ Capacidad para soportar múltiples servicios. Tener suficiente memoria, así como capacidad de procesamiento y un sistema operativo de alto desarrollo y robusto que pueda trabajar como multitarea, las pasarelas residenciales deberían tener la capacidad de ejecutar varias aplicaciones a la vez en cualquier instante de tiempo, donde cada una de ellas esta previamente configurada a dar un servicio diferenciado de las demás. La conexión de banda ancha será

compartida entre todos estos servicios con la multiplexación de datos, ya sea a nivel IP o nivel de aplicaciones.

- ❖ Monitorización usando páginas Web. Con poder hacerlo de forma local, como también de forma remota, el usuario podrá acceder al sistema de pasarela residencial y realizar alguna configuración si es necesaria, poder suprimir servicios como realizar una supervisión de su estado. Para ello las pasarelas deberán ser insertadas en pequeños protocolos como son HTTP y guardar la página en un servidor Web.

1.3. Protocolo de comunicación Modbus

1.3.1. Generalidades

MODBUS es un protocolo de comunicación serial estructurado en el modelo maestro/esclavo, a la fecha es un estándar de facto, es público, muy seguro, no requiere licencias y su implementación es relativamente fácil en dispositivos electrónicos, para múltiples proyectos suelen usar el microcontrolador PIC1F2550 que se lo puede programar para que responda comandos MODBUS, es por ello que en la mayoría de las cosas es acogido para realizar proyectos de control, y lo más destacable es que ya tiene más de 30 años en la industria.

En su origen el uso de MODBUS estaba exclusivamente orientado a la programación de los controladores lógicos PLCs, es uno de los protocolos de comunicación más usado en los entornos de la industria. Lo que significa de forma implícita que: tanto a nivel local como a nivel de red, en su versión TCP/IP, seguirá conocido como uno de los protocolos de referencia en las llamadas Smart Grids, redes de sensores, telecontrol, y un gran número de sistemas de información que transmiten señales para ser procesadas (Domotizando.wordpress.com, 2016).

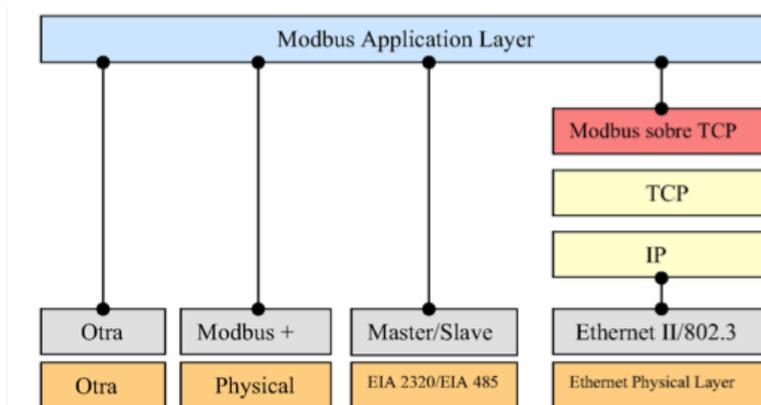


Figura 1-5 Arquitectura MODBUS

Fuente: <http://ricveal.com/contenido/uploads/2014/10/arquitectura-modbus.png>

En sus inicios estaba orientado para realizar una conectividad a través de líneas serie como pueden ser RS-232 o RS485 pero con el paso del tiempo han aparecido algunos cambios como nuevas variantes como la MODBUS TCP, que permite el encapsulamiento del MODBUS serie en tramas Ethernet TCP/IP de forma sencilla. Es por que su uso es muy relevante que se ha convertido en un protocolo estandarizado dentro del sector industrial esto se debe a varias razones diferenciales con respecto a los demás protocolos de comunicación.

El estándar MODBUS es público lo que permite a los fabricantes desarrollar dispositivos tanto master como esclavo. Este hecho facilita el acceso a la información y estructura del protocolo que, además es muy básica pero funcional para su objetivo.

Desde el punto de vista técnico, su implementación no es complicada y sobre todo el tiempo de desarrollo es corto respecto a otros protocolos en los que se complica la estructura de las tramas y en consecuencia el acceso a los datos que no están almacenados en estructuras complejas, y no se puede realizar una supervisión concreta.

1.3.2. Comunicación maestro-esclavo

MODBUS funciona en todos sus entornos con un maestro y uno o más esclavos, donde el maestro es de vital importancia porque controla en todo momento el inicio de la comunicación con todos los esclavos donde se encuentran las acciones programadas.

El esclavo por otro lado no puede reenviar los datos solicitados por el maestro, así se basa la estructura de la comunicación usando el MODBUS, el maestro se encarga de enviar los datos y los esclavos son los encargados de responder. En cada esclavo esta insertado una única dirección, de esta manera el maestro sabrá a quien enviar un dato correspondiente para comunicarse. A continuación veamos cómo funciona el envío de mensajes entre el maestro y el esclavo:

- Cada esclavo tiene su propia dirección y se puede conectar hasta 247 esclavos.
- El maestro siempre inicia la comunicación enviando un paquete de datos bien estructurado a todos los esclavos del sistema, entre otras muchas cosas también se inserta el número del esclavo con su respectiva dirección.
- El esclavo elegido responde, enviando lo que se le pide por medio también de un paquete de información bien estructurado.

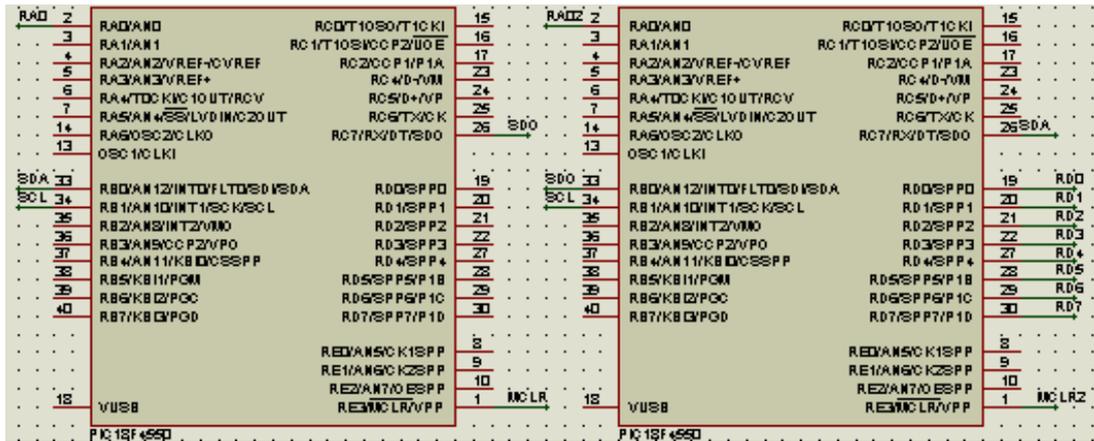


Figura 1-6 Maestro y esclavos

Fuente: http://1.bp.blogspot.com/_t8Ozc8jIT9I/SqeSUP03PqI/AAAAAAAAAEQ/ZNNkyJpIHVA/s400/RS485+2+Esclavos.JPG

1.3.3. Comunicación serie rs485

RS485 requiere de un protocolo "half dúplex" para operar, dado que las mismas líneas son usadas tanto para enviar como para recibir datos.

RS-485, es una mejora sobre el RS-422 puesto que se aumenta el número de dispositivos de 10 a 32 y es en donde se define las características eléctricas que se necesitan para asegurar de manera correcta los voltajes de señales bajo la carga máxima. Con el incremento en esta capacidad, se pueden crear redes de dispositivos conectados a un solo puerto serial RS-485. La inmunidad al ruido y la gran capacidad hacen que RS-485 sea la conexión serial que es más común en aplicaciones industriales que necesitan diversos dispositivos distribuidos en red a una PC o algún otro controlador para colección de datos como puede ser una placa con un microcontrolador, HMI, u otras operaciones. RS-485 es un gran conjunto de RS-422; por lo tanto, todos los dispositivos RS-422 pueden ser controlados mediante el serial RS-485. Por lo tanto se puede utilizar el hardware RS-485 para comunicación por puerto serial para cables de hasta 4000 pies.

Entre las aplicaciones que más se resaltan en cuanto a la norma RS485, se encuentran los sistemas de control de acceso, los sistemas distribuidos de control industrial y los sistemas de seguridad casera. También podemos encender una simple lámpara o un relé hasta una discoteca completa o un sistema de proceso industrial.

A través del puerto serie de un dispositivo se puede aprovechar todos los beneficios que te brinda el estándar RS485 para poder entrar en conexión con el mundo real, de manera física, y poder de esta forma modificar y controlar dispositivos y procesos remotos que estén a larga distancia.

Operar una cámara de vigilancia, las aberturas de puertas y ventanas, así como también persianas, todo manejado desde tu computador o un celular mediante su respectiva aplicación.

El protocolo MODBUS se maneja mediante el concepto de tablas de datos donde se puede almacenar la información en un esclavo, una tabla de datos es un bloque de memoria usado para almacenar datos en el esclavo, algunas tablas de datos que se usa en MODBUS son cuatro y se muestran a continuación:

Tabla 1-1 direcciones MODBUS

Dirección MODBUS	Dirección Usada en el protocolo	Nombre de la Tabla de Datos
1 - 9999	0000 – 9998	Output Coils (Lectura/escritura)
10001 – 19999	0000 – 9998	Inputs Contact (Lectura)
30000 – 39999	0000 – 9998	Inputs Registers (Lectura)
40001 – 49999	0000 – 9998	Holding Registers (Lectura/Escritura)

Realizado por: Tecnología digital de bajo

Fuente: <http://tecdigitaldelbajo.com/blog/27-modbus-parte-iii-que-es-el-modbus.html>

Observe que hay dos direcciones, una es usada para diferenciar entre las tablas de datos, la llamamos dirección MODBUS y la otra es la dirección usada en la comunicación entre el maestro y el esclavo, está es la dirección usada en la trama o paquetes de datos enviados entre maestros y esclavos.

Se puede afirmar que el bus permite una velocidad de datos de 10 y hasta 20 Mbps (a 12 metros de distancia), y de 100 Kbps cuando se conectan terminales o módulos separados 1200 metros entre sí. El sistema tiene la facilidad de “colgar” del bus hasta 32 terminales, pero actualmente ya se están utilizando sistemas de 128 y hasta 256 dispositivos que se pueden conectar entre si a una misma red de sólo dos hilos trenzados. Sin embargo es recomendable que el par de cables que transporta la información sea blindado, pero si este montaje no es posible, y debemos utilizar cables que trabajen de manera individual, por ello lo más efectivo es tener un tercer cable que trabaje a forma de referencia de tierra o GND. Un cable blindado ayudaría a atenuar los ruidos eléctricos que pueden afectar en la transmisión de los datos del sistema diferencial que utiliza el estándar RS485.

1.3.3.1. Output Coils

En MODBUS un coil representa un valor booleano típicamente usado para representar una salida, solo hay dos estados para el coil el ON y el OFF. Por ejemplo la tarjeta de relevadores usa dos Coils, para activar o desactivar precisamente dos relevadores.

El coil entonces, puede verse como una celda o un bloquecito de memoria que me permite almacenar el estado de un bit, que puede tener solo dos valores como se vio arriba, el ON o el OFF, o en lógica binaria, un 1 o un 0, o más cerca de la realidad física, tierra (0 Volts) y voltaje (5 Volts), como el lector quiera verlo, eso sí debe quedar claro que son solo dos valores por cada coil.

Hablemos de las direcciones, el coil según la tabla de datos puede tener una dirección MODBUS, la cual yo elijo y según la tabla anterior, puede ser entre 1 y 9999 valores, por simplicidad, se puede elegir el coil 1 y el coil 2 para, activar o desactivar, según nuestro caso, así como también se puede elegir el coil 9998 y coil 9999, eso está dependido de lo que el usuario designe para cada aplicación, por lo tanto cualquier dirección en ese rango se puede usar sin ningún problema.

Note como el esclavo responde a la petición del maestro, colocando el coil en el estado que se le pide, pero además regresa una trama de información al maestro, que prácticamente es lo mismo que se le pidió.

Sobra decir que la trama de información enviada por el maestro y la respuesta enviada por el esclavo, son números binarios.

1.3.3.2. Holding Registers

Los registros Holding, representan cantidades de 16 bits, se puede ver cada registro como un bloque o celda de memoria, que es capaz de almacenar números entre 0 y 65535.

Las direcciones usadas para los registros holding inician en la dirección 40001, pero puede usarse cualquiera de las mencionadas en la tabla para los registros holding.

Estos registros son de lectura/escritura, es decir que el maestro puede hacer peticiones de información para leer el dato que está almacenado, o puede enviar peticiones para escribirle un valor al registro holding en el momento que se requiera.

1.3.4. Modos de transmisión

1.3.4.1. Modo RTU

En este modo a los controladores se configuran en modo RTU (Remote Terminal Unit) cada byte de 8 bits en un mensaje que contiene dos caracteres hexadecimales de 4 bits respectivamente. Utiliza el sistema maestro-esclavo para el intercambio de mensajes. Permite hasta 247 esclavos, más solamente un maestro. Toda comunicación inicia con el maestro haciendo una solicitud a un esclavo, y este contesta al maestro el que fue solicitado. En ambos los telegramas (pregunta y respuesta), la estructura utilizada es la misma: Dirección, Código de la Función, Datos y Checksum. Solo el contenido de los datos posee tamaño variable.

El formato para cada byte en el modo RTU es el siguiente:

Sistema codificación:

8 bits en binario, hexadecimales 0-9, A-F.

2 caracteres hexadecimales contenidos en cada campo de 8 bits del mensaje.

Bits por byte:

1 bit de start.

8 bits de datos, el menos significativo se envía primero.

1 bit para paridad par o impar, no se envía BIT si no hay paridad.

1 bit de stop, si se usa paridad; 2 bits si no hay paridad.

1.3.4.2. Modo ASCII

Este modo particular se da cuando los controladores están configurados para comunicarse con la red MODBUS utilizando la codificación estándar ASCII. Para este modo cada byte de 8 bits en un mensaje es enviado como dos caracteres ASCII. El cambio que se registra en este modo radica en que los intervalos de tiempo los mismos que están por encima de un segundo, y son resultado que se da entre dos caracteres no generan ningún error.

El formato para cada byte en el modo ASCII es el siguiente:

Sistema de codificación:

Hexadecimal, caracteres ASCII van del 0 al 9, e incluyen letras de A-F

Un carácter+ hexadecimal es contenido en cada carácter ASCII del mensaje.

Bits por cada byte:

1 bit start.

7 bits de datos. El bit menos significativo se envía primero.

1 bit para paridad par o impar. No hay bit si no hay paridad.

Campo de chequeo de error:

1.4. Aplicación web

1.4.1. Introducción

Es claro ver que vivimos en una evolución diaria de la tecnología es así que el éxito espectacular de la web se basa en dos puntales fundamentales: el protocolo HTTP y el lenguaje HTML. El primero permite la elaboración de la web simple y sencilla de un sistema de comunicaciones que permita al usuario enviar todo tipo de ficheros de una forma sencilla, reduciendo el funcionamiento de un servidor y dando paso a que servidores de poca potencia puedan atender a miles de solicitudes y mejore los costes de despliegue. El segundo proporciona una variedad de mecanismos de elaboración de páginas entrelazadas simple y fácil, altamente eficiente y de uso muy sofisticado.

1.4.2. Protocolo http

HTTP se basa en el paradigma pedido/respuesta. La comunicación suele llevarse a cabo sobre una conexión TCP/IP en Internet. El puerto por defecto es el 80, pero otros puertos también suelen estar disponibles para usarse. Sin embargo no significa que el protocolo se implemente arriba de algún otro protocolo de Internet, siempre y cuando se garantice la seguridad y confiabilidad. Un

programa ejecutado por un cliente establece la conexión con un programa del servidor y solicita un pedido a este último a través de un método request, URL y la versión de protocolo, seguido por un mensaje donde se encuentran los modificadores del pedido, información sobre el cliente y contenido. El servidor responde con una línea de estado, donde incluye su versión de protocolo y un mensaje de éxito o error, así como también de un mensaje que contiene información del servidor, meta información y contenido.

De acuerdo con centro de investigación de la Web (2008, p.56), Nos indica que HTTP establece un dialogo con el navegador que estos pueden ser (Internet Explorer, Mozilla Firefox, etc.) a su vez el servidor web en el otro extremo una vez que están conectados. Este protocolo permite intercambiar contenidos de todo tipo como puede ser texto, páginas web, imágenes, audio, video, etc. Si considera que toda página Web está basada en el lenguaje HTTP.

1.4.3. Lenguaje HTML

Otra clave fundamental del WWW ha sido el lenguaje HTML (hypertext Mark-up language). Se trata de un lenguaje de marcas (se utiliza insertando marcas en el interior del texto) que nos permite represen- Software libre, de tal de forma que tenga estética el contenido y también referenciar otros recursos (imágenes, etc.), mandar enlaces a otros documentos (la característica más destacada del WWW), visualizar formularios para finalmente procesarlos, etc. HTML es el lenguaje mediante el cual se define el contenido de las páginas web. Generalmente se trata de un conjunto de etiquetas que se utilizan para definir un texto y algunos elementos de los cuales está compuesto una página web, como imágenes, listas, vídeos, etc.

En sus inicios fue creado con objetivos divulgativos de información con texto y algunas imágenes, no se pensó que llegara a ser utilizado para crear área de ocio y consulta con carácter multimedia (lo que es actualmente la web), de modo que, el HTML se creó sin pensar en todos los posibles usos que se le iba a dar y a todos los colectivos de gente que lo utilizarían en un futuro. Sin embargo, pese a esta deficiente planificación, sí que se han ido incorporando modificaciones con el tiempo, se han ido convirtiendo en estándares del lenguaje HTML.

1.4.3.1. Características

Slider control

Nuevo en HTML5 es el control deslizante. En realidad, es el elemento de "entrada", pero con un nuevo atributo tipo denominado "rango". Ejemplo de código HTML.

```
<form action = método "slider-control.php" = "post">
<input type = "rango" min = "1" max = "9" paso = "1" name = "rating" />
<input type = "submit" />
</ form>
```

Botón

Para crear un botón en HTML lo que tenemos que hacer son los siguientes pasos.

```
<hml>
<head>
<title>Botón 3D con CSS3</title>
</head>
<body>
<button class="boton3d">Cristalab</button>
</html>
```

Color picker

Es una sencilla Javascript y HTML selector de color para sus formularios HTML. Se extiende todas deseados <input> campos de un diálogo de selección de color. Puede pickerize sus campos de entrada en dos sencillos pasos.

El selector es una capa oculta, no utiliza ningún pop-ups y se puede colocar en cualquier lado del campo de entrada. Tiene 2 modos; VHS (saturación matiz, valor) o HVS, se puede configurar para mostrar un color por defecto en la carga inicial y su aspecto se pueden personalizar.

Hay funciones previstas para mostrar / ocultar el selector de color y puede generar RGB o HSV valores de una entrada de HEX.

1.4.4. Java script

JavaScript se trata de un lenguaje de programación que está basado principalmente para la creación y programación de páginas web dinámicas. Se considera una página web dinámica debido a que en el diseño se incorpora, animaciones de textos que aparecen y desaparecen a manera de efectos programados, acciones que se pueden activan al pulsar botones y abrir ventanas

despegables que pueden contener mensajes como puede ser de alerta al usuario (Eguíluz, 2008, p.5).

JavaScript es un lenguaje de programación técnicamente interpretado, de allí la necesidad de que primero se deba compilar los programas para luego proceder a ejecutarlos. Lo que quiere decir es que, los programas creados y escritos con JavaScript tiene la gran facilidad de que se puedan probar directamente en cualquier navegador web ya que no hay la necesidad de otros procesos ajenos para enfocar su resultado.

A pesar de su nombre, JavaScript no guarda ninguna relación directa con el lenguaje de programación Java, sin embargo a JavaScript lo interpreta directamente cualquier navegador, es por ello que el código se incrusta directamente dentro del código de HTML de la página (Sánchez, 2003, p.2).

De igual manera como es un lenguaje de programación, pues es de naturaleza saber que al momento de programar se pueden utilizar los conocimientos básicos de programación, es decir que podemos hacer uso de los operadores aritméticos como suma, resta, multiplicación, división, etc. También operadores lógicos como and, not, or. Y finalmente operadores comparativos que son utilizados para realizar procesos condicionales en el programa, estos pueden ser: mayor, menor, igual, mayor o igual, menor o igual.

1.4.5. Lenguaje Php

PHP prácticamente es un lenguaje con código abierto muy utilizado hoy en día, específicamente para desarrollo de páginas web y que puede ser combinado y insertado en HTML. Es muy utilizado debido a que un gran número de páginas y portales web viendo siendo creadas con PHP. El código abierto nos da a entender que es de uso libre y gratuito para todos los programadores que quieran hacer uso de él. Insertado en HTML significa que en un mismo archivo vamos a poder combinar tanto código PHP con código HTML.

PHP se utiliza para generar páginas web dinámicas. Recalquemos que llamamos página estática a aquella cuyos contenidos permanecen siempre igual, mientras que las páginas dinámicas a aquellas cuyo contenido no es el mismo siempre está cambiando. Por ejemplo, los contenidos pueden haber cambios, en base a los cambios que haya en una base de datos, de búsquedas o aportaciones de los usuarios, (Aprenderaprogramar.com, 2016).

1.4.5.1. Características

- Es un lenguaje multiplataforma.
- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información que se encuentre almacenada en una base de datos.
- El código fuente que es escrito en PHP es invisible al navegador web y al cliente puesto que es el servidor el que se encarga de ejecutar el código y enviar un resultado HTML al navegado. Hace que la programación en PHP tenga seguridad y confiabilidad.
- Máxima conexión con la mayor parte de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Capacidad de incrementar su potencial utilizando módulos (llamados ext's o extensiones).
- Contiene una amplia documentación en los sitios web oficial, entre algunas de ellas se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos los usuarios.
- Permite aplicar algunas técnicas de programación las mismas que están orientadas a objetos.
- Se acopla una biblioteca nativa de funciones sumamente amplia e incluida.
- No necesita definición de tipos de variables sin embargo sus variables se pueden evaluar mediante el tipo que estén manejando en tiempo de ejecución.
- Tiene manejo de excepciones (desde PHP5).

1.4.5.2. Principales tareas

Funciones de correo electrónico

Podemos con una facilidad asombrosa enviar un e-mail a una persona o lista parametrizando toda una serie de aspectos tales como el e-mail de procedencia, asunto, persona a responder. Otras

funciones menos frecuentes pero para mayor utilidad para gestionar correos electrónicos son incluidas en sus librerías.

Gestión de bases de datos

Resulta difícil concebir un sitio actual, potente y rico en contenido que no es gestionado por una base de datos. En el lenguaje PHP ofrece interfaces para el acceso a la mayoría de las bases de datos comerciales y por ODBC a todas las bases de datos posibles en sistemas Microsoft, a partir de las mismas podremos editar el contenido en nuestro sitio con absoluta facilidad.

Gestión de archivos

Crear, borrar, mover, modificar...cualquier tipo de operación más o menos razonable que se nos pueda ocurrir tiene la posibilidad de ser realizada a partir de una alta gama de librerías de funciones para gestionar los archivos por PHP. También se puede transferir archivos por FTP mediante sentencias en nuestro código, y un protocolo para el cual PHP ha previsto también gran cantidad de funciones.

1.4.6. Estructura de una aplicación web

Por lo general pueden existir variaciones posibles, una aplicación web esta generalmente estructurada como una aplicación de tres-capas. Es su forma común, el navegador web ofrece la primera capa y un motor capaz de usar alguna tecnología web dinámica (por ejemplo PHP, JavaServlets o ASP, ASP. NET, CGI, ColdFusion, embPerl, Python [Programming language] o Ruby on Rails) constituye la capa intermedia. Por último la base de datos constituye la tercera y última capa.

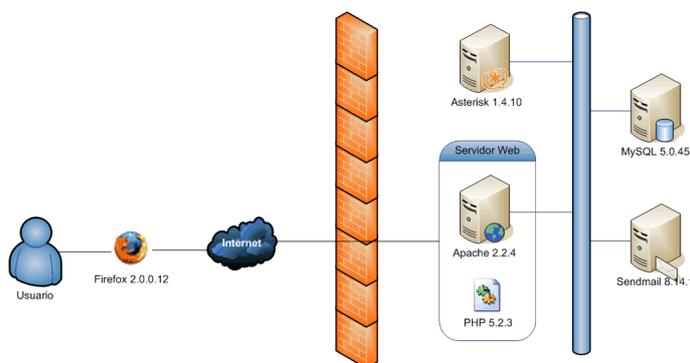


Figura 1-7 Estructura Web

Fuente: <http://i29.tinypic.com/21fep.gif>

El navegador web envía peticiones a la capa media que ofrece servicios valiéndose de consultas y actualizaciones a la base de datos y a su vez proporciona una interfaz de usuario para que acceda a la Web.

En la ingeniería de software se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador. En otras palabras, es una aplicación software que se codifica en un lenguaje soportado por los navegadores web en la que se confía la ejecución al navegador (Alemán, 2011).

El cliente se ocuparía de la capa de presentación, el servidor tienen la base de datos (capa de almacenamiento), y la lógica de negocio (nivel de aplicación) sería en uno de ellos o en ambos. Aunque esto aumenta la escalabilidad de las aplicaciones y separa la pantalla y la base de datos, aun así no se permite la verdadera especialización de las capas, por lo que la mayor parte de las aplicaciones superan este modelo.

1.4.6.1. Ventajas

- **Ahorra tiempo:** Suele estar listo para realizar tareas sencillas sin necesidad de descargar e instalar ningún programa adicional al que se está usando.
- **No hay problemas de compatibilidad:** Con tener un simple navegador actualizado para poder correr el programa.
- **Actualizaciones inmediatas:** Como el software lo gestiona el propio desarrollador, si cuando nos conectamos en cualquier instante estamos usando siempre la última versión que haya terminado.
- **Consumo de recursos bajo:** Dado que la mayor parte de la aplicación no se encuentra en nuestro ordenador, y la mayoría de las tareas que realiza el software no consumen recursos nuestros ya que dicha operación se realizan desde otro ordenador.
- **Multiplataforma:** Hay la posibilidad de usar desde cualquier sistema operativo porque lo que sólo se necesita tener un navegador.
- **Portables:** Es independiente del ordenador donde se utilice, porque se accede mediante una página web (sólo es necesario disponer de acceso a Internet). La reciente tendencia al acceso

a las aplicaciones web a través de teléfonos móviles requiere sin embargo un diseño específico de los ficheros CSS para no dificultar el acceso de estos usuarios.

- **La disponibilidad:** suele tener un alto grado de disponibilidad, porque el servicio se ofrece desde múltiples localizaciones para asegurar la continuidad del mismo.
- **Los virus:** Estos datos no suelen dañarse por la presencia de un virus porque éstos están guardados en el servidor de la aplicación.
- **Colaboración:** Gracias a que todos los accesos al servicio se realiza desde una única ubicación es sencillo el acceso y compartición de datos por parte de varios usuarios. Tiene un amplio sentido, por ejemplo, en aplicaciones online de calendarios como de oficina.

1.4.6.2. Desventajas

- Ofrecen menos funcionalidades que las aplicaciones de escritorio. Se debe a que las funciones que se pueden ejecutar desde un navegador son más limitadas que las que se pueden realizar desde un sistema operativo. Sin embargo los navegadores están más listos para mejorar en este aspecto. La aparición de HTML5 representa un gran avance en este sentido. Hay gran posibilidad de añadir nuevas funciones a estas aplicaciones gracias al uso de Aplicaciones de Internet con alto grado de desarrollo.
- La disponibilidad de que la información es que depende de un tercero, el proveedor de la conexión a internet o quien provee la conexión entre el servidor de la aplicación y el cliente. Así que la disponibilidad del servicio está sujeto al proveedor de servicio web.

1.4.6.3. Cliente

El cliente es aquel proceso que facilita al usuario formular los requerimientos y enviarlos al servidor, se le conoce con el término front-end. El Cliente es quien maneja todas las funciones que están relacionadas con la modificación y despliegue de los datos, por lo que están desarrollados sobre múltiples plataformas que permiten la construcción de interfaces gráficas de usuario (GUI), y a más de acceder a los servicios distribuidos en cualquier parte de una red. Las funciones que lleva a cabo el proceso cliente se resumen en los siguientes puntos:

- Administrar la interfaz de usuario.

- Interactuar con el usuario.
- Procesar la lógica de la aplicación y hacer validaciones locales.
- Generar requerimientos de bases de datos.
- Recibir resultados del servidor.
- Formatear resultados.

1.4.6.4. Servidor

Este proceso es el encargado de atender a múltiples clientes que hacen diferentes peticiones de algunos recursos administrados por él. Al proceso servidor se le conoce con el término back-end. El servidor por lo general maneja todas las funciones relacionadas con la mayoría de las reglas del negocio y los recursos de datos. Las funciones que lleva a cabo el proceso del servidor son los siguientes puntos:

- Aceptar los requerimientos de bases de datos que hacen los clientes.
- Procesar requerimientos de bases de datos.
- Formatear datos para transmitirlos a los clientes.
- Procesar la lógica de la aplicación y realizar validaciones a nivel de bases de datos.

1.4.7. Entornos web

El entorno web basa en un ambiente de desarrollo y/o ejecución de programas o servicios en el espacio de la web en general. El entorno web es una como una manera de interfaz de usuario gráfico. Por ejemplo, para recibir un email se puede utilizar una aplicación (como Outlook de Microsoft), pero a su es muy usual utilizar un “entorno web” para la recepción y envío de correos electrónicos, como el que ofrecen Gmail de Google o Hotmail de Microsoft. Existen herramientas, programas, lenguajes de programación y desarrollo que son específicos para el diseño de aplicaciones dentro de un entorno web. De hecho se cree que poco a poco las aplicaciones e incluso gran parte del sistema operativo irán migrando hacia un entorno web.

1.5. Microcontrolador

Un microcontrolador es conocido como un circuito integrado que se puede programar que esta encapsulado en un solo chip las unidades de memoria para el almacenamiento de datos, aritmética – lógica para el cálculo de operaciones, las unidades de entrada y salida para comunicación con otros periféricos, temporizadores y el controlador de interrupciones. La memoria generalmente está constituida por memoria RAM compuesta por registros que almacena datos temporales, memoria EEPROM para el almacenamiento del programa que se debe ejecutar. La unidad aritmética lógica ALU es la encargada de realizar las operaciones aritméticas suma, resta y multiplicación y las operaciones lógicas como And, Or, Or- exclusivo.

Para funcionar el microcontrolador dispone de un conjunto de instrucciones que son traducidas a lenguaje de máquina (1's y 0's) por un programa que se llama Ensamblador.

Igualmente existen Compiladores que se encargan de traducir un lenguaje de alto nivel como el lenguaje C a lenguaje o código de máquina. En ambos casos es el código ejecutable que se debe grabar en la memoria del micro (EEPROM) para que se ejecute el programa y desarrolle la aplicación que se quiere (Anon, 2016).

1.5.1. Características de un microcontrolador

Las principales características de los microcontroladores son:

- ❖ **Unidad de Procesamiento Central (CPU):** Típicamente de 8 bits, pero también las hay de 4, 32 y hasta 64 bits con arquitectura Harvard, con memoria/bus de datos separada de la memoria/bus de instrucciones de programa, o arquitectura de von Neumann, también llamada arquitectura Princeton, con memoria/bus de datos y memoria/bus de programa compartidas.
- ❖ **Memoria de Programa:** Es una memoria ROM (Read-Only Memory), EPROM (Electrically Programable ROM), EEPROM (Electrically Erasable/Programable ROM) o Flash que almacena el código del programa que típicamente puede ser de 1 kilobyte a varios megabytes.
- ❖ **Memoria de Datos:** Es una memoria RAM (Random Access Memory) que típicamente puede ser de 1, 2 4, 8, 16, 32 kilobytes.

- ❖ **Generador del Reloj:** Usualmente un cristal de cuarzo de frecuencias que genera una señal oscilatoria de entre 1 a 40 MHz, o también resonadores o circuitos RC.
- ❖ **Otras opciones:** – Conversores Análogo-Digitales (A/D, analog-to-digital) para convertir un nivel de voltaje en un cierto pin a un valor digital manipulable por el programa del microcontrolador. Moduladores por Ancho de Pulso (PWM, Pulse-Width Modulation) para generar ondas cuadradas de frecuencia fija pero con ancho de pulso modificable.

1.5.2. Proceso de desarrollo

1.5.2.1. Desarrollo de software

En la siguiente etapa es el lugar donde debemos escribir y compilar/ensamblar nuestro programa que determinaría los controles del μC y su funcionamiento. Hoy en día existen distintas maneras de desarrollar un programa, lógicamente dependerá del lenguaje inicial que se utiliza para escribir el programa:

- ❖ Lenguaje Assembly - Lenguaje de Máquina/Código Objeto (.asm) → ensamblador → (.hex, .o, .bin, .coff).
- ❖ Lenguaje de Alto Nivel - Lenguaje Assembly - Lenguaje de Máquina/Código Objeto (.c, .cpp) → compilador → (.asm) → ensamblador → (.hex, .o, .bin, .coff).

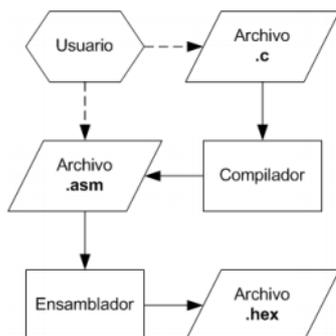


Figura 1-8 Programación de un microcontrolador

Fuente: <http://4.bp.blogspot.com/-lr1iR28RB20/T0F8x6OVLVI/AAAAAAAAA9w/vEGf3hPkJwk/s320/proceso.PNG>

1.5.2.2. Programación del microcontrolador

Es el proceso que corresponde a utilizar un programa en el PC que recoge un código ensamblado (.hex, .o, .bin, .coff) para el microcontrolador específico, y envía los datos mediante algún puerto

(serial, paralelo, USB, etc.) a un dispositivo que lo escribe en la memoria del microcontrolador. Se suele denominar programador tanto al software como al hardware involucrado para este propósito, lo cual puede prestarse para alguna confusión.

El software programador también se lo conoce con el nombre de downloader, ya que su objetivo es descargar o transferir desde el PC al microcontrolador el código ensamblado. Es muy importante recalcar que no deben confundirse los términos desarrollo o programación del software y programación del microcontrolador, siendo el primero el encargado de escribir el programa, de tal manera que el segundo se refiere a transferir el código de máquina a la memoria del microcontrolador.

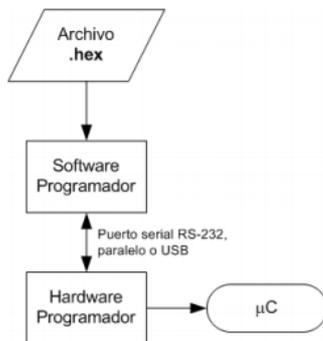


Figura 1-9 Proceso de programación de un PIC

Fuente: <http://2.bp.blogspot.com/-IYW-dZrCWtE/T0F87nXAdTI/AAAAAAAAA98/YFkGznus1II/s400/prog.PNG>

1.5.2.3. Prueba y verificación

Una vez programado el microcontrolador, se puede colocar en el circuito final para comprobar si el resultado coincide con el adecuado funcionamiento. Existen variedad de herramientas de software que permiten simular el comportamiento de un microcontrolador, muy fundamentales cuando el programa alcanza cierta complejidad y no podemos montar físicamente nuestro diseño. Para para solucionar los problemas en un circuito real, el instrumento más utilizado es el analizador lógico, como también algunas herramientas donde se puede simular un proceso que esté programado en un microcontrolador.

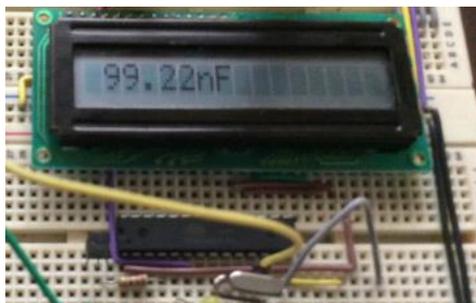


Figura 1-10 Pruebas de un microcontrolador

Fuente: <https://nomadaselectronicos.files.wordpress.com/2015/03/2015-04-02-08-46-23.jpg?w=618>

1.5.3. Comunicación uart

De sus siglas que significan (recepción-transmisión asíncrona universal) es uno de los protocolos serie más utilizados. La mayoría de los controladores disponen de hardware UART. Usa una línea de datos simple para transmitir y otra para recibir datos. Comúnmente, 8 bits de datos son transmitidos de la siguiente forma: un bit de inicio, a nivel bajo, 8 bits de datos y un bit de parada a nivel alto. El bit de inicio a nivel bajo y el de parada a nivel alto indican que siempre hay una transmisión de alto a bajo para iniciar la transmisión. Para largas distancias de comunicación, utilizar 5V en UART no es fiable, por lo que es convertido a un voltaje más alto de +12V para un 0 lógico y 12V para un 1. El formato de datos se conserva. Es entonces cuando tienes RS-232 (Blog Patagonia Technology, 2014).

Las funciones principales de chip UART son de manejar las interrupciones de los dispositivos conectados al puerto serie y de convertir los datos en formato paralelo, transmitidos al bus de sistema, a datos en formato serie, para que puedan ser transmitidos a través de los puertos y viceversa.

Mediante esta comunicación podemos realizar la comunicación de dos o más microcontroladores, a través de un bus de datos mediante un integrado max485 o su reemplazo, en este caso se realizará una comunicación Modbus la cual se habló anteriormente.

1.5.4. Integrado max485

RS-485 es una norma de comunicación serie asíncrona que utiliza dos líneas (A y B) de manera que la tensión diferencial (sin tierra absoluta) entre ambas, marca el nivel lógico que se está enviando. La transferencia es semi-dúplex (al trabajar con 2 hilos solamente) ya que sólo es posible que un equipo envíe información gobernando las líneas de datos (A y B) y otro u otros equipos reciban. Está pensada para una 11 comunicación multipunto. El medio físico es un par de hilos trenzados entre sí para reducir el posible ruido electromagnético inducido. El estándar RS-485 permite la interconexión de hasta 32 dispositivos sobre un único par de hilos, con velocidades de hasta 10Mbits por segundo y una distancia máxima de 1200 metros.

Al tratarse de un estándar bastante abierto permite muchas y muy diferentes configuraciones y utilidades. Esta norma sólo especifica características eléctricas de una unidad, pero no especifica o recomienda ningún protocolo de datos como tampoco algún conector específico. Los circuitos integrados que manejan esta norma además pueden soportar “colisiones”, es decir que más de un circuito transmisor esté emitiendo. Al transmitir en modo diferencial, si el terminal A está a una

tensión superior a B (con un valor diferencial superior a 0,2V) se estará recibiendo un “1” y en caso contrario (tensión de B superior en más de 0,2V a A) se interpreta un “0”.

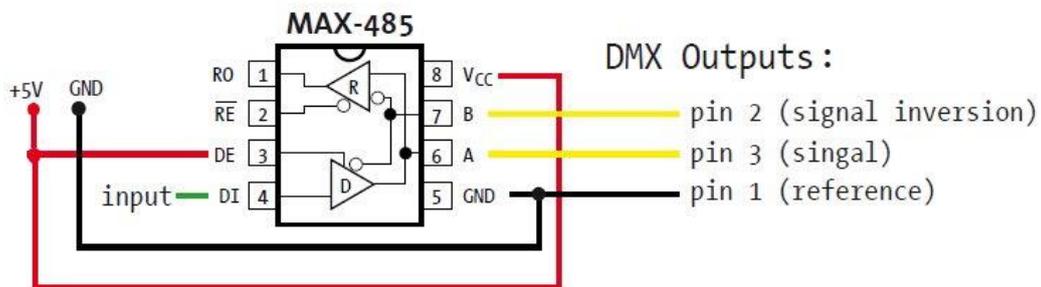


Figura 1-11 Conexión max-485

Fuente: <http://www.pablogindel.com/imagenes/art36.jpg>

Tabla 1-2 Pines de conexión MAX-485

Pin	Nombre	Descripción
1	RO	Receiver Output
2	RE	Receiver Output Enable Active LOW
3	DE	Driver Output Enable Active HIGH
4	DI	Driver Input
5	GND	Ground Connection.
6	A	Driver Output/Receiver Input Non-inverting
7	B	Driver Output/Receiver Input Inverting
8	Vcc	Positive Supply 4.75V<Vcc< 5.25V

Realizado por: Edison Lema

Fuente: Propia del autor

Al trabajar con un integrado MAX485 en modo half-dúplex la transmisión debe de ser compartida, en un momento uno envía y luego este escucha. Para ejecutar una transmisión a través de estos integrados es necesario controlar el terminal DE (habilita envío) y RE (habilita recepción), como podemos ver indica una lógica opuesta se puede unir para ser controlados mediante una solo línea. Cuando transferimos información hay que colocar el pin 2 y el 3 en nivel alto, de esta forma se habilita el integrado para enviar información por el pin 4 y de on mediato se bloquea para recibir información por el pin 1, en caso contrario, para recibir información hay que colocar el pin 2 y el 3 en nivel bajo, en este caso se bloquea el envío de información por el pin 4 y se activa la recepción de información por el pin1.

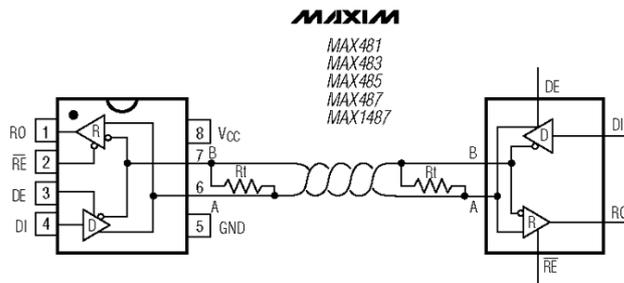


Figura 1-12 Conexión Max-485

Fuente: <http://www.extremadura-web.es/Blog/wp-content/uploads/2013/04/imax485.png>

Muchos enlaces con RS-485 trabajan con una resistencia de 120 ohm (R_t) a través de las líneas A y B en cada extremo de la línea, para proveer a los dispositivos de una adaptación de impedancia al medio de transmisión, pero también se puede pasar por alto dicha resistencia si ocupamos un reemplazo del integrado MAX485 como es el caso del SN7517 (Neoteo, 2009).

1.5.5. PIC18F2550

Ideal para baja potencia (nanoWatt) y aplicaciones de conectividad que se benefician de la disponibilidad de tres puertos serie: FS-USB (12 Mbit / s), I²C™ y SPI™ (hasta 10 Mbit / s) y una asíncrona (LIN capaces) de serie puerto (EUSART). Las grandes cantidades de memoria RAM para el almacenamiento intermedio y memoria del programa FLASH mejorada hacen que sea ideal para el control integrado y monitoreo aplicaciones que requieren conexión periódica con un (legado gratis) ordenador personal a través de USB para la carga de datos / descarga y / o actualizaciones de firmware. Mientras opera hasta 48 MHz, el PIC18F2550 es también sobre todo el software y hardware compatible con los dispositivos OTP USB PIC16C745 baja velocidad.



Figura 1-13 PIC18F2550

Fuente: http://www.microchip.com/_images/ics/medium-PIC18F2550-SPDIP-28.png

1.5.5.1. Características

Microcontrolador CMOS de 8 bits de la familia media alta de microchip y tiene las siguientes especificaciones.

- ❖ Procesador: microcontrolador de alto rendimiento, multifunciones PIC18F2550-I/SP de 48 Mhz, 28 pines encapsulado DIP, de Microchip.
- ❖ Arquitectura: Harvard, memoria de código de 16 bits, separada de la memoria de datos de 8 bits. Procesamiento “pipeline”.
- ❖ Tecnología: RISC (reduced instruction set computer), con 75 instrucciones.
- ❖ Puerto USB v2.0: transceptor integrado al microcontrolador. 12 Mb/s
- ❖ Memoria: 16K localidades de 16 bits de FLASH (ó 32 Kbytes), 2 Kb localidades (8 bits) de RAM, 256 localidades (8 bits) de EEPROM.
- ❖ Auto programación de la memoria FLASH: a través del puerto USB, por medio de un firmware bootloader residente.
- ❖ Puertos digitales: puerto A de 5 bits, puerto B de 8 bits, puerto C de 8 bits un total de 21 bits programables como entradas o como salidas.
- ❖ Capacidad de salidas: cada bit de salida puede tomar (“sink”), ó generar (“source”), hasta 25 miliamperios.
- ❖ Puertos seriales: USART compatible RS232. SSP Puerto serial síncrono con 2 modos de operación: SPI (Serial Peripheral Interface, modos Master/Slave) eI2C (Integrated, Integrated Circuit. Modo Slave)
- ❖ Temporizadores: 4 temporizadores de 16 bits. Un generador de PWM
- ❖ Convertidores A/D: 10 canales, con 10 bits de resolución.
- ❖ Funciones adicionales: power-on reset, brown out reset, power up timer, watchdog, code protection, sleep (bajo consumo).

1.6. Raspberry pi modelo b

Es considerado como un pequeño ordenador que tiene el tamaño de tarjeta de crédito que se conecta a vía HDMI a una pantalla donde podemos realizar las configuraciones correspondientes y un teclado (López, 2016). Se encuentra colocado sobre una placa que soporta abundantes componentes necesarios en una computadora común. Es una pequeña computadora capaz que tiene todas las funcionalidades de una pc normal, que puede ser utilizado para realizar varias de las cosas que su PC de escritorio hace, como hojas de cálculo, procesadores de texto y juegos y obviamente reproduce vídeo de alta definición (ABC.es, 2016).

RASPBERRY PI MODEL B

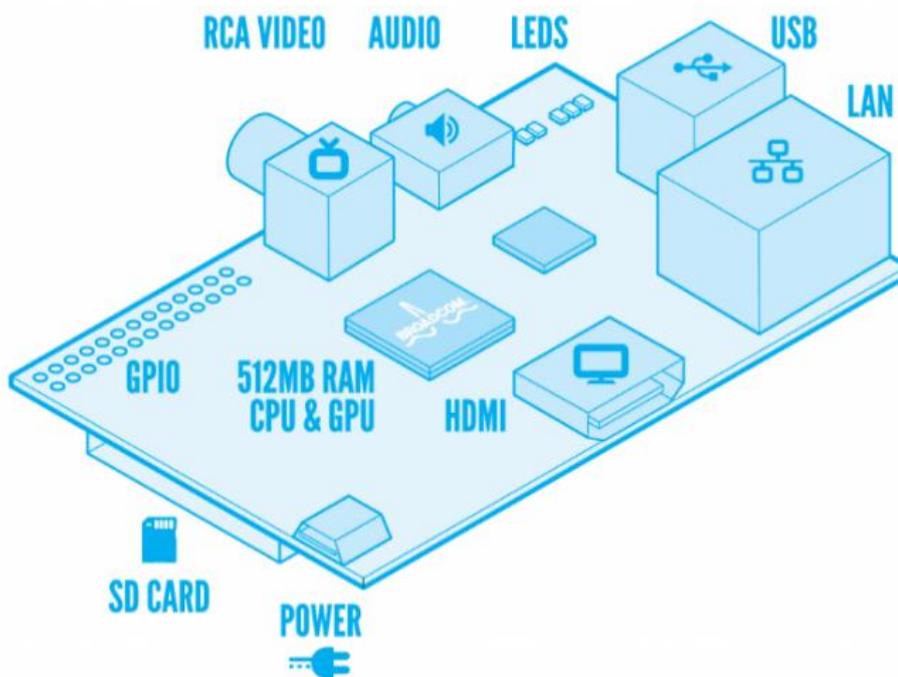


Figura 1-14 Raspberry pi B

Fuente: <http://www.raspberrypi.org/wp-content/uploads/2011/07/RaspiModelB-1024x902.png>

La placa, que tiene un tamaño muy reducido consta de varios puertos y entradas, dos USD, uno de Ethernet y salida HDMI. Estos puertos permiten conectar el miniordenador a otros dispositivos, teclados, ratones y pantallas.

También posee un System on Chip que contiene un procesador ARM que corre a 700 Mhz, un procesador gráfico VideoCore IV y hasta 512 MG de memoria RAM. Es posible instalar sistemas operativos libres a través de una tarjeta SD (ABC.es, 2016).

1.6.1. Especificaciones hardware

- CPU: ARM1176JZF-S a 700 MHz (familia ARM11) perteneciente a la versión 6, por lo que puede soportar RISC OS.
- GPU: Broadcom VideoCore IV, es capaz de reproducir vídeos en 1080p a 30fps. Hay reviews sobre esta placa que dicen que hasta puede asimilarse a unos gráficos de la primera XBOX.
- RAM: Tiene una versión de 256MB (modelo A) y una de 512MB (modelo B). Comparte una parte de la memoria con la GPU.
- Salidas de vídeo, HDMI y audio.
- 1/2 Puertos USB dependiendo el modelo.
- Puerto Ethernet (sólo el modelo B).
- Consumo energético 500 mA y 2,5W en su modelo A y 700mA y 3,5W en el modelo B.
- Alimentación de 5V por el puerto micro USB.

1.7. Altium Designer

Es un software y a su vez una aplicación única y unificada que incorpora todas las tecnologías y capacidades necesarias para el desarrollo completo de productos electrónicos.

Altium Designer integra:

- Diseño de tarjetas y sistemas basados en FPGAs.
- Software de desarrollo para procesadores basados en FPGA.
- Diseño, edición y fabricación de PCB.

Su entorno de diseño único. Y vez combinado con la capacidad de gestión de datos de diseño, se convierte en la solución completa para el desarrollo de productos electrónicos - una solución que satisface las necesidades de diseño actuales y futuras.

Altium Designer incluye la tecnología de productos anteriores como Protel y P-CAD. Altium Designer es un entorno de desarrollo electrónico unificado que incorpora las herramientas necesarias para desarrollar los productos electrónicos inteligentes más avanzados. Basado en una aplicación simple suministra la tecnología de diseño mucho más avanzada, la gestión de la integridad de los datos de diseño y un nuevo enfoque innovador para el diseño (User, 2016).

1.7.1. Productos

Altium es líder en el desarrollador de software de diseño electrónico para:

- Edición de esquemas
- Diseño de PCB
- Diseño de FPGAs
- Software Embedded
- Gestión de datos inteligente

Altium Designer proporciona un entorno unificado de desarrollo de productos electrónicos, para todos los aspectos del proceso de desarrollo electrónico, incluyendo:

- Creación de diseño, generación de esquema
- Diseño físico de PCB
- Simulación mixta del circuito
- Simulación / análisis de integridad de señal
- Fabricación de PCB
- Diseño de hardware FPGA
- Desarrollo del software para CPUS integradas en FPGAs
- Depuración del sistema basado en FPGA mediante Altium NanoBoard

1.7.2. Herramientas de diseño

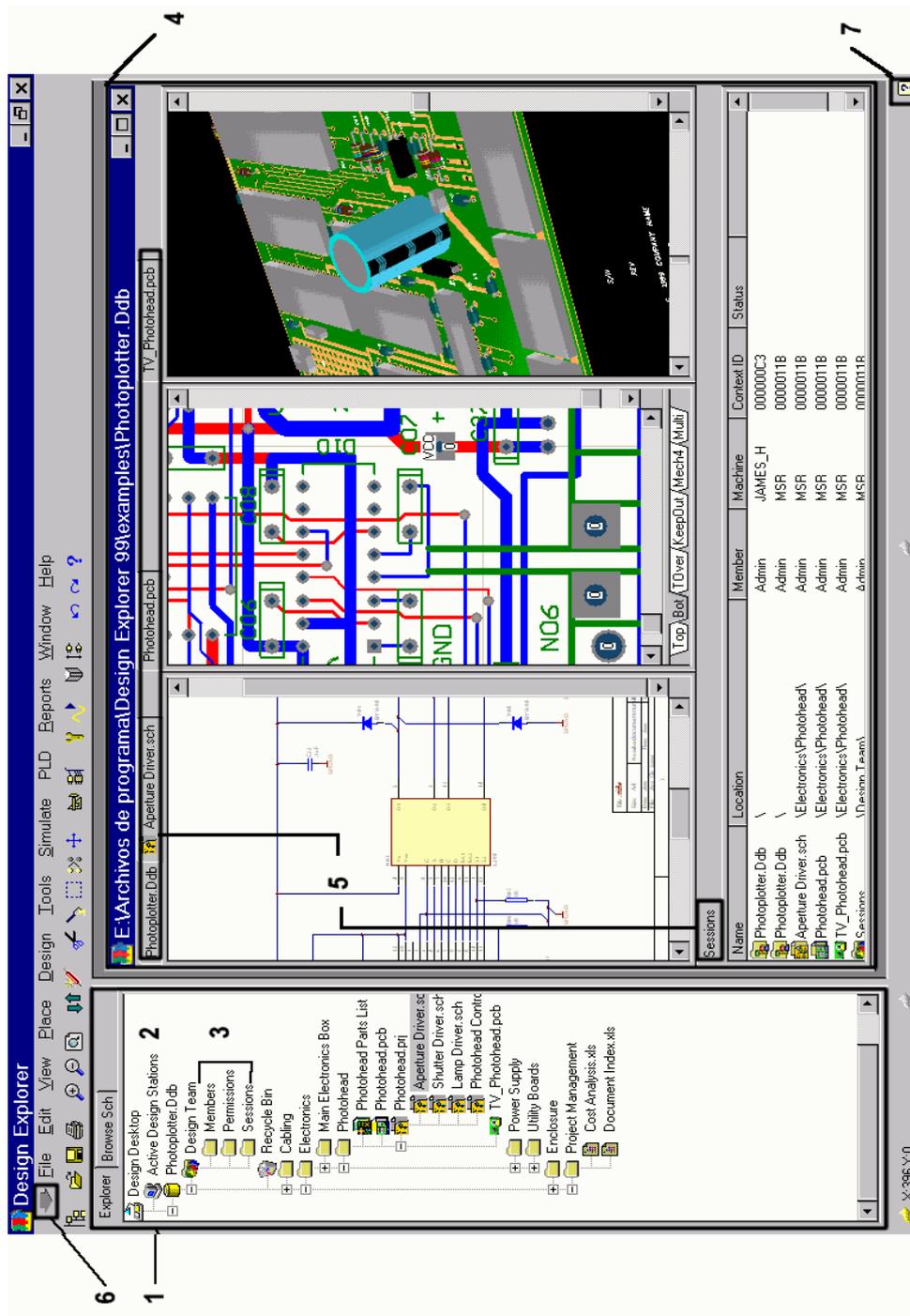


Figura 1-15 Partes de Altium Designer

Fuente: http://www.uhu.es/manual_sanchez/tutoriales/xilinx/Image2.gif

- 1) **Gestor de Diseño.** Este panel muestra el árbol de navegación similar al Explorador de Windows, nos ofrece una vista de las bases de datos de diseño que tenemos abiertas y los documentos que contienen. Haciendo clic en los iconos + y – podemos expandir y contraer las carpetas. Haciendo clic en un documento o carpeta podemos abrirlo o verlo en la ventana

de diseño principal. Si hacemos clic con el botón derecho en un documento o carpeta aparece un menú de opciones. También se muestran en el panel del gestor de diseño paneles de edición específicos que aparecen como pestañas cuando se editan varios tipos de documentos.

- 2) **Estaciones de diseño activas.** Haciendo clic en el símbolo + podemos ver una lista de ordenadores de la red que están ejecutando Protel.
- 3) **Equipo de diseño.** Esta es una carpeta especial dentro de cada base de datos de diseño que nos permite configurar y mostrar información sobre las propiedades de compartición del diseño.
- 4) **Ventana de diseño.** Cada base de datos de diseño abierta dispone de una ventana de diseño asociada empleada para mostrar documentos de diseño abiertos y carpetas.
- 5) **Pestañas de diseño.** Los documentos abiertos y las carpetas aparecen como pestañas en la ventana de diseño. Haciendo clic en una pestaña hace a ese documento o carpeta activa. Al hacer clic con el botón derecho sobre la pestaña aparece un menú de opciones, incluyendo la posibilidad de dividir la ventana para mostrar varias pestañas al mismo tiempo.
- 6) **Menú del Explorador de diseño.** Haciendo clic en la flecha de la esquina izquierda de la barra de menús activa el menú del Explorador, que ofrece varias opciones del sistema.
- 7) **Asistente de ayuda.** Haciendo clic en este icono nos ofrece ayuda simplemente tecleando la pregunta. Existe un sistema de análisis de preguntas en lenguaje natural (naturalmente en inglés) que presenta una lista de temas de ayuda relacionados.

1.7.3. Partes importantes para elaborar un PCB

- ❖ **Esquemático** vendría a ser una copia del circuito de la hoja de datos del componente esto es para poder exportar el circuito al PCB.
- ❖ **PCB** hacemos referencia al circuito en sí, esto abarca el diseño de las pistas y los componentes que utilizamos.
- ❖ **Bottom Layer:** En el programa nos marca como van a estar las pistas de cobre.
- ❖ **Top Layer:** Vendría a ser el contorno de los componentes con sus designaciones.

- ❖ **Multi Layer:** Nos marca los pads, es donde van los huecos para poner los pins de los componentes.
- ❖ **KeP-Out Layer:** Marca el contorno de la plaqueta.
- ❖ **FootPrint:** Tamaño del componente con sus pines.
- ❖ **Pines:** Patas de los componentes.
- ❖ **Net List:** cada pista del Bottom Layer que sea continua, no importa la forma tiene un código o Net List.

1.7.4. Pasos para crear un proyecto de PCB

1. Vamos a la ventana de File.
2. Escogemos la opción new.
3. Y posteriormente cogemos Project
4. Escogemos PCB Project y le damos un nombre al proyecto.
5. Se nos crea el proyecto damos clic izquierdo sobre el proyecto y escogemos la opción add to new Project.
6. Luego le damos en donde dice Schematic, y se nos crea la parte esquemática.
7. Para crear el PCB vamos a File y luego a la opción PCB Board Wizard.
8. Arrastramos a nuestro proyecto y tenemos las ventanas para realizar nuestro PCB.

1.8. MikroC pro for Pic

La programación en el compilador MikroC PRO se basa en PROYECTOS. Cada proyecto está conformado por un conjunto de archivos interrelacionados entre sí. Para crear un programa primero se debe crear un Proyecto (básicamente se debe seleccionar el microcontrolador, la frecuencia de operación y establecer los bits de configuración); a continuación se procede a

escribir las instrucciones del programa en lenguaje C (código fuente); y por último se genera el código de máquina .HEX (código ejecutable) que será almacenado en la memoria de programa del microcontrolador.

Este lenguaje es muy similar al C estándar, no obstante en determinados aspectos difiere del ANSI estándar en algunas características. Algunas de estas diferencias se refieren a las mejoras, destinadas a facilitar la programación de los microcontroladores PIC, mientras que las demás son la consecuencia de la limitación de la arquitectura del hardware de los PIC (Verle, 2016).

Cada una de las partes que se mencionan, son importantes para la ejecución de proyecto, y se pueden aplicar todas las herramientas de programación, como pueden ser instrucciones, interrupciones, subrutinas, etc.

Luego pasar el archivo hexadecimal a nuestro microcontrolador, que es el encargado de guardar y ejecutar el programa en la parte física donde se va aplicar.

1.8.1. Características principales

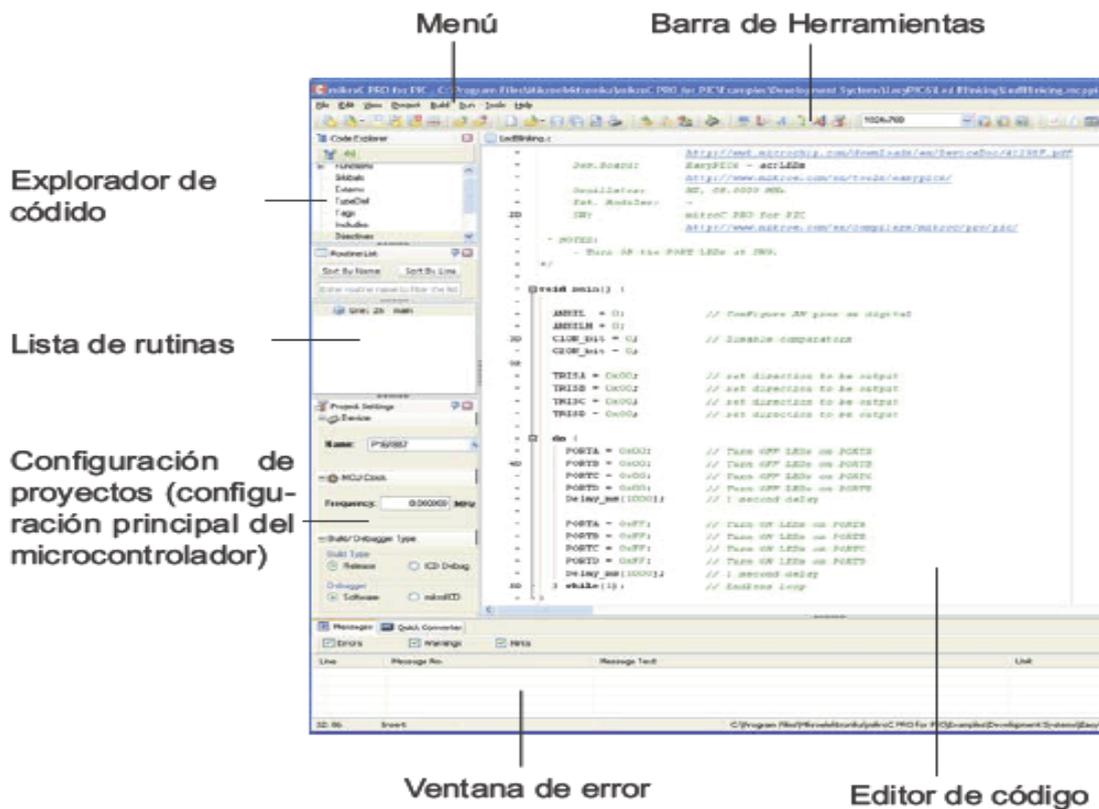


Figura 1-16 Partes de MikroC Pro

Fuente: <http://www.mikroe.com/img/publication/spa/pic-books/programming-in-c/chapter/02/fig2-20.gif>

1.9. Componentes y elementos del sistema

1.9.1. Sensor magnético

Puede ser usado en puertas, ventanas, cajones o placares. Es ampliamente utilizado para la vigilancia y sistemas de seguridad, con una alta fiabilidad.

Esta utilizado un contacto normalmente cerrado el cual para su uso se lo debe realizar un circuito de control y obtener una salida la cual puede ser usada en cualquier microcontrolador para mandar a ejecutar los actuadores.



Figura 1-17 Sensor magnético

Fuente: <http://tanyx.com.ar/images/site/farm1/1359754237.jpg?w=380&h=310>

Características

- Fácil de instalar
- Tamaño pequeño
- Detección de puerta abierta
- Modo de conexión: NC
- Instalar exteriormente

Especificaciones

Distancias de detección: de 15-25 mm

Housing: Plástico ABS blanco

Instalación: Instalar exteriormente

Dimensión: 27x14x8 mm

1.9.2. Optoacoplador moc3021

Son conocidos como optoaisladores o dispositivos de acoplamiento óptico, basan su funcionamiento en el empleo de un haz de radiación luminosa para pasar señales de un circuito a otro sin conexión eléctrica. Estos son muy útiles cuando se utilizan por ejemplo, Microcontroladores PIC si queremos proteger nuestro microcontrolador este dispositivo es una buena opción. En general pueden sustituir los relés ya que tienen una velocidad de conmutación mayor, así como, la ausencia de rebotes.

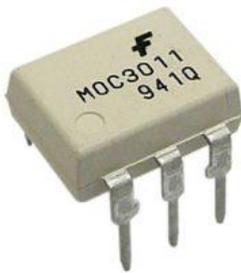


Figura 1-18 Optoacoplador

Fuente: http://mlm-s2-p.mlstatic.com/electronicos-12133-MLM20055690242_022014-Y.jpg

La gran ventaja de un Optoacoplador reside en el aislamiento eléctrico que puede establecerse entre los circuitos de entrada y salida. Fundamentalmente este dispositivo está formado por una fuente emisora de luz, y un fotosensor de silicio, que se adapta a la sensibilidad espectral del emisor luminoso, todos estos elementos se encuentran dentro de un encapsulado que por lo general es del tipo DIP.

Características:

- Salida a fototriac
- Fase aleatoria
- Voltaje de salida pico repetitivo Max: 400 V

- Voltaje de aislamiento: 5000 VRMS
- Rata crítica de crecimiento del voltaje de estado off alta: 100 V/ μ s min.
- Corriente de disparo del LED: 8 mA típico, 15 mA Max.
- Corriente del LED en directo Max: 50 mA
- Voltaje inverso del LED Max: 6 V
- Aprobaciones de seguridad: UL, CSA, FIMKO
- Encapsulado: PDIP de 6 pines

1.9.3. Fusibles

Los fusibles y disyuntores son dispositivos que protegen los circuitos eléctricos contra daños causados por sobrecargas de corriente, que pueden provocar hasta incendios, explosiones y electrocuciones.

El funcionamiento del fusible se basa en el principio según el que una corriente que pasa por un conductor genera calor proporcional al cuadrado de su intensidad. Cuando la corriente alcanza la intensidad máxima tolerable, el calor generado no se disipa con rapidez suficiente, derritiendo un componente e interrumpiendo el circuito.



Figura 1-19 Fusible

Fuente: <http://fisica.laguia2000.com/wp-content/uploads/2011/05/FUSIB1.jpg>

1.9.4. Resistencia eléctrica

Cuando un cuerpo material es sometido a una diferencia de potencial entre dos de sus puntos, se establece una Corriente Eléctrica de una determinada magnitud. La intensidad de esta corriente,

depende de diversos factores, algunos de ellos propios del material en cuestión y otros más bien externos.

La resistencia se define como el cociente entre la diferencia de potencial entre dos puntos de un objeto material y la corriente establecida como consecuencia de esa diferencia de potencial. En términos matemáticos la resistencia es:

$$R = \Delta V / I$$

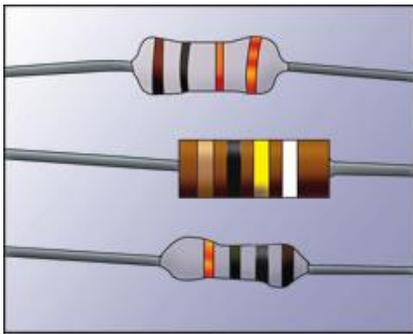


Figura 1-20 Resistencia

Fuente: <http://www.rena.edu.ve/TerceraEtapa/Fisica/Imagenes/RESISTENCIAS.jpg>

1.9.5. Regulador LM7805

El 7805 tienen un patillaje bien recubierto y sólido, donde tiene una entrada de voltaje para reducirlo, el pin numero 1 siempre es la entrada de la tensión a estabilizar, como he dicho antes esta tensión debe ser mayor a los +5V, la patita 2 es el negativo, y por la patita 3 obtendremos la tensión ya estabilizada.

LM7805 PINOUT DIAGRAM

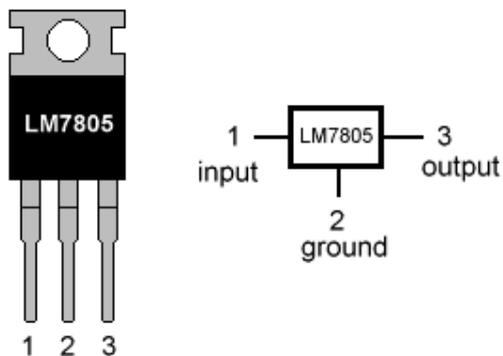


Figura 1-21 LM7805

Fuente: <http://www.puntofotante.net/REGULADOR-DE-VOLTAJE-L7805.jpg>

1.9.6. Regulador LM317

El LM317 es un regulador de tensión positivo con sólo 3 terminales y con un rango de tensiones de salida desde los 1.25 hasta 37 voltios.

Las patillas son: Entrada (IN), Salida (OUT), Ajuste (ADJ).

Para lograr esta variación de tensión sólo se necesita de 2 resistencias externas (una de ellas es una resistencia variable).

Entre sus principales características se encuentra la limitación de corriente y la protección térmica contra sobrecargas.

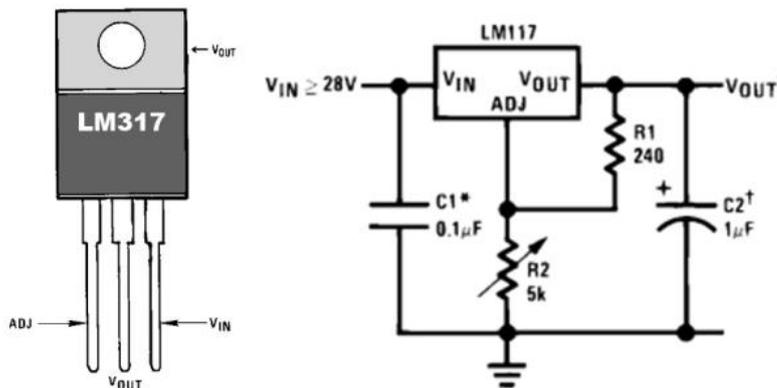


Figura 1-22 LM317

Fuente: <http://electronilab.co/wp-content/uploads/2013/03/LM317-y-esquema.png>

La tensión entre la patilla ADJ y OUT es siempre de 1.25 voltios (tensión establecida internamente por el regulador) y en consecuencia la corriente que circula por la resistencia R1 es: $IR1 = V / R1 = 1.25/R1$.

Con el propósito de optimizar la regulación el resistor R1 se debe colocar lo más cercano posible al regulador, mientras que el terminal que se conecta a tierra del resistor R2 debe estar lo más cercano posible a la conexión de tierra de la carga.

1.9.7. Sensor de corriente Acs714

Este sensor de corriente es un tablero de soporte o ruptura de placa ACS714LLCTR 30A-Allegro-T de efecto Hall con sede actual del sensor lineal, por lo tanto, el sensor funciona a 5 V y tiene una sensibilidad de la salida de 66 mV / A.

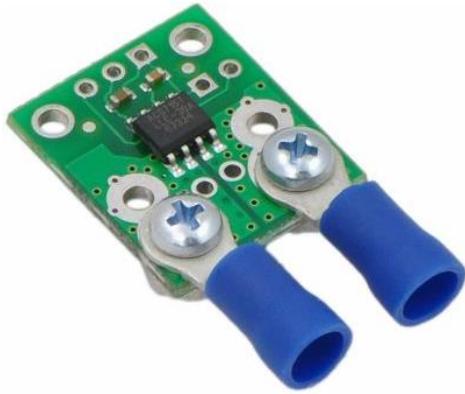


Figura 1-23 Sensor de corriente

Fuente: https://mlv-s2-p.mlstatic.com/925801-MLV20402244701_092015-C.jpg

Características

La lista siguiente detalla algunas de las características clave del sensor:

- Diseñado para la entrada de corriente bidireccional de -30 a 30 A (aunque el sensor robusto IC puede sobrevivir hasta cinco veces la condición de sobrecorriente).
- El uso de un sensor de efecto Hall: el CI es capaz de aislar eléctricamente la trayectoria actual de la electrónica del sensor (hasta 2.1 kV RMS), que permite que el sensor se inserta en cualquier lugar a lo largo de la ruta actual y que deben utilizarse en aplicaciones que requieren eléctrica aislamiento.
- El sensor requiere una tensión de alimentación de 4,5 a 5,5 V para ser conectado a través de las almohadillas de Vcc y GND, que están marcadas en la parte inferior de serigrafía. El sensor hace salir una señal analógica que es linealmente proporcional a la corriente de entrada. Esta tensión de salida se centra en el 2,5 V y los cambios en un 66 mV por amperio de corriente de entrada, con la corriente positiva el aumento de la tensión de salida y la corriente negativa la disminución de la tensión de salida.
- La corriente de entrada se puede conectar a la tarjeta en una variedad de maneras. Para aplicaciones de baja corriente, puede soldar 0.1 "pasadores encabezado macho a la tarjeta a través de la pequeña a través de agujeros en el lado de entrada de corriente de la placa. Para aplicaciones de alta corriente, se puede soldar los cables directamente a los agujeros cuyos tamaños mejor se adapten a los cables, o puede usar conectores sin soldadura terminal de anillo

1.9.8. Triac

Es un dispositivo semiconductor que pertenece a la familia de los dispositivos de control: los tiristores. El triac es en esencia la conexión de dos tiristores en paralelo pero conectados en sentido opuesto y compartiendo la misma compuerta.

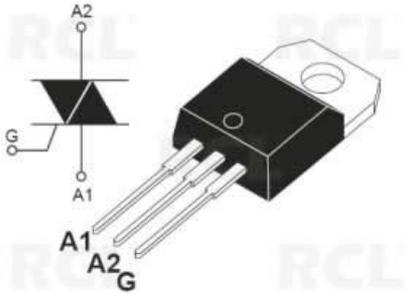


Figura 1-24 Triac

Fuente: http://yourduino.com/sunshop2/images/products/large_330_BT136-1.jpg

A1: Ánodo 1, A2: Ánodo 2, G: Compuerta.

El triac sólo se utiliza en corriente alterna y al igual que el tiristor, se dispara por la compuerta. Como el triac funciona en corriente alterna, habrá una parte de la onda que será positiva y otra negativa.

Características:

Corriente total RMS max: 8 A

Corriente pico max: 65 A

Voltaje pico repetitivo en estado apagado max: 600 V

Corriente pico de compuerta max: 2 A

Operación en cuatro cuadrantes

Alta sensibilidad de compuerta, puede ser conectado directamente a μ C's o circuitos lógicos

Voltaje de disparo de compuerta: $\leq 1.5V$

Voltaje de disparo de compuerta: 2.5 mA típico

Encapsulado: TO-220AB

Aplicaciones:

Control de motores y calefactores

Control AC por fase

Control de luminosidad (Dimmer)

Interruptor de estado sólido

1.9.9. Mosfet

Es un transistor de potencia que soporta altas corrientes de acuerdo a las necesidades del usuario a diferencia de los transistores normales es que estos son activados mediante y no por corriente, aquí se exponen algunas características que son de mucha utilidad para el diseño de un circuito.

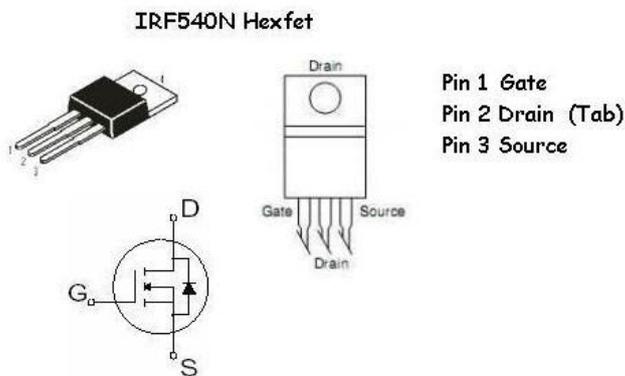


Figura 1-25 Transistor de potencia

Fuente: http://www.electrocave.com/_/rsrc/1275636574185/home/projects/irf540n-hexfet-info/montage2.JPG

Características

Polaridad del transistor: Canal N

Drenaje continúa Id actual: 33A

Fuente de drenaje Voltaje Vds: 100V

En Resistencia RDS (on): 44mohm

RDS (on) Tensión de prueba Vgs: 10V

Tensión umbral Vgs Typ: 4V

Transistor Case Style: TO-220AB

N.º de Pins: 3

Corriente, máx ID .: 33A

Temperatura actual: 25 ° C

Marcador: IRF540N

Completo Temperatura Potencia nominal: 25 ° C

Unión a encapsulado Resistencia térmica R: 1,1 ° C / W

Espaciado de conductor: 2,54 mm

Disipación de energía Pd: 94W

Disipación de energía Pd: 94W

Pulso Idm actual: 110A

Tensión Vds Typ: 100V

Tensión Vgs Max: 4V

Tensión Vgs RDS Medición: 10V

1.9.10. Trimpot

Potenciómetro (Trimpot) vertical sin switch de 1/2 Watt. Los potenciómetros de ajuste (Trim o Trimpot), son potenciómetros que se ajustan de una vez por todas, con la ayuda de un destornillador. Sirven para que el circuito al que pertenecen, actúe adecuadamente (en su punto) y compensar las tolerancias en otros componentes.



Figura 1-26 Trimpot

Fuente: <http://www.bluetip.com.uy/image/cache/data/Componentes/Potenci%C3%B3metros/3386P%20SERIES-500x500.jpg>

Especificaciones

- Tipo: Trimpot
- Número de pines: 3
- Dimensiones de la base: 0.9 cm x 0.4 cm x 0.9 cm
- Separación de los pines: 0.2 cm
- 1 Vuelta
- Cuadrado
- Vertical
- Modelo: 3386C-1

1.9.11. Cámara Wireless

Características:

- ❖ Sensor CMOS 1/4 de alta sensibilidad
- ❖ Rotación e inclinación por control de software y visión nocturna por infrarrojos
- ❖ Frecuencia de video: 30fps (QVGA), 15fps (VGA)
- ❖ Resolución 640 x 480 (VGA), 320 x 240 (QVGA)
- ❖ Frecuencia de luz 50HZ, 60HZ o luz exterior
- ❖ Compresión de video MJPEG optimizado
- ❖ Admite la gestión de múltiples usuarios y definición de contraseñas
- ❖ Servidor Web integrado para monitorizar desde Internet Explorer
- ❖ Alerta en caso de detección de movimiento
- ❖ Permite captura de imágenes
- ❖ Soporta Múltiples protocolos de red; HTTP/TCP/IP/UDP/STMP/DDNS/SNTP/DHCP/FTP



Figura 1-27 Cámara Wireless

Fuente: http://www.sure.com.bo/296-thickbox_default/camara-ip-wireless-omniview-vision-nocturna-.jpg

1.9.12. Motor 12 V DC

Características

Corriente Continua (A): 8^a

Uso: Coche

Potencia de salida: 20W

Características protegidas: Antigoteo

Tipo: Motor reductor

Construcción: Imán Permanente

Torsión: 3N.m

Velocidad (RMP): 95

Voltaje (V): 12/24V



Figura 1-28 Motor DC

Fuente: <http://g02.a.alicdn.com/kf/HTB1VZpNIVXXXXczaXXXq6xXFXXxi/1PCS-DC-12V-67rpm-Gear-font-b-Motor-b-font-Automotive-font-b-Glass-b-font.jpg>

1.9.13. Tira de diodos Led 12V

Modelo RGB que incluye los tres colores (ROJO + VERDE + AZUL). El modelo RGB tiene un cable para cada color, y es capaz de tener un total de 256 colores distintos. Alimentación de 12V.



Figura 1-29 Tira led RGB

Fuente: http://tienda.bricogeek.com/2219-large_default/tira-de-led-rgb-indexable-2m-60-leds-m-ws2812b.jpg

Tipos de led

SMD3528: luminosidad media

SMD5050: luminosidad alta

Tipo de encapsulado

IP65: Resistente al agua (pero no sumergible). Película fina de resina para protegerle.

IP68: Muy resistente al agua. Película gorda de resina. La resina ocupa cierto espacio, por lo que este modelo está recomendado para exteriores.

CAPÍTULO II

2. MARCO APLICATIVO

2.1. Diseño del sistema domótico

En el siguiente esquema podemos ver cuáles son las distintas etapas que va seguir nuestro sistema domótico el mismo que controlará las distintas acciones que se mencionan a continuación.

Nos será de gran ayuda para guiarnos en el desarrollo del proyecto siguiendo cada etapa en el orden indicado, y a su vez nos ayudará a comprender el funcionamiento del sistema.

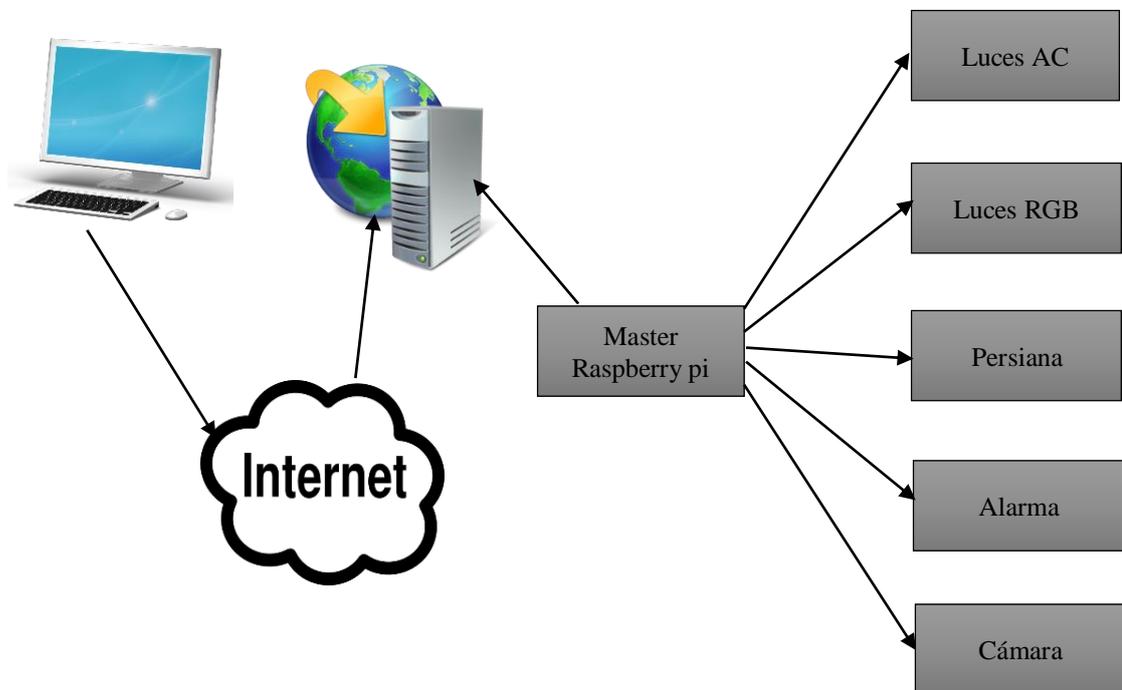


Figura 2-1 Diseño del sistema

Fuente: Lema, Edison, 2016.

2.1.1. Estructura del sistema

El sistema será instalado en una vivienda, en primera instancia será probado en una mesa donde se conectarán todas las placas y analizar su funcionamiento, para luego instalar en la vivienda y controlar las acciones que se mencionaron anteriormente.

La comunicación entre las placas se lleva a cabo mediante el bus Rj45 los mismos que transportaran los datos de la comunicación Modbus entre el master y los distintos esclavos, a su

vez también viaja la alimentación para las placas, tanto voltaje 12 DC y también 5 DC, y la conexión a tierra de todo el sistema.

2.1.2. Etapa de control

El sistema va ser controlado desde la Web por lo que se necesita de un computador para conectar a la Aplicación Web que se encuentra en la Raspberry la misma que está colocada en la Placa master y desde aquí mediante comunicación serial hacia los modulos esclavos que forman el sistema para lograr la comunicación Modbus en los microcontroladores insertados en las placas.

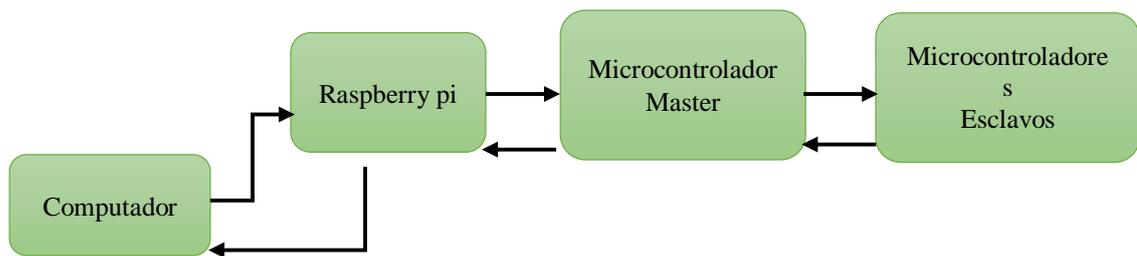


Figura 2-2 Diagrama de bloques del sistema

Fuente: Lema, Edison, 2016.

2.1.3. Etapa de adquisición de datos

Los datos se van adquirir en la Raspberry pi para ellos se diseña una placa conjuntamente con un microcontrolador y estos datos son enviados a todas las placas esclavo para su proceso, a su vez hay entradas de sensores y pulsantes que son receptadas por las placas consideradas como esclavos del sistema.

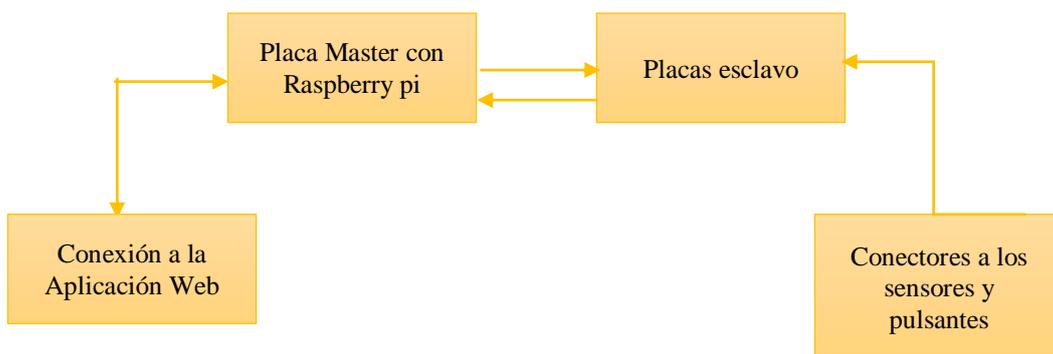


Figura 2-3 Diagrama de bloques de la adquisición de datos

Fuente: Lema, Edison, 2016.

2.1.4. Etapa de potencia

Como en el sistema vamos a tener un manejo de corriente considerable es preferible proteger nuestros equipos, mediante el acoplamiento de la señal hacia el circuito de potencia, además hay que tomar en cuenta que se van a usar actuadores de alta potencia y es necesario proteger las placas que se van a utilizar, y así mismo tener que acoplar la señal para poder utilizar en la salida final del sistema.

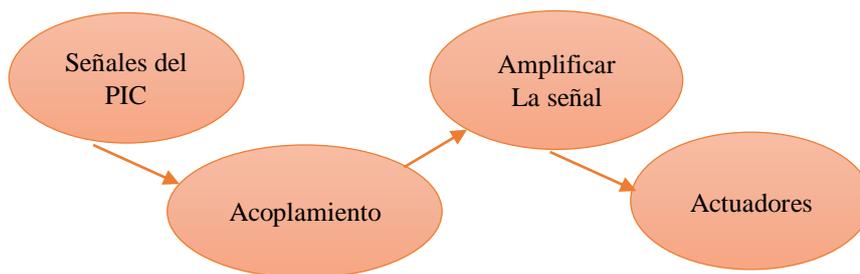


Figura 2-4 Diagrama de bloques de la etapa de potencia

Fuente: Lema, Edison, 2016.

2.2. Etapa de alimentación del sistema

En la etapa de alimentación del sistema, está diseñada con una fuente de 12 V con un amperaje de 15 Am, además se necesita una fuente de 5 V con un circuito regulador de voltaje para alimentar el microcontrolador, ya que de acuerdo a las características del PIC se alimenta con este valor de voltaje.

2.3. Método de programación del microcontrolador

2.3.1. Programación por interrupción

Es favorable trabajar con este tipo de programación debido a que detiene la ejecución del programa, para realizar operaciones que están previamente programadas y luego continuar con el proceso que se estuvo ejecutando, las mismas que deben estar escritas dentro de una función que en MikroC for PIC, la misma que tiene la siguiente sintaxis:

```
Void interrupt () {  
    Cnt++; // Al producirse una interrupción la cnt se incrementa en 1  
    PIR1.TMR1IF = 0; // Poner a 0 el bit TMR1IF }  
}
```

Se debe también tomar en cuenta las prioridades de las interrupciones al momento de programar debido a que el bit más significativo tiene la prioridad más alta dependiendo del programa que estemos ejecutando.

2.3.2. Programación Uart

Se ha utilizado la programación serial debido a que tiene una respuesta más rápida en la ejecución de los datos puesto que viajan por cable UTP y conectores RJ45, tanto para recibir como para enviar, para esto se utiliza los pines del microcontrolador que son Rx, Tx, y en primera instancia se debe definir cuál es el PIC considerado como master, y cuáles son los esclavos que se han designado en el sistema, tomando en cuenta el estándar de direcciones Modbus que se pueden utilizar. En este caso la velocidad que se utilizó es de 9600 BPS, con osciladores de 16 MHz en las placas.

2.4. Interfaz de control con la aplicación web

Para interactuar con el sistema domótico, lo que se debe utilizar es una computadora o a su vez un teléfono inteligente, con el cual nos podemos conectar a la aplicación Web, la misma que consta de una cuenta que contiene nombre de usuario y password para autenticarse y además consta de un menú de opciones que muestra las acciones que se pueden ejecutar dentro de la vivienda donde se instaló el sistema.

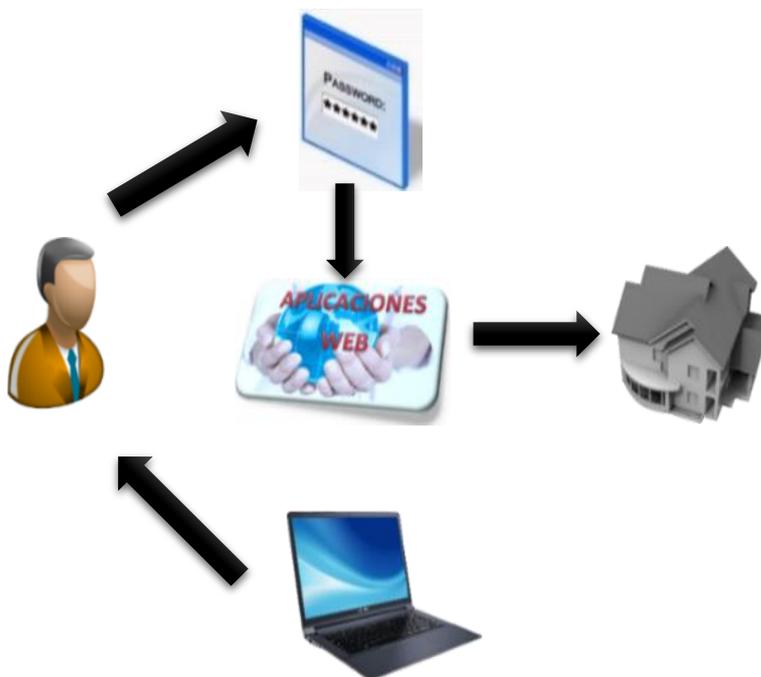


Figura 2-5 Interfaz de control con la aplicación Web

Fuente: Lema, Edison, 2016.

2.5. Plano domótico de la instalación

Tabla 2-1 Identificación del plano domótico

PLANO DOMÓTICO	
Caja de distribución	
Caja domótica	
Fase	
Neutro	
Leds RGB	
Luces AC	
Persiana	
Alarma	
Cámara	
Pulsantes	

Realizado por: Edison Lema

Fuente: Lema, Edison, 2016.

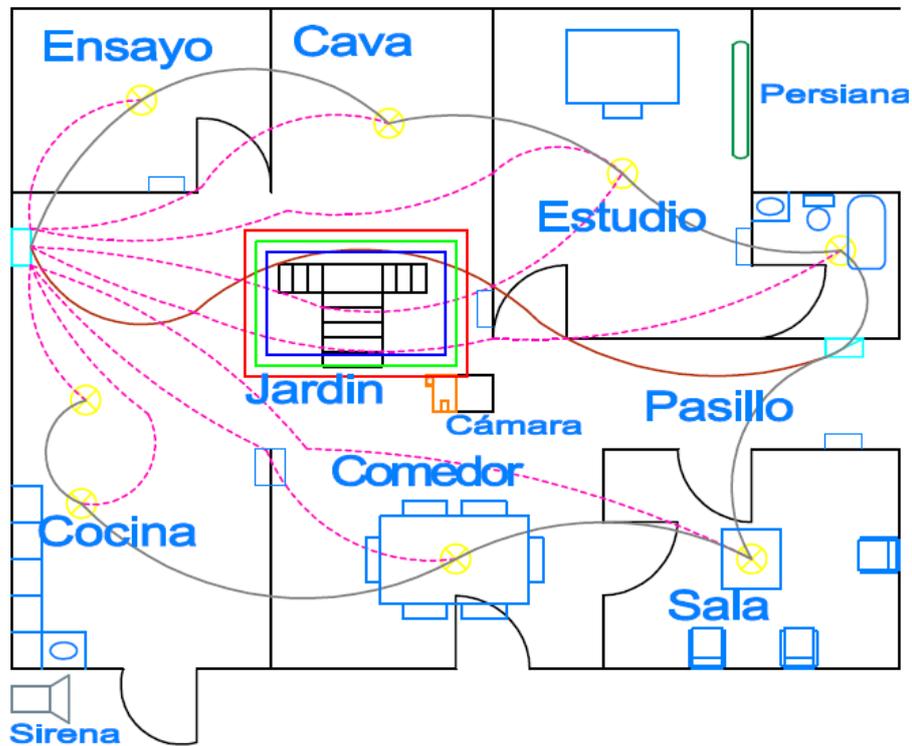


Figura 2-6 Plano domótico

Fuente: Lema, Edison, 2016.

El plano de domótica sirve para el cableado centralizado en la vivienda, para ello cada uno de los puntos importantes a llevar a cabo este proceso se muestra en la figura 2-6 donde podemos observar el número de luces a ser controladas desde la Web, de igual manera están identificados las distintas fases a ser conmutadas por las placas Ac, colocadas en la caja domótica, las luces RGB se encuentran en el espacio de un jardín interno. La persiana está ubicada en el espacio del estudio junto a una luz que tiene la característica de ser dimmer, la siguiente etapa del cableado es la de la alarma que está ubicado a un costado de la cocina en la parte externa de la vivienda, los sensores magnéticos están colocados en las puertas y ventanas que se encuentran cableadas a la placa de la alarma como una entrada digital que lee el microcontrolador.

La alimentación principal del sistema está dado por la caja de distribución principal de 120 V AC, hacia la caja de domótica. Cabe indicar que todo el cableado está realizado con cable UTP categoría 5 y a su vez las líneas de electricidad de las luces AC se utiliza cable # 16 AWG flexible, de igual manera para los pulsantes que accionan los dispositivos.

En la parte central de la estructura tenemos instalado una cámara wireless la misma que tiene un giro de 270 grados con la cual se puede observar ciertas áreas de la vivienda y observar los cambios que se presentan. Debido a las características del dispositivo se pueden grabar ciertas posiciones predefinidas y almacenar en la aplicación web para luego visualizarlas.

2.6. Construcción de las placas

2.6.1. Partes de control de las placas

La etapa de control está diseñada con un PIC18F2550 en todas las placas, es una parte importante del sistema porque aquí se encuentra programado las acciones que debe realizar cada módulo. El micro controlador recibe las datos que se envían desde la aplicación Web a través de una interacción entre la Raspberry pi mediante la comunicación Modbus.

Se diseñó el circuito adecuado para implementar nuestro PCB, está formado por un pulsante que sirve para resetear la placa, además dispone de un oscilador de 16 Mhz.

El microcontrolador es también quien recibe las señales de los sensores y pulsantes en los módulos de domótica para poder encender una luz o accionar algún dispositivo. Es por ello que el sistema está diseñado para ser controlado desde la Web y también desde el interior de la vivienda mediante los pulsantes.

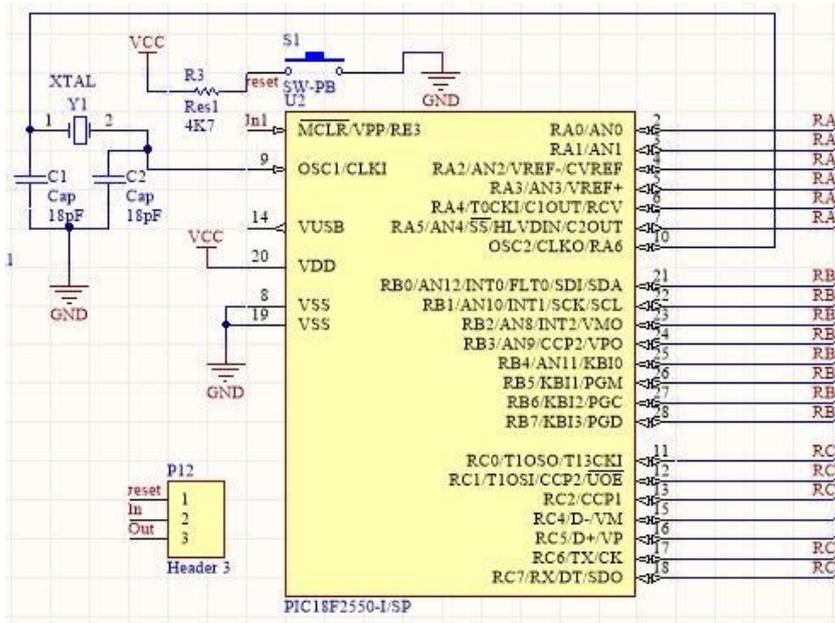


Figura 2-7 Etapa comunicación de la placa master

Fuente: Lema, Edison, 2016.

2.6.2. Etapa de comunicación

En esta etapa es donde se encuentra el integrado Max485 que realiza la comunicación serial Modbus a través de los pines A y B por donde viajan los datos desde la placa central hacia los modulos esclavos. El sistema está diseñado para que se comunique mediante conectores RJ45 como se mostrará en la figura 2-11. Este integrado es alimentado con 5 voltios los cuales vienen del regulador de voltaje, se debe tener cuidado con este integrado debido a que es muy sensible a sobre voltajes y puede provocar errores en la comunicación impidiendo el control y manejo del sistema.

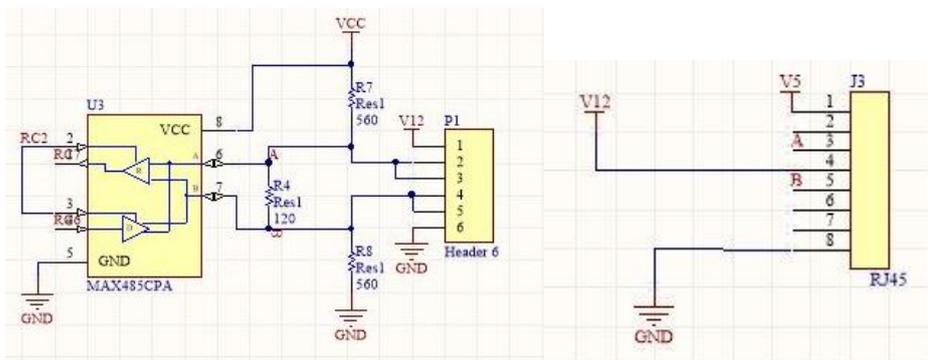


Figura 2-8 Etapa comunicación de la placa master

Fuente: Lema, Edison, 2016.

2.6.3. Placa master

La placa master es aquella que se encarga de la transmisión de los datos los mismos que son receptados desde la aplicación Web, mediante la comunicación Modbus. Estos datos viajan constantemente desde la placa central hacia los esclavos, y a la placa Raspberry pi que envía los datos al PIC18F2550 que se encarga de interpretar los datos recibidos y luego enviar a los distintos módulos del sistema.

2.6.3.1. Etapa de la placa Raspberry pi

En esta etapa es donde va insertada la placa Raspberry pi a la placa master para que envíe los datos hacia el microcontrolador. Cada placa esclavo tiene su dirección Modbus para realizar la comunicación, esta etapa es importante debido a que funcionan con el cerebro del sistema que tiene la salida hacia el internet y a su vez es la entrada hacia la aplicación Web para el control del sistema.

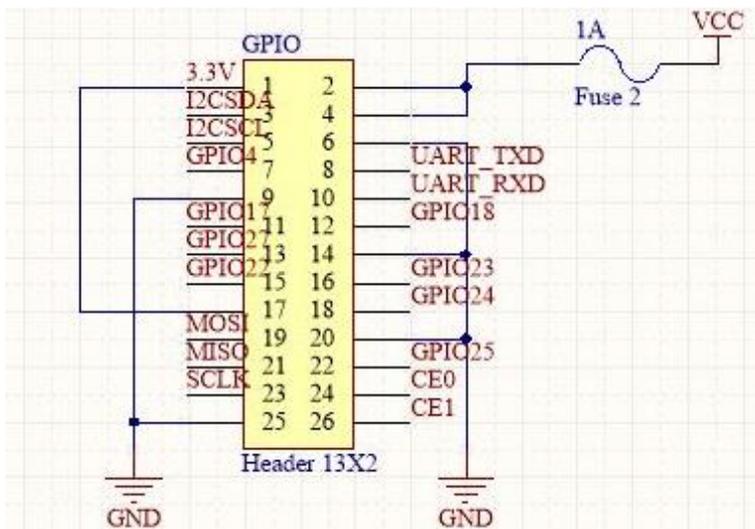


Figura 2-9 Etapa de la placa Raspberry pi

Fuente: Lema, Edison, 2016.

Los pines de la placa Raspberry pi envían señales al microcontrolador por medio de los pines GPIO que sirven para la transmisión y recepción de datos para el control y gestión de los procesos en el sistema. Antes de que la señal llegue al PIC esta debe ser amplificada por medio de un circuito amplificador para mejorar la señal del sistema para que el microcontrolador reciba una señal TTL.

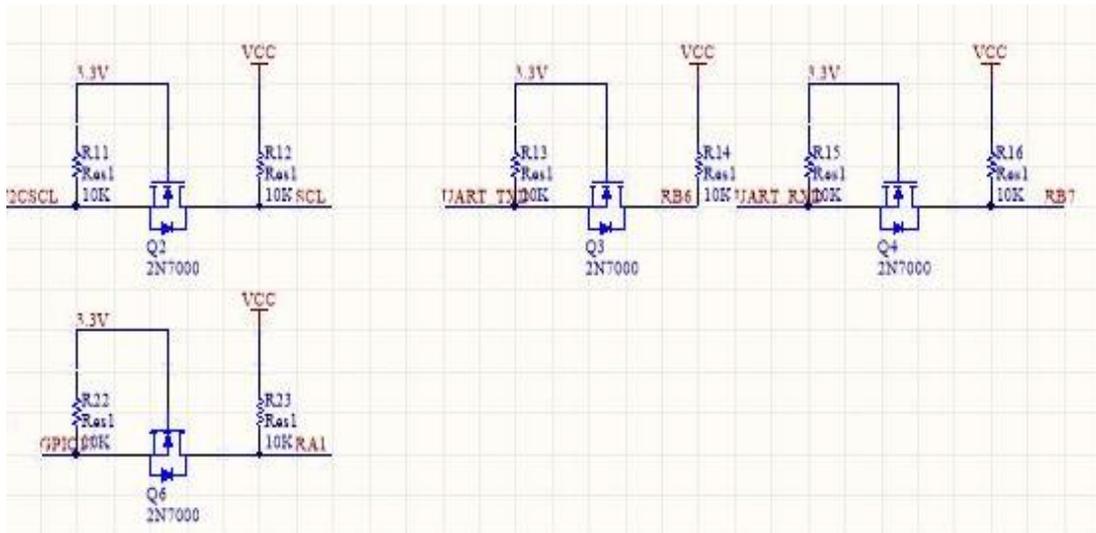


Figura 2-10 Pines de conexión de la placa Raspberry pi

Fuente: Lema, Edison, 2016.

2.6.3.2. Etapa de alimentación

La entrada de alimentación es de 12 voltios los cuales provienen de una fuente externa que alimenta a todo el sistema, este voltaje es procesado y reducido por un regulador de voltaje como el LM317 a 5 voltios que sirve para alimentar al microcontrolador de la placa master, y a sus distintos componentes que forman la misma.

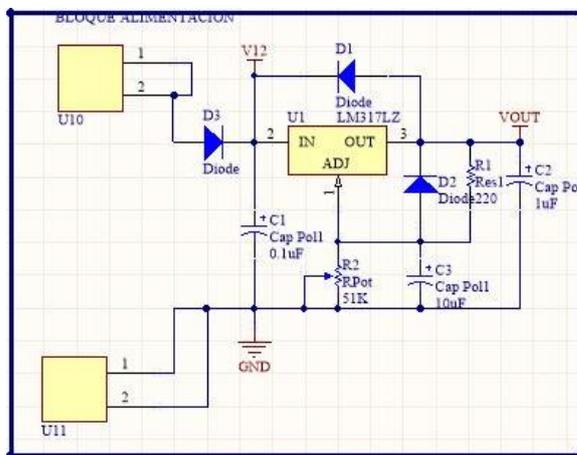


Figura 2-11 Etapa de alimentación de la placa master

Fuente: Lema, Edison, 2016.

2.6.3.3. PCB master

Para el diseño de PCB se procedió a pasar del circuito esquemático al circuito PCB con todos sus elementos electrónicos propios del sistema.

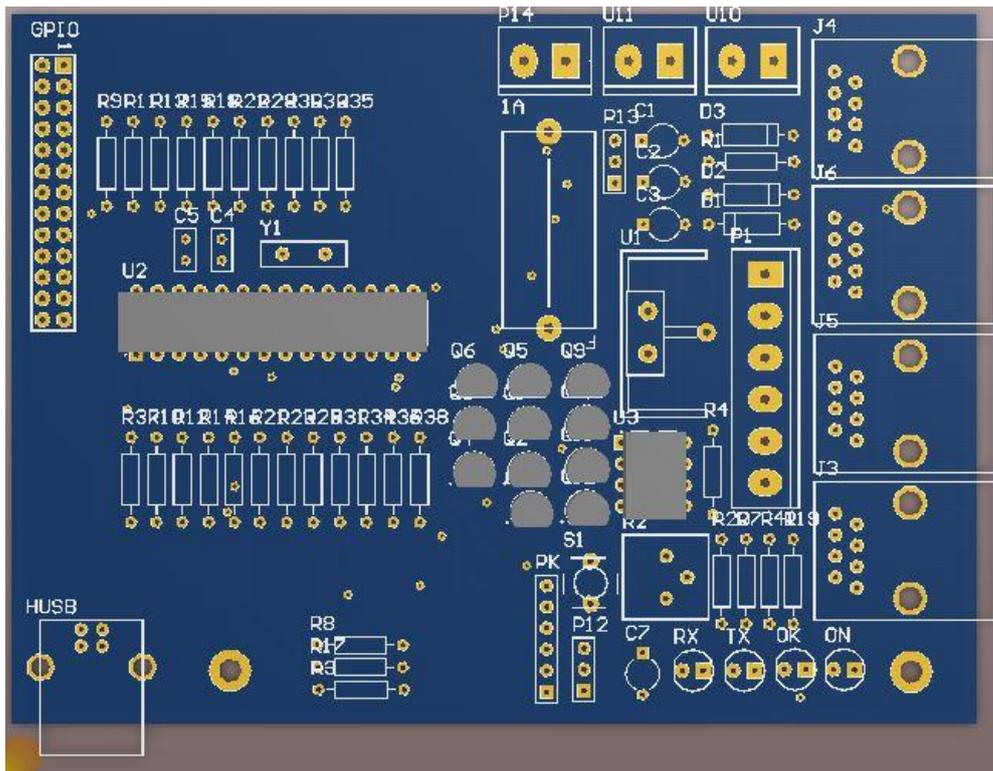


Figura 2-12 PCB de la placa master

Fuente: Lema, Edison, 2016.



Figura 2-13 PCB de la placa master implementado

Fuente: Lema, Edison, 2016.

2.6.4. Placa de luces AC

La placa de luces AC está diseñada con la finalidad de que realice el encendido de una luz dimmer y ON/OFF, la cual varía su luminosidad de acuerdo a las necesidades del usuario. La placa es el primer esclavo del sistema que está en constante comunicación con la placa master. La placa central envía los datos a la placa AC para que se activen los actuadores.

A continuación vamos a detallar cada una de partes de las que está compuesta nuestra placa Ac.

2.6.4.1. Etapa de cruce por cero

Está diseñado con un integrado operacional LM358 que consta de un transformador que está conectado a la línea de 120 Ac seguido de ello un diodo rectificador para mantener la señal en un solo sentido, con una configuración de resistencias en serie tenemos la primera entrada del operacional, luego se configuró la segunda entrada del OPAM mediante un divisor de voltaje y resistencias con una fuente de 5 V DC.

Lo que hace el operacional es comparar y enviar una señal cada vez que la línea de 120 pasa por 0, la misma que es enviada al microcontrolador para su procesamiento. A continuación se muestra el circuito del cruce por cero.

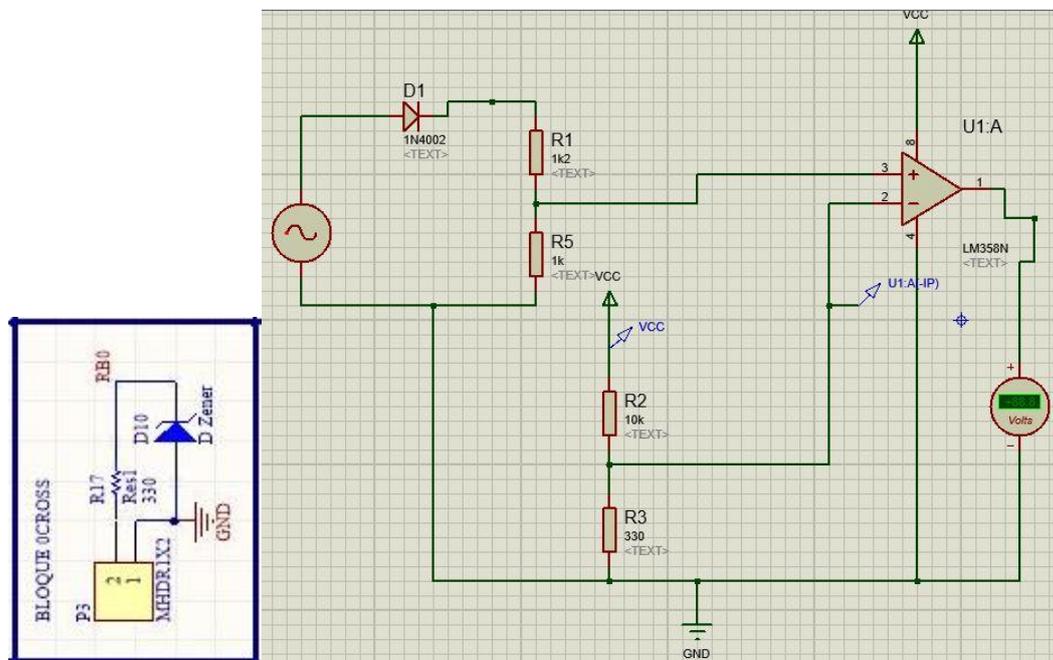


Figura 2-14 Circuito cruce por cero y entrada al microcontrolador

Fuente: Lema, Edison, 2016.

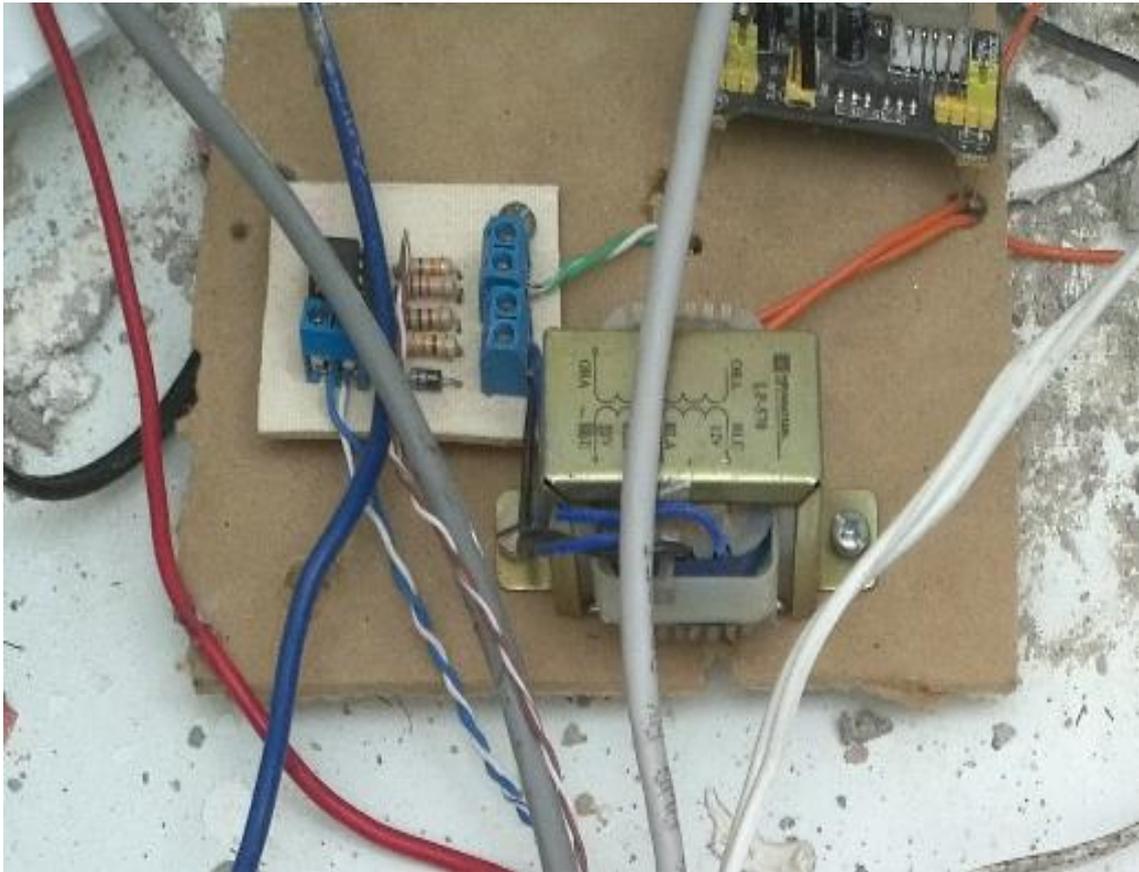


Figura 2-15 Circuito cruce por cero implementado

Fuente: Lema, Edison, 2016.

2.6.4.2. Etapa de potencia

La parte más importante del diseño es la separación de la etapa de control y de la etapa de potencia, esto se ha logrado mediante un aislamiento, utilizando un relé de estado sólido conocido como Optoacoplador como el MOC3021, cuyas características permiten aislar circuitos dando seguridad a nuestros equipos y protegiéndolos.

Por un lado se tiene la parte de control que envía al microcontrolador una señal para activar el Optoacoplador, y por otro lado tenemos la parte de potencia que es la más sobresaliente y está diseñado con Triacs que son transistores de potencia y por sus características permiten el control de cargas Ac de variedad de corrientes, también soportan activar una luz dimmer Ac es por ello que hemos escogido para nuestro circuito. Dispone de una protección adicional que consiste en un fusible conectado en serie para nuestro caso un foco.

En nuestro diseño se han insertado entradas de pulsantes para encender los focos de la vivienda en un número de 4, pueden también adicionarse algunos al sistema por ser modular es decir si se necesitará una luz más se podría hacer una placa y dar una nueva dirección Modbus a la misma.

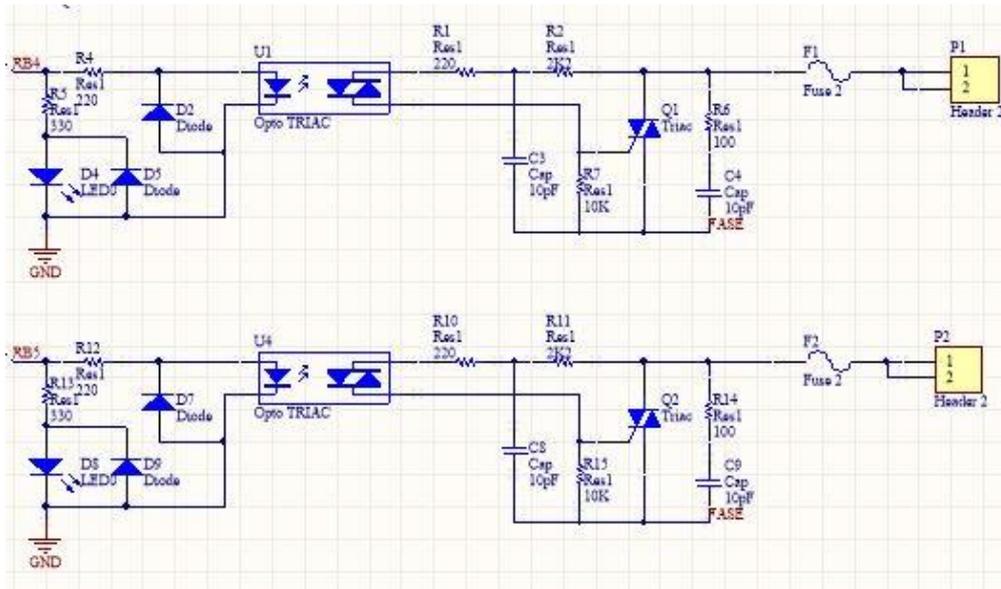


Figura 2-16 Etapa de potencia de la placa de luces AC

Fuente: Lema, Edison, 2016.

2.6.4.3. PCB de la placa de Luces AC

Del diseño esquemático se pasó al diseño PCB colocando todos los elementos electrónicos en orden según el esquema del diseño el cual se muestra en la figura 2-17.

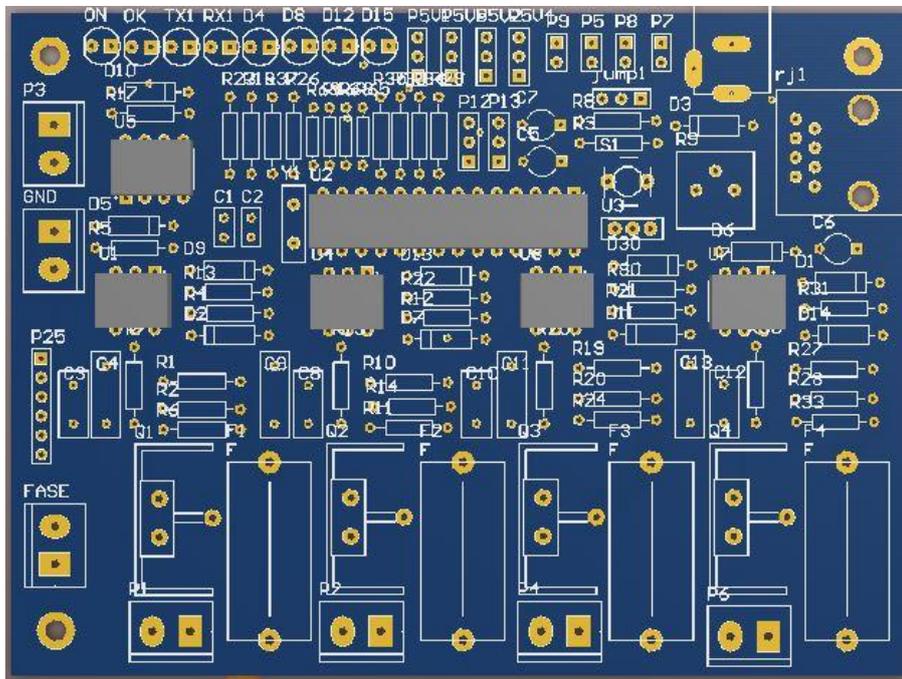


Figura 2-17 PCB de la placa de luces Ac

Fuente: Lema, Edison, 2016.

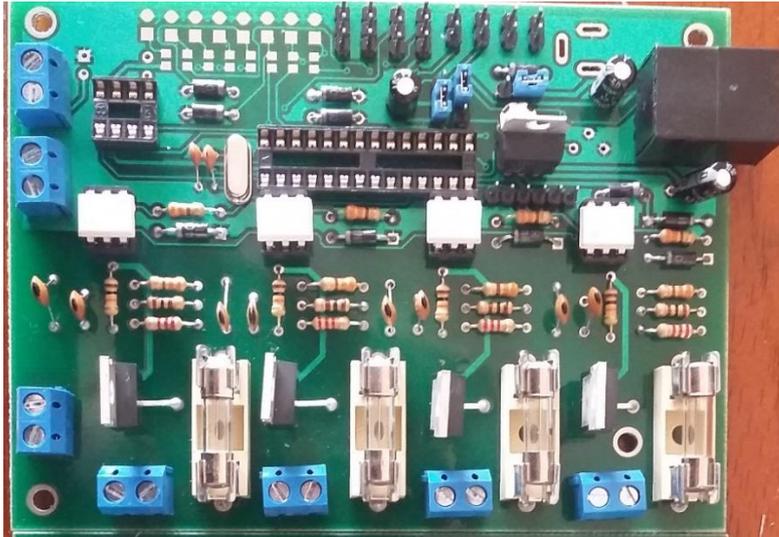


Figura 2-18 PCB de la placa de luces AC implementado

Fuente: Lema, Edison, 2016.

2.6.5. Placa de luces RGB

La placa está diseñada con el objetivo de activar luces RGB es decir para la combinación de colores, lo que hace es recibir los datos que proporciona la aplicación Web, para que la placa Master analice y redireccione a la placa RBG de acuerdo a la dirección MODBUS. Las luces trabajan en una combinación de colores (señales PWM) desde el PIC hacia los actuadores (luces RGB) con una alimentación de 12 V. a continuación se describen cada una de las fases que contiene esta placa.

2.6.5.1. Etapa de alimentación

La etapa de alimentación de la placa de conmutación RGB, es alimentado por el integrado LM7805 a 5 V y éste alimenta a los microcontroladores que trabajan en un voltaje estándar. En esta etapa el integrado recibe un voltaje de 12 V que proviene de la placa master mediante el puerto RJ45 hacia la placa RGB.

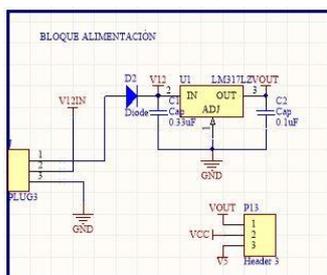


Figura 2-19 Etapa de control de la placa de luces RGB

Fuente: Lema, Edison, 2016.

2.6.5.2. Etapa de entradas de pulsantes

En esta etapa fue necesario adicionar entradas de pulsantes, para combinar los colores de las luces RGB en el interior de la vivienda donde se encuentran instalados. Tiene un circuito que acciona una señal al microcontrolador para activar los puertos de control: RA0, RA1, RA2, los cuales son respectivamente R, G, B.

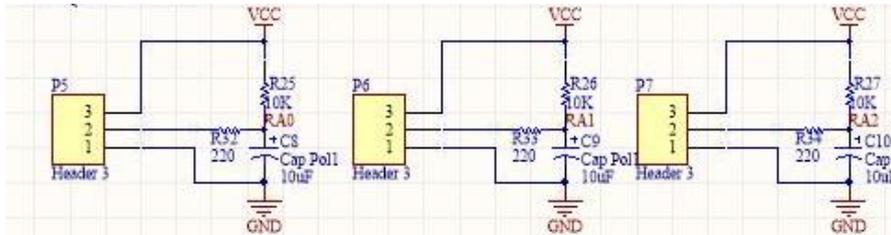


Figura 2-20 Entradas de pulsantes

Fuente: Lema, Edison, 2016.

2.6.5.3. Etapa de potencia

Dispone de dos fuentes distintas de alimentación, debido a que los actuadores manejan voltajes y corrientes diferentes, una para la parte de control y otra para alimentar la tira de diodos RGB que trabajan a 12 V DC como se muestra en la Figura 2-20.

En esta etapa se utilizó transistores de potencia como el IRF540N, con el objetivo de que la señal PWM que es enviada desde el PIC18F2550 al transistor 2n3904 y mediante un divisor de voltaje para activar el Mosfet. El Mosfet deja pasar la señal PWM con su voltaje 12 V DC a las distintas salidas de la Placa para conectar las luces RBG.

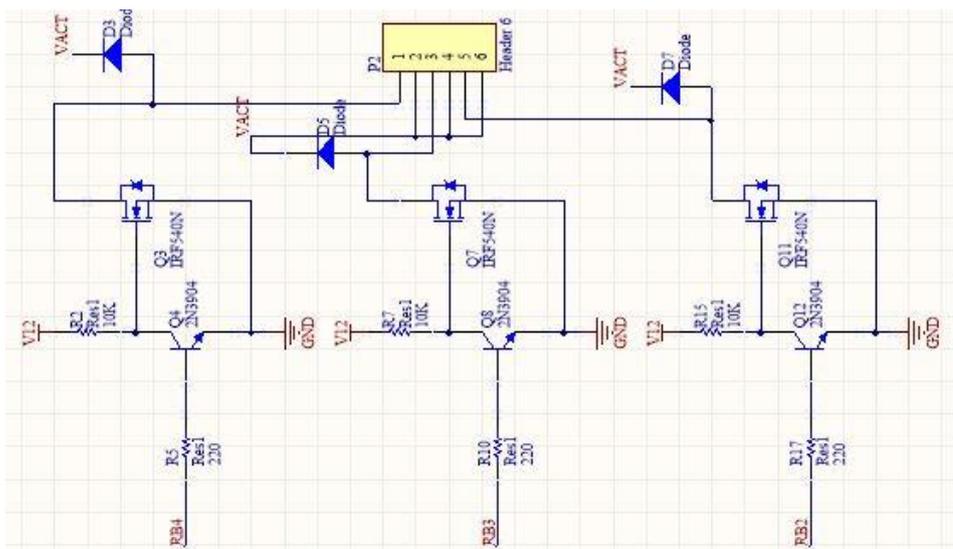


Figura 2-21 Circuito de potencia de la placa de luces RGB

Fuente: Lema, Edison, 2016.

2.6.5.4. PCB placa de luces RGB

Se diseñó el circuito PCB en base al diagrama esquemático, con todos los elementos de electrónica que contiene la placa, para su correcto funcionamiento del sistema.

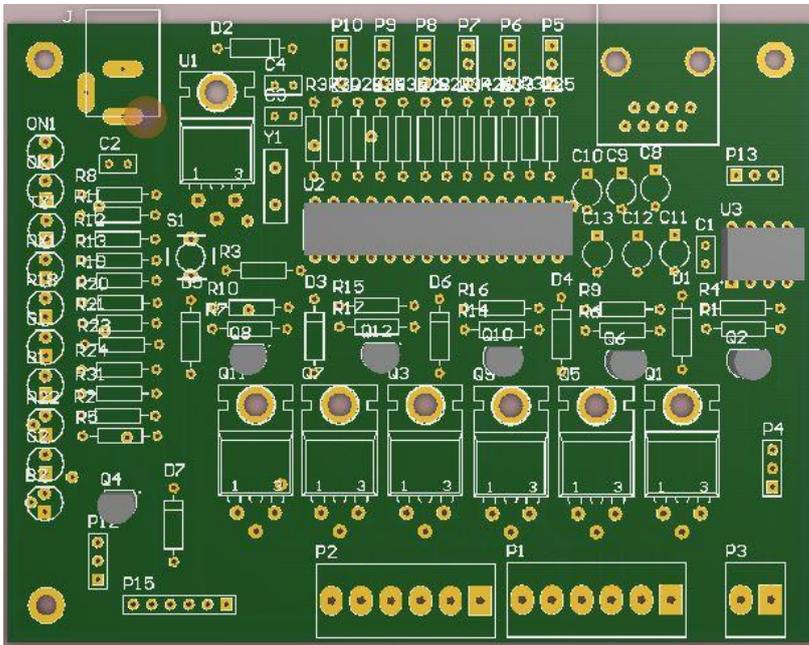


Figura 2-22 PCB de la placa de luces RGB

Fuente: Lema, Edison, 2016.

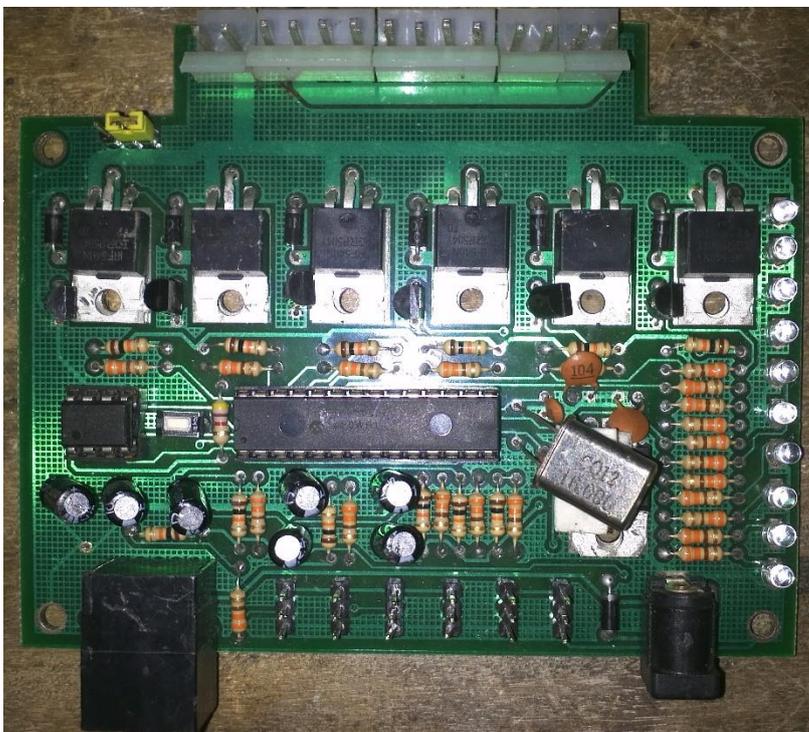


Figura 2-23 PCB de la placa de luces RGB implementado

Fuente: Lema, Edison, 2016.



Figura 2-24 Luces RGB colocadas en el jardín

Fuente: Lema, Edison, 2016.

2.6.6. Placa del sistema de alarma

Esta placa está diseñada para que trabaje como medida de seguridad del sistema de domótica, cuenta con sensores magnéticos que han sido instalados en las puertas y ventanas de la vivienda. Desde la aplicación Web se envían señales a la placa de alarma y activar el actuador que es una sirena que funciona a 12 V. La comunicación de esta placa con la placa master se lo realiza mediante el protocolo Modbus que dispone de una dirección que la define esclavo, en esta placa tenemos algunas etapas que se describen a continuación.

2.6.6.1. Etapa de sensores magnéticos

El siguiente esquema muestra la etapa de entradas de señales de los sensores magnéticos que utiliza el sistema. Las señales de entradas son enviadas a la etapa de control para ser procesadas por el microcontrolador para activar la sirena a través de los pines RB0 hasta RB5.

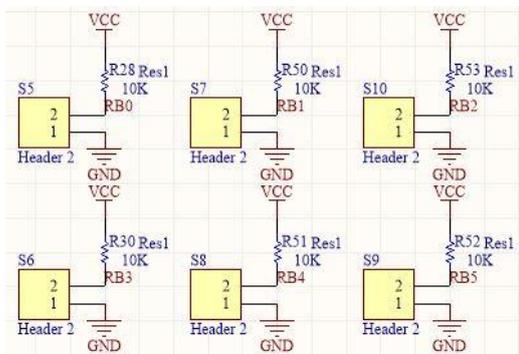


Figura 2-25 Etapa de entrada de sensores magnéticos

Fuente: Lema, Edison, 2016.



Figura 2-26 Colocación de sensores magnéticos en las ventanas y puertas

Fuente: Lema, Edison, 2016.

2.6.6.2. Etapa de potencia

Esta diseñada por un transistor de potencia, que conmuta la señal que proviene del microcontrolador para amplificar esta señal a 12 V mediante el Mosfet IRF540N.

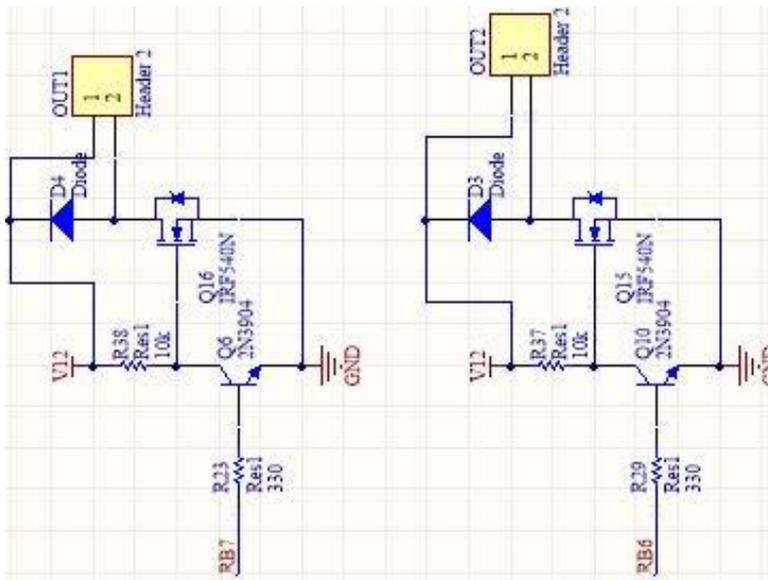


Figura 2-27 Etapa de potencia de la placa de alarma

Fuente: Lema, Edison, 2016.



Figura 2-28 Sirena instalada en la vivienda

Fuente: Lema, Edison, 2016.

2.6.6.3. PCB Placa del sistema de alarma

Pasamos del circuito esquemático al circuito PCB para implementar y soldar todos los elementos electrónicos que contiene la misma, para finalmente realizar las pruebas necesarias de funcionamiento.

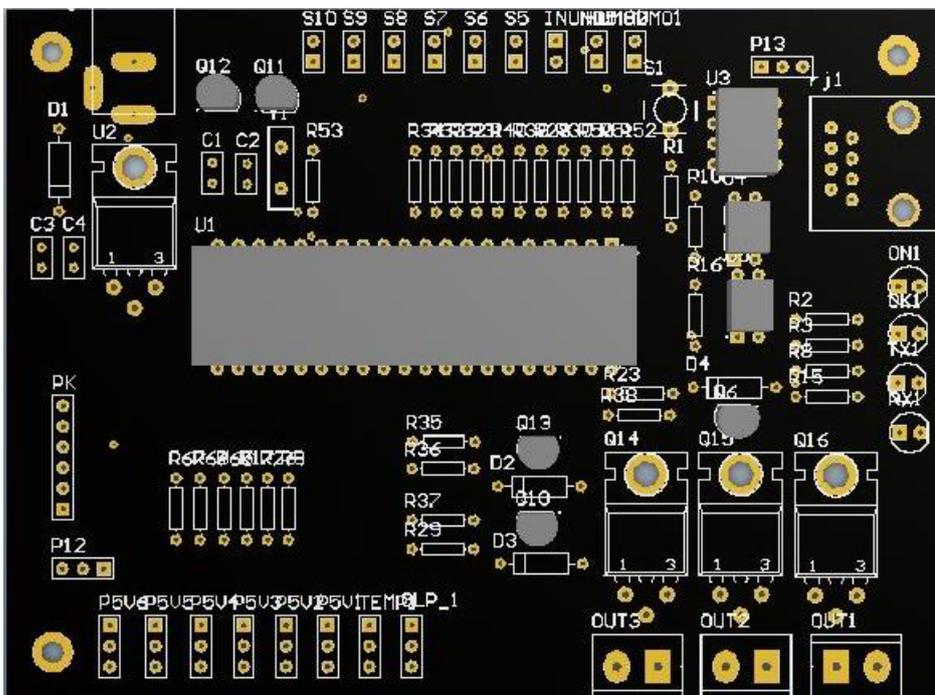


Figura 2-29 PCB de la placa de la alarma

Fuente: Lema, Edison, 2016.

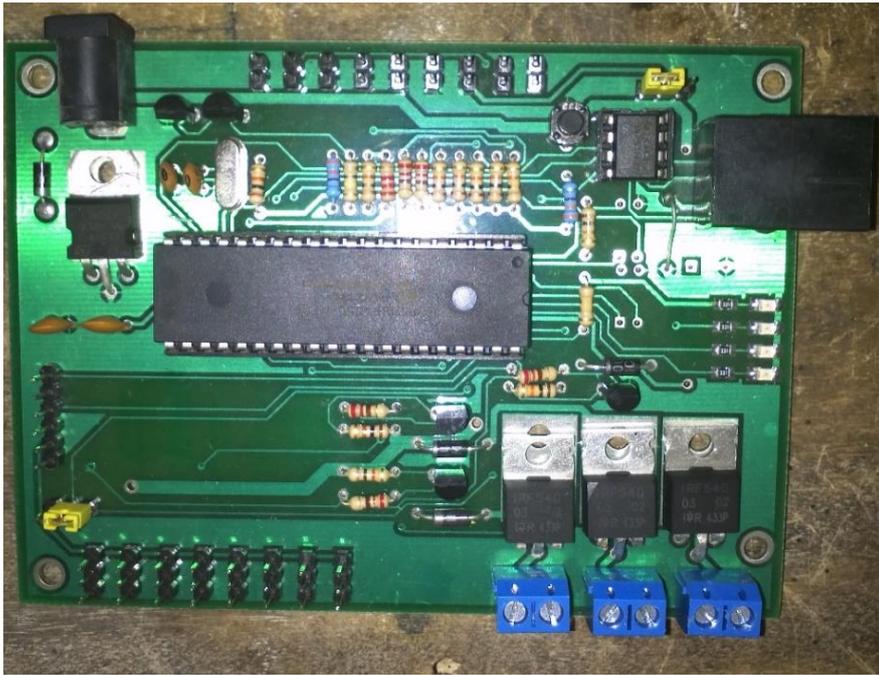


Figura 2-30 PCB de la placa de la alarma implementado

Fuente: Lema, Edison, 2016.

Como medida de seguridad de la placa de la alarma dispone de dos alternativas, una cuando el usuario está en la vivienda y otra cuando está fuera de ella. En este caso se ha implementado un sistema de envío de mensajes de texto vía celular al usuario de la vivienda. Esto ocurre cuando se abre una puerta o una ventana por parte de personas extrañas o intrusos, el sensor magnético envía la señal a la placa de alarma. La otra alternativa es el uso del botón de pánico que esta implementado en la aplicación Web.



Figura 2-31 Módulo GSM para el envío de mensajes al usuario

Fuente: Lema, Edison, 2016.

2.6.7. Placa de control de la persiana

El sistema domótico dispone de una persiana con movimiento, es decir, para subir y bajar. Trabaja con un motor de 12 V DC. A continuación se detalla cada una de las etapas del diseño de esta placa.

2.6.7.1. Etapa de entradas

En esta etapa se han insertado unas entradas en la placa de la persiana para colocar pulsantes y enviar señales al microcontrolador que permita subir y bajar la persiana. Cada entrada tiene un circuito para obtener 5 V TTL al PIC, como se muestra en la figura 2-31.

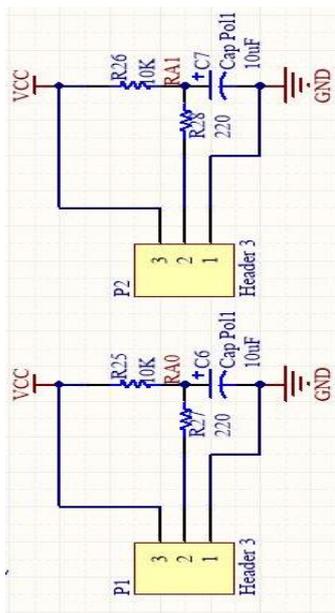


Figura 2-32 Etapa de entradas para pulsantes

Fuente: Lema, Edison, 2016.

2.6.7.2. Etapa de potencia

Es una etapa importante donde se ha realizado un diseño de un puente h; es decir, un controlador de giro para el motor que se encargará del movimiento de la persiana. Este puente h está diseñado con transistores de potencia IRF540N y el IRF9540, son adecuados para el giro del motor de 12 V a una corriente de 5 A y una potencia de 60 Watt.

El diseño tiene como objetivo aislar al circuito del puente h, separando la parte de control y la parte de potencia para garantizar y proteger los elementos electrónicos del circuito. Para el aislamiento de la etapa de potencia se ha utilizado un opto aislador PC817 que recibe una señal del microcontrolador, para activar el Mosfet mediante un divisor de voltaje. El diseño dispone de una fuente de voltaje para la alimentación del motor, conectores y diodos rectificadores.

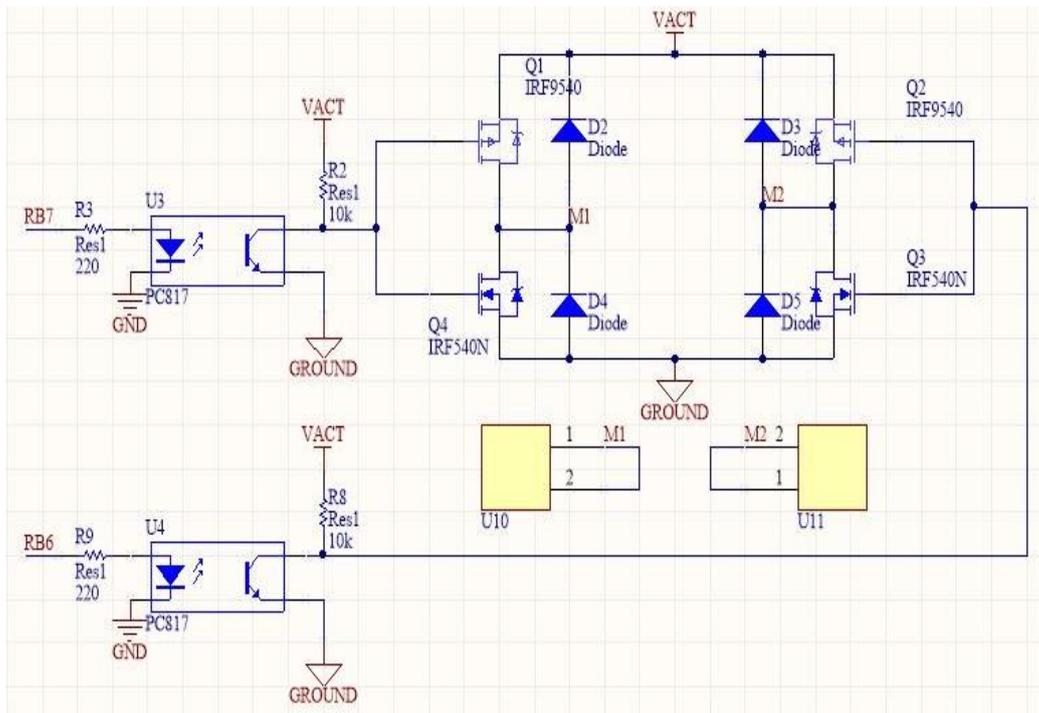


Figura 2-33 Etapa de potencia de la placa de la persiana

Fuente: Lema, Edison, 2016.



Figura 2-34 Persiana colocada en el estudio

Fuente: Lema, Edison, 2016.

2.6.7.3. PCB de la persiana

El PCB de la placa de la persiana cuenta con todos los elementos electrónicos diseñados en el circuito esquemático, como se muestra en la figura 2-35.

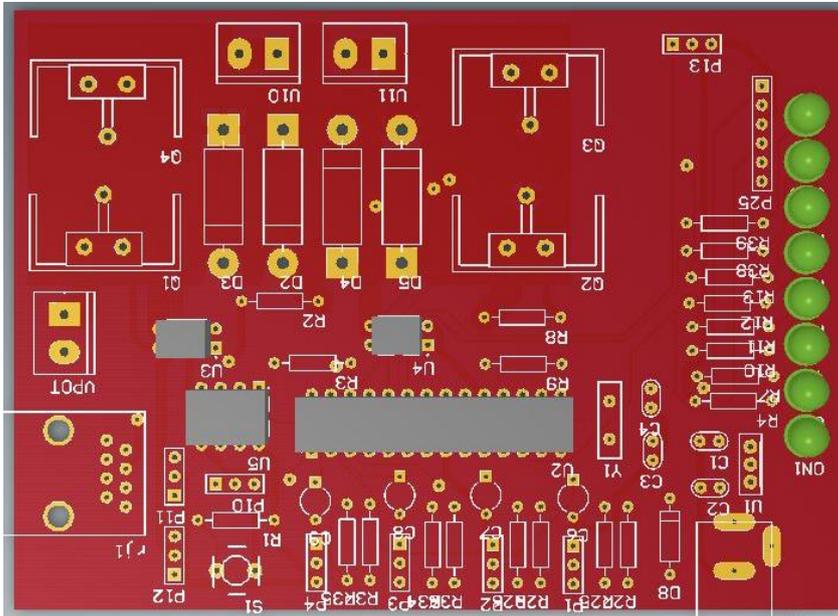


Figura 2-35 PCB de la de la placa de la persiana

Fuente: Lema, Edison, 2016.

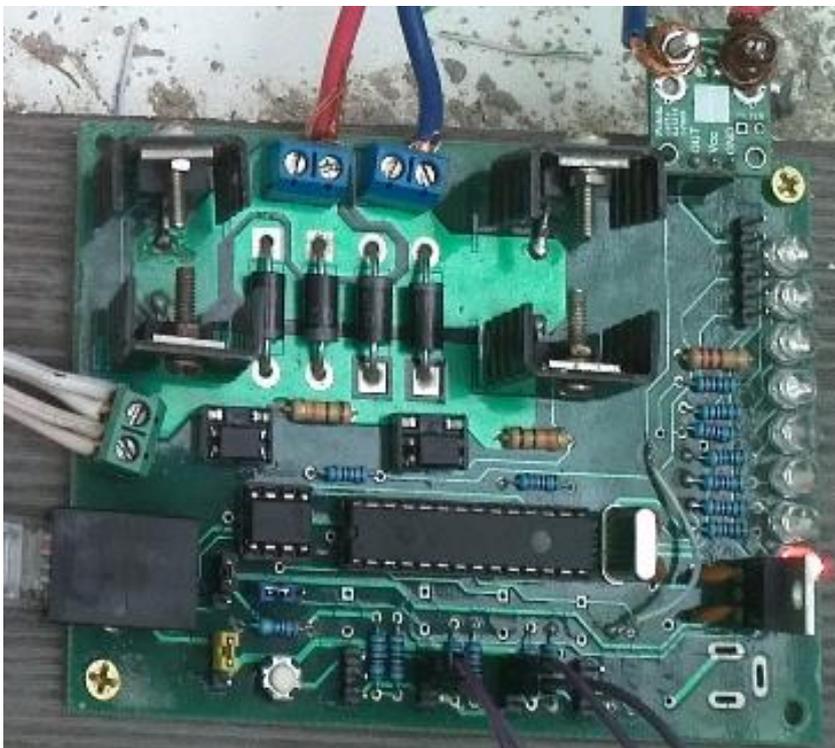


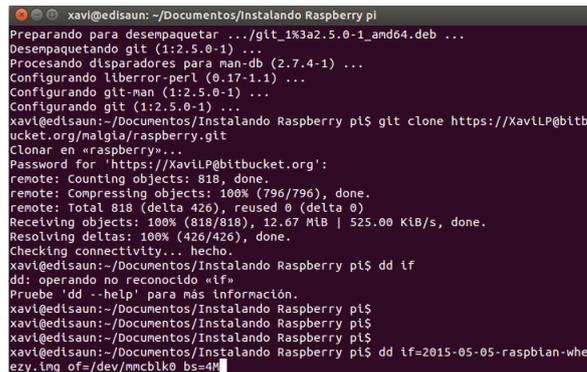
Figura 2-36 PCB de la placa de la persiana implementado

Fuente: Lema, Edison, 2016.

2.7. Programación de la placa Raspberry pi

2.7.1. Instalación del sistema operativo Ubuntu en la placa Raspberry pi

Para la instalación del sistema operativo en la placa Raspberry pi se utilizó el sistema operativo Ubuntu que por sus características y servicios permiten aprovechar al máximo las capacidades de la placa y los otros elementos del sistema para su funcionamiento.



```
xavi@edisaun: ~/Documentos/Instalando Raspberry pi
Preparando para desempaquetar .../git_1%3a2.5.0-1_and64.deb ...
Desempaquetando git (1:2.5.0-1) ...
Procesando disparadores para man-db (2.7.4-1) ...
Configurando liberror-perl (0.17-1.1) ...
Configurando git-man (1:2.5.0-1) ...
Configurando git (1:2.5.0-1) ...
xavi@edisaun:~/Documentos/Instalando Raspberry pi$ git clone https://XaviLP@bitbucket.org/malgia/raspberry.git
Clonar en «raspberry»...
Password for 'https://XaviLP@bitbucket.org':
remote: counting objects: 818, done.
remote: Compressing objects: 100% (796/796), done.
remote: Total 818 (delta 426), reused 0 (delta 0)
Receiving objects: 100% (818/818), 12.67 MiB | 525.00 KiB/s, done.
Resolving deltas: 100% (426/426), done.
Checking connectivity... hecho.
xavi@edisaun:~/Documentos/Instalando Raspberry pi$ dd if
dd: operando no reconocido «if»
Pruebe 'dd --help' para más información.
xavi@edisaun:~/Documentos/Instalando Raspberry pi$
xavi@edisaun:~/Documentos/Instalando Raspberry pi$
xavi@edisaun:~/Documentos/Instalando Raspberry pi$
xavi@edisaun:~/Documentos/Instalando Raspberry pi$ dd if=2015-05-05-raspbian-wheezy.img of=/dev/mmcblk0 bs=4M
```

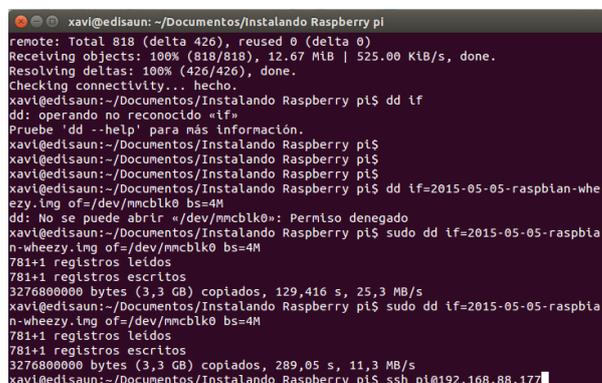
Figura 2-37 Instalación del SO Ubuntu

Fuente: Lema, Edison, 2016.

Una vez finalizada la instalación del sistema operativo, se procedió a programar en la placa Raspberry pi tanto para la transmisión como para la recepción de datos que van a ser enviados desde la aplicación Web hacia placa master y esta a los módulos esclavos.

2.7.2. Instalación del protocolo SSH

Se instaló esta librería, debido a que nos permite conectarnos remotamente hacia la Raspberry pi que está conectada a la Red de la vivienda, los datos que se envían y se reciben son encriptados para mantener la integridad y seguridad de los mismos. En este caso se va a utilizar para poder monitorear la vivienda mediante el sistema domótico.



```
xavi@edisaun: ~/Documentos/Instalando Raspberry pi
remote: Total 818 (delta 426), reused 0 (delta 0)
Receiving objects: 100% (818/818), 12.67 MiB | 525.00 KiB/s, done.
Resolving deltas: 100% (426/426), done.
Checking connectivity... hecho.
xavi@edisaun:~/Documentos/Instalando Raspberry pi$ dd if
dd: operando no reconocido «if»
Pruebe 'dd --help' para más información.
xavi@edisaun:~/Documentos/Instalando Raspberry pi$
xavi@edisaun:~/Documentos/Instalando Raspberry pi$
xavi@edisaun:~/Documentos/Instalando Raspberry pi$
xavi@edisaun:~/Documentos/Instalando Raspberry pi$ dd if=2015-05-05-raspbian-wheezy.img of=/dev/mmcblk0 bs=4M
dd: No se puede abrir «/dev/mmcblk0»: Permiso denegado
xavi@edisaun:~/Documentos/Instalando Raspberry pi$ sudo dd if=2015-05-05-raspbian-wheezy.img of=/dev/mmcblk0 bs=4M
781+1 registros leídos
781+1 registros escritos
3276800000 bytes (3,3 GB) copiados, 129,416 s, 25,3 MB/s
xavi@edisaun:~/Documentos/Instalando Raspberry pi$ sudo dd if=2015-05-05-raspbian-wheezy.img of=/dev/mmcblk0 bs=4M
781+1 registros leídos
781+1 registros escritos
3276800000 bytes (3,3 GB) copiados, 289,05 s, 11,3 MB/s
xavi@edisaun:~/Documentos/Instalando Raspberry pi$ ssh pi@192.168.88.177
```

Figura 2-38 Instalación de ssh

Fuente: Lema, Edison, 2016.

2.7.3. Instalación de la librería wiring pi

En esta etapa se instaló la librería Wiring pi, que permite leer y escribir en los 26 pines de la Raspberry pi. Ofrece otras características como: General Purpose Input output (GPIO) en especial los pines correspondientes a los de RX y TX, entre otros. Los pasos de instalación se muestran en la figura 2-38.

```
pi@raspberrypikabazon: ~
Ign http://archive.raspberrypi.org wheezy/main Translation-en_GB
Ign http://archive.raspberrypi.org wheezy/main Translation-en
Get:7 http://mirrordirector.raspbian.org wheezy/contrib armhf Packages [23.6 kB]
Get:8 http://mirrordirector.raspbian.org wheezy/non-free armhf Packages [49.3 kB]
]
Get:9 http://mirrordirector.raspbian.org wheezy/rpi armhf Packages [592 B]
Ign http://mirrordirector.raspbian.org wheezy/contrib Translation-en_GB
Ign http://mirrordirector.raspbian.org wheezy/contrib Translation-en
Ign http://mirrordirector.raspbian.org wheezy/main Translation-en_GB
Ign http://mirrordirector.raspbian.org wheezy/main Translation-en
Ign http://mirrordirector.raspbian.org wheezy/non-free Translation-en_GB
Ign http://mirrordirector.raspbian.org wheezy/non-free Translation-en
Ign http://mirrordirector.raspbian.org wheezy/rpi Translation-en_GB
Ign http://mirrordirector.raspbian.org wheezy/rpi Translation-en
Fetched 7,149 kB in 49s (146 kB/s)
Reading package lists... Done
pi@raspberrypikabazon ~ $ git clone git://git.drogon.net/wiringPi
Cloning into 'wiringPi'...
remote: Counting objects: 914, done.
remote: Compressing objects: 100% (748/748), done.
remote: Total 914 (delta 653), reused 217 (delta 142)
Receiving objects: 100% (914/914), 285.87 KiB | 243 KiB/s, done.
Resolving deltas: 100% (653/653), done.
pi@raspberrypikabazon ~ $
```

Figura 2-39 Instalación de librería Wiring pi

Fuente: Lema, Edison, 2016.

2.7.4. Instalación de la base de datos

Para poder almacenar los datos que viajan en nuestro sistema domótico se utilizó mysql-server phpmyadmin, que permite almacenar los datos enviados desde la aplicación Web, analizarlos y acceder a ellos en cualquier momento que el usuario lo desee.

```
pi@raspberrypikabazon: ~/wiringPi
[Link]
[Install]

All Done.

NOTE: To compile programs with wiringPi, you need to add:
  -lwiringPi
  to your compile line(s) To use the Gertboard, MaxDetect, etc.
code (the devLib), you need to also add:
  -lwiringPiDev
  to your compile line(s).

pi@raspberrypikabazon ~/wiringPi $ sudo apt-get install mysql-server phpmyadmin
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  apache2-mpm-prefork apache2-utils apache2.2-bin apache2.2-common
  dbconfig-common heirloom-mailx libaio1 libapache2-mod-php5 libapr1
  libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libdbd-mysql-perl
  libdbi-perl libhtml-template-perl libmcrypt4 libmysqlclient16
  libmysqlclient18 libonig2 libqdbm14 lsof mysql-client-5.5 mysql-common
  mysql-server-5.5 mysql-server-core-5.5 php5-cli php5-common php5-gd
  php5-mcrypt php5-mysql ssl-cert
```

Figura 2-40 Instalación de mysql-server

Fuente: Lema, Edison, 2016.

Luego se abre una pantalla donde nos indica las configuraciones de mysql-server, una de las primeras configuraciones es insertar una contraseña la cual va ser usada únicamente por el usuario para la administración del dispositivo, esta contraseña no se podrá cambiar a menos que se desinstale de la placa Raspberry pi, el mysql-server o el usuario mismo la cambie.

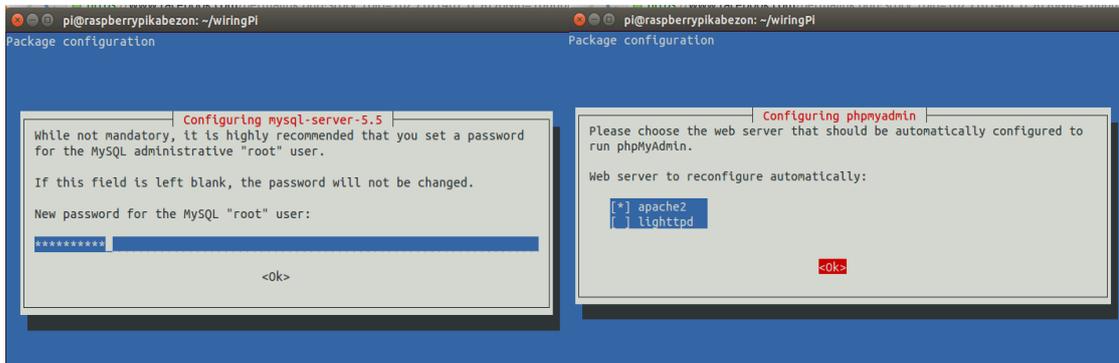


Figura 2-41 Configuración mysql-server y phpmyadmin

Fuente: Lema, Edison, 2016.

2.8. Creación de la aplicación web

La aplicación Web está desarrollada en HTML y JavaScript, por lo que puede ser usada desde cualquier navegador Web y desde cualquier parte del mundo a través del Internet. Si el usuario se encuentra dentro de la red local, puede acceder sin necesidad del uso del Internet.

Esta aplicación está dividida en 5 secciones, mediante las cuáles el usuario final puede controlar los diferentes módulos instalados en la vivienda, así como la configuración de su cuenta. A continuación se detallan cada una de las secciones:

2.8.1. Opción alarma

La pantalla para el control de la Alarma permite desactivar o activar remotamente. Así como verificar el estado de cada uno de los sensores conectados al sistema. la pantalla principal consta de 3 tarjetas (información general, on/off y pánico).

Para agregar un botón de tipo on/off se utilizó un input de tipo checkbox (Materialize). El mismo que se encarga de activar o desactivar la alarma cuándo es presionado. Cuándo se quiere desactivar la alarma se hace además uso de un Modal que permite el ingreso de la contraseña:

```
<input id="inputAlarm"
onchange="if(this.checked){ callAlarm(43,98,";)}else{ $('#password').openModal();}"
type="checkbox">
```

El botón de Pánico ubicado en la tercera opción de la pantalla, permite activar la sirena en caso de que exista una emergencia. Para agregarlo usamos un link de HTML.

```
<a onclick="callAlarm(43,99,";">Pánico</a>
```

En la pantalla dónde se muestran los sensores magnéticos, usamos inputs de tipo radio. Los mismos que siempre están desactivados y únicamente presentan el estado actual de los sensores.

```
<input class="with-gap" id="mag0" disabled="" type="radio">
```

Por último, tenemos una pantalla que nos permite activar o desactivar la alarma ciertos días de la semana de manera automática, según la necesidad del usuario de la vivienda. Se usó un inputs de tipo checkbox para marcar o desmarcar cada uno de los días.

```
<input id="allDays"
onchange="if(this.checked){ checkDays(this.id,'all');}else{ checkDays(this.id,'none');}"
type="checkbox">
```



Figura 2-42 Diseño en la Web para la alarma

Fuente: Lema, Edison, 2016.

2.8.2. Opción luces RGB

En la sección RGB tenemos la opción para el control de Luces LED de colores en donde el usuario puede escoger cualquiera de los 16777215 combinaciones de colores que tiene la gama RGB.

Para agregar una tarjeta tipo RGB usamos la librería colorpicker de jquery-ui de la siguiente manera.

```

<div style="background-color: rgb(226, 102, 0);" id="31-1"></div>

$('#31-1').colpick({
  layout:'rgbhex',
  color:'e26600',
  onSubmit:function(hsb,hex,rgb,el) {
    red=rgb['r']>254?254:rgb['r']<1?1:rgb['r'];
    green=rgb['g']>254?254:rgb['g']<1?1:rgb['g'];
    blue=rgb['b']>254?254:rgb['b']<1?1:rgb['b'];
    callClient(31, 1, blue);
    callClient(31, 2, green);
    callClient(31, 3, red);
    $(el).css('background-color', '#'+hex);
    $(el).colpickHide(); });

```

El código anterior, se encarga tanto de mostrar el botón para escoger el color (primera parte), como de enviar el color seleccionado al módulo central (segunda parte).



Figura 2-43 Espacio para luces RGB

Fuente: Lema, Edison, 2016.

2.8.3. Opción luces Ac

En la tercera pantalla, tenemos la opción para el control de todas las luces de la vivienda. Debido a que se han instalado tanto luces On/Off como luces Dimmer se dispone de dos tipos de controles, en el caso de ser on/off tenemos un botón para encender, mientras que para realizar un control dimmer se usa un slider para cambiar la luminosidad de los focos, en las siguiente figura 2-43 podemos observar el panel de luces.

2.8.3.1. Botones On/Off

Contienen un input de tipo checkbox, el mismo que permite encender o apagar una luz, la misma sintaxis se usa para mover la persiana instalada en el sistema determinada:

```
<inputonchange="if(this.checked){ callClient(10,1,2);}else{ callClient(10,1,1);}"  
type="checkbox">
```

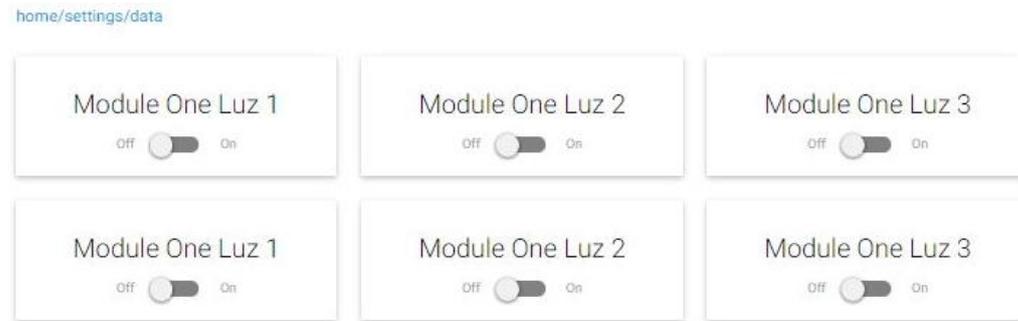


Figura 2-44 Panel de luces on/off

Fuente: Lema, Edison, 2016.



Figura 2-45 Panel de la persiana on/off

Fuente: Lema, Edison, 2016.

2.8.3.2. Botón Dimmer

Contienen un input de tipo checkbox, igual al de los botones On/Off (En caso de que se desee prender o apagar la luz por completo), y además un input de tipo range mediante el cual se pueden escoger valores entre 1 y 100.

```
<input id="dimmer21_1" min="1" max="100" onchange="callClient(21,1,this.value);"  
type="range">
```

Configuración

La cuarta pantalla permite configuración del sistema. Aquí el usuario puede crear, modificar o eliminar usuarios y ver los módulos que tiene instalado en el mismo.

Función callClient

Una de las funciones más importantes de la aplicación Web es el callClient de JavaScript. Esta función nos permite enviarle información al módulo central. Como se puede ver en la parte software cada uno de los botones de la aplicación hacen un llamado a esta función.

```
function callClient(id, device, value){

var data = {"id":id,"device":device,"value":value};

$.ajax({

url: '../api/v1/send',

headers: {

"Authorization": "Basic " + btoa(user + ":" + apiKey)},

contentType: 'application/json',

type: 'POST',

dataType: "json",

data: JSON.stringify(data),

success: function (data) {

console.log(data);},

error: function (data) { console.log(data.responseText);}});}
```



Figura 2-46 Espacio luces Dimmer
Fuente: Lema, Edison, 2016.

2.8.3.3. Opción para la cámara

La aplicación Web contiene un menú de opciones, una de ellas con el nombre de cámara sirve para visualizar algunos sectores de la vivienda, observando los cambios que se ejecutan mediante las acciones que están programadas en los módulos. Gracias a las características que tiene la cámara se pueden almacenar movimientos previamente definidos, los cuales se encuentran almacenados en la aplicación mediante botones con sus respectivos nombres de las áreas de la vivienda.



Figura 2-47 Opción para la cámara
Fuente: Lema, Edison, 2016.

Para poder visualizar la cámara se tiene que agregar en el código del programa una imagen en el cuál se cargarán las imágenes enviadas por la misma.

```
<img id="cam_draw" src="" width="400" height="300"/>
```

El contenido de esta imagen inicialmente esta vacío, por lo que al momento de ejecutar el botón “Cámara” cargamos el mismo usando el código:

```
$('#cam_draw').attr('src','http://'+ip+':'+puerto+'/videostream.cgi?user=user&pwd=pwd&resoluci  
on=32&rate=0');
```

Para poder mover la cámara usamos un frame, en el cuál ejecutamos peticiones web al momento de dar click en los botones.

```
<iframe name="action_zone" id="action_zone" hidden></iframe>
```

La función que se encarga de ejecutar las peticiones es moveCam(codigo) y ejecuta el siguiente código:

```
action_zone.location='http://'+ip+':'+puerto+'/decoder_control.cgi?user=user&pwd=pwd&com  
mand='+pos;  
(data),
```



Figura 2-48 Cámara instalada en la vivienda

Fuente: Lema, Edison, 2016.

Una vez terminada la aplicación Web y el ensamblado de las placas de domótica, se han instalado en una caja de distribución a donde llega todo el cableado del sistema según el plano de domótica e interconectamos todos los módulos entre si formando un solo sistema con la pasarela residencial que está formado por una PC y la Raspberry pi para la salida a la nube del internet para monitorear desde cualesquier parte donde el usuario se encuentre.

CAPÍTULO III

3. COSTOS, PRUEBAS Y RESULTADOS

Para la elaboración del Sistema domótico de la vivienda, intervinieron algunos componentes electrónicos que implican gastos en la implementación del sistema, así como los distintos dispositivos electrónicos utilizados en el mismo.

3.1. Costo de materiales

Tabla 3-1 Costos de materiales del sistema

PRECIOS DE LOS MATERIALES DEL SISTEMA DOMÓTICO RESIDENCIAL			
Materiales	Cantidad	P.unitario	Total
Programador Pickit 3	1	96,00	96,00
PIC18F2550	10	9,00	90,00
PIC18F4550	2	11,,00	22,00
Raspberry pi modelo B	2	80,00	160,00
Cámara wireless	2	120,00	240,00
Motor 12V DC	1	70,00	70,00
Tarjeta Sd card y memoria de 8GB	2	10,00	20,00
Quemador de Pic	1	5,00	5,00
Tira Led RGB	5	10,00	50,00
Persiana Giratoria	1	180,00	180,00
Cable UTP	1	150,00	150,00
Cable HDMI	1	12,00	12,00
Cable AWG #14(Blanco, Negro, Rojo)	3	45,00	135,00
Caja domótica	1	50,00	50,00
Placa master(Sin materiales)	1	45,00	45,00
Placa Luces RGB(Sin materiales)	1	45,00	45,00
Placa Luces AC(Sin materiales)	2	45,00	90,00
Placa De alarma(Sin materiales)	1	45,00	45,00
Placa de la persiana(Sin materiales)	1	45,00	45,00
Sirena para exterior	1	30,00	30,00

Foco incandescente	12	1,50	18,00
Sensor de rotura de vidrio	6	5,00	30,00
Sensor magnético	5	4,50	22,50
Fuente de alimentación	2	40,00	80,00
Resistencias	120	0,15	18,00
Opto acopladores	8	0,60	4,80
Capacitores	20	0,15	3,00
Triac BTA12	10	2,00	20,00
Mosfet IRF9540	6	2,50	15,00
Mosfet IRF540	11	2,50	27,50
Reguladores de voltaje 7805CV	5	0,75	4,25
Reguladores de voltaje 7812CV	4	1,20	4,80
Diodos led SMD	24	0,40	9,60
Resistencias SMD	24	0,25	6,00
Integrado max485	10	8,00	80,00
Conectores RJ-45	10	0,80	8,00
Zócalos RJ-45	10	0,50	5,00
Peineta Macho	4	0,80	3,20
Peineta Hembra	2	0,80	1,60
Fusibles y Porta fusible	6	1,50	9,00
Borneras	12	0,60	7,20
Diodos a 5 A	6	0,80	4,80
Diodos a 2 A	20	0,30	6,00
Potenciómetros	5	0,70	3,50
Gastos varios	5	20,00	100,00
Total			2071,75

Realizado por: Lema, Edison, 2016.

Fuente: Lema, Edison, 2016.

3.2. Pruebas realizadas

Se ha visto importante hacer un análisis, en cuanto a la confiabilidad que el sistema domótico ha logrado en la vivienda para los usuarios, para ello se registran algunos datos que nos permiten hacer el respectivo análisis. Se ha visto pertinente obtener datos para hacer un análisis en cuanto a los tiempos de respuesta de los esclavos al ser activados desde la aplicación Web, estos serán medidos en base a la distancia de donde nos encontremos de la vivienda, las distancias son variadas, para medir la respuesta del sistema, y saber cuál es su resultado.

3.3. Análisis estadístico

3.3.1. Confiabilidad del sistema

Para determinar el análisis en cuanto a la confiabilidad del sistema se ha visto necesario realizar una encuesta a 30 personas para que den la opinión acerca del sistema domótico, esto se ha hecho con aquellas personas que tienen conocimiento en cuanto a tecnología, para esto se han medido ciertos parámetros que se pueden ver a continuación en la siguiente tabla.

Tabla 3-2 Datos para la confiabilidad del sistema

Número de personas	Facilidad de manejo			Seguridad			Domótica en una vivienda	
	Fácil	Regular	Difícil	Alta	Media	Baja	Confort y comodidad	Ahorro energético
Oswaldo	x				x		x	
Gabriela		x			x			x
Gabriel	x					x	x	
Total	1	2	0	0	2	1	2	1
Porcentaje								

Realizado por: Lema, Edison, 2016.

Fuente: Lema, Edison, 2016.

Los usuarios han calificado que el manejo de la aplicación Web es fácil, por lo que se puede manipular desde cualquier lugar donde tengamos acceso a internet.

En la seguridad con la que cuenta el sistema, los usuarios de la vivienda han dispuesto que es el sistema domótico cuenta con un grado de seguridad medio, ya que cuenta con un aviso de mensajes de texto al usuario en caso que no se encuentre en la vivienda.

Uno de los parámetros escogidos por los usuarios de la vivienda es el confort y comodidad, porque los actuadores se pueden accionar desde cualquier parte de la vivienda.

3.3.2. Tiempo de respuesta del sistema

Tabla 3-3 Tabla de tiempos para la respuesta del sistema

Muestras del exterior de la vivienda.				Muestras en el interior de la vivienda.			
Nota: Los datos están tomados cada media hora.							
Respuesta del tiempo del sistema				Respuesta del tiempo del sistema			
Número	Tiempo (s)				Número	Tiempo (s)	
	Encendido	Apagado				Encendido	Apagado
1	6,2	9,2			1	1,3	1,5
2	6,3	7,5			2	1,4	1,6
3	8	7,3			3	1,1	1,7
4	7,8	6,5			4	1,6	1,5
5	6,3	7,4			5	1,3	1,1
6	8,4	7,6			6	1,2	1,6
7	7,8	6,3			7	1,3	1,5
8	8,6	7,9			8	1,1	1,6
9	6,8	7,7			9	1,5	1,3
10	7,6	7,4			10	1,6	1,2
11	8,5	7,9			11	1,4	1,1

12	6,6	7,3			12	1,3	1,6
13	7,5	8,8			13	1,7	1,2
14	8,6	7,2			14	1,5	1,1
15	6,5	7,4			15	1,6	1,5
16	6,8	7,6			16	1,5	1,2
17	8,5	6,9			17	1,3	1,4
18	6,7	8,1			18	1,1	1,5
19	7,5	8,3			19	1,7	1,4
20	7,6	6,9			20	1,3	1,6
21	7,8	6,8			21	1,2	1,4
22	6,4	8,3			22	1,1	1,6
23	7,3	8,6			23	1,4	1,2
24	6,6	7,9			24	1,5	1,3
Promedio	7,36	7,62				1,38	1,40

Respuesta del tiempo del sistema				Respuesta del tiempo del sistema			
Número	Tiempo (s)				Número	Tiempo (s)	
	Encendido	Apagado				Encendido	Apagado
1	7,4	5,3			1	1,8	1,5
2	5,1	4,7			2	1,2	1,9
3	4,2	3,9			3	1,1	1,5
4	4,7	4,1			4	1,9	1,6
5	5,2	6,4			5	1,4	1,2
6	5,3	3,8			6	1,6	1,8

7	6,2	5,9			7	1,7	1,3
8	4,8	6,2			8	1,6	1,2
9	3,7	5,3			9	1,3	1,5
10	7,3	6,2			10	1,6	1,1
11	7,2	7,1			11	1,8	1,4
12	4,2	6,3			12	1,9	1,3
13	4,4	3,9			13	1,5	1,3
14	5,3	6,3			14	1,2	1,7
15	4,2	5,6			15	1,4	1,9
16	5,8	4,7			16	1,3	1,5
17	6,1	5,9			17	1,6	1,1
18	5,2	5,6			18	1,2	1,6
19	3,8	4,6			19	1,7	1,9
20	4,3	5,3			20	1,6	1,2
21	3,9	4,8			21	1,4	1,7
22	5,1	4,7			22	1,5	1,1
23	4,6	3,9			23	1,8	1,4
24	5,8	6,4			24	1,6	1,3
Promedio	5,16	5,29				1,53	1,46

Realizado por: Lema, Edison, 2016.

Fuente: Lema, Edison, 2016.

Para el análisis del tiempo de respuesta del sistema, se vio necesario tomar datos tanto para el encendido como para el apagado de los actuadores, estos tiempos fueron tomados durante 12 horas de dos días consecutivos, cada media hora, lo cual nos ayuda a determinar qué tan real es nuestro sistema en cuanto a la distancia de donde nosotros estamos monitoreando nuestra vivienda.

Con estos datos se ha logrado hacer un promedio de los distintos parámetros que se han medido, para luego poder comparar, los promedios de encendido como de apagado, ya que estos datos se han tomado utilizando distintos anchos de banda de internet en los días expuestos. Además se han tomado estos datos desde la parte interna de la vivienda, y también se ha tomado desde partes externas para poder hacer un monitoreo y obtener datos y saber cuál es la variación del tiempo que tan rápido es el sistema y que acogida le da el usuario con la respuesta.

Muestras del exterior de la vivienda.

Se ha hecho un promedio de los datos obtenido el cual es de 7,36 segundos cuando encendemos un actuador, mientras que el promedio de apagado es de 7,62 segundos, estos datos son del primer día. En el segundo día se han tomado datos que haciendo promedio arroja un resultado de 5,16 segundos cuando encendemos y de 5,29 cuando apagamos, esto debido a la calidad del internet con que estamos trabajando en la web.

Muestras del interior de la vivienda.

Los promedios obtenidos en el interior de la vivienda son datos muy cortos, por lo cual tenemos en el primer día los siguientes valores en el encendido 1,38 segundos y 1,40 segundos en el apagado, promedios muy cortos que nos indican que el sistema responde rápidamente. En los datos del segundo día tenemos 1,53 segundos en el encendido y de 1,46 segundos en el apagado, como se puede observar hay una diferencia muy corta y se puede afirmar que el sistema tiene una excelente respuesta.

Luego de hacer los promedios de los datos, se puede concluir en que el sistema tiene un tiempo respuesta mayor cuando estamos en algún lugar ajeno a la vivienda una diferencia de 6,36 segundos encendiendo y 6,45 segundos apagando esto en el primer día. En el segundo día tenemos los siguientes resultados que son los tiempos en el interior de la vivienda, 1,45 segundos en el encendido de los actuadores y 1,43 segundos en el apagado.

Tabla 3-4 Retardo de tiempo desde el interior y exterior de la vivienda

	Exterior	Interior
	Primer día	Segundo día
Encendido	6,36	1,45
Apagado	6,45	1,43

Realizado por: Lema, Edison, 2016.

Fuente: Lema, Edison, 2016.

Hay un tiempo de diferencia de los dos días de acuerdo al promedio en el encendido y en el apagado, con lo cual hay un retardo para poder ver la reacción en tiempo real, con lo cual se puede ver que hay diferencia cuando estamos en la vivienda y fuera de ella.

CONCLUSIONES

- La implementación del sistema domótico está diseñada con la placa Raspberry pi, que es un dispositivo que permite el uso de un sistema operativo libre como es Ubuntu, gracias a sus características se puede tener la salida al internet y comunicación serial para el envío y recepción de datos que permiten monitorear las áreas de la vivienda mediante una aplicación.
- El diseño e implementación de los módulos del sistema domótico está compuesta por una placa master y 4 placas esclavos que controlan a los distintos actuadores del sistema.
- La comunicación entre la placa central y los módulos esclavos de domótica se realizó mediante el protocolo de comunicación MODBUS que brinda un alto grado de confiabilidad, además permite trabajar mediante un sistema Maestro-esclavo, hasta un máximo de 247 esclavos.
- Las escenas de control del sistema domótico están previamente definidas mediante una aplicación Web con una interfaz gráfica amigable para el usuario, el mismo que permita activar distintos dispositivos en el mismo instante de tiempo.
- El diseño centralizado de un sistema domótico es eficiente a comparación del descentralizado debido a que el diseño descentralizado necesita controladores individuales para tener acceso y activar un dispositivo desde el exterior de la vivienda, mientras que el sistema centralizado utiliza un solo dispositivo para el control de la vivienda.
- El sistema domótico tiene un tiempo de respuesta el cual varía de acuerdo a la distancia de donde se está monitoreando. Presenta oscilaciones de tiempo con anchos de banda de distinta frecuencia, lo cual puede ocasionar un desbalance en el tiempo real a la respuesta del sistema.
- El sistema domótico modular brinda múltiples beneficios en cuanto a instalación del sistema, puesto que permite incorporar en cualquier momento nuevos módulos de control y asignar una dirección Modbus como esclavo.
- El sistema contiene como medida de seguridad el envío de mensajes de texto a su celular, para informar al dueño de la vivienda si hay alguna anomalía o hay personas ajenas a la vivienda, estas señales son enviadas mediante los sensores magnéticos colocados en las puertas y ventanas de vivienda, y se activarán inmediatamente la alarma.

- La placas del sistema domótico se han diseñado con Altium Designer para su mejor estética y funcionamiento, puesto que brinda gran cantidad de características para elaboración de una placas profesionales, además da la facilidad de que los archivos que se crean puedan ser entendidos por la maquina CNC donde se fabrican los modulos.
- El uso de una cámara Wireless ayuda al diseño del sistema, mediante la cual podemos visualizar cada una de las acciones que están programadas en la aplicación Web.
- La pasarela residencial está formada por un computador u otro dispositivo con el que nos podamos conectar a la Web y conjuntamente con la tarjeta Raspberry pi para conectarnos a la nube del internet y monitorear desde el exterior de la vivienda.

RECOMENDACIONES

- En el desarrollo de las placas de control es de gran importancia tener claro el concepto de acoplamiento de los circuitos tanto de la parte de control, como la parte potencia del circuito, así se protege los elementos del sistema, además conocer que elemento utilizar para realizar el acoplamiento adecuado.
- Cuando se trabaja con una Raspberry pi y se utiliza los pines de GPIO, lo recomendable es realizar un circuito amplificador para mejorar la señal enviada al microcontrolador y sea TTL, esto se puede hacer con transistores de potencia Mosfet.
- Es factible utilizar la comunicación serial RS485, que permite la transmisión de datos de lectura y escritura para la comunicación MODBUS que utiliza el sistema, mediante un master y varios esclavos instalados.
- El sistema domótico es confiable para una vivienda, por ello es necesario que esté se acople a la estética de la misma, es necesario instalar en una estructura que está en proceso de construcción de esta manera el cableado quedará totalmente cubierto basándonos en un plano domótico, el cual servirá para la instalación.
- En cuanto a la comunicación entre el master y los distintos esclavos, es necesario que el cableado que se esté utilizando sea aislado, así podemos evitar retardos y pérdidas de información.
- Cuando se diseña un circuito en Altium Designer debemos conocer el valor del voltaje y corriente que se manejan en la placa debido a esto será el tamaño del grosor de las pistas de cobre, por lo general se recomienda utilizar 1mm de grosor por cada amperio que vaya a circular en la placa.
- En la aplicación web es necesario usar WebSocket, ya que permite una interfaz de comunicación con la Raspberry pi, y así mostrar en la aplicación, si habido un cambio en el sistema es decir si se acciona algún botón para prender las luces, esto sirve de gran ayuda al usuario para saber que sucede en su casa de no estar en ella.

BIBLIOGRAFÍA

ABC TECNOLOGÍA. *Que es raspberry pi y para qué sirve* [Web]. Madrid, España, 2013.

[Consulta: 15 diciembre de 2015].

Disponible en: <http://www.abc.es/tecnologia/informatica-hardware/20130716/abci-raspberry-como-201307151936.html>

ALEGSA, Leandro. *Definición de entorno web* [Web]. Argentina, 2011.

[Consulta: 7 de febrero de 2015].

Disponible en: <http://www.alegsa.com.ar/Dic/entorno%20web.php>

ALVAREZ, Miguel. *Que es HTML* [Web]. 2001.

[Consulta: 29 de noviembre 2015].

Disponible en: <http://www.desarrolloweb.com/articulos/que-es-html.html>

Casas inteligentes [Web]. 2015.

[Consulta: 25 noviembre 2015].

Disponible en: <http://www.construtips.com/articulos-utiles/casas-inteligentes/casas-inteligentes.html>

CENTRO DE INVESTIGACIÓN DE LA WEB. *Cómo funciona la web*. 1ra edición. Santiago, Chile. Gráfica de LOM. 2008. pp. 47 – 48.

[Consulta: 20 noviembre 2015].

Disponible en: <http://www.ciw.cl/libroWeb-NV.pdf>

CISTERNA, Alexander. *DESARROLLO DE LA ELECTRONICA DE COMUNICACIÓN DE DATOS Y VIDEO, PARA UNA CAMARA SUBMARINA SOSTENIDA POR CABLE* (Tesis).

Universidad Austral de Chile, Facultad de Ciencias de la Ingeniería, Escuela de Ingeniería Civil Electrónica. Valdivia, Chile. 2009. pp. 3-5.

[Consulta: 2 de febrero de 2015].

Disponible en: <http://cybertesis.uach.cl/tesis/uach/2009/bmfcic579d/doc/bmfcic579d.pdf>

CHÁVEZ, Samuel. *Aplicaciones Web* [blog]. 2011.

[Consulta: 1 diciembre de 2015].

Disponible en: <http://rendimientodesistemas.blogspot.com/2011/12/aplicaciones-web.html>

COLLOGO, Jorge. *Características del lenguaje PHP* [blog]. 2011.

[Consulta: 2 diciembre de 2015].

Disponible en: <http://lenguajephpjc.blogspot.com/2011/05/caracteristicas-del-lenguaje-php.html>

CASTILLO, Leonardo. *USB TTL puerto de comunicación UART* [blog]. 2014.

[Consulta: 26 diciembre de 2015].

Disponible en: <http://saber.patagoniatecnology.com/usb-ttl-puerto-de-comunicacion-uart-arduino-argentina-ptec/>

CORDOVA, Mario. *Domótica al alcance de tu mano* [Web]. Neoteo, 2009.

[Consulta: 4 diciembre de 2015].

Disponible en: <http://www.neoteo.com/rs485-domotica-al-alcance-de-tu-mano-15810#prettyPhoto>

DIGNANI, Jorge. *ANÁLISIS DEL PROTOCOLO ZIGBEE* (Tesis). Universidad Nacional de la Plata, Facultad de Informática. La plata, Argentina. 2011. pp. 2.

[Consulta: 5 de marzo de 2016].

Disponible en: http://postgrado.info.unlp.edu.ar/Carreras/Especializaciones/Redes_y_Seguridad/Trabajos_Finales/Dignanni_Jorge_Pablo.pdf

DUARTE, Manuel. *Programación de PHP con ejemplos*. Sevilla-España. 2012 pp. 2-6.

[Consulta: 6 de abril de 2016].

Disponible en: http://servicio.uca.es/softwarelibre/publicaciones/apuntes_php

GONZALES, Enrique. *Que es PHP* [Web]. 2011.

[Consulta: 2 de diciembre 2015].

Disponible en: http://aprenderaprogramar.com/index.php?option=com_content&view=article&id=492:ique-es-php-y-para-que-sirve-un-potente-lenguaje-de-programacion-para-crear-paginas-web-cu00803b&catid=70:tutorial-basico-programador-web-php-desde-cero&Itemid=193

GONZALES, Mónica. *Electricidad-fusibles y disyuntores* [Web]. 2011.

[Consulta: 15 diciembre de 2015].

Disponible en: <http://fisica.laguia2000.com/energia/electricidad-fusibles-y-disyuntores>

HERNÁNDEZ, Ramón. *TECNOLOGÍA DOMÓTICA PARA EL CONTROL DE UNA VIVIENDA* (Tesis). Universidad Politécnica de Cartagena, Tecnologías de la Información y las Comunicaciones, Escuela Técnica Superior de Ingeniería de Telecomunicación. Murcia, España. 2012. pp. 7-12.

[Consulta: 23 de abril de 2016].

Disponible

en:

<http://repositorio.upct.es/bitstream/handle/10317/2793/pfc4381.pdf;jsessionid=F1222E8DD461C66FE551084E9BA2ED9C?sequence=1>

Introducción a MODBUS [Web]. 2012.

[Consulta: 25 noviembre 2015].

Disponible en: <https://domotizando.wordpress.com/tag/modbus/>

JUNESTRAND, Stefan. *Pasarela residencial* [Web]. 2001.

[Consulta: 23 abril 2016].

Disponible en: <https://www.casadomo.com/noticias/pasarela-residencial>

LAVADO, Paco. *Interfaces de comunicación* [blog]. 2013.

[Consulta: 4 diciembre de 2015].

Disponible en: <http://geekenformacion.blogspot.com/2013/01/interfaces-de-comunicacion-spi-i2c-uart.html>

MARIN, Juliana. *Sistema domótico y su funcionamiento* [blog]. Blogspot, 2012.

[Consulta: 25 marzo 2016].

Disponible en: <http://julianamarin04.blogspot.com/>

Microcontroladores PIC [Web]. 2013.

[Consulta: 3 de diciembre 2015].

Disponible en: <http://www.ceduvirt.com/resources/Microcontroladores.pdf>

Programación en C [Web]. 2011.

[Consulta: 18 de diciembre de 2015].

Disponible en: <http://www.mikroe.com/chapters/view/80/capitulo-2-programacion-de-los-microcontroladores/>

Que es domótica [Web]. Barcelona. 2000.

[Consulta: 24 noviembre 2015].

Disponible en: <http://www.cedom.es/sobre-domotica/que-es-domotica#gkPageTop>

Que es raspberry pi [blog]. 2013.

[Consulta: 15 diciembre de 2015].

Disponible en: <http://arduprojects.blogspot.com/2013/10/definicion-y-usos-de-raspberry-pi.html>

RODRIGUEZ, Raúl. *MODBUS parte III* [Blog]. 2012

[Consulta: 29 de noviembre 2015].

Disponible en: <http://tecdigitaldelbajio.com/blog/27-modbus-parte-iii-que-es-el-modbus.html>

TORRES, Roger. *IMPLEMENTACIÓN DEL PROTOCOLO EN MICROCONTROLADOR* (Tesis). Universidad Industrial de Santander, Facultad de ingenierías fisicomecánicas, Escuela de Ingeniería eléctrica, electrónica y telecomunicaciones. Bucaramanga. 2006. pp. 22-25 [Consulta: 13 de enero 2016].

Disponible en: <http://repositorio.uis.edu.co/jspui/bitstream/123456789/3176/2/119444.pdf>

TORRITI, Miguel. Tutorial de microcontroladores PIC. 2007, pp. 3-6.

[Consulta: 8 de marzo de 2016].

Disponible en: http://web.ing.puc.cl/~mtorrest/downloads/pic/tutorial_pic.pdf

VAQUERO, Miguel. *Características del lenguaje HTML* [Web]. Web docente experimental, 2010.

[Consulta: 01 de diciembre 2015].

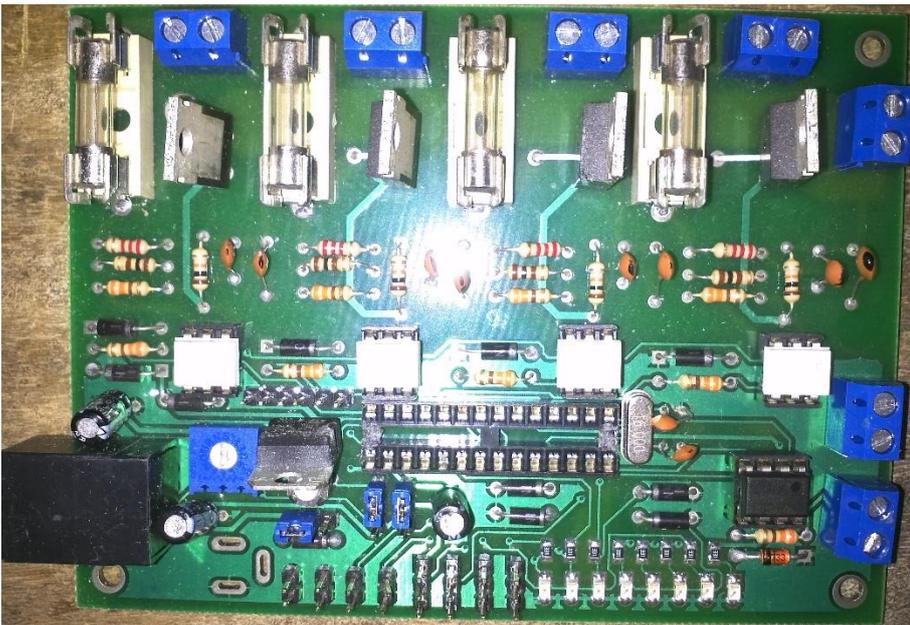
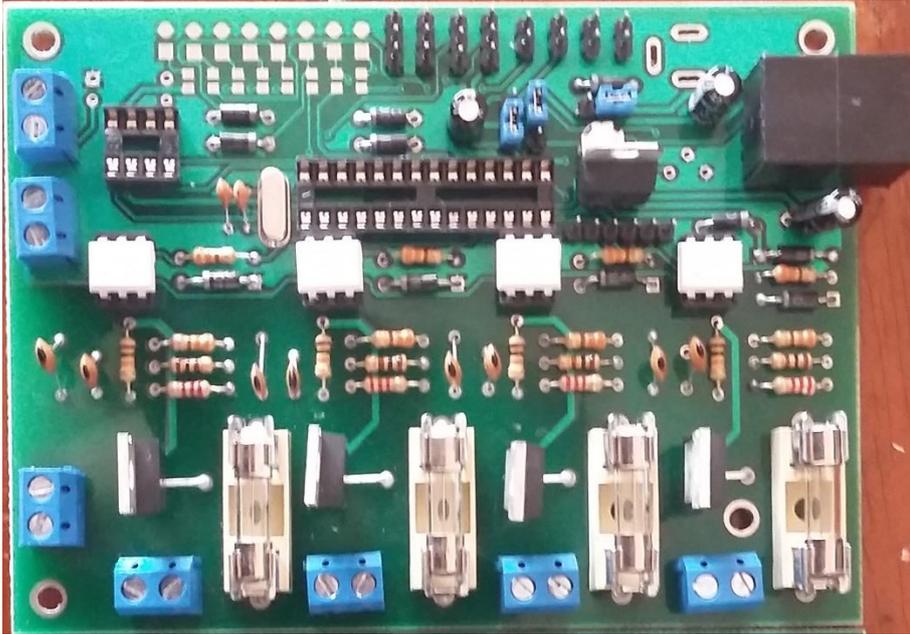
Disponible en: http://www.deciencias.net/disenoweb/elaborardw/paginas/intro_html.htm

ZAMORA, Marlen. *Estructura de las aplicaciones Web* [blog]. 2012.

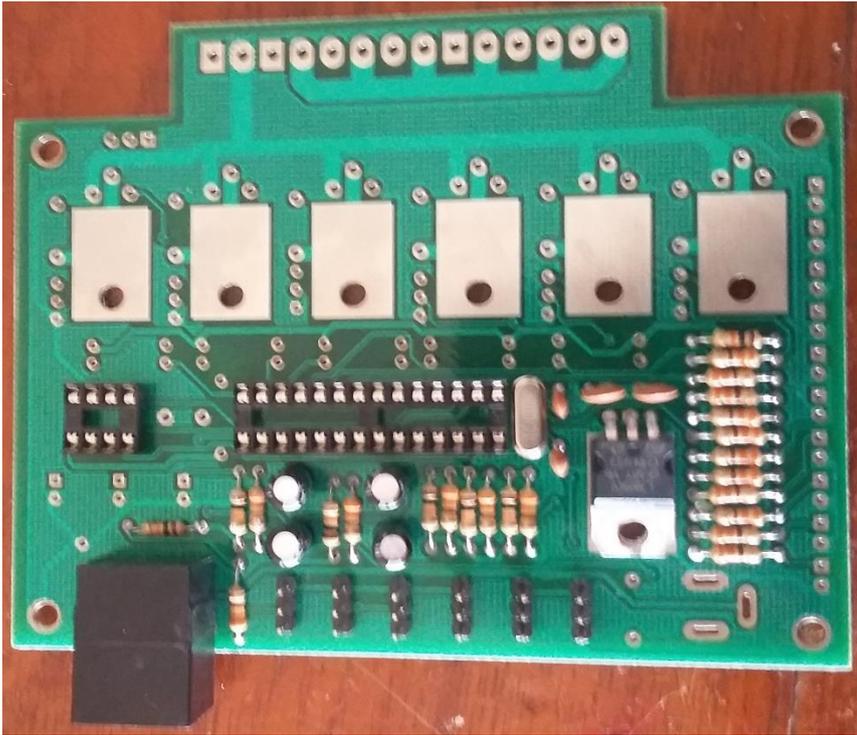
[Consulta: 20 de enero de 2016].

Disponible en: <http://unidad6aplicacionesweb1.blogspot.com/2012/10/estructura-de-las-aplicaciones-web.html>

ANEXOS



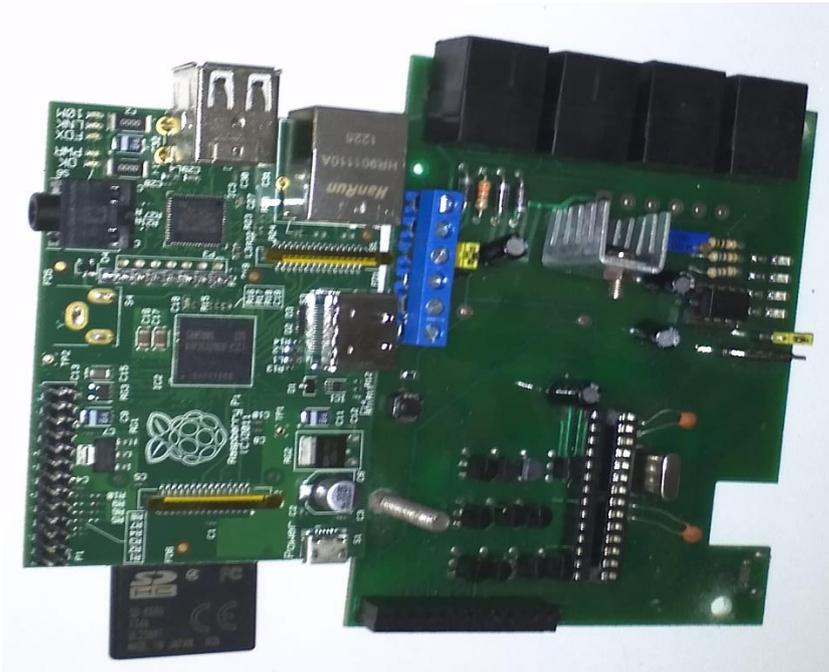
Anexo A: Placas de conmutación Ac elaboradas con TRIAC BT-A12 y opto acopladores Moc3011.



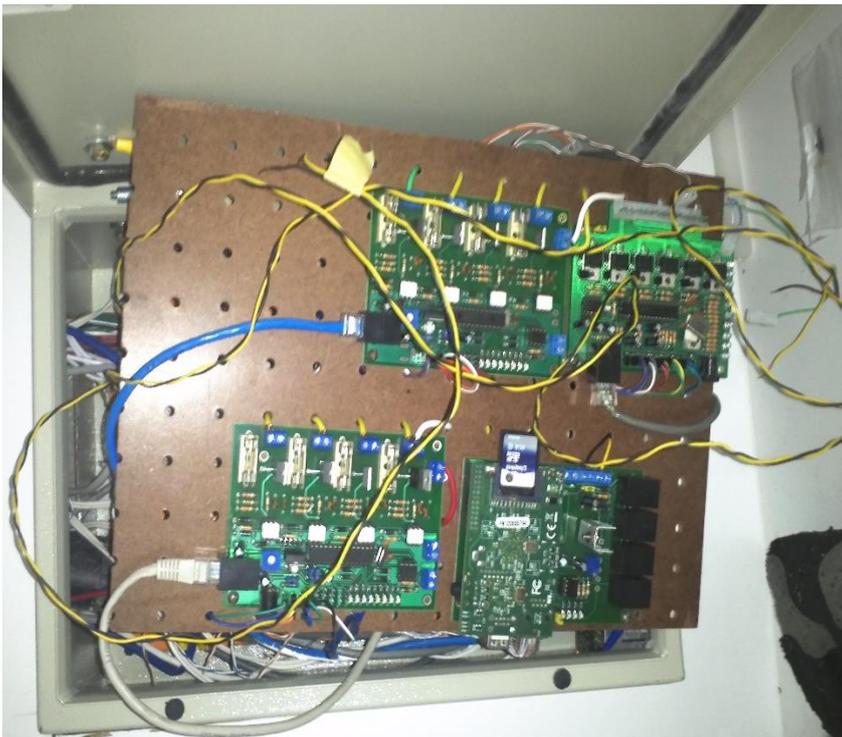
Anexo B: Placa de luces RGB con PWM amplificado para 12V con Mosfet IRF540.



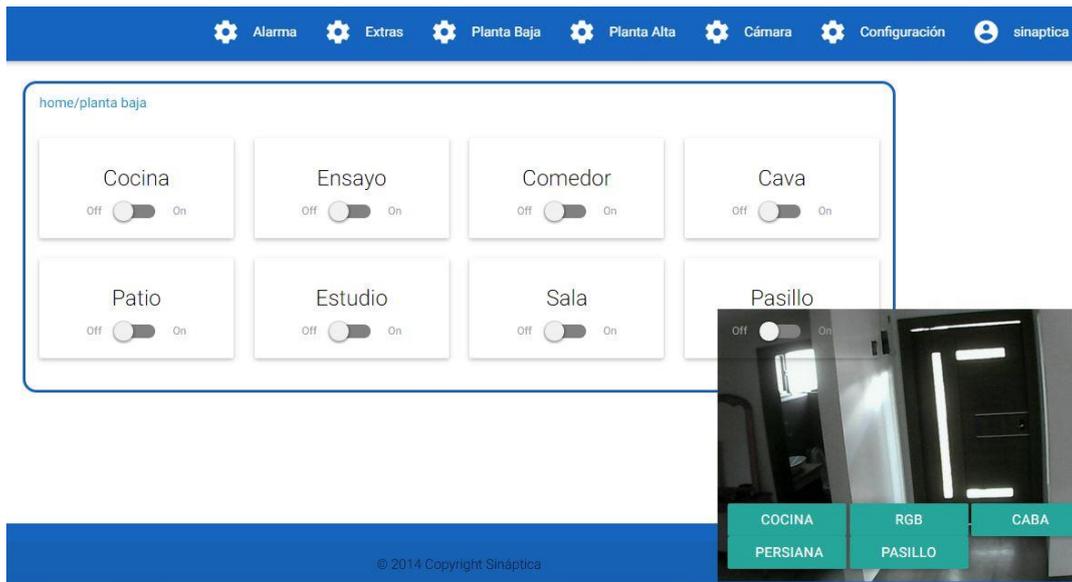
Anexo C: Motor utilizado para el movimiento de la persiana.



Anexo D: Placa master, y salida al internet mediante una Raspberry Pi modelo B.



Anexo E: Caja domótica donde están ubicados los módulos para el sistema domótico.



Anexo F: Aplicación Web junto a la cámara Ip con los botones de las luces Ac de la vivienda.

```

        <p class="range-field">
          <input type="range" id="dimmer21_1" min="1" max="100" onchange="callclient(21,1,this.value);" />
        </p>
      </div>
    </div>
  </div>
</div>
:div>
:div class="col s12 m6 l3">
  <div class="card white hoverable">
    <div class="card-content center-align">
      <span class="card-title black-text" lang="en">Module Three Luz 1</span>
      <!-- Switch -->
      <div class="switch">
        <label>
          Off
          <input type="checkbox" onchange="if(this.checked){callclient(21,2,100);}else{callclient(21,2,1)};">
          <span class="lever"></span>
          On
        </label>
      </div>
      <div>
        <p class="range-field">
          <input type="range" id="test5" min="1" max="100" onchange="callclient(21,2,this.value);" />
        </p>
      </div>
    </div>
  </div>
</div>
:div>
:div class="col s12 m6 l3">
  <div class="card white hoverable">
    <div class="card-content center-align">
      <span class="card-title black-text" lang="en">Module Three Luz 1</span>
      <!-- Switch -->
      <div class="switch">
        <label>
          Off
          <input type="checkbox" onchange="if(this.checked){callclient(21,3,100);}else{callclient(21,3,1)};">
          <span class="lever"></span>
        </label>
      </div>
    </div>
  </div>
</div>

```

Anexo G: Código para el envío de datos desde la aplicación para la luz dimmer, con dirección Modbus 21, el rango esta dado entre 1 y 100.

```

<div class="row">
  <div class="col s12 m6 l3">
    <div class="card white hoverable">
      <div class="card-content center-align">
        <span class="card-title black-text" lang="en">RGB</span>
        <br>
        <div class="rgb-button"><div id="31-1"></div></div>
      </div>
    </div>
  </div>
  <div class="col s12 m6 l3">
    <div class="card white hoverable">
      <div class="card-content center-align">
        <span class="card-title black-text" lang="en">Persiana</span>
        <!-- Switch -->
        <div class="switch">
          <label>
            Bajar
            <input type="checkbox" onchange="if(this.checked){callClient(35,131,2);}else{callClient(35,130,1);}">
            <span class="lever"></span>
            Subir
          </label>
        </div>
      </div>
    </div>
  </div>
  <div class="col s12 m6 l3">
    <div class="card white hoverable">
      <div class="card-content center-align">
        <span class="card-title black-text" lang="en">Estudio</span>
        <!-- Switch -->
        <div class="switch">
          <label>
            Off

```

Anexo H: Código para la persiana con dirección Modbus 35 y envío de datos para subir y bajar.

```

<
  <include 'login.php';
  >
  <?php
  <script type="text/javascript">
  <?php
  if ((substr($_SERVER['REMOTE_ADDR'],0,10) == "192.168.1.") || ($_SERVER['REMOTE_ADDR'] == "127.0.0.1")){
    echo 'var ip = "192.168.1.106";';
  }else{
    echo 'var ip = "ocalle.ddns.net";';
  }
  <?>
  $( document ).ready(function() {
    // Handler for .ready() called.
    $(".button-collapse").sideNav();
    $(".collapsible").collapsible({
      accordion : false // A setting that changes the collapsible behavior to expandable instead of the default
    });
    connectWebSocket(ip);
    $('#31-1').colpick({
      layout: 'rgbhex',
      color: 'e26600',
      onSubmit: function(hsb,hex,rgb,e1) {
        red=rgb['r']>254?254:rgb['r']<1?1:rgb['r'];
        green=rgb['g']>254?254:rgb['g']<1?1:rgb['g'];
        blue=rgb['b']>254?254:rgb['b']<1?1:rgb['b'];
        callClient(31, 1, blue);
        callClient(31, 2, green);
        callClient(31, 3, red);
        $(e1).css('background-color', '#'+hex);
        $(e1).colpickHide();
      }
    }).css('background-color', '#e26600');
  });

```

Anexo I: Código para las luces RGB con PWM, se envían 3 datos de R, G, B, respectivamente.

```

<div class="col s12 m6 l3">
  <div class="card white hoverable">
    <div class="card-content center-align">
      <span class="card-title black-text" lang="en">LED Cocina</span>
      <!-- Switch -->
      <div class="switch">
        <label>
          Off
          <input type="checkbox" onchange="if(this.checked){callClient(31,5,100);}else{callClient(31,5,1);}" id="31-1":
          <span class="lever"></span>
          On
        </label>
      </div>
      <p class="range-field">
        <input type="range" id="dimmer31-1" min="1" max="100" onchange="callClient(31,1,this.value);" />
      </p>
    </div>
  </div>
</div>
</div>
</div>
<?php
include 'paginas/footer.php';
}
else
{
include 'login.php';
}
?>
<script type="text/javascript">
<?php
if ((substr($_SERVER['REMOTE_ADDR'],0,10) == "192.168.1.") || ($_SERVER['REMOTE_ADDR'] == "127.0.0.1")){
echo 'var ip = "192.168.1.106"';
}
}

```

Anexo J: Continuación del código de luces RGB, con dirección Modbus 31 y rango entre 1 y 100 para el PWM.

```

      <span id="panic" class="card-title green-text">&nbsp;</span>
    </div>
    <div class="card-action center-align">
      <a onclick="callAlarm(43,99, '');" class="waves-effect white-text btn red darken-2">Pánico
    </div>
  </div>
</div>
</div>
</div>
<div id="sensors" class="col s12">
  <div class="row">
    <div class="col s12 m12 l12">
      <div class="card white hoverable">
        <div class="card-content center-align">
          <span class="card-title black-text">Mágneticos</span>
          <p hidden>
            <input class="with-gap" type="radio" id="mag0" disabled/>
            <label for="mag0">VESTIDOR</label>
          </p>
          <p hidden>
            <input class="with-gap" type="radio" id="mag1" disabled/>
            <label for="mag1">PUENTE</label>
          </p>
          <p>
            <input class="with-gap" type="radio" id="mag2" disabled/>
            <label for="mag2">Puerta de Cocina<br />Ventana Cocina</label>
          </p>
          <p>
            <input class="with-gap" type="radio" id="mag3" disabled/>
            <label for="mag3">Ventana Gabriel<br />Ventana de Star</label>
          </p>
          <p hidden>
            <input class="with-gap" type="radio" id="mag4" disabled/>
            <label for="mag4">PRINCIPAL</label>
          </p>

```

Anexo K: Código para la alarma y nombre de las ubicaciones donde están instalados los sensores, a su vez configuración de la opción pánico, con dirección Modbus 43.

```

<div class="col s12 m6 l3">
  <div class="card white hoverable">
    <div class="card-content center-align">
      <span class="card-title black-text" lang="en">Baño Social</span>
      <!-- Switch -->
      <div class="switch">
        <label>
          Off
          <input type="checkbox" onchange="if(this.checked){callClient(13,1,2);}else{callClient(13,1,1);}" id="13-1">
          <span class="lever"></span>
          On
        </label>
      </div>
    </div>
  </div>
</div>
<div class="col s12 m6 l3">
  <div class="card white hoverable">
    <div class="card-content center-align">
      <span class="card-title black-text" lang="en">Gradas</span>
      <!-- Switch -->
      <div class="switch">
        <label>
          Off
          <input type="checkbox" onchange="if(this.checked){callClient(13,2,2);}else{callClient(13,2,1);}" id="13-2">
          <span class="lever"></span>
          On
        </label>
      </div>
    </div>
  </div>
</div>
...

```

Anexo L: Código para las luces AC On/Off, con dirección Modbus 13.

```

<div class="col s12 m6 l3">
  <div class="card white hoverable">
    <div class="card-content center-align">
      <span class="card-title black-text" lang="en">Cocina</span>
      <!-- Switch -->
      <div class="switch">
        <label>
          Off
          <input type="checkbox" onchange="if(this.checked){callClient(10,1,2);}else{callClient(10,1,1);}" id="10-1">
          <span class="lever"></span>
          On
        </label>
      </div>
    </div>
  </div>
</div>
<div class="col s12 m6 l3">
  <div class="card white hoverable">
    <div class="card-content center-align">
      <span class="card-title black-text" lang="en">Ensayo</span>
      <!-- Switch -->
      <div class="switch">
        <label>
          Off
          <input type="checkbox" onchange="if(this.checked){callClient(10,2,2);}else{callClient(10,2,1);}" id="10-2">
          <span class="lever"></span>
          On
        </label>
      </div>
    </div>
  </div>
</div>

```

Anexo M: Código para las demás luces AC, pero con dirección Modbus 10.

```

<div class="valign-wrapper">
  <div class="row center">
    <form class="col s12" action="" method="POST">
      <div class="row">
        <div class="input-field col s12">
          <!--<i class="material-icons prefix">account_circle</i-->
          <input id="username" type="text" name="username" class="validate" required>
          <label for="username">Username</label>
        </div>
      </div>
      <div class="row">
        <div class="input-field col s12">
          <!--<i class="material-icons prefix">security</i-->
          <input id="password" type="password" name="password" class="validate" required>
          <label for="password">Password</label>
        </div>
      </div>
      <div class="row">
        <input type="checkbox" class="filled-in" id="remember" name="remember" />
        <label for="remember">Remember</label>
      </div>
      <div class="row">
        <button class="btn waves-effect waves-light blue" type="submit" name="action">Login
          <!--<i class="material-icons">send</i-->
        </button>
      </div>
      <div class="row">
        <a href="pages/forgotPassword.php">Forgot Password</a>
      </div>
    </form>
  </div>
</div>
</main>

```

Anexo N: Código para ingresar a la aplicación con nombre de usuario y contraseña.

```

<br>
<span class="card-title black-text">Nuevo Usuario</span>
<form>
  <div class="row">
    <div class="input-field col s12 m6 l6 offset-m3 offset-l3">
      <input name="username" id="username" type="text" class="validate" autofocus required>
      <label for="username">Username</label>
    </div>
  </div>
  <div class="row">
    <div class="input-field col s12 m6 l6">
      <input id="first_name" type="text" class="validate" required>
      <label for="first_name">Nombre</label>
    </div>
    <div class="input-field col s12 m6 l6">
      <input id="last_name" type="text" class="validate" required>
      <label for="last_name">Apellido</label>
    </div>
  </div>
  <div class="row">
    <div class="input-field col s12 m6 l6">
      <input id="password" type="password" class="validate" required>
      <label for="password">Password</label>
    </div>
    <div class="input-field col s12 m6 l6">
      <input id="password_again" type="password" class="validate" required>
      <label for="password_again">Repeat Password</label>
    </div>
  </div>
  <div class="row">
    <div class="input-field col s12 m6 l6">
      <input id="email" type="email" class="validate" required>
      <label for="email">Email</label>
    </div>
  </div>

```

Anexo Ñ: Código para crear un nuevo usuario y pueda logarse en la aplicación.

```

int conterror=0,error=0;
char receivedData[9];           // buffer for receving/sending messages
char sendData[9];              // buffer for receving/sending messages
char i,j;
char RS232, byte_read; //variable usada para la comunicación RS232C
char read_action[5]={255,255,255,255,255};
sbit  rs485_rxtx_pin at RC2_bit;           // set transcieve pin
sbit  rs485_rxtx_pin_direction at TRISC2_bit; // set transcieve pin direction
//-----
//Nuevas variables en array
const int SIZE = 4;
const int SIZE_MOD = 8;

int h,k,l,m,o;
//int modulos[SIZE_MOD]={21};
int modulos[SIZE_MOD]={10,13,21,31,35,43};
//int modulos[SIZE_MOD]={41};
int info_modulos[SIZE_MOD]={0,0,0,0,0,0,0};
int statusMod[SIZE_MOD][SIZE];
int error_mod[SIZE_MOD]={0,0,0,0,0,0,0};

```

Anexo O: Código de la placa master donde se declara las direcciones Modbus de los diferentes esclavos utilizados en el sistema.

```

//-----
//Inicialización del modulo RS485 en modo SLAVE ADDRESS 44
UART1_Init(9600);           // initialize UART1 module
Delay_ms(100);
RS485Slave_Init(35);       // Intialize MCU as slave, address 28
Delay_ms(100);

receivedData[0] = 0;
receivedData[1] = 0;
receivedData[2] = 0;
receivedData[4] = 0;           // ensure that message received flag is 0
receivedData[5] = 0;           // ensure that error flag is 0
receivedData[6] = 0;

sendData[0] = 0;
sendData[1] = 0;
sendData[2] = 0;
sendData[4] = 0;           // ensure that message received flag is 0
sendData[5] = 0;           // ensure that error flag is 0
sendData[6] = 0;

```

Anexo P: Comunicación UART con dirección Modbus esclavo 35 para la persiana.

```

//-----INT RS485
UART1_Init(9600);          // initialize UART1 module
Delay_ms(100);
RS485Slave_Init(21);      // Intialize MCU as slave, address 10-11-12-13

received[4] = 0;          // ensure that message received flag is 0
received[5] = 0;          // ensure that message received flag is 0
received[6] = 0;          // ensure that error flag is 0

PORTB.RB7 = 1; PORTB.RB6 = 1; PORTB.RB5 = 1; PORTB.RB4 = 1;
PORTB.RB2=1;
PORTA.RA5=1;
delay_ms(500);
PORTB.RB7 = 0; PORTB.RB6 = 0; PORTB.RB5 = 0; PORTB.RB4 = 0;
PORTB.RB2=0;
PORTA.RA5=0;

```

Anexo Q: Comunicación UART con dirección Modbus esclavo 21 para la luz dimmer AC.

```

//-----INT RS485
UART1_Init(9600);          // initialize UART1 module
Delay_ms(100);
RS485Slave_Init(31);      // Intialize MCU as slave, address 34

received[4] = 0;          // ensure that message received flag is 0
received[5] = 0;          // ensure that message received flag is 0
received[6] = 0;          // ensure that error flag is 0

RCIE_bit = 1;             // enable interrupt on UART1 receive
TXIE_bit = 0;             // disable interrupt on UART1 transmit
PEIE_bit = 1;             // enable peripheral interrupts
GIE_bit = 1;              // enable all interrupts

PORTB=0x00; //Se encienden todos lo RGB
PORTC=0xFF; //Se encienden los leds indicadores
delay_ms(1000);
PORTB=0xFF; //Se apagan todos los RGB
PORTC=0x00; //Se apagan los indicadores
while(1)

```

Anexo R: Comunicación UART con dirección Modbus esclavo 31 para las luces RGB.

```

UART1_Init(9600);           // initialize UART1 module
Delay_ms(100);
RS485Slave_Init(10);       // Intialize MCU as slave, address 34

received[4] = 0;           // ensure that message received flag is 0
received[5] = 0;           // ensure that message received flag is 0
received[6] = 0;           // ensure that error flag is 0

PORTB.RB7 = 1; PORTB.RB6 = 1; PORTB.RB5 = 1; PORTB.RB4 = 1;
PORTB.RB2=1;
PORTB.RB3=1;
PORTA.RA5=1;
delay_ms(500);
PORTB.RB7 = 0; PORTB.RB6 = 0; PORTB.RB5 = 0; PORTB.RB4 = 0;
PORTB.RB2=0;
PORTB.RB3=0;
PORTA.RA5=0;

```

Anexo S: Comunicación UART con dirección Modbus esclavo 10 para las luces On/Off Ac del sistema.

```

UART1_Init(9600);           // initialize UART1 module
Delay_ms(100);
RS485Slave_Init(43);       // Intialize MCU as slave, address 30
sendData[0] = 0;
sendData[1] = 0;
sendData[2] = 0;
sendData[4] = 0;           // ensure that message received flag is 0
sendData[5] = 0;           // ensure that error flag is 0
sendData[6] = 0;
//PORTC.RC1=0;
return;
}
void main() {
ADCON1 = 0x0F;             //all inputs are digital
CMCON = 0x07;
//-----
// INTCON.IPEN=1;
RCIE_bit = 1;             // enable interrupt on UART1 receive
TXIE_bit = 0;             // disable interrupt on UART1 transmit
PEIE_bit = 1;             // enable peripheral interrupts

```

Anexo T: Comunicación UART con dirección Modbus esclavo 43 para la alarma del sistema.