

UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería

Departamento de Ingeniería Mecatrónica



**Control a distancia del sistema eléctrico en instalaciones
domésticas**

Trabajo de graduación presentado por

José Andrés Quintanilla López

**para optar al grado académico de Licenciado en Ingeniería
Mecatrónica.**

Guatemala

2014

Control a distancia del sistema eléctrico en instalaciones domésticas

UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería

Departamento de Ingeniería Mecatrónica

**Control a distancia del sistema eléctrico en instalaciones
domésticas**

Trabajo de graduación presentado por

José Andrés Quintanilla López

**para optar al grado académico de Licenciado en Ingeniería
Mecatrónica.**

Guatemala

2014

Vo. Bo.:

(f) 

Ing. Gustavo Prera
Asesor


Tribunal Examinador:

(f) 

Ing. Gustavo Prera
Ing. Electrónico

(f) 

Ing. Roberto Delgado
Director Ing. Mecatrónica

(f) 

Ing. Carlos Esquit
Director Ing. Electrónica

Fecha de aprobación: Guatemala 7 Enero de 2014

CONTENIDO

	Página
LISTA DE ILUSTRACIONES.....	VIII
LISTA DE TABLAS.....	XI
RESUMEN.....	XII
I. INTRODUCCIÓN	1
II. OBJETIVOS.....	4
III. JUSTIFICACIÓN.....	5
IV. MARCO TEÓRICO	6
1. Raspberry Pi.....	6
2. Interfaz Web.....	8
3. Protocolo X10.....	9
4. Micro controlador PIC16F887.....	13
5. Dimmer.....	15
6. Transistor Amplificador de señales pequeñas.....	17
V. METODOLOGÍA	20
1. Comunicación de Micro controladores	20
a. Detector de cruce por cero..	21
b. Generador de pulsos a 120KHz.....	23
c. Detector de pulsos 120KHz..	30

2.	Recepción y envío de comandos	38
3.	Servidor Web.....	40
4.	Control de Intensidad, encendido y apagado	41
5.	PCB.....	45
VI.	RESULTADOS.....	47
1.	Comunicación de micro controladores	47
a.	Detector de cruce por cero.	50
b.	Generador de pulsos a 120KHz.....	51
c.	Detector de pulsos a 120KHz... ..	53
2.	Distancia efectiva de comunicación	57
3.	Recepción y ejecución de comandos	62
4.	Website	66
VII.	CONCLUSIONES	69
VIII.	RECOMENDACIONES	70
IX.	BIBLIOGRAFÍA	71
X.	APÉNDICE.....	74
1.	Código de micro controlador central.....	74
2.	Código de micro controlador receptor	82
3.	Código Website	89
XI.	GLOSARIO	99

LISTA DE ILUSTRACIONES

	Página
1. ESTRUCTURA GENERAL DEL SISTEMA	3
2. COMPUTADORA RASPBERRY PI MODELO B	6
3. GPIO DEL RASPBERRY PI	7
4. FUNCIONAMIENTO DEL PROTOCOLO X-10	9
5. DIAGRAMA DE PINES DEL PIC16F887	14
6. SÍMBOLO ESQUEMÁTICO DE TRIAC	15
7. FUNCIONAMIENTO DEL TRIAC.....	16
8. DIAGRAMA DE AMPLIFICADOR CE.....	17
9. CIRCUITO DE 4 TERMINALES PARA TRANSISTORES	18
10. CIRCUITO EQUIVALENTE PARÁMETROS-H EN AMPLIFICADOR CE.	19
11. ESTRUCTURA INTERNA DE PINES I/O	21
12. CIRCUITO DETECTOR CRUCE POR CERO.....	22
13. TRANSISTOR INVERSOR	23
14. DISEÑO PRE-ELIMINAR INYECTOR DE PULSOS 120KHZ.....	26
15. DISEÑO INYECTOR DE PULSOS 120KHZ [3].....	26
16. ALGORITMO PARA ENVÍO DE SEÑALES.....	28
17. DISEÑO PRE-ELIMINAR CIRCUITO DETECTOR DE PULSOS 120KHZ.....	30

18. CONFIGURACIÓN DE AMPLIFICADOR CE UTILIZADO.	33
19. GANANCIA VRS CORRIENTE COLECTOR 2N3904	34
20. DISEÑO CIRCUITO DETECTOR DE PULSOS 120KHZ	36
21. ALGORITMO PARA RECEPCIÓN DE SEÑALES	37
22. ESTRUCTURA FRAMEWORK WEBIOPI	41
23. CONEXIÓN DE TRIAC	44
24. NOMENCLATURA PCB	45
25. ESQUEMÁTICO CIRCUITO EMISOR	47
26. DIAGRAMA PCB EMISOR	48
27. DIAGRAMA PCB RECEPTOR	48
28. ESQUEMÁTICO CIRCUITO RECEPTOR	49
29. MÓDULOS CONSTRUIDOS	49
30. INTERRUPCIÓN GENERADA POR EL CRUCE POR CERO.	50
31. PULSOS DE 120 KHZ GENERADOS EN EL CRUCE POR CERO	51
32. DURACIÓN DE PULSO 120KHZ	52
33. MEDICIONES CIRCUITO EMISOR	52
34. COMANDO COMPLETO GENERADO	53
35. SALIDA DEL FILTRO PASA ALTA	54
36. SALIDA OBTENIDA DEL AMPLIFICADOR TRANSISTORIZADO	54
37. SALIDA CIRCUITO DETECTOR ENVOLVENTE	55
38. SEÑALES COMPARADAS	56

39. SEÑAL FINAL.....	56
40. DISTANCIA VRS PORCENTAJE DE COMUNICACIÓN BAJO CONDICIONES CONTROLADAS.....	57
41. MEDICIONES BAJO CONDICIONES CONTROLADAS	58
42. DISTANCIA VRS PORCENTAJE DE COMUNICACIÓN BAJO CONDICIONES COTIDIANAS EN RESIDENCIAS.....	59
43. MEDICIONES BAJO CONDICIONES COTIDIANAS EN RESIDENCIAS	60
44. CONEXIONES PARA SIMULACIÓN COTIDIANA DE RESIDENCIAS	61
45. DISTANCIA VRS PORCENTAJE DE COMUNICACIÓN PRUEBAS EN AULA	62
46. CONEXIONES PARA PRUEBAS EN AULA.....	62
47. TIEMPO DE ESPERA PARA LECTURA.....	63
48. INTENSIDAD LUMÍNICA MÁXIMA	64
49. INTENSIDAD LUMÍNICA AL 75%.....	64
50. INTENSIDAD LUMÍNICA AL 50%.....	65
51. INTENSIDAD LUMÍNICA AL 25%.....	65
52. PUNTOS DE OPERACIÓN DE TRIAC.....	66
53. PANTALLA PRINCIPAL DE INTERFAZ.....	67
54. PANTALLA DE INFORMACIÓN DE WEBSITE	68

LISTA DE TABLAS

	Página
1. CÓDIGO DE CASA	10
2. CÓDIGOS DE DISPOSITIVOS X10.	11
3. EJEMPLO ENVÍO DE COMANDOS X10	12
4. CARACTERÍSTICAS PIC16F887.....	13
5. FORMULAS PARA RESOLUCIÓN ANALÍTICA DE AMPLIFICADOR CE.	19
6. CÓDIGOS UTILIZADOS	39
7. SECUENCIA DE ENVÍO.	39
8. SECUENCIA DE ENVÍO COMANDOS PROTOCOLO DISEÑADO	40
9. CALIBRE DE COBRE PARA PCB.....	46
10. DISTANCIA VRS PORCENTAJE COMUNICACIÓN BAJO CONDICIONES CONTROLADAS.....	57
11. DISTANCIA VRS PORCENTAJE COMUNICACIÓN BAJO CONDICIONES COTIDIANAS EN RESIDENCIAS.....	59
12. DISTANCIA VRS PORCENTAJE COMUNICACIÓN PRUEBAS EN AULA	61

RESUMEN

En las últimas generaciones de sistemas de automatización, las aplicaciones pueden comunicarse transmitiendo información a través del cableado eléctrico ya existente. Por lo tanto no existe la necesidad de realizar instalación de nuevos conductores de información, mientras exista un cableado eléctrico con alimentación AC será posible la comunicación. Esto genera que los costos de instalación sean minimizados, ya que el único requisito es que exista instalación eléctrica de 120VAC a 60 Hz. A esta aplicación se le conoce como Power Line Communication (PLC).

El presente trabajo describe el desarrollo de un sistema de automatización de tomacorrientes y luminarias, mediante la comunicación de micro controladores utilizando el cableado eléctrico AC como bus de datos, es decir PLC y a su vez permitir ser controlados desde computadores conectados al LAN (Local Area Network) en donde se encuentran los dispositivos.

Para realizar la comunicación se implementaron circuitos receptores e inyectores de señales en el cableado de alimentación AC, en los cuales sus señales resultantes son acopladas a los micros controladores, y el control desde la LAN se realiza utilizando la interfaz web programada que se encuentra en la computadora Raspberry Pi la cual se utiliza como servidor, indicando al micro controlador central que comando se debe enviar.

Al tener la comunicación establecida Raspberry Pi- Micro controlador- Micro controlador se logra la automatización con la programación adecuada de cada elemento.

I. INTRODUCCIÓN

Se le conoce Domótica al conjunto de sistemas que automatizan las diferentes instalaciones de una residencia. Domótica o automatización residencial puede incluir sistemas centrales de control HVAC (calefacción, ventilación y aire acondicionado), aplicaciones de seguridad, control de iluminación para conveniencia y eficiencia de energía. La domótica tiene como propósito facilitar el control integral de las residencias desde un solo punto. También tiene aplicación en notificaciones de eventos no deseados como aumento de temperatura, gases, humedad, accesos no autorizados, etc. estas notificaciones normalmente proveen una interfaz remota por medio de telefonía o internet.^[1]

Un ejemplo de aplicación de notificaciones puede ser cuando el detector de gases es activado, se activan las luminarias en modo intermitente para alertar a los ocupantes y desconectar los equipos eléctricos para evitar el riesgo de fuego eléctrico. En el área de HVAC, este se coloca en modo de ahorro de energía cuando no hay ocupantes en la residencia y se activa momentos antes de que regresen los ocupantes. En el área de iluminación se puede ahorrar energía cuando existen sensores de intensidad lumínica y ultrasónica, ya que la iluminación se activa únicamente cuando estos sensores lo indican.

Bajo esta arquitectura de redes Power Line Communications han surgido varios protocolos, tales como UPB(Universal Powerline Bus), Insteon, ZigBee, Z-Wave y X-10 por mencionar los mas utilizados.

X-10 es el protocolo más antiguo en este sector. Fue desarrollada entre 1976 y 1978 en Escocia y la transmisión de sus señales es por medio de las líneas de alimentación, aunque esto más que cerrarle el mercado residencial, la

ha vuelto líder en él ya que no es necesario crear una nueva infraestructura en las residencias las cuales no fueron diseñadas con fines de automatización. Este protocolo también tiene sus limitaciones, trabaja a 60bps y tiene un conjunto de comandos limitados. Sin embargo el protocolo para implementarlo es abierto, razón por la cual se eligió dicho protocolo como base de funcionamiento para el presente trabajo. Esto ha permitido el surgimiento de nuevas empresas que desarrollan productos con este protocolo. [1]

Power Line Communication, también conocido como PLC, utiliza el cableado existente de distribución de voltaje para enviar datos. Existen 2 aplicaciones importantes para PLC, uno es usado para introducir el acceso a internet a las residencias en conjunto con la señal de telefonía y el otro se utiliza para crear redes de comunicación. Este trabajo se enfoca en el uso del cableado para crear redes de comunicación.

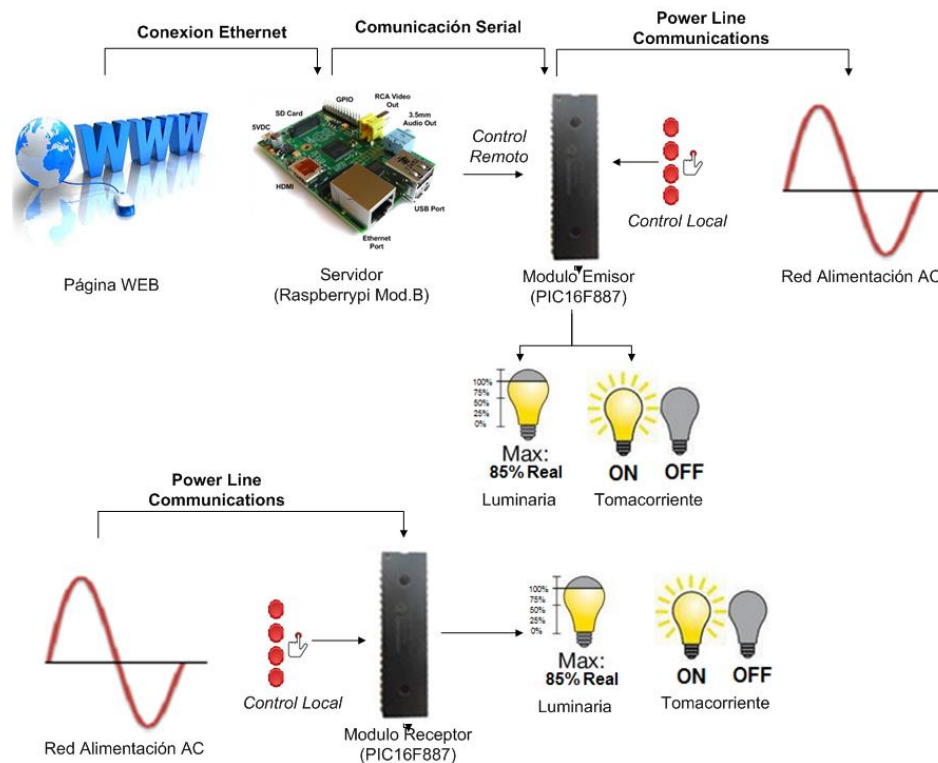
Dispositivos wireless han abarcado la mayor parte de las aplicaciones de comunicación, pero estos son afectados por las distancias y obstrucciones como paredes. Uno de los mayores beneficios de Power Line Communication, es la habilidad que mientras exista un tomacorriente en la instalación con alimentación AC, existe un punto de conexión a la red.

La idea detrás de X-10 es poder utilizar la red de alimentación para controlar equipo eléctrico existente, sin necesidad de modificarlos para su automatización. X-10 es un protocolo de comunicación diseñado para enviar señales sobre el cableado de 120VAC. Este protocolo utiliza pulsos de 120Khz de frecuencia, los cuales están sincronizados por el cruce por cero de la señal de 120VAC.

Además existe la posibilidad de controlar los dispositivos desde internet, pero para esto es necesario un módulo servidor, lo cual normalmente es un computador. El uso del computador para este fin no es práctico, ya que es necesario que se mantenga encendida todo el tiempo aunque no se esté utilizando, provocando desaprovechamiento de energía, y uno de los objetivos de la domótica es el ahorro de energía. Para minimizar el desaprovechamiento de energía por parte del servidor, se utiliza el computador Raspberry PI, la cual utiliza únicamente 700mA o 3.5Watts a 5 VDC, en comparación a las computadoras que utilizan en promedio 3.5A o 66.5Watts.

La siguiente ilustración muestra una representación gráfica general de como el sistema interconecta sus componentes.

Ilustración 1: Estructura general del sistema



II. OBJETIVOS

1. General

El objetivo general de este trabajo es desarrollar un sistema de control de tomacorrientes y luminarias.

2. Específicos:

- a. Implementar la estructura Power Line Communications (PLC) en micro controladores.
- b. Implementar control de intensidad en las luminarias por medio de PLC.
- c. Utilizar el computador Raspberry Pi como servidor para utilizar módulos portátiles.
- d. Controlar el sistema desde una interfaz web.
- e. Lograr una distancia efectiva de comunicación entre micro controladores mayores a 15 metros.

III. JUSTIFICACIÓN

En la actualidad existen instalaciones eléctricas que han sido diseñadas y construidas sin la posibilidad de ser automatizadas, debido a que estas no tienen la infra estructura necesaria para su automatización inmediata. Para lograr automatizar la mayoría de instalaciones eléctricas existentes, es necesario la remodelación, adaptación y modificaciones en la estructura de obra civil.

Debido al problema existente, surge la motivación de realizar e implementar una solución de control inmediato sin necesidad de realizar modificaciones a la infraestructura en el que se encuentra la instalación eléctrica.

Otro aspecto importante en automatización de sistemas, es el alto costo de realizar la instalación debido a los equipos y cableados necesarios para su funcionamiento. Es por esto que se decide utilizar Power Line Communications para eliminar por completo la necesidad de instalar nuevos cableados de comunicación.

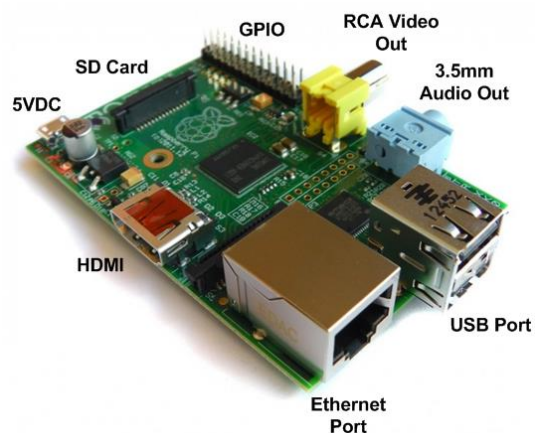
IV. MARCO TEÓRICO

1. Raspberry Pi

El Raspberry Pi modelo B es una computadora con dimensiones de 85.60mm x 56mm x 21mm, con un peso de 45g, posee el procesador Bradcom BCM2835 ARM11 a una velocidad de 700Mhz, 512MB RAM, con capacidad de reproducción de video 1080p con HDMI, 2 puertos USB, 3.5mm Audio Jack, 10/100Mb Ethernet port, SD, MMC, SDIO Flash memory card slot, conectores de 26-pins, con un consumo de potencia de 3.5Watts (Ptak, 2013).

La idea de crear esta mini computadora portátil nació en el 2006, cuando Eben Upton y sus colegas Rob Mullins, Jack Lang y Alan Mycroftn de la universidad de Cambridge, se preocuparon del decaimiento en las habilidades que presentaban los alumnos año a año cuando aplicaban al área de ciencias de la computación. En el año 2011 iniciaron la producción en masa de su prototipo al que llamaron Raspberry Pi. La Ilustración 2 muestra el resultado de la idea de crear una computadora portátil más accesible en precio y consumo de energía (Ptak, 2013).

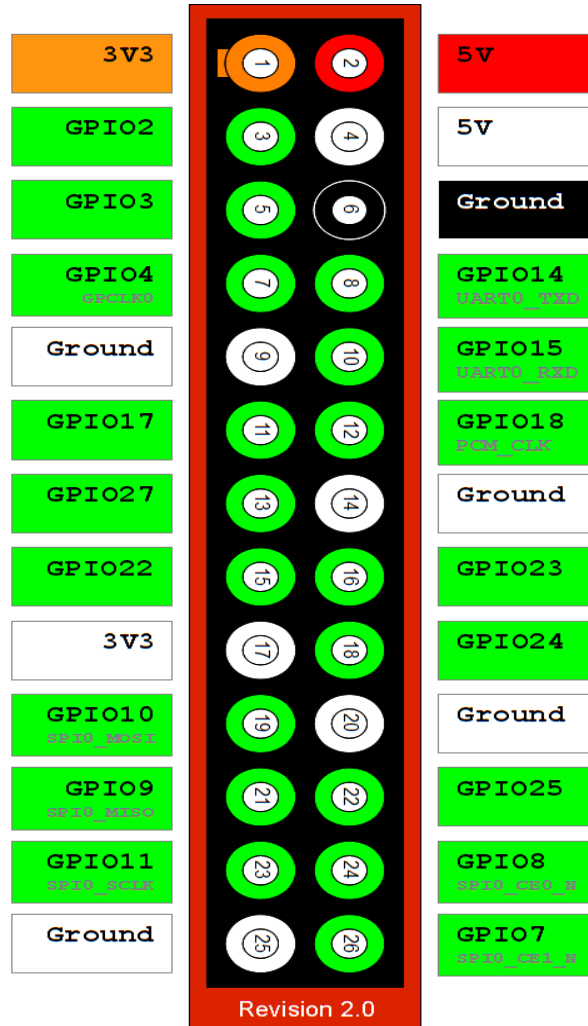
Ilustración 2. Computadora Raspberry Pi modelo B



El Raspberry Pi utiliza el sistema operativo Debian por default, pero es posible usar cualquier otra versión de ARM Linux. El sistema operativo se encuentra en la tarjeta SD.

Esta computadora posee GPIO, pines de propósito general de entrada y salida, por sus siglas en inglés, los cuales se utilizan para interactuar con otros circuitos o computadoras. La Ilustración 3 muestra el posicionamiento de dichos pines e indica las funcionalidades que poseen.

Ilustración 3: GPIO del Raspberry Pi



2. Interfaz Web

El lenguaje de publicación utilizado por la World Wide Web es HTML (Hyper Text Markup Language). HTML es un lenguaje para describir la estructura de páginas Web y le da al autor la posibilidad de publicar documentos en línea con encabezados, textos, tablas, listas, fotos, etc.

HTML y CSS(Cascading Style Sheets) son dos núcleos de tecnología para la construcción de páginas web. HTML provee la estructura de la página y CSS se encarga de lo visual y formato. CSS es un lenguaje para la descripción de presentaciones de páginas web, incluyendo los colores, formato y fuente. Permite adaptar la presentación para diferentes tipos de dispositivos.

Un documento HTML debe comenzar con la declaración de tipo de documento `<html>`. Luego sigue la etiqueta `<head></head>` que contiene información sobre el documento tales como los títulos y palabras clave, seguido de `<body></body>` con el contenido del documento organizado por inline-elements y block-level elements y al final se debe cerrar la marca `</html>` (W3C Org.,2013).

A continuación se presenta un ejemplo del formato del lenguaje HTML.

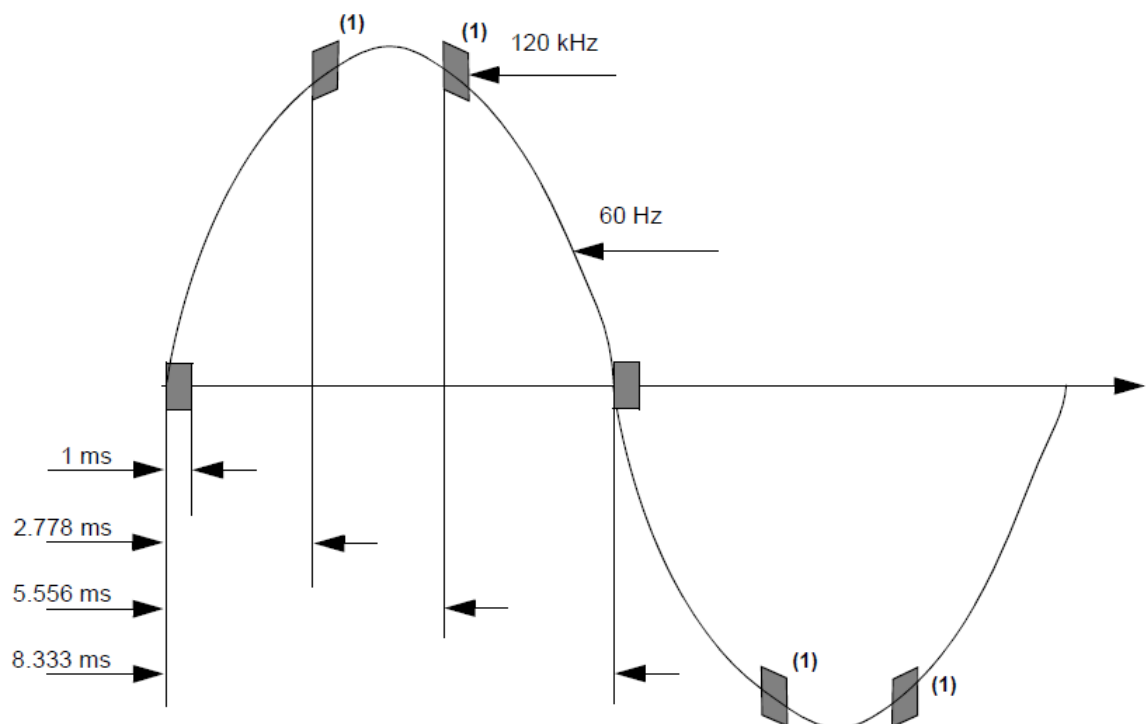
```
<html>
  <head>
<title> Título del documento </title>
  </head>
  <body>
Contenido del documento </body>
</html>
```

3. Protocolo X10

La transmisión y recepción de datos en el protocolo del X-10, están sincronizados con el cruce por cero de la señal de alimentación 120AC. Los dispositivos X-10 utilizan el cruce por cero para sincronizar el instante en que deben transmitir o recibir información (Burroughs, 2010).

El '1' binario es representado por una señal cuadrada con frecuencia 120Khz con duración de 1ms. Un '0' binario se considera como la ausencia de dicha señal cuadrada. Estas señales de 120kHz pueden estar desfasadas con el cruce por cero. La Ilustración 4 muestra el funcionamiento del protocolo.

Ilustración 4: Funcionamiento del protocolo X-10



Un mensaje completo de X-10 está compuesto por un código de inicio(1110), seguido por un código de casa, seguido de un código dispositivo y por último el código de comando. El código de inicio siempre es constante y no cambia, los posibles códigos de casa y dispositivos se muestran en las siguientes tablas (Burroughs, 2010).

Tabla 1: Código de casa

House Addresses	House Codes			
	H1	H2	H4	H8
A	0	1	1	0
B	1	1	1	0
C	0	0	1	0
D	1	0	1	0
E	0	0	0	1
F	1	0	0	1
G	0	1	0	1
H	1	1	0	1
I	0	1	1	1
J	1	1	1	1
K	0	0	1	1
L	1	0	1	1
M	0	0	0	0
N	1	0	0	0
O	0	1	0	0
P	1	1	0	0

Para transmitir los códigos de casa y dispositivo se utiliza un formato específico, se necesitan 2 cruces por cero para transmitir un bit con su bit complementario, es decir un '0' es representado '01' y para representar '1' se usa '10'. Por ejemplo para transmitir el código de casa P(1100) que se encuentra en la tabla 1, el código a transmitir será en formato complementario '10100101'.

Tabla 2: Códigos de dispositivos X10.

Unit Addresses	Key Codes					
	D1	D2	D4	D8	D16	
1	0	1	1	0	0	
2	1	1	1	0	0	
3	0	0	1	0	0	
4	1	0	1	0	0	
5	0	0	0	1	0	
6	1	0	0	1	0	
7	0	1	0	1	0	
8	1	1	0	1	0	
9	0	1	1	1	0	
10	1	1	1	1	0	
11	0	0	1	1	0	
12	1	0	1	1	0	
13	0	0	0	0	0	
14	1	0	0	0	0	
15	0	1	0	0	0	
16	1	1	0	0	0	
Function Codes	All Units Off	0	0	0	0	1
	All Units On	0	0	0	1	1
	On	0	0	1	0	1
	Off	0	0	1	1	1
	Dim	0	1	0	0	1
	Bright	0	1	0	1	1
	All Lights Off	0	1	1	0	1
	Extended Code	0	1	1	1	1
	Hail Request	1	0	0	0	1
	Hail Acknowledge	1	0	0	1	1
	Pre-set Dim	1	0	1	X	1
	Extended Code (Analog)	1	1	0	0	1
	Status = On	1	1	0	1	1
	Status = Off	1	1	1	0	1
	Status Request	1	1	1	1	1

En los códigos de dispositivos, los cuales tienen 5 bits de longitud, se tomará 10 bits o cruces por cero para enviar un código de dispositivo en formato complementario. Debido a que los códigos de dispositivos terminan en '0' y los códigos de función terminan en 1, el último bit es considerado como un sufijo para diferenciar los códigos de funciones de los códigos de dispositivos (Burroughs, 2010).

Un bloque completo de datos consiste del código de inicio, código de casa, código de dispositivo y comando, el primer bloque envía el identificador de dispositivo y el segundo bloque envía la acción a realizar. Cada bloque es enviado 2 veces, con 3 ciclos completos de AC o 6 cruces por cero de separación entre cada envío de bloque y por último se esperan 6 cruces por cero para poder enviar un comando nuevo. A continuación se presenta un ejemplo.

Tabla 3: Ejemplo envío de comandos X10

Inicio	Casa P	Unidad 1/Off	Sufijo
1110	10100101	01101001	01
1110	10100101	01101001	01
000000			
1110	10100101	01011010	10
1110	10100101	01011010	10
000000			

Para mandar un comando completo siguiendo el protocolo X10 se necesitan 94 cruces por cero, lo que equivale a 783.3ms. Este tiempo que toma enviar un comando completo se considera como una desventaja de este protocolo.

4. Micro controlador PIC16F887

El micro controlador PIC16F877 posee varias cualidades que lo hacen versátil tales como oscilador interno de precisión, capacidad de interrupciones, modo ahorro de energía, módulo PWM, Modulo USART, convertidor análogo-digital, los cuales permiten al micro controlador realizar aplicaciones de automatización

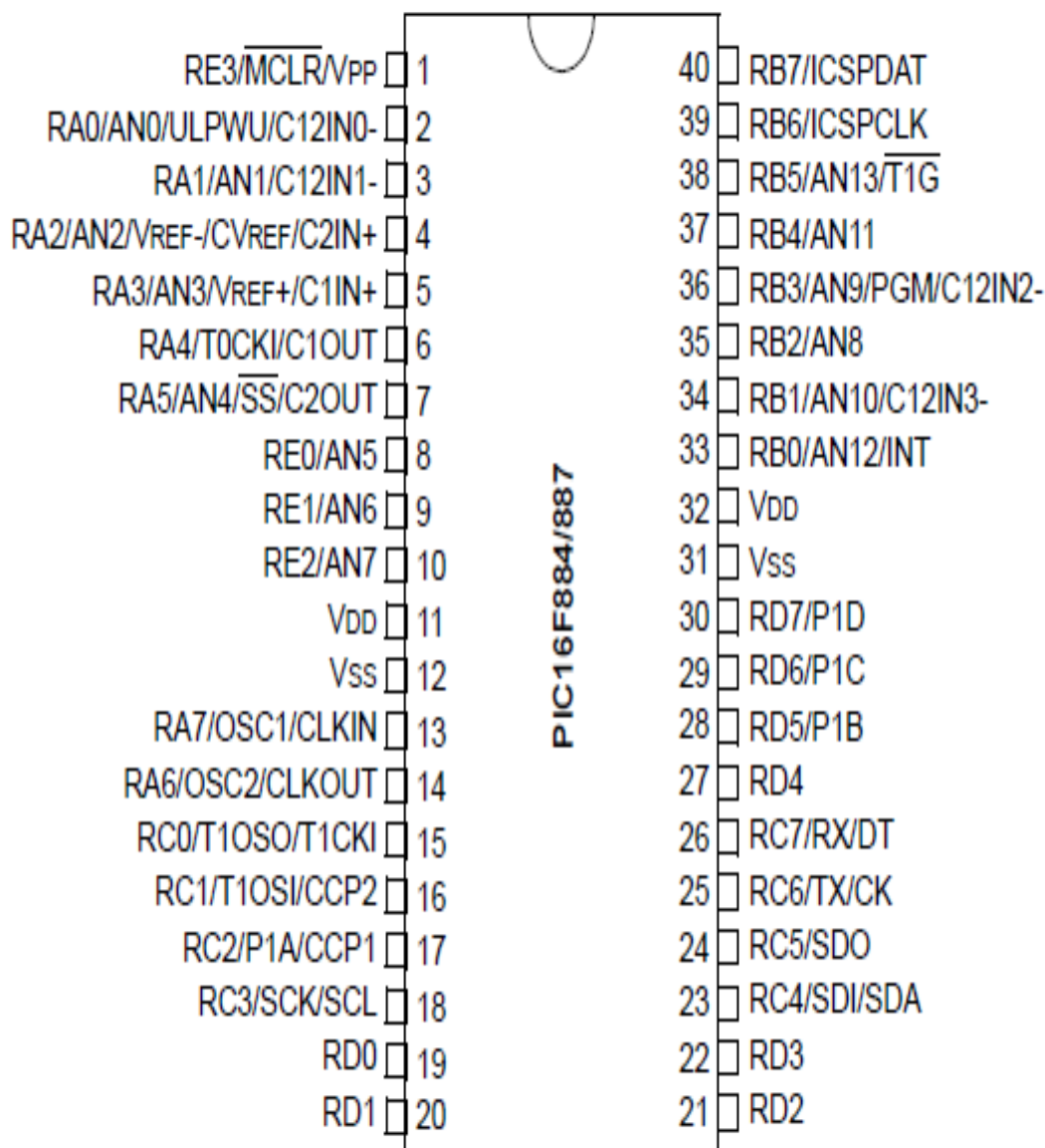
Para esta aplicación de domótica se utilizó el micro controlador del fabricante Microchip PIC16F887 de 40 pines. Para la selección del micro controlador se deben considerar varios factores importantes tales como interrupciones que maneja y módulos disponibles de comunicación. Este micro controlador posee varias cualidades que son indispensables para dicha aplicación, por mencionar algunas:

Tabla 4: Características PIC16F887

Capacidad para interrupciones	Módulo USART
Stack de 8 niveles de profundidad	14 canales de conversión Análogo-Digital de 10-bit
Oscilador interno de precisión	Módulo PWM
Modo Sleep para ahorro de energía	2 Timers de 8-bit y 1 de 16-bit
Amplio rango de voltaje de operación (2.0V-5.5V)	Corriente de operación de 220 μ A @ 4MHz
Protección de código	2 Comparadores

La siguiente ilustración muestra el diagrama de pines así como las funciones que poseen los pines del PIC16F887.

Ilustración 5: Diagrama de pines del PIC16F887.

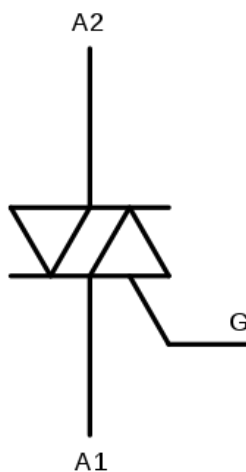


5. Dimmer

Los Dimmer (del inglés Dim: atenuar) son dispositivos reguladores de la intensidad lumínica. Existen varios tipos de Dimmer, pero los más utilizados son los resistivos y los electrónicos. Los resistivos utilizan resistencias variables y su funcionamiento es idéntico al de un divisor de voltaje. Al aumentar la resistencia variable, el voltaje disponible para las cargas disminuye, variando así la intensidad.

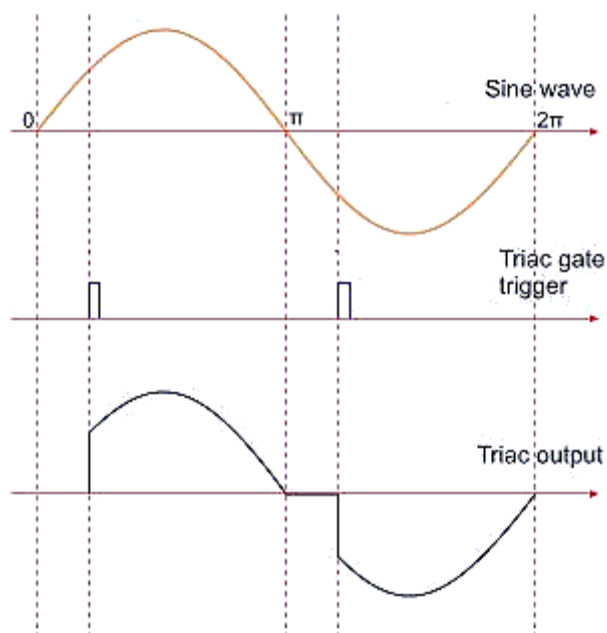
Ahora los Dimmer electrónicos, a diferencia de los reóstatos, estos funcionan como un switch y no desperdician energía para controlar la intensidad lumínica. Estos Dimmer electrónicos son conocidos como TRIAC (Triode for Alternating Current), estos pueden conducir corriente en ambas direcciones cuando son activados. Los TRIAC pertenecen a la familia de los thyristor. Estos dispositivos conducen corriente cuando un flanco de activación es aplicado a su terminal Gate, y continúan conduciendo hasta que la corriente cae a un nivel de cierta referencia, la cual es llamada corriente de mantenimiento (Burmester, 2012). La Ilustración 6 muestra el símbolo esquemático de los TRIAC.

Ilustración 6: Símbolo esquemático de TRIAC



Los TRIACs son usados en aplicaciones de baja potencia aparte de control de iluminación, tales como control de velocidades en motores eléctricos, sistemas de control computarizado de elementos caseros (Burmester, 2012). La Ilustración 8 muestra el funcionamiento de los TRIACs, en lo cual se observa que al momento de que se activa el gate, este empieza a conducir. Para lograr una reducción en el consumo eléctrico de la carga, se debe sincronizar el gate trigger con el cruce por cero de la señal sinusoidal, y partir de dicha sincronización, se hace un retardo en la activación del gate.

Ilustración 7: Funcionamiento del TRIAC

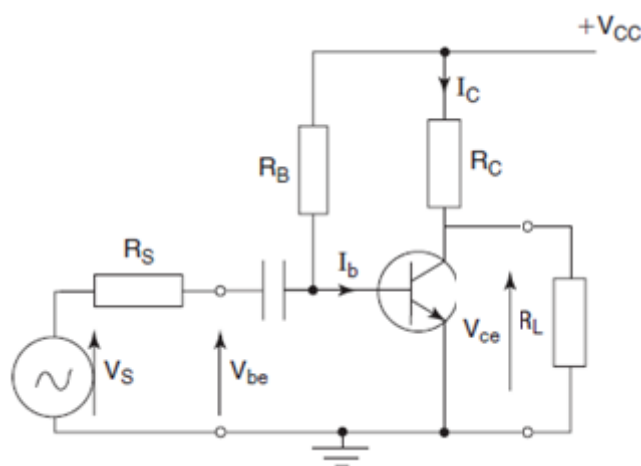


Se debe mencionar que para el control de encendido y apagado de sistemas eléctricos, no se debe sincronizar la activación del gate con el cruce por cero de la señal AC, ya que estos equipos necesitan un flujo de voltaje constante.

6. Transistor Amplificador de señales pequeñas

El transistor en configuración de amplificador de señales pequeñas que se muestra en la siguiente ilustración, es normalmente conocido como amplificador de emisor común o simplemente CE (Common Emitter). Este se caracteriza por amplificar la señal tanto en voltaje como en corriente y su impedancia de entrada y salida son altos (Esquit, 2011).

Ilustración 8: Diagrama de amplificador CE.



La amplificación es realizada gracias a la propiedad de los transistores, cuando pequeños cambios de voltaje ocurren en el nodo Base-Emisor ($\sim mV$), la corriente en el emisor sufre grandes cambios ($\sim 100\mu A$) y dicha corriente es transferida al colector del transistor (Taylor & Francis, 2013).

La ganancia en voltaje y corriente en transistores amplificadores de emisor común son definidas por las siguientes ecuaciones.

Ecuación 1: Ganancia en corriente del amplificador CE.

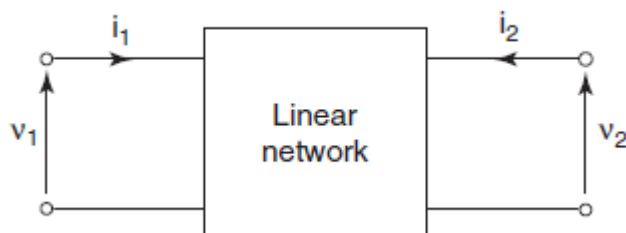
$$A_i = \frac{h_f}{1 + h_o R'_L} = \frac{i_2}{i_1}$$

Ecuación 2: Ganancia en voltaje del amplificador CE.

$$A_v = \frac{h_f R'_L}{h_i(1 + h_o R'_L)} = \frac{A_i R'_L}{h_i} = \frac{v_2}{v_1}$$

Si el transistor se encuentra correctamente polarizado y la señal de entrada es suficientemente pequeña para causar cambios de corriente y voltaje para permanecer en la región lineal del transistor, entonces se puede considerar el transistor como un circuito simple de cuatro terminales tal como se muestra a continuación.

Ilustración 9: Circuito de 4 terminales para transistores



El análisis del circuito involucra variables que son conocidas como híbridas o parámetros-h, las cuales se representan de la siguiente manera.

Ecuación 3: Impedancia de entrada en amplificador CE

$$h_i = \frac{v_1}{i_1} \text{ ohm}$$

Ecuación 4: Admitancia de salida en amplificador CE.

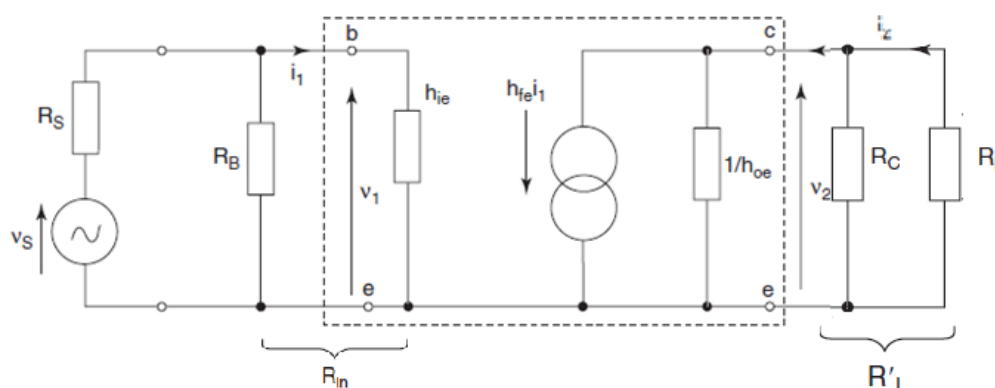
$$h_o = \frac{i_2}{v_2} \text{ siemen}$$

Ecuación 5: Corriente de ganancia DC en amplificador CE

$$h_f = \frac{i_2}{i_1}$$

Otra alternativa para encontrar los parámetros-h es analizar las gráficas de comportamiento en la hoja de datos técnicos para el transistor que se utiliza. El circuito equivalente para los parámetros-h es el siguiente.

Ilustración 10: Circuito equivalente parámetros-h en amplificador CE.



Para la resolución del circuito en forma analítica se puede utilizar la siguiente tabla para encontrar las variables necesarias.

Tabla 5: Formulas para resolución analítica de amplificador CE.

Variable	Fórmula	Dimensional
V_1	$\frac{(R_{in})(V_s)}{R_s + R_{in}}$	Volts
i_1	$\frac{V_1}{h_{ie}}$	Amperios
i_2	$\frac{\left(\frac{1}{h_{oe}}\right)(h_{fe} * i_1)}{\frac{1}{h_{oe}} + R'_l}$	Amperios
V_2	$R'_l * i_2$	Volts

V. METODOLOGÍA

El micro controlador a utilizar en esta aplicación es el PIC16F877, debido a sus propiedades que se muestran en la Tabla 4. Estas características son comunes en el amplio rango de micro controladores disponibles, pero se utilizará el micro controlador PIC16F887 debido a que su ambiente de programación se ha utilizado con anterioridad.

La comunicación entre micro controladores es basada en los principios de transmisión y recepción de datos del protocolo X10, por lo que los micro controladores estarán basados en estos principios únicamente y no se comunicarán como lo hace un componente puro de X10.

Los módulos de transmisión y recepción de datos que en este caso son los micro controladores deben ser conectados a los tomacorrientes de 120VAC pero estos módulos necesitan alimentación de 5VDC además deben ser compactos y portátiles, es por esto que se usaran fuentes de alimentación 5VDC sin transformador.

1. Comunicación de Micro controladores

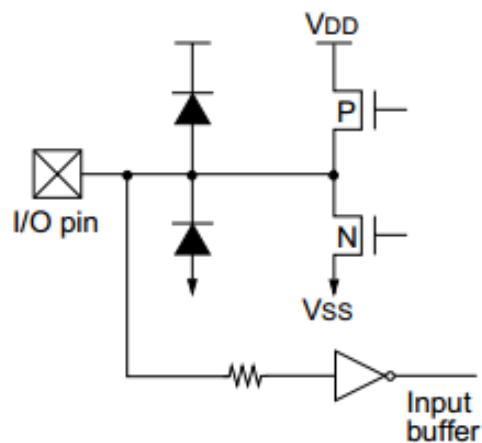
La comunicación entre micro controladores será utilizando el cableado de alimentación 120VAC como bus de datos y para lograrlo se decidió dividirlo en tres etapas.

- a. Detector de cruce por cero.
- b. Generador de pulsos a 120KHz.
- c. Detector de pulsos a 120KHz.

a. Detector de cruce por cero. El sistema funciona en sincronización con el cruce por cero de la señal de 120VAC, para realizar dicha sincronización se utilizó la interrupción de cambio de estado del puerto B del micro controlador, en el pin7 en específico. Este circuito detector se implementa conectando la línea viva AC al micro controlador a través de un único componente externo, una resistencia. Este método toma ventaja de la protección en circuitería existente en todos los pines de entrada/salida en la familia de los CMOS PIC16Fxxx. Estas protecciones están diseñadas para ajustar el voltaje presente en los pines al voltaje de alimentación en caso exista una sobre carga por lo que protegiendo al chip de picos en voltaje mayores a $V_{DD}(+5V)$ o menores a $V_{SS}(0V)$.

La Ilustración 11 muestra la estructura interna de los pines con la protección.

Ilustración 11: Estructura interna de pines I/O



El Input buffer es un Schmitt trigger, esto hace que el umbral de entrada en el flanco de subida sea de $0.8V_{DD} = 4V$, y el umbral en el flanco de bajada sea de $0.2V_{DD} = 1V$. Por lo tanto, debido a los diodos de protección, cuando el voltaje AC

este en el semi-ciclo negativo, el pin RB7 se encontrara a $V_{SS} - 0.6V$, lo cual es menor al umbral de flanco de bajada e interpretado como un cero lógico. Cuando el voltaje AC sea mayor al umbral del flanco de subida, este será anclado al V_{DD} e interpretado como uno lógico (Cox, 2002).

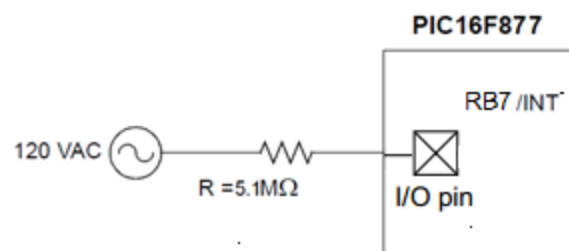
Una señal sinusoidal de 120VAC y 60Hz, le toma 32uS en la transición de 0 a 2V, por lo tanto un umbral típico de 2V en los I/O permitirá una precisión en la detección de cruce por cero de alrededor de 30uS. Si la capacitancia típica en los pines I/O es de 5pF, entonces la resistencia a usar debe cumplir con la siguiente ecuación (Cox, 2002).

Ecuación 6: Tiempo de precisión en detección de cruce por cero.

$$t = RC$$

Obteniendo un valor de $R = 6M\Omega$, o menor para mejorar la precisión en la detección de cruce por cero. Una $R = 5.1 M\Omega$ provee una precisión de 25uS y limita la corriente de entrada a 33uA con un voltaje pico máximo de 165V, cumpliendo a la vez con la limitación máxima de corriente continua soportada por los pines I/O de 500uA. La Ilustración 12 muestra la configuración resultante.

Ilustración 12: Circuito detector cruce por cero.

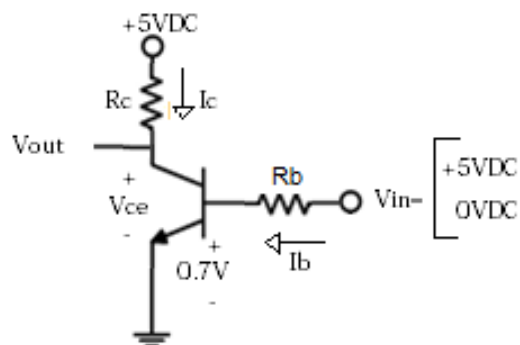


Al momento de tener la interrupción generada por cruce por cero, se hace el manejo de variable necesarias para emitir o leer señales que se encuentran en la línea de transmisión.

b. Generador de pulsos a 120KHz. Esta etapa es la encargada de transmitir en la línea de 120VAC a 60Hz, un pulso de señal cuadrada con ciclo de trabajo del 50% a una frecuencia de 120kHz con duración de 1ms, cada vez que se detecte el cruce por cero y sea necesario mandar '1' lógico. Se utiliza la frecuencia de 120kHz, ya que la comunicación se basa en el protocolo X10 y este utiliza dicha frecuencia con el fin de poder separarlas fácilmente en la etapa de recepción. Esta señal cuadrada puede ser generada por un circuito externo, pero será aprovechado el módulo CCP del micro controlador realizando una señal de PWM.

Como diseño inicial del circuito acoplador de pulsos DC a la red de alimentación AC, se utilizó un transistor en configuración de inversor con el objetivo que la corriente de los pulsos no sea obtenida del micro controlador. La configuración de transistor inversor de emisor común se muestra en la Ilustración 13.

Ilustración 13: Transistor inversor



El análisis para determinar los valores de los componentes en el inversor es el siguiente. Debido a que la salida se encuentra en el colector, se necesita la mayor corriente de colector posible, sin exceder los rangos máximos del transistor, para obtener mayores distancias de comunicación emisor-receptor. Se utiliza el transistor NPN 2N2222, que tiene una corriente de colector máxima de 1A, por lo que se escoge un $R_c=10\Omega$ para obtener una corriente de colector $I_c=500\text{mA}$, con lo cual según la hoja de datos de dicho transistor se obtiene una ganancia de corriente $\beta=40$.

En el caso que la entrada sea '0' lógico, sabemos que no existirá corriente de base, debido a que no se polariza el diodo base-emisor con un voltaje menor a 0.7V. Por lo tanto el transistor se encuentra en modo cut-off, es decir que la corriente de base es menor a la del emisor y la corriente de colector es mayor a la corriente de base, lo que significa que el transistor no permite el paso de corriente desde el colector hacia el emisor, por lo tanto voltaje presente en la salida será 5V.

En el caso que la entrada sea '1' lógico, necesitamos que el transistor se encuentre en la región de saturación, para esto debemos escoger una resistencia de base tal que la corriente de base sea mayor a la corriente de emisor. En la región saturada se debe cumplir que:

Ecuación 7: Ecuación a cumplir de transistor en saturación.

$$\beta * I_b > I_c$$

Ahora sustituyendo valores de la ecuación 7 para encontrar R_b , se obtiene la ecuación 8:

Ecuación 8: Valor Rb para transistor saturado.

$$(40) \left(\frac{V_{in} - 0.7V}{R_b} \right) > 500mA$$

$$\frac{(40)(4.3V)}{500mA} > R_b$$

$$344\Omega > R_b$$

Lo cual indica que cualquier valor menor a 344Ω para R_b , por lo que se escoge una $R_b=220\Omega$, hará que el transistor se encuentre en la región saturada, tal que se permite el paso de corriente desde el colector hacia el emisor. Ahora podemos resolver el circuito mediante la ecuación 9.

Ecuación 9: Ecuaciones solución para transistor saturado.

$$I_e = I_b + I_c$$

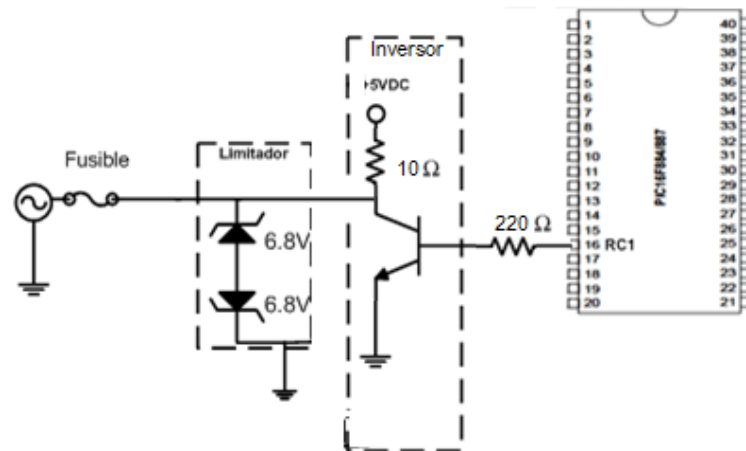
$$5V - 220\Omega * I_b - 0.7v = 0$$

$$5V - I_c * 10\Omega - V_{ce} = 0$$

Sabiendo que $V_{ce}=0.3V$ en región saturado según la hoja de datos técnicos del transistor, obtenemos una $I_c=470mA$, lo cual a través de $R_c=10\Omega$ genera una caída de voltaje de $4.7V$, por lo tanto en la salida tendremos $0.3V$ que es interpretado como un '0' lógico.

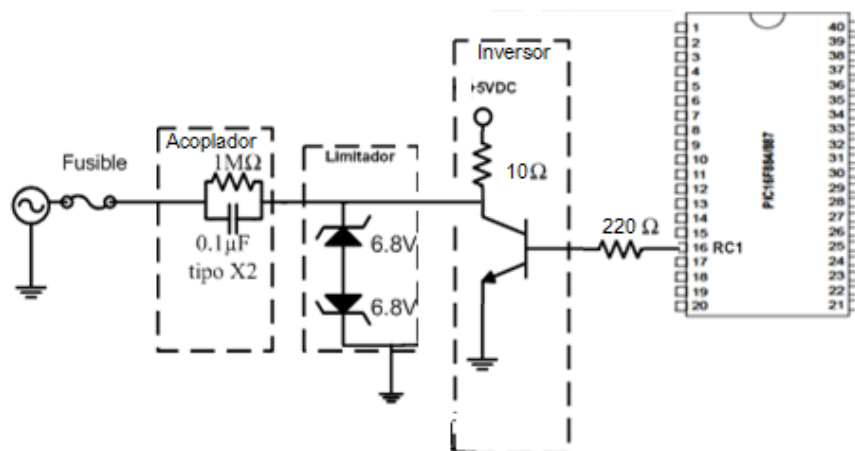
Seguido del transistor se colocaron diodos zener limitadores de voltaje para evitar que voltajes altos interactúen con el transistor. Dicho circuito se muestra en la Ilustración 14.

Ilustración 14: Diseño pre-eliminar inyector de pulsos 120KHz.



El diseño de la Ilustración 14 no resultó satisfactorio, ya que al momento de conectar la línea AC en el emisor del transistor, este se daña debido a que la señal de 120VAC a 60Hz está presente en el emisor del transistor. Por dicha razón se procede a implementar un capacitor y resistencia en paralelo para que represente una impedancia alta a señales con frecuencias de 60Hz y así limitar la corriente de la línea AC. El circuito resultante se muestra en la Ilustración 15.

Ilustración 15: Diseño inyector de pulsos 120KHz [3]



Para determinar los valores apropiados se encuentra la impedancia del capacitor mediante la ecuación 10 para luego sumarla en paralelo con la impedancia de la resistencia:

Ecuación 10: Impedancia de capacitor.

$$Z_c = \frac{1}{2\pi * f * C}$$

Se escoge un valor común de capacitancia de 0.1 μ F, para luego determinar el valor de resistencia que genere el comportamiento deseado. Con los valores utilizados, el capacitor y la resistencia representan una impedancia baja de 13.25 Ω a frecuencias de 120kHz y alta impedancia de 25.84K Ω a frecuencias de 60Hz.

En el área de programación del micro controlador se debe seguir un algoritmo para que el envío de datos sea correcto. La Ilustración 18 muestra el algoritmo construido para el envío de comandos.

Ilustración 16: Algoritmo para envío de señales.



El algoritmo para envío de señales solo indica la lógica que posee el programa, por lo que a continuación se explica detalladamente que registros y variables deben ser modificados en cada paso.

En el segmento de preparación se inicia la configuración de cruce por cero, lo cual se debe configurar el pin respectivo como entrada digital con los registros

ANSEL y ANSELH, deshabilitar los bits comparadores C1ON y C2ON y habilitar el bit RBIE del registro INTCON el cual se encarga del control de interrupciones de port-on-change de puerto B. Además se debe habilitar el bit correspondiente al pin que genera la interrupción en el registro IOCB.

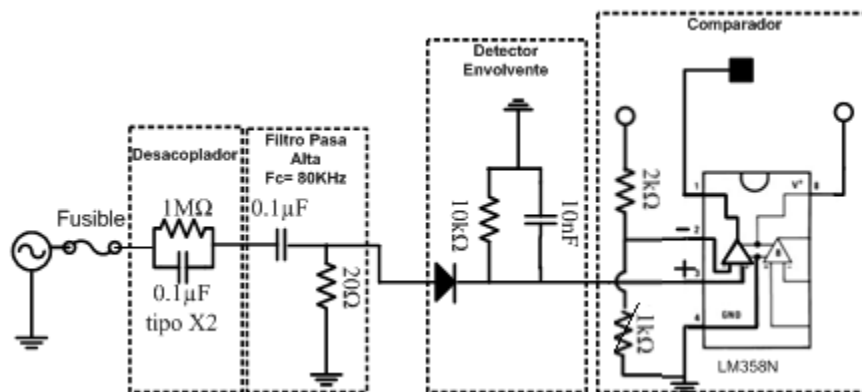
En la configuración de puertos y banderas, se indican que puertos son entradas y salidas, en este caso el pin encargado de la interrupción por cero del puerto B debe configurarse como entrada en el registro TRISB. Las banderas encargadas del flujo de envío de comandos son inicializadas en 1, para permitir el envío de comandos en el encendido del micro controlador.

La configuración del PWM se facilita debido a que el programa MikroC solo requiere que se indique la frecuencia y el ciclo de trabajo deseado. De lo contrario se tiene que configurar el módulo CCP en modo de operación PWM configurando cada registro necesario para obtener dicho modo de operación.

Luego de que se ha realizado la configuración del micro controlador, este es programado para que se encuentre en loop infinito, chequeando únicamente las interrupciones generadas por el cruce por cero y los comandos recibidos del servidor. En la interrupción se envía el arreglo de comandos solamente si se han cumplido las condiciones para poder enviar. Dichas condiciones incluyen los 3 cruces por cero de separación entre cada comando y si se ha recibido información del servidor.

c. Detector de pulsos 120KHz. Para poder detectar los pulsos de 120KHz, es necesaria la sincronización de cruce por cero del micro controlador que estará recibiendo, debido a que se utiliza el cruce por cero como reloj general de comunicación en los módulos de emisor y receptores. Esta recepción de pulsos de 120KHz que se encuentran en el cableado AC, se hizo posible por medio del circuito que se explica a continuación.

Ilustración 17: Diseño pre-eliminar circuito detector de pulsos 120KHz



La parte del capacitor des-acoplador se utiliza con el mismo objetivo para el que fue utilizado en el circuito generador de pulsos, de poder mantener efectivamente el voltaje que se encuentra en las frecuencias de Khz-Mhz. Además este des-acoplador reduce el ruido causado por otros elementos. [3]

Se utilizará filtro pasa alta RC lo cual para encontrar la frecuencia de corte de este filtro se utiliza la siguiente ecuación:

Ecuación 11: Frecuencia de corte filtro RC

$$f_c = \frac{1}{(2\pi * R * C)}$$

Con esta configuración de filtro pasa alta RC, se escoge un valor común de capacitor de $C=0.1\mu\text{F}$, una frecuencia de corte de 80Khz y la resistencia se despeja de la ecuación 11, con lo que se obtiene $R=20\Omega$, de este modo se logra atenuar la frecuencia de 60Hz y obtener efectivamente las señales 120kHz.

Luego se utiliza un circuito detector envolvente, la función principal de este circuito es recobrar la señal envolvente de la señal filtrada, mediante el seguimiento de su variación lenta y el filtrado de las variaciones rápidas de la alta frecuencia. Este proceso está relacionado con la demodulación, ya que las variaciones de la señal envolvente son de hecho la señal modulante (González, 2010).

Este circuito carga el capacitor cuando hay picos de subida de voltaje en la entrada y el momento de flancos de bajada en su voltaje, el capacitor se descarga a través de la resistencia, obteniendo así el envolvente de la señal de entrada. Para valores altos de RC, el capacitor se descargara muy lentamente y la salida del circuito no será exactamente la envolvente. Si la constante de tiempo RC es muy pequeña en relación al periodo de la frecuencia mayor en la señal modulada, este circuito no podrá seguir los picos para obtener un envolvente estable. Por lo que es requerido que la ecuación 12 se cumpla (Freir, 2008).

Ecuación 12: Calculo de circuito de envolvente

$$T \ll RC \ll \frac{(1 - m^2)^{1/2}}{m * Wm}$$

Donde T indica el periodo de la señal portadora, Wm es la más alta frecuencia de la señal modulada y “m” es el índice de modulación y vienen dados por la ecuación 13 y 14 respectivamente.

Ecuación 13: Frecuencia señal modulada

$$Wm = 2\pi f \text{ (rad)} = 753,982.2369 \text{ rad.}$$

Ecuación 14: Índice de Modulación

$$m = \frac{Vm}{Vc} = \frac{5v}{5v} = 1$$

Con V_c representando la amplitud en voltaje de la onda portadora y V_m es la amplitud en voltaje de onda en variación. Al tener los datos adecuados se cumplen los requisitos tal como lo muestra la ecuación 15.

Ecuación 15: Requisito detector envolvente

$$\frac{1}{120kHz} \ll (10nF)(10K\Omega) \ll \frac{(1 - 1^2)^{1/2}}{1 * 753,982 \text{ rad}}$$

$$8.333\mu S \ll 0.1mS \ll 0$$

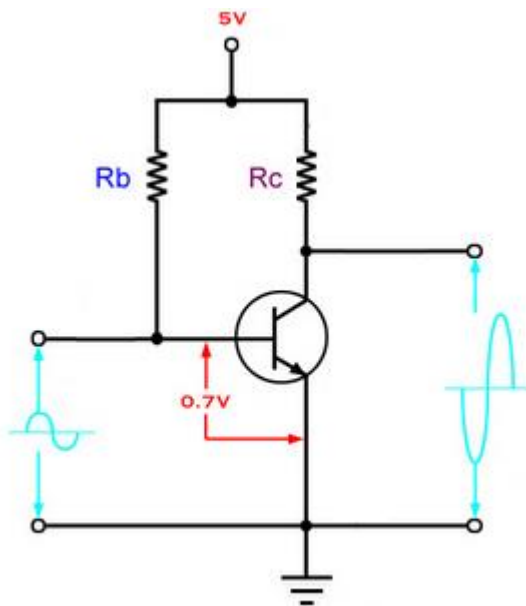
En general RC debe ser mucho mayor al periodo del portador y mucho menor al periodo de la señal modulada (Freir, 2008).

Como último paso se utiliza un amplificador operacional LM358N en configuración de comparador. Se escoge este amplificador operacional en específico debido a que este no requiere de voltajes negativos para su funcionamiento. Este amplificara la señal solo cuando la señal que se encuentra en la terminal no invertida sea mayor al voltaje que se encuentra en la terminal invertida. Dicho voltaje de comparación es generado por un divisor de voltaje que podrá variar de 0 – 0.5Vcc mediante la modificación de un potenciómetro de

precisión. Esto es debido a que mientras aumenta valor de voltaje de comparación, aumenta la eliminación de ruido eléctrico presente en las señales, pero al mismo tiempo disminuye la distancia de comunicación.

Luego de comprobar el funcionamiento del circuito receptor se logró observar que con el diseño pre-eliminar se obtienen las señales correctamente únicamente cuando el circuito detector se encuentra conectado en el mismo toma corriente que el circuito generador de señales. Al realizar pruebas con mayor distancia entre el micro controlador inyector y el micro controlador detector, se observa que la salida del filtro pasa alta posee amplitud menor a 200mV, la cual no es suficiente para que el detector envolvente muestre el envolvente. Por dicha razón se procede a implementar un transistor amplificador de señales pequeñas a la salida del filtro pasa alta. La siguiente ilustración muestra la configuración del circuito.

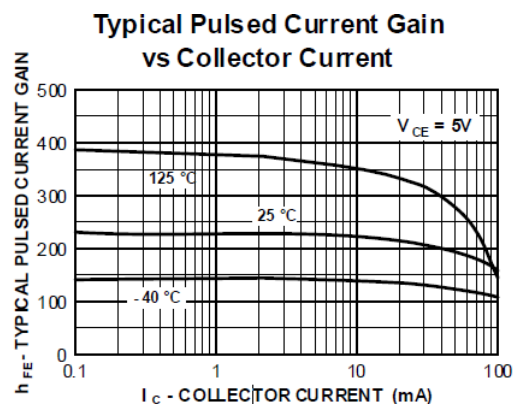
Ilustración 18: Configuración de amplificador CE utilizado.



Para determinar los valores de los componentes se realiza el siguiente análisis. Se analiza la situación cuando el voltaje de entrada sea 0v. El voltaje de colector será igual al valor del voltaje de alimentación en la ausencia de señales, de este modo el voltaje de colector puede variar entre el voltaje de alimentación y 0v cuando una señal esta finalmente presente en la base. La resistencia R_b provee el voltaje necesario para polarizar el diodo base-emisor, debido a que se necesita que el diodo sea polarizado incluso cuando la señal de entrada sea menor a 0.7V, de no ser así, el colector no estará conectado a tierra y el voltaje de colector siempre será igual al voltaje de alimentación.

Para determinar los valores correctos de R_b y R_c necesitamos decidir cuanta corriente es necesaria que fluya entre el emisor y colector para mantener el transistor activo cuando no existan señales en la base, esta corriente es llamada corriente inactiva, la cual no debe ser mayor a la necesaria para conservar energía. El transistor 2N3904 opera satisfactoriamente con una corriente inactiva de 1mA en el colector. Ahora debemos saber cuánta corriente de base nos generará la corriente inactiva. Utilizando la Ilustración 19 obtenida de la hoja técnica del transistor, podemos determinar la ganancia de corriente colector (h_{fe}).

Ilustración 19: Ganancia vrs Corriente Colector 2N3904



Con una corriente de colector de 1mA obtenemos una $h_{fe}=225$ aproximadamente. Ahora para determinar el valor adecuado para R_B necesitamos encontrar la corriente que estará presente en la base mediante la ecuación 16.

Ecuación 16: Corriente de base en amplificador CE.

$$Corriente\ base = \frac{Corriente\ colector}{Hfe} = \frac{1mA}{225} = 4.44\mu A$$

Ahora con una alimentación de 5V en la base a través de R_b , necesitamos saber qué valor de R_b es necesario para proveer $4.44\mu A$ en la base. Se debe mencionar que el voltaje en la base no es exactamente igual al voltaje de alimentación debido a que se tiene una caída de voltaje de 0.7V para polarizar el diodo. Utilizando la ley de ohm para obtener un valor de R_b tal como lo muestra la ecuación 17.

Ecuación 17: Valor R_b necesario en amplificador

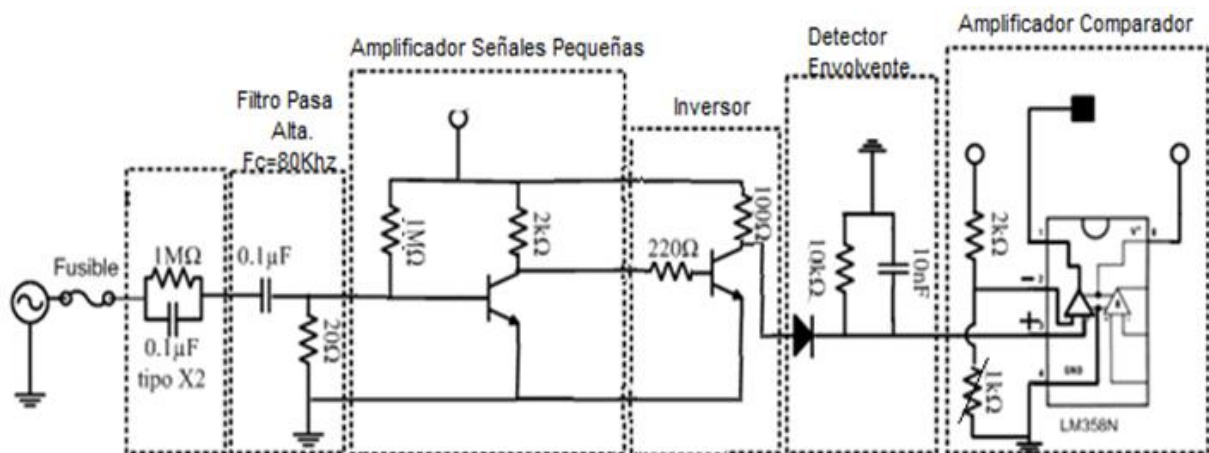
$$R_b = \frac{V_b}{I_b} = \frac{4.3V}{4.44\mu A} = 967.5K\Omega$$

Si no hay señal aplicada en la base, la corriente de colector siempre será 1mA sin importar que valor de R_c se utilice. El valor de R_c solo influye en el valor de voltaje de colector. Con una alimentación de 5V el transistor siempre tratara de mantener una resistencia combinada de $5,000\Omega(5v/1mA)$ entre el voltaje de

alimentación y tierra. Debido a esto el valor a utilizar de $R_c=2k\Omega$, por lo que el transistor estará actuando como una resistencia de $3k\Omega$ para complementar la resistencia combinada necesaria.

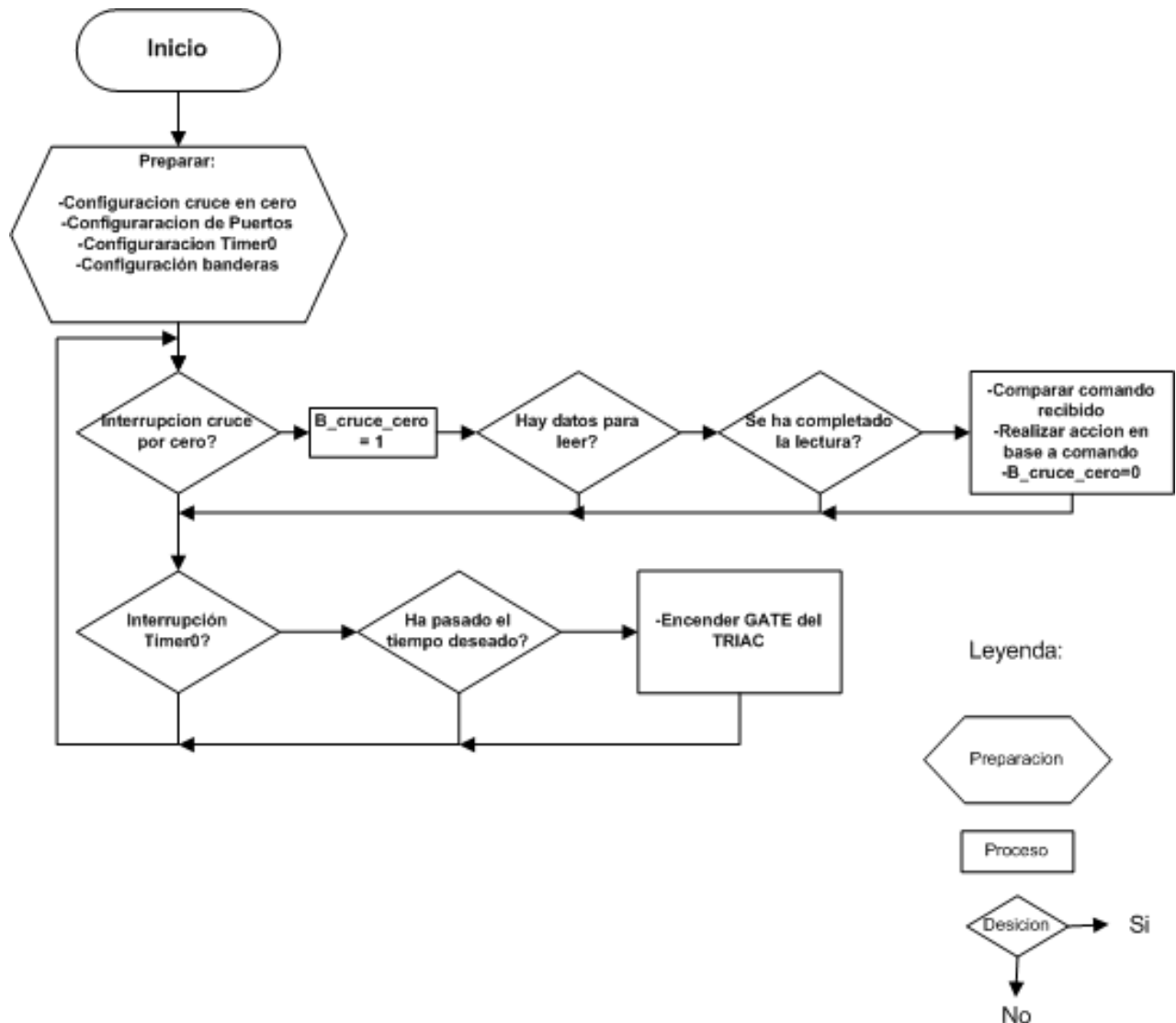
La Ilustración 20 muestra la configuración final del circuito detector de pulsos de 120KHz de frecuencia.

Ilustración 20: Diseño circuito detector de pulsos 120KHz



En el área de programación del micro controlador se debe seguir un algoritmo para que la recepción de datos sea interpretada correctamente. La Ilustración 21 muestra el algoritmo construido para la recepción de comandos.

Ilustración 21: Algoritmo para recepción de señales



En el algoritmo de recepción se debe configurar el Timer0 para el control de iluminación, ya que este controla el Gate del TRIAC, determina cuanto tiempo después del cruce por cero se debe activar el Gate. Cuando se genera la interrupción por el cruce por cero se levanta una bandera, la cual es utilizada para indicarle al Timer0 que debe iniciar su conteo. Se debe mencionar que al momento del cruce por cero, se hace un delay de 500ms antes de realizar la lectura de datos, esto para asegurar que la lectura sea correcta.

2. Recepción y envío de comandos

La comunicación entre micro controladores se basa en los principios del protocolo X10. Para esta etapa, se buscó tener un protocolo más eficiente en el envío y recepción de señales, es por esto que se hicieron modificaciones al formato que utiliza el protocolo X10 para enviar sus señales, por lo que los microcontroladores no son programados para enviar e interpretar el formato de comunicación de dicho protocolo.

Una de las desventajas de dicho protocolo es el tiempo que toma en enviar un comando completo. Este protocolo es lento, debido a que es un protocolo conservador con respecto al ruido y es por esto que envía dos veces cada comando.

Las modificaciones fueron realizadas tomando en cuenta que el ruido que posee la red de alimentación de voltaje AC puede llegar a introducir señales no deseadas y de esta manera realizar acciones no deseadas o simplemente no realizar acciones cuando se envían los comandos. Además luego de la etapa de pruebas se logró determinar que mientras más bits se utilicen para enviar un comando, la distancia efectiva de separación entre micro controladores se reduce.

Para determinar cuántos bits son necesarios para formar un comando completo de la aplicación, como primer paso se debe decidir cuantos dispositivos se van a controlar y cuantas funciones tendrá cada dispositivo. Se decidió que se tendrán tres dispositivos automatizados, cada uno con control para luminaria y tomacorriente. Ya que uno de los dispositivos automatizados será el que estará conectado directamente con el servidor Raspberry Pi, se tomaran en cuenta únicamente dos dispositivos con ocho funciones cada uno para fabricar el protocolo necesario de comunicación a través de la línea AC. Al tener definido lo anterior se procede a fabricar la estructura del protocolo, el cual se muestra en la Tabla 6.

Tabla 6: Códigos utilizados

Nombre	Código
Comando inicio	11
Dispositivo 1	01
Dispositivo 2	10
Todos los dispositivos	11
Tomacorriente	1
Luminaria	0
ON	001
OFF	010
Incrementar intensidad	011
Disminuir intensidad	100
Intensidad al 50%	101
Intensidad al 25 %	110

Las funciones de incrementar y disminuir intensidad realizan la variación en una tasa de 25%. La Tabla 7 muestra la secuencia de envío de códigos.

Tabla 7: Secuencia de envío.

Comando	Dispositivos		Elemento		Función	
Inicio						
11	Disp. 1	01	Toma.	1	ON	001
	Disp. 2	10	Luminaria	0	OFF	010
	Todos	11	--	-	Incrementar	011
				-		
					Disminuir	100
					50%	101
					25%	110

Se debe remarcar que el elemento tomacorriente solo puede utilizar las funciones ON-OFF. Cuando el comando es enviado, inmediatamente se inicia la transmisión del inverso completo del comando sin incluir el comando de inicio. Es decir un comando completo contiene 14 bits o 14 cruces por cero, equivalente a 116.66 ms. La secuencia de envío se muestra en la Tabla 8.

Tabla 8: Secuencia de envío comandos protocolo diseñado

Inicio	Dispostivo1	Elemento	Función
11	01	1	001
	10	0	110
	000000		

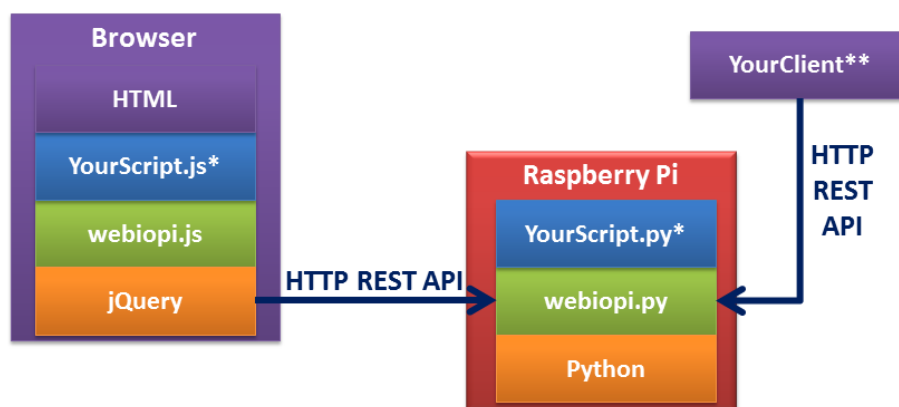
3. Servidor Web.

Para el módulo del servidor se utilizó el computador RaspberryPi, el cual se comunica al micro controlador por medio de comunicación serial. Para lograr controlar los GPIO desde la web se utilizó el Framework Webiopi.

Webiopi es un framework completo de servidores de internet para el RaspberryPi, permite controlar los GPIO local y remotamente desde cualquier explorador de internet. Webiopi instala completamente la función que permite al RaspberryPi funcione como un servidor y a su vez permitiendo interactuar con sus GPIO desde la web.

La estructura y funcionamiento del framework se muestra en la Ilustracion 22.

Ilustración 22: Estructura Framework Webiopi



* Optional, allows you to customize the interface and the behavior

** Use REST API from any technology/language

Basándose en dicho Framework se procedió a programar la interfaz web en lenguaje HTML. Para ingresar a la interfaz se utiliza la dirección IP o nombre del computador del raspberry para acceder a las páginas HTML que tiene publicadas, para esto el acceso debe realizarse en la LAN. Se debe mencionar que se necesita una dirección de IP pública o un servicio de Hostname para poder controlar el sistema desde afuera de la LAN.

4. Control de Intensidad, encendido y apagado

Luego de que los micro controladores se logran comunicar, se debe controlar las intensidades y encendidos-apagados de las luminarias y tomacorrientes respectivamente. Este control se realiza mediante el TRIAC.

Para lograr una conexión segura entre el TRIAC y el micro controlador se utiliza un opto acoplador (moc3020) para separar por completo el micro controlador de las cargas que presentan las luminarias y tomacorrientes. Luego de tener des acoplado el micro controlador de las cargas, se procede a introducir el TRIAC para el control de intensidad lumínica. Este control de intensidad

lumínica es realizado mediante la activación de Gate en el TRIAC, en el caso de las luminarias dicha activación es controlada por las interrupciones del Timer0 en el micro controlador.

En términos generales cada vez que ocurre una interrupción del cruce por cero se modifica la variable que controla el overflow del Timer0 causante de la interrupción. Cada vez que ocurre dicha interrupción del Timer0, se consulta cuantas interrupciones de Timer0 deben ocurrir antes de activar el Gate del Triac y de esta manera controlar la intensidad lumínica.

Para controlar el tiempo que transcurre entre cada interrupción del timer0, se procede hacer el siguiente análisis. El micro controlador, el cual controlará el TRIAC, utilizará el reloj interno con frecuencia de 4 Mhz. Según la hoja técnica del micro controlador se necesitan 4 ciclos de reloj por instrucción, por lo tanto el micro controlador receptor estará realizando 1MIPS (Millones de Instrucciones Por Segundo). Luego se escogió una tasa de 1:8 en el prescaler, esto quiere decir que el Timer0 genera 8 overflows antes de generar una interrupción, lo cual modifica a 125,000 instrucciones por segundo. Ahora se puede obtener el tiempo necesario para realizar una instrucción el cual es de $8\mu\text{s}$.

En domótica, regularmente los controladores de luminarias limitan la intensidad máxima a un 85%, esto es debido a que se ha comprobado que el ojo humano no logra diferenciar el cambio de intensidad de una luminaria al 100% con una de 85%. Al limitar la intensidad máxima se puede aumentar el ahorro de consumo eléctrico por parte de las luminarias.

Se sabe que medio ciclo de señal AC tiene una duración de 8.333ms, si el Triac es activado durante esta cantidad de tiempo la luminaria estaría a su

intensidad máxima real. Pero se usará el método utilizado en domótica para el ahorro de consumo eléctrico, por lo tanto el Triac solo podrá estar activado durante 7.08ms por cada medio ciclo para intensidad máxima. Luego se decide que la intensidad se manipulara a una tasa de 25%, teniendo cinco niveles desde 0% al 100%(85% real). Al tener cinco niveles de intensidad lumínica, es necesario que el timer0 genere su interrupción cinco veces durante el semi ciclo, por lo que tendrá que generar dicha interrupción cada 1.4166ms.

Teniendo los datos respectivos se procede a realizar la siguiente relación.

Ecuación 18: Relación para obtener ajuste de Timer0.

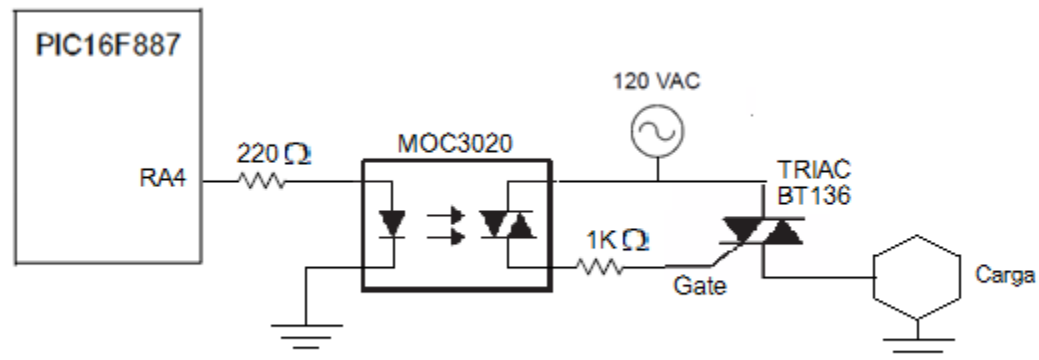
$$1.41661ms = 8\mu s * (255 - \text{ajuste})$$

Donde 255 indica la resolución de 8 bits que posee el Timer0. De la relación anterior se obtiene un ajuste de 77.92, lo cual estará utilizando 78. Este valor será impuesto al Timer0 en software cada vez que termine la interrupción para obtener el control de resolución deseado del Dimmer por medio del TRIAC.

Para el control de encendido y apagado de toma corrientes, no es necesario sincronizar la activación del gate en el Triac con el cruce por cero.

A continuación se presenta el circuito utilizado para esta etapa.

Ilustración 23: Conexión de TRIAC



El valor de la resistencia que se encuentra en el Gate del TRIAC se escoge para cumplir la siguiente relación.

Ecuación 19: Resistor limitante de TRIAC

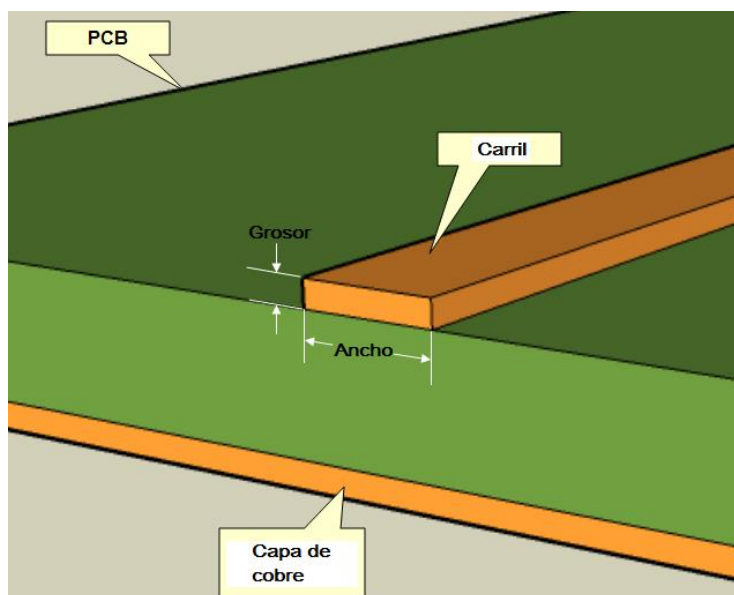
$$R = \frac{Vp \text{ linea AC}}{I_{tm}}$$

Donde V_p es el voltaje pico de la línea AC, I_{tm} es la corriente máxima que soporta el Gate del TRIAC, que en este caso es de 1A según lo indica su hoja técnica. Con el valor de la resistencia colocada se obtiene una corriente de 0.12A en el Gate, cumpliendo la relación sin exceder el máximo permitido. [2]

5. PCB

PCB (Printed Circuit Board) es el método utilizado para la presentación de los módulos emisores y receptores. Antes de iniciar el diseño del esquemático para el circuito impreso en placa de cobre, se deben tomar en cuenta parámetros como espesor de cobre en la placa a utilizar, el ancho de los carriles conductores, espaciamiento entre carriles, medidas de los pad's, número de capas, etc. La Ilustración 25 muestra la nomenclatura en una PCB.

Ilustración 24: Nomenclatura PCB



Las placas de cobre utilizadas para PCB tienen diferentes designaciones en el calibre de cobre con las que son construidas, esto depende de las necesidades de la aplicación. La Tabla 9 muestra los diferentes valores de calibre de cobre según el tipo de placa que se va a utilizar.

Tabla 9: Calibre de cobre para PCB

Foil Designation	Common Industry Terminology	Metric		English		
		Area Weight (g/m ²)	Nominal Thickness (μm)	Area Weight (oz./ft. ²)	Area Weight (g/254 in ²)	Nominal Thickness (mils)
E	5 μm	45.1	5.1	0.148	7.4	0.20
Q	9 μm	75.9	8.5	0.249	12.5	0.34
T	12 μm	106.8	12.0	0.350	17.5	0.47
H	1/2 oz	152.5	17.1	0.500	25.0	0.68
M	3/4 oz	228.8	25.7	0.750	37.5	1.01
1	1 oz	305.0	34.3	1	50.0	1.35
2	2 oz	610.0	68.6	2	100.0	2.70
3	3 oz	915.0	102.9	3	150.0	4.05
4	4 oz	1220.0	137.2	4	200.0	5.40
5	5 oz	1525.0	171.5	5	250.0	6.75
6	6 oz	1830.0	205.7	6	300.0	8.10
7	7 oz	2135.0	240.0	7	350.0	9.45
10	10 oz	3050.0	342.9	10	500.0	13.50
14	14 oz	4270.0	480.1	14	700.0	18.90

El calibre de placas de cobre comercial comúnmente contienen 1 oz. de cobre por cada pie² de área, la cual es la utilizada en este diseño. Como primer paso se debe calcular el área del carril con la ecuación 20, utilizando una corriente máxima de 2 amperios, debido a que el moc3020 tiene la menor corriente de manejo, la cual es de 2.5amps.

Ecuación 20: Área de carril de cobre

$$Area (mils^2) = \frac{Corriente (amps)}{(k * temperatura(^{\circ}C))^{\frac{1}{c}}}$$

En donde el estándar IPC-2221 (Generic Standard on Printed Board Design) indica los valores de k=0.048, b=0.44, c=0.725 para capas externas. Luego de tener el área del carril, se procede a calcular el ancho con la ecuación 18.

Ecuación 21: Ancho necesario en carril de cobre

$$Ancho (mils) = \frac{Area (mils^2)}{Grosor (oz) * 1.378 \left(\frac{mils}{oz}\right)}$$

VI. RESULTADOS

1. Comunicación de micro controladores

Las ilustraciones muestran diagramas esquemáticos, PCBs y módulos construidos.

Ilustración 25: Esquemático Circuito Emisor

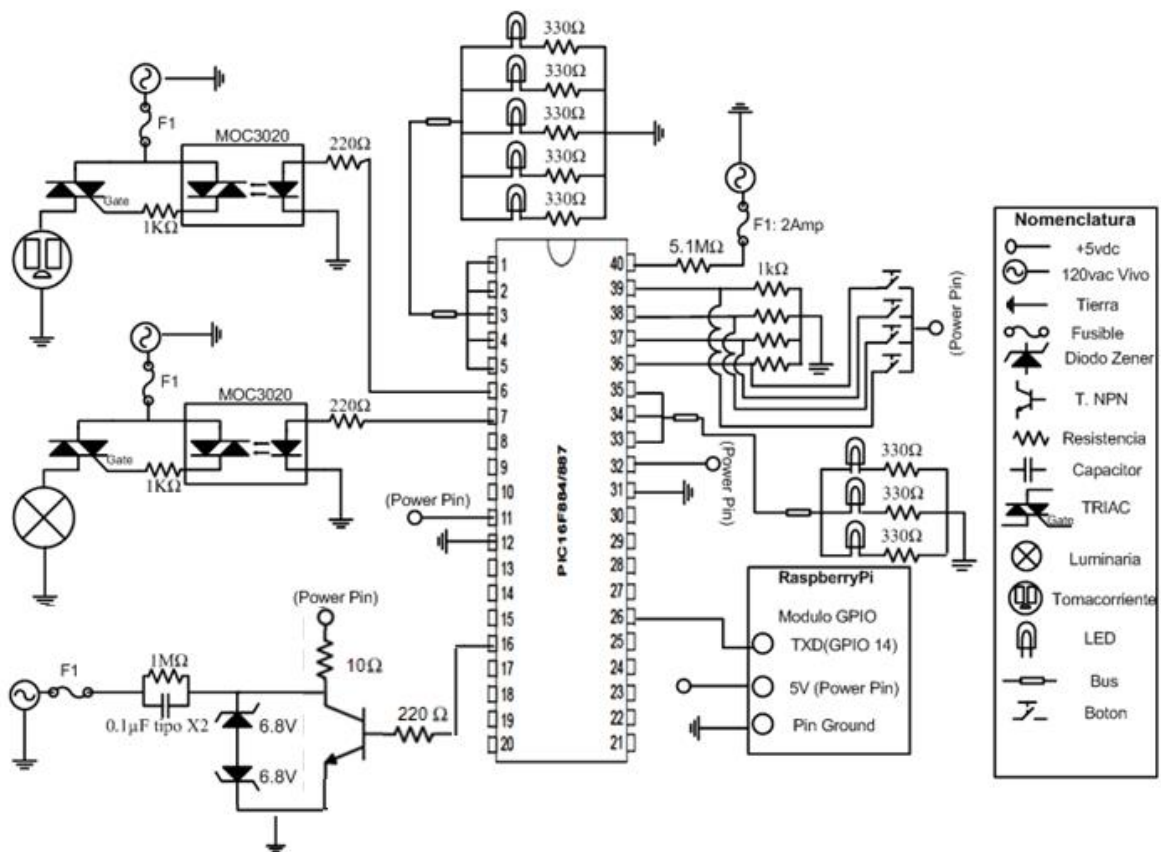


Ilustración 26: Diagrama PCB emisor

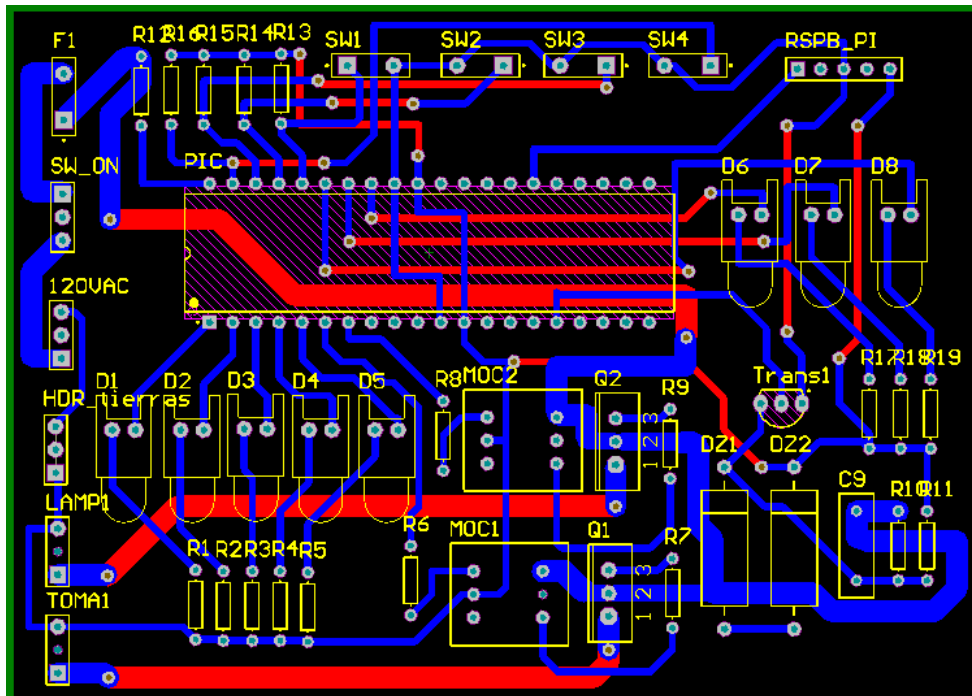
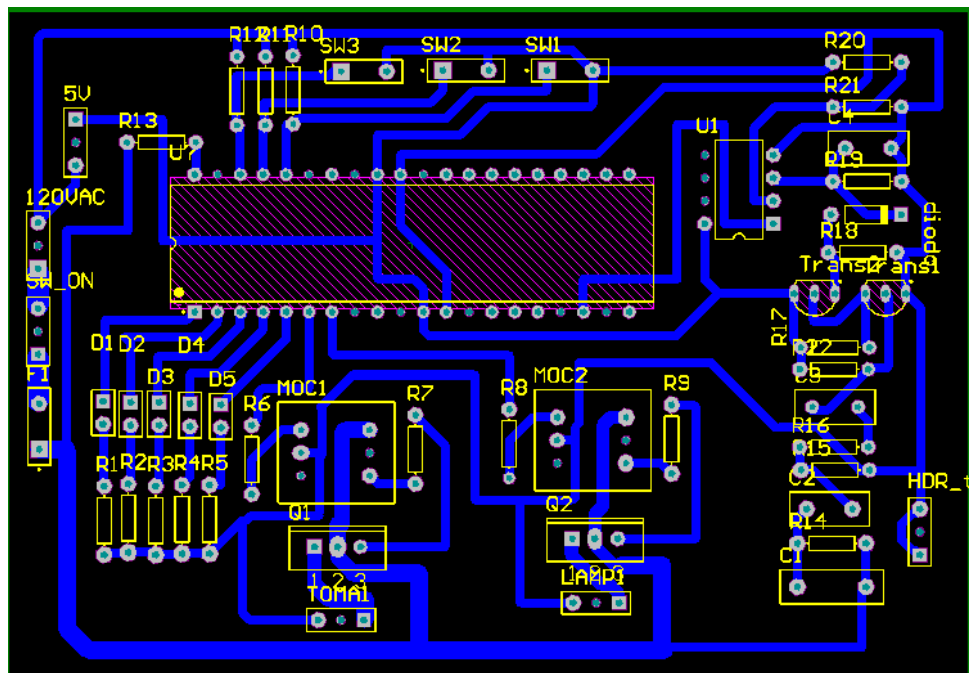


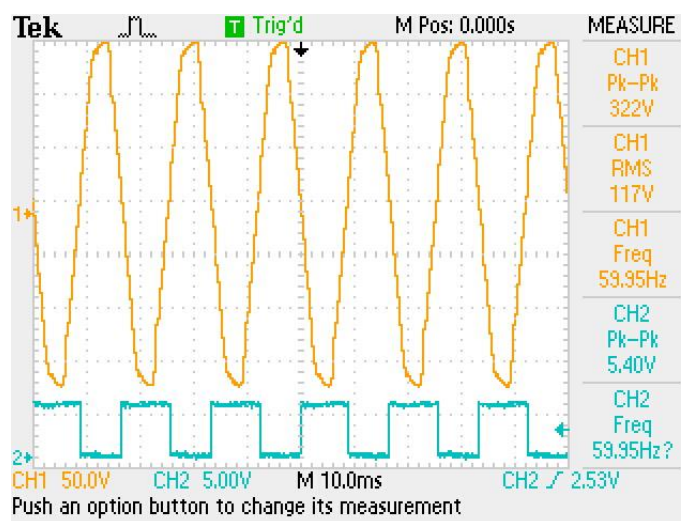
Ilustración 27: Diagrama PCB receptor



Los resultados obtenidos se muestran de forma independiente para cada etapa, debido a que cada una fue puesta a prueba de forma independiente para optimizarlas y luego unificarlas para obtener el resultado final de todo el sistema.

a. Detector de cruce por cero. En esta etapa las mediciones se realizaron en el micro controlador central, es decir el que se comunica en forma serial con el servidor Raspberry Pi. Se desarrolló un código de prueba, en el cual el micro controlador cambia el estado de un pin en el momento que el circuito detector de cruce por cero genera la interrupción por el Port-On-Change. La siguiente imagen muestra el resultado obtenido.

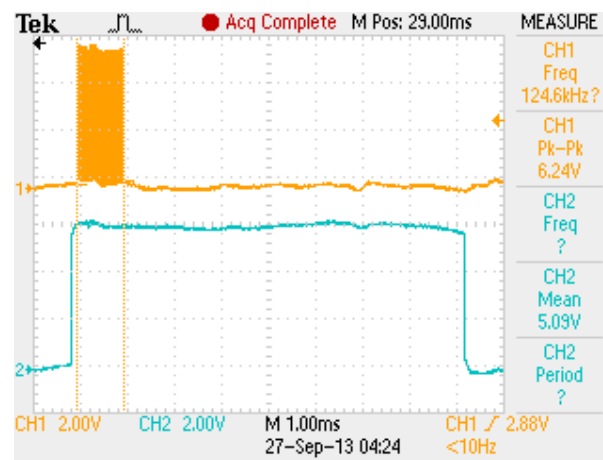
Ilustración 30: interrupción generada por el cruce por cero.



Cada cambio de estado del pin, el cual es representado por la señal 2, cuadrada azul, indica cuando el micro controlador ha entrado a la interrupción. Se observa que ocurre la interrupción cada vez que la señal AC cruza por cero. A partir de esta interrupción se generan los pulsos de 120kHz cuando sea necesario.

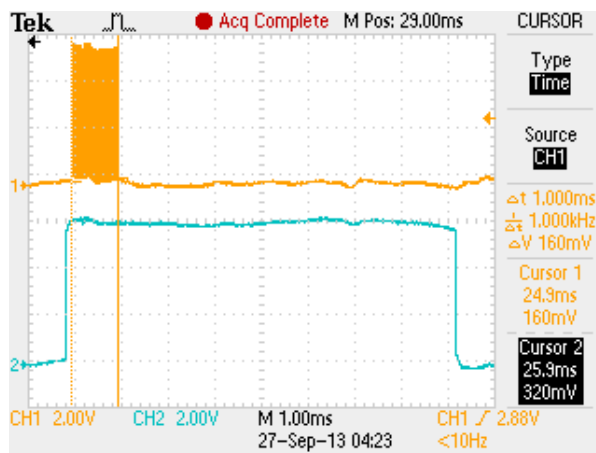
b. Generador de pulsos a 120KHz. En esta etapa se generan los pulsos de 120kHz cada vez que el micro controlador entra a la interrupción de cruce por cero. Cuando sea necesario representar '1' lógico en el cruce por cero, se genera la señal de 120kHz, el PWM es iniciado durante 1ms, luego de este tiempo el PWM es detenido. En la Ilustración 31 se muestra un ejemplo del pulso de 120kHz en los cruces por cero.

Ilustración 31: Pulsos de 120 kHz generados en el cruce por cero.



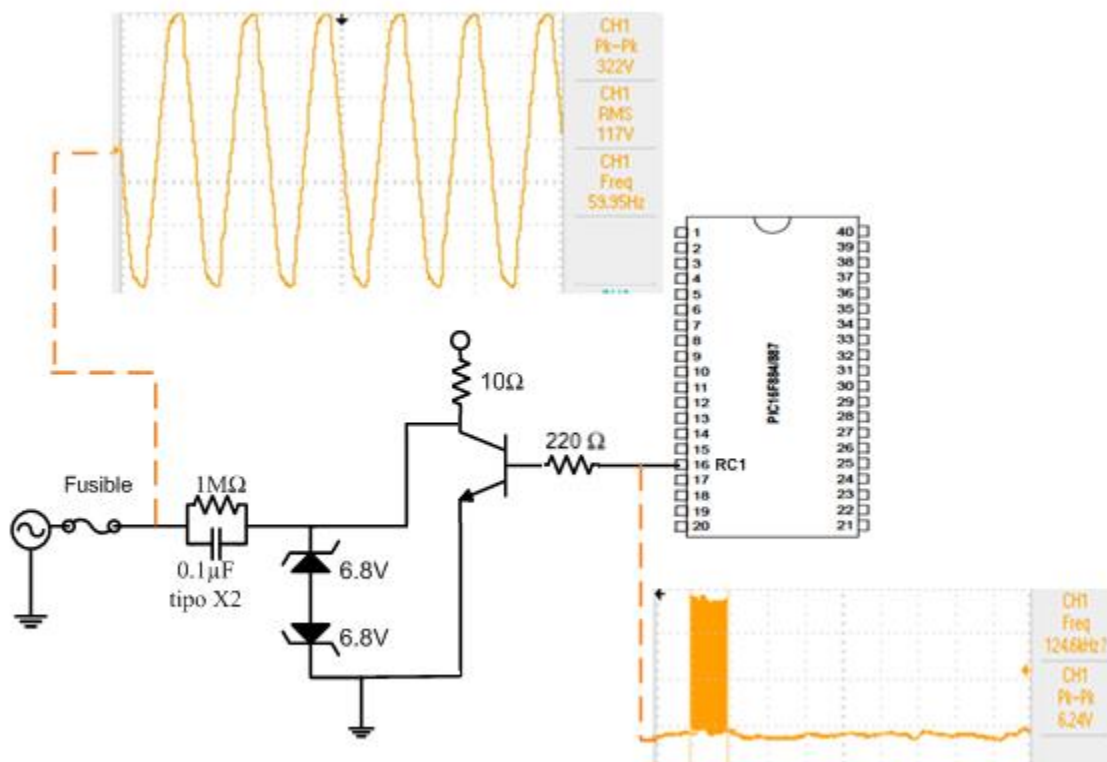
Este resultado fue obtenido realizando la medición en el pin que genera el pulso de 120kHz del micro controlador central. La señal 1 representa el pulso pwm y la señal 2 los flancos en cada interrupción del port on change debido al cruce por cero de la señal AC. Se observa que la amplitud de la señal inyectada en la red AC tiene una amplitud de voltaje zener 6.24V. La siguiente ilustración muestra la duración de los pulsos generados.

Ilustración 32: Duración de pulso 120kHz



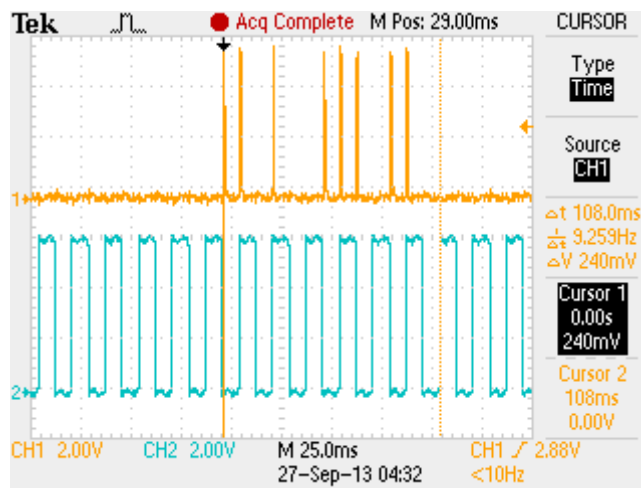
La Ilustración 33 muestra los puntos de medición en el circuito emisor.

Ilustración 33: Mediciones circuito emisor



Luego de generar pulsos en los cruces por cero se procede a transmitir comandos completos incluyendo el código de inicio y el inverso del comando. La Ilustración 34 muestra el resultado obtenido.

Ilustración 34: Comando completo generado

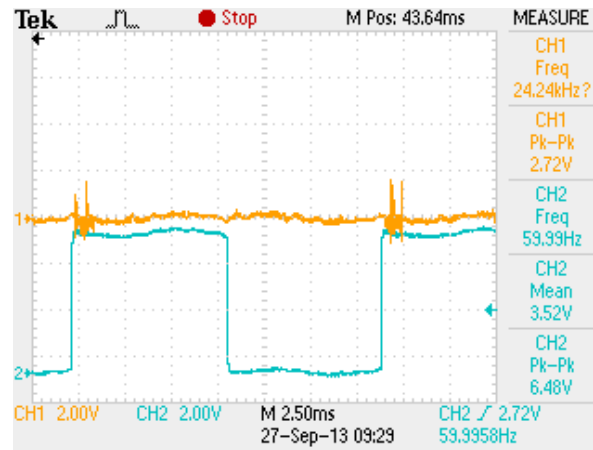


En la Ilustración 35, la señal 1 representan los pulsos a ser introducidos en la red AC y la señal 2 son las interrupciones generadas por el cruce por cero. El comando corresponde a: Inicio-Dispositivo 1-Luminaria-Incrementar-inverso equivalente en software a transmitir: 11010011101100.

c. Detector de pulsos a 120KHz. Nuevamente se realizan mediciones de forma independiente para cada etapa, para obtener una señal resultante que el micro controlador pueda interpretar correctamente. En esta etapa las mediciones se realizan en los circuitos detectores de pulsos que se encuentran conectados a los micro controladores remotos.

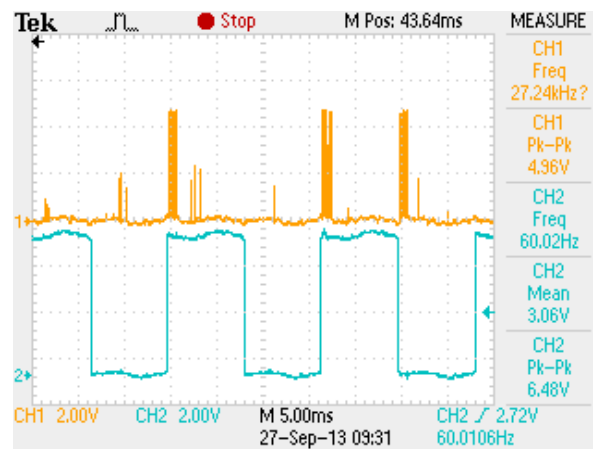
En la Ilustración 35 se observa el salida del filtro pasa alta.

Ilustración 35: Salida del filtro pasa alta



En la Ilustración 35 la señal 1 es la salida del filtro pasa alta, la señal es de referencia para el cruce por cero. Se observa que las señales con frecuencias bajas son atenuadas y las frecuencias altas no son afectadas. A pesar que la medición es realizada a menos de 1 metro de distancia del circuito emisor, se observa que la amplitud de la señal inyectada se ha reducido a 2.72V. Para obtener mejores resultados cuando la distancia aumenta, se implementa el amplificador transistorizado de señales pequeñas. La Ilustración 36 muestra su comportamiento teniendo como entrada la señal 1 de la Ilustración 35.

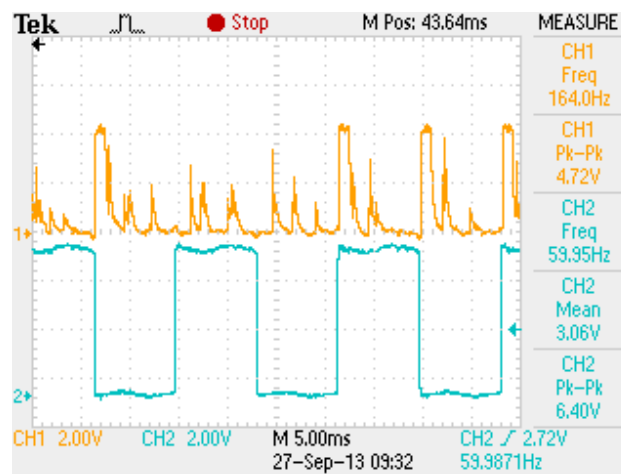
Ilustración 36: Salida obtenida del amplificador transistorizado



Este amplificador provee resultados esenciales para la recepción correcta de comandos, pero tiene la desventaja que además de amplificar las señales de comunicación también amplifica el ruido que se encuentra en la red AC. La señal 1 muestra su salida en la que se puede observar que la amplitud de las señales de comunicación ha aumentado a 4.96V, así como se observa el ruido amplificado.

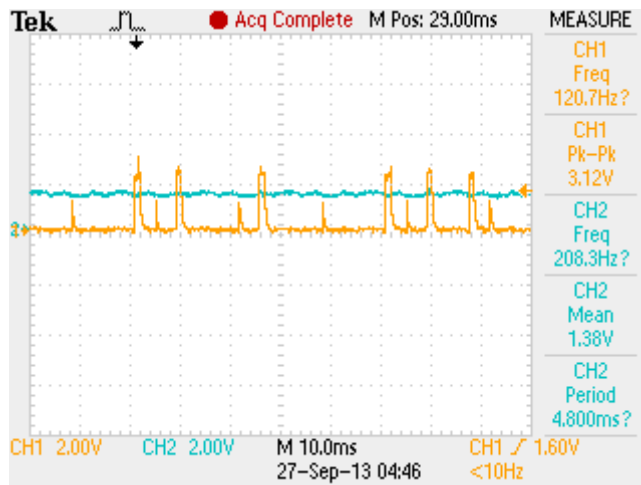
Luego esta señal es utilizada en la entrada del circuito detector envolvente, el cual se observa su resultado en la Ilustración 37.

Ilustración 37: Salida circuito detector envolvente



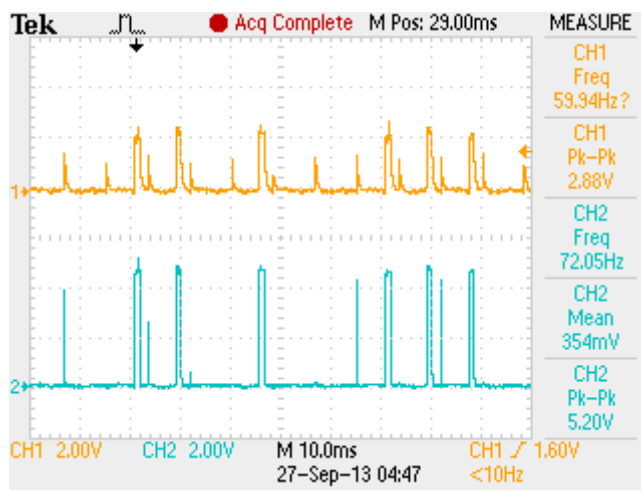
En la salida del detector envolvente se logra obtener el resultado deseado, ya que desde este punto se trabajara con una señal cuadrada con duración de 1ms en los cruces por cero. Tal como se observa en la Ilustración 29, el ruido también es representado con su envolvente y para reducir este efecto se introduce esta señal en un comparador, el cual solo transmitirá la parte de la señal que sea mayor al voltaje utilizado para comparación, este voltaje de comparación podrá ser aumentado o disminuido mediante el potenciómetro de precisión del divisor de voltaje. La Ilustración 38 muestra las señales a ser comparadas.

Ilustración 38: Señales comparadas



La señal 2 muestra el voltaje utilizado en comparación para reducir el ruido por parte de la señal 1. La Ilustración 39 muestra el resultado de la comparación.

Ilustración 39: Señal final



La Ilustración 39 muestra la señal que es introducida al micro controlador, esta señal siempre tendrá amplitud igual al voltaje de alimentación del amplificador operacional utilizado para la comparación, en este caso es de 5.20V, la cual es suficiente para que el micro controlador logre diferenciar entre “0” y “1” lógicos.

2. Distancia efectiva de comunicación

Los siguientes resultados muestran el porcentaje de comunicación efectivo entre el módulo generador y el módulo receptor de pulsos, en función de la distancia de separación de estos. La Tabla 10 y la Ilustración 40 muestran los resultados obtenidos en pruebas bajo condiciones controladas, es decir en un área donde no se encuentren equipos y motores eléctricos que induzcan ruido eléctrico a la red de alimentación AC.

Tabla 10: Distancia vrs porcentaje de comunicación bajo condiciones controladas

Distancia (metros)	10	20	30	40	50	60	70	80
Comandos enviados	15	15	15	15	15	15	15	15
Comandos recibidos	15	15	15	15	15	15	10	0
% Comunicación	100%	100%	100%	100%	100%	100%	67%	0%

Ilustración 40: Distancia vrs porcentaje de comunicación bajo condiciones controladas

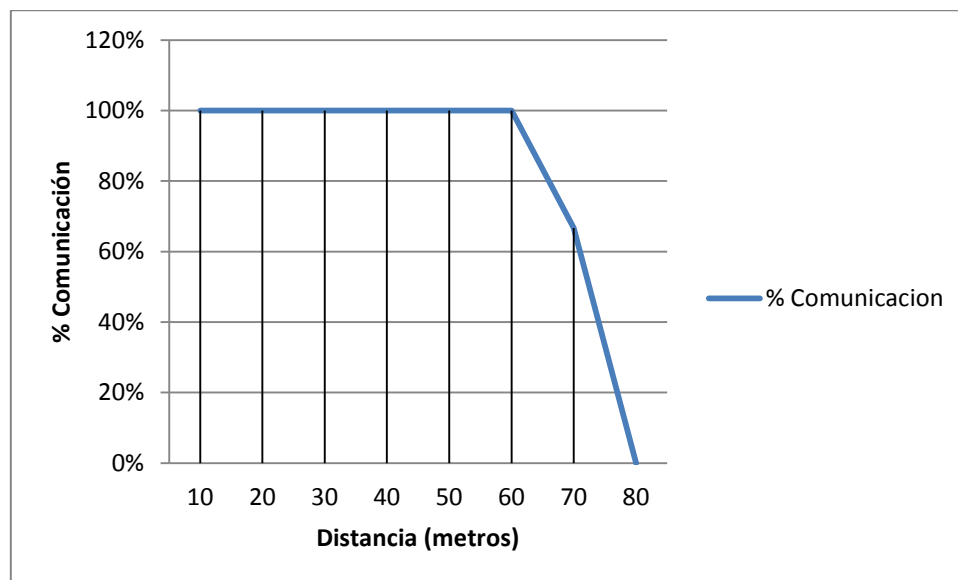
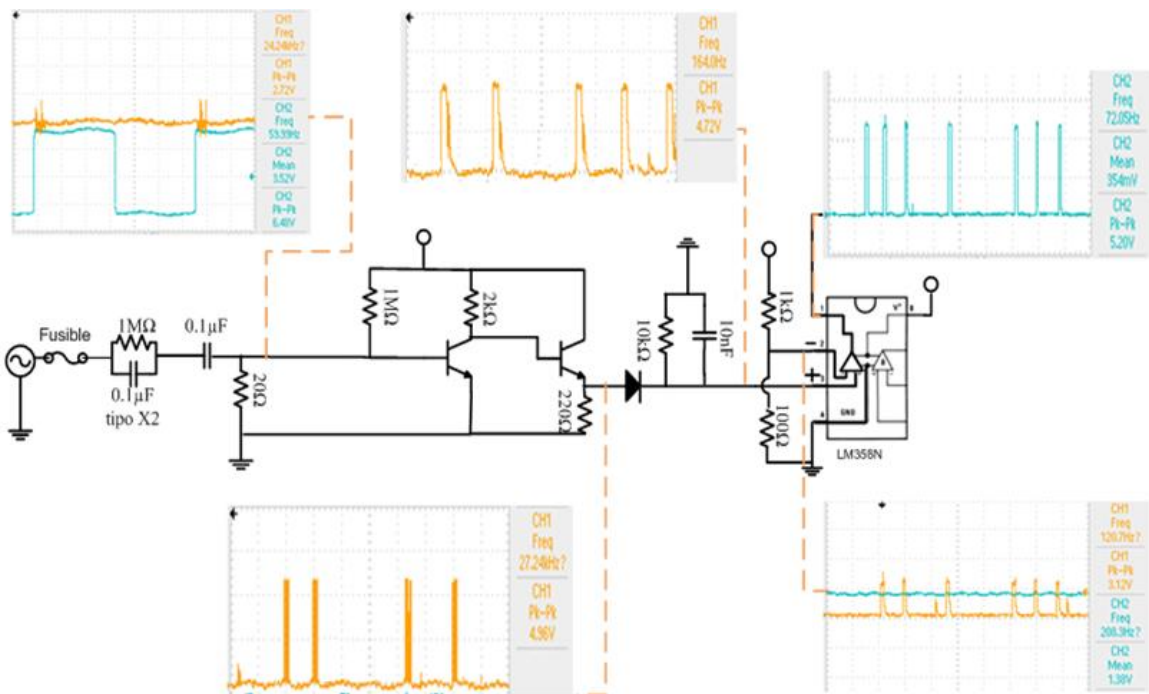


Ilustración 41: Mediciones bajo condiciones controladas



La Tabla 11 y las Ilustraciones 42-43 muestran los resultados obtenidos bajo simulación de condiciones cotidianas en uso de equipo eléctrico en residencias. En estas condiciones se encontraban trabajando a máxima capacidad un barreno de potencia de 700watts y una lavadora con un motor de 120VAC con 9.8 amperios de consumo, conectados al mismo tomacorriente que los módulos emisor y receptor.

Tabla 11: Distancia vrs porcentaje de comunicación bajo condiciones cotidianas en residencias

Distancia (metros)	10	20	30	40	50	60
Comandos enviados	15	15	15	15	15	15
Comandos recibidos	15	15	15	15	8	0
% Comunicación	100%	100%	100%	100%	53%	0%

Ilustración 42: Distancia vrs porcentaje de comunicación bajo condiciones cotidianas en residencias

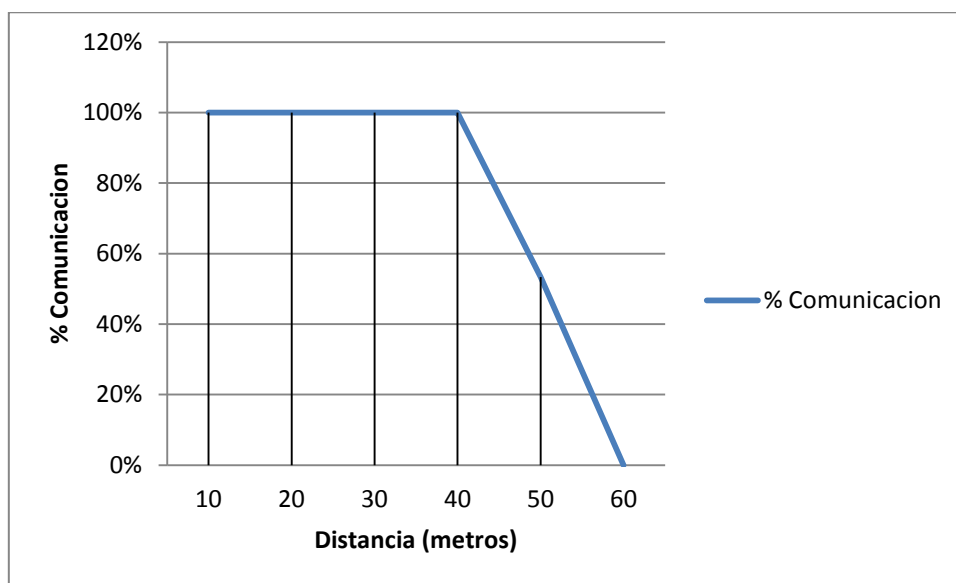
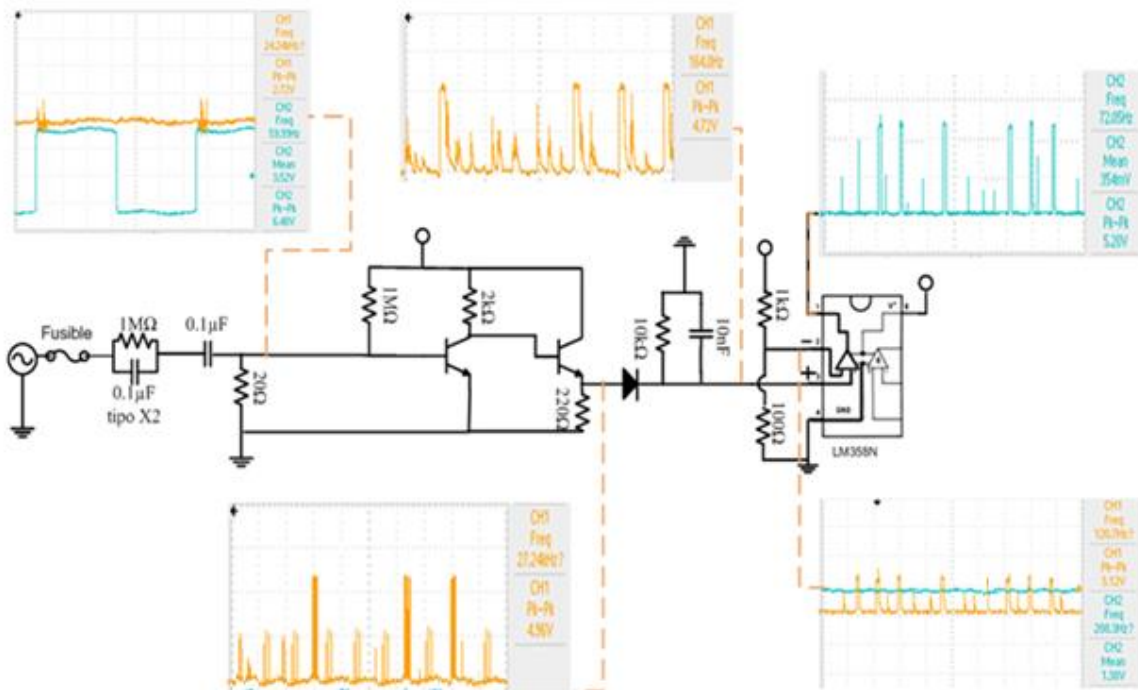


Ilustración 43: Mediciones bajo condiciones cotidianas en residencias



Para realizar las pruebas de distancia bajo condiciones cotidianas en residencias, se dejó conectado en el mismo tomacorriente el módulo emisor en conjunto con la lavadora y el barreno, los cuales se encontraban trabajando a máxima capacidad. Al mismo tiempo el módulo receptor se conectó en distintos tomacorrientes con el fin de aumentar la distancia de separación emisor-receptor. Cada vez que el módulo receptor se conectaba en un tomacorriente distinto, se midió la distancia de separación emisor-receptor, luego se enviaron comandos por medio del módulo emisor para obtener mediciones del lado del circuito receptor y de este modo encontrar la distancia efectiva bajo condiciones que simulan las cargas cotidianas en una residencia.

Ilustración 44: Conexiones para simulación cotidiana de residencias



La Tabla 12 y la Ilustración 45 muestran los resultados obtenidos en pruebas realizadas en un aula de 54m² ubicada en los laboratorios de Mecatrónica de la Universidad del Valle de Guatemala, en la cual durante las pruebas se encontraban encendidos equipos de cómputo y medición, donde además se conectaron a los tomacorrientes de los módulos directamente, dos ventiladores de 77 watts de potencia cada uno.

Tabla 12: Distancia vrs porcentaje de comunicación pruebas en aula

Distancia (metros)	2	4	10	12	14	16	18
Comandos enviados	15	15	15	15	15	15	15
Comandos recibidos	15	15	14	11	0	0	0
% Comunicacion	100%	100%	93%	73%	0%	0%	0%

Ilustración 45: Distancia vrs porcentaje de comunicación pruebas en aula

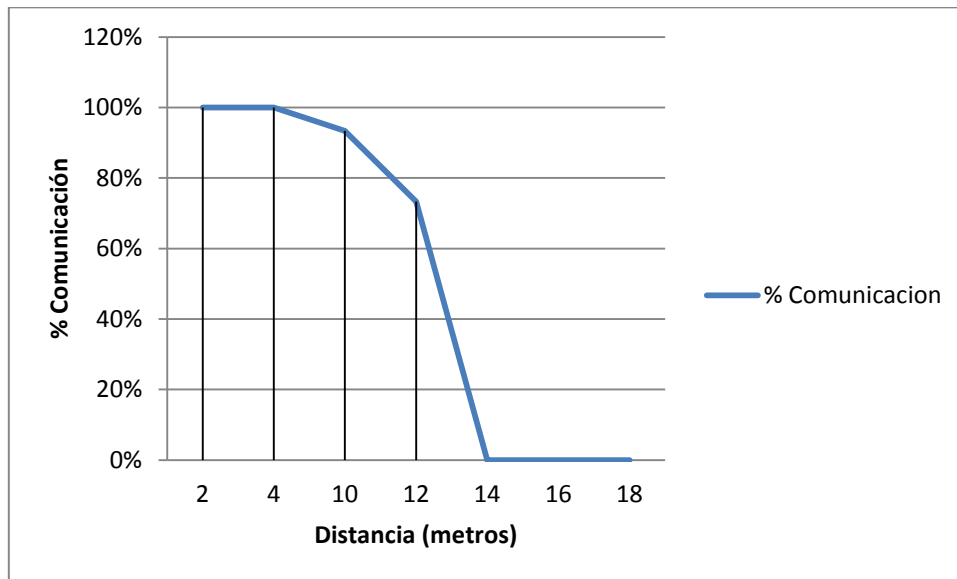


Ilustración 46: Conexiones para pruebas en aula

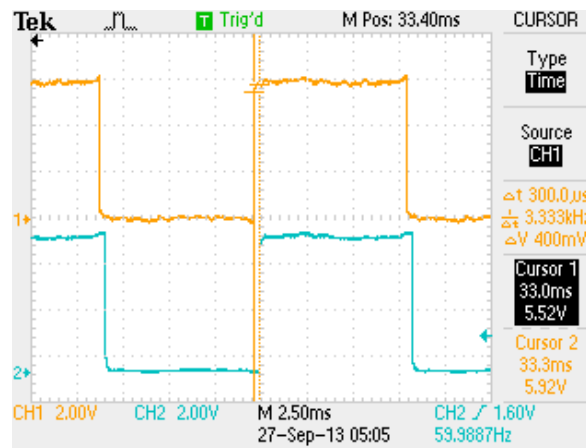


3. Recepción y ejecución de comandos

Luego de lograr enviar y recibir comandos con micro controladores usando Power Line Communications, es necesario realizar la adecuada programación de software para la interpretación correcta de los comandos. Una parte necesaria es esperar cierto tiempo después del cruce por cero para indicarle al micro

controlador que puede leer datos. La Ilustración 47 muestra el tiempo de espera antes de chequear si hay datos disponibles.

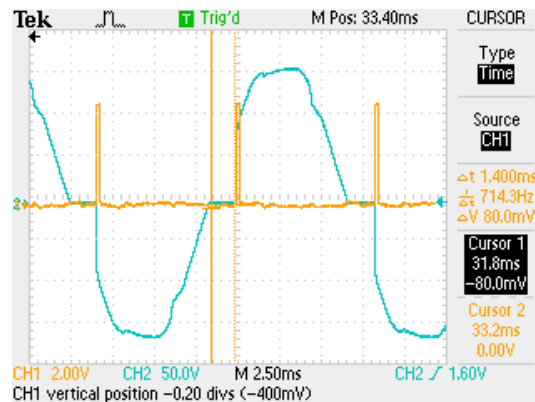
Ilustración 47: Tiempo de espera para lectura



La señal 1 representa la interrupción de cruce por cero, la señal 2 representa el instante en el que el micro controlador hace el cheque de datos. Tal como se observa en este caso el tiempo de espera es de 0.3ms, lo cual permitirá obtener una lectura fuera del rango de transición de señales. Esta espera ocurre en cada cruce por cero.

Luego de la correcta obtención de comandos, se procede a almacenarlos por medio de software y compararlos con la base de datos de todos los comandos para determinar que elemento debe realizar cierta función. A continuación se presenta el comportamiento de las luminarias realizando el envío de comandos desde la interfaz web.

Ilustración 48: Intensidad lumínica máxima



En la Ilustración 43, la señal cuadrada representa la activación del Gate en el Triac y la señal sinusoidal es la que alimenta a las luminarias. En dicha ilustración se muestra la intensidad máxima a la cual las luminarias estarán expuestas. Se observa que hay un retardo después del cruce por cero de 1.40ms, por lo tanto la luminaria es alimentada un aproximado de 7 ms por semi ciclo, equivalente a un 85% de 8.33ms. Este tiempo de 7ms por semi ciclo es utilizado como punto de referencia del 100% de intensidad, cada nivel de intensidad varía por un tiempo de 1.40ms para obtener 5 niveles de intensidad. Las siguientes ilustraciones muestran las intensidades de 75%, 50% y 25% respectivamente.

Ilustración 49: Intensidad lumínica al 75%

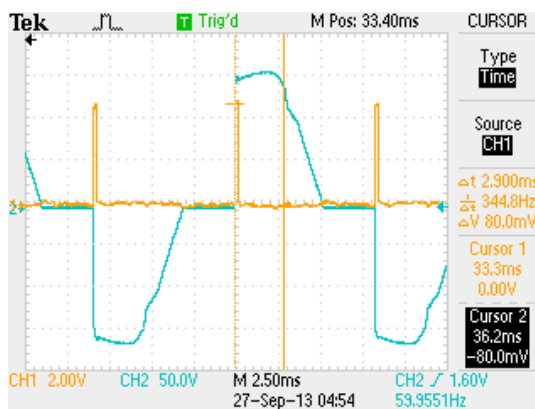


Ilustración 50: Intensidad lumínica al 50%

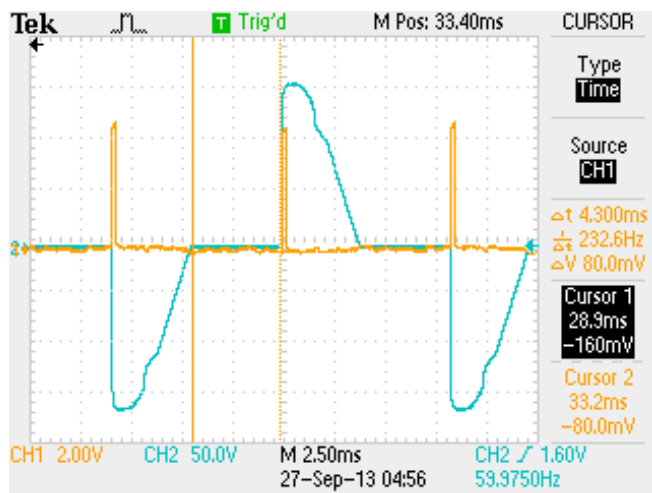
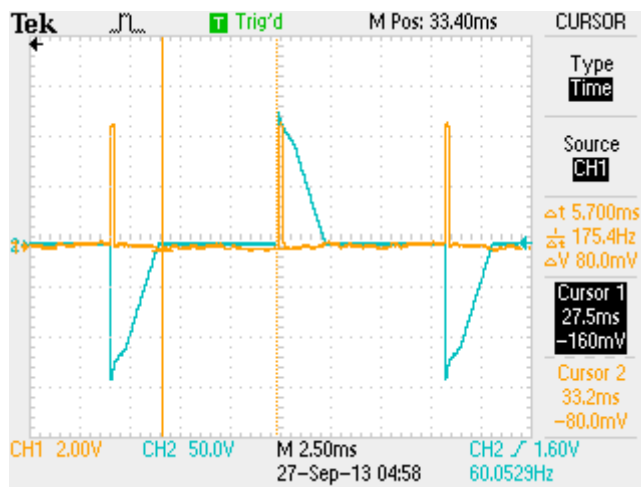
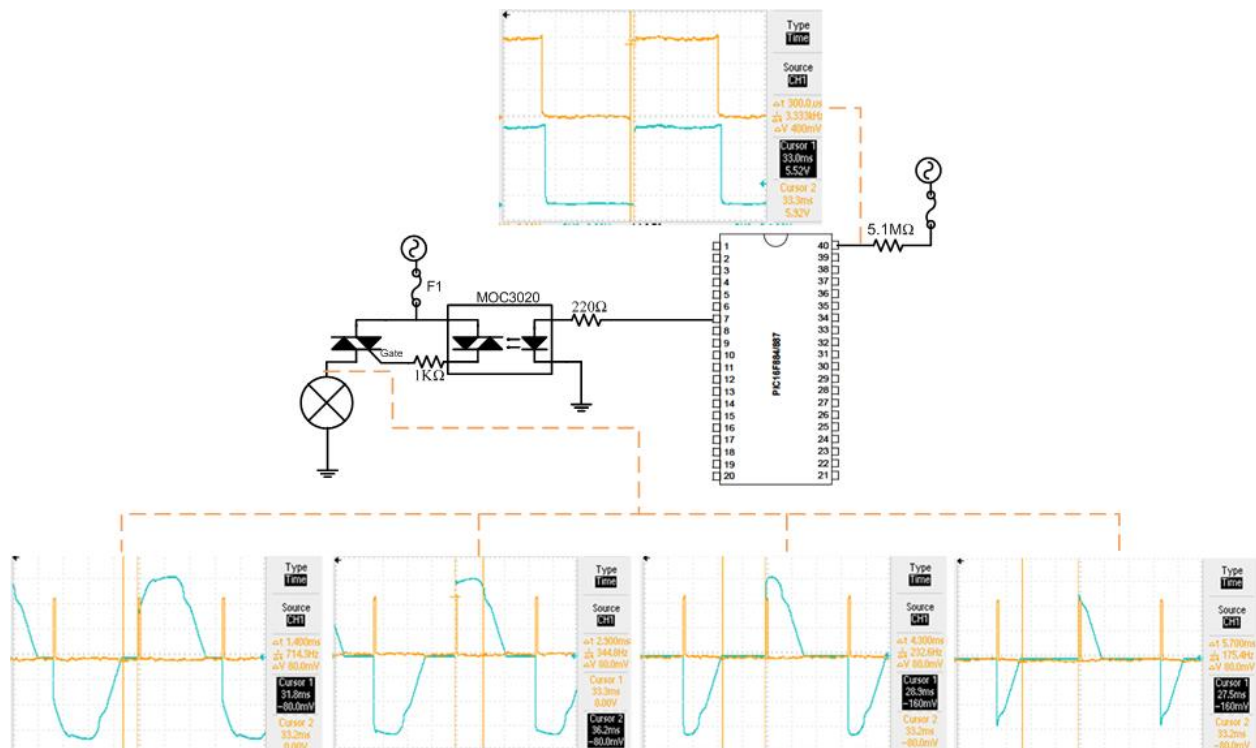


Ilustración 51: Intensidad lumínica al 25%



La Ilustración 52 muestra el punto de medición de la salida del TRIAC, con sus distintos modos de operación.

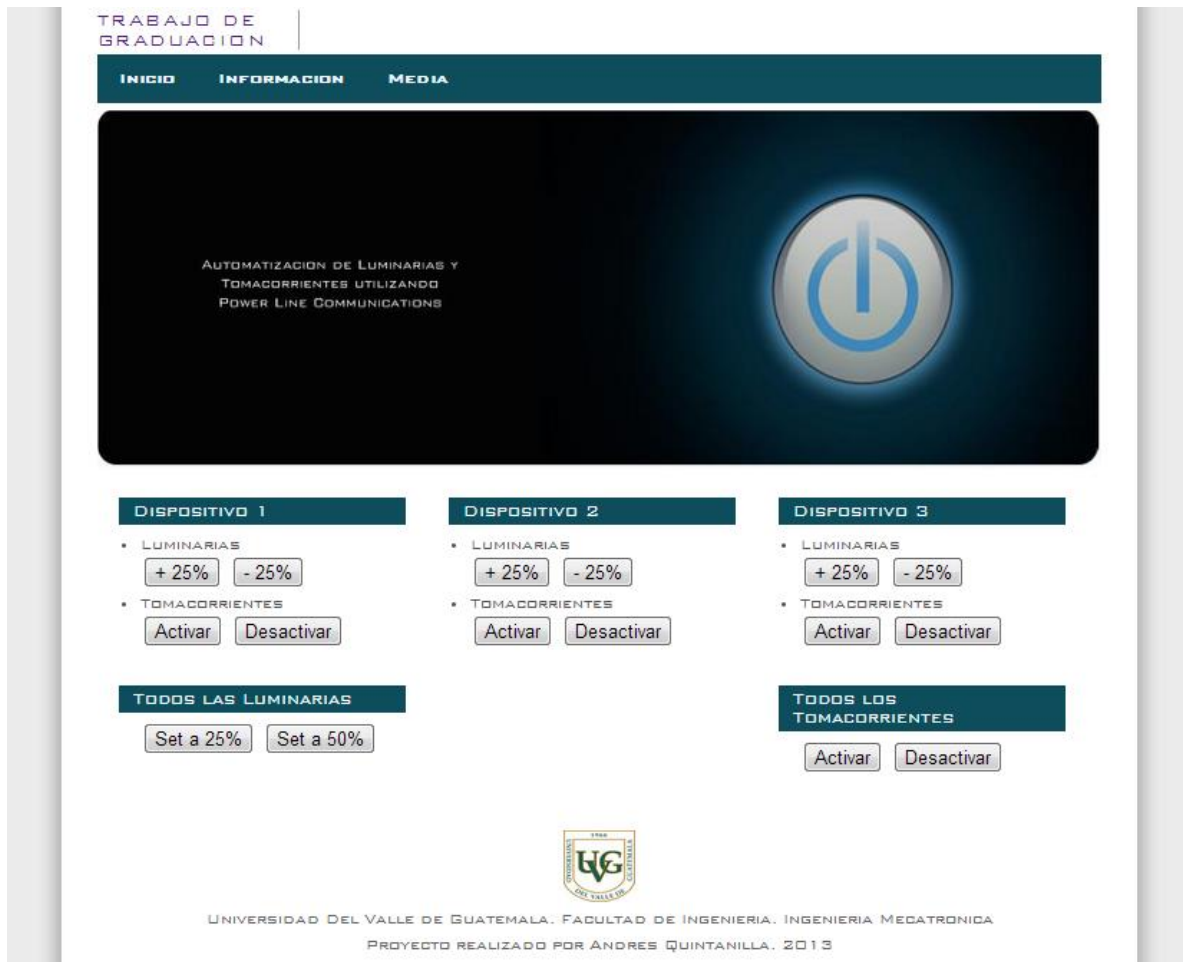
Ilustración 52: Puntos de operación de TRIAC



4. Website

La interfaz web se diseñó utilizando el lenguaje para HTML versión 4. Se programaron tres pantallas diferentes, en la cual la principal contiene los comandos para control del sistema y se muestra en la Ilustración 53.

Ilustración 53: Pantalla principal de interfaz



La Ilustración 54 muestra la pantalla de información que contiene los objetivos y conclusiones del proyecto.

Ilustración 54: Pantalla de información de website

TRABAJO DE GRADUACION

INICIO INFORMACION MEDIA

AUTOMATIZACION DE LUMINARIAS Y TOMACORRIENTES UTILIZANDO POWER LINE COMMUNICATIONS



TRABAJO DE GRADUACION PRESENTADO POR
JOSE ANDRES QUINTANILLA LOPEZ
PARA OPTAR AL GRADO ACADEMICO DE
LICENCIADO EN INGENIERIA MECATRONICA

EL OBJETIVO GENERAL DE ESTE PROYECTO ES DESARROLLAR UN SISTEMA DE AUTOMATIZACION DE TOMACORRIENTES Y LUMINARIAS, EN SISTEMAS ELECTRICOS CON ALIMENTACION 120VAC A 60HZ.

LOS OBJETIVOS ESPECIFICOS SON:

- *IMPLEMENTAR LA ESTRUCTURA POWER LINE COMMUNICATIONS EN MICRO CONTROLADORES.
- *IMPLEMENTAR EL CONTROL DE INTENSIDAD EN LUMINARIAS POR MEDIO DE POWER LINE COMMUNICATIONS.
- *COMUNICAR DE FORMA SERIAL EL DISPOSITIVO RASPBERRYPI CON MICROCONTROLADORES.
- *UTILIZAR EL RASPBERRYPI COMO SERVIDOR DEL SISTEMA.

LUEGO DE COMPLETADO EL DESARROLLO E IMPLEMENTACION DEL PROYECTO EN 3 DISPOSITIVOS, CADA UNO CON CONTROL LOCAL Y REMOTO PARA LUMINARIA Y TOMACORRIENTE, SE LLEGARON A LAS SIGUIENTES CONCLUSIONES:

- *EL TIEMPO NECESARIO PARA TRANSMITIR UN COMANDO COMPLETO DE MICRO CONTROLADOR A MICRO CONTROLADOR A TRAVES DE PLC, ES DE 116.66MS O 14 CRUCES POR CERO DE LA SEÑAL DE ALIMENTACION AC.
- *EL PROTOCOLO DE COMUNICACION QUE SE FABRICO, DEDICA 2 BITS PARA DIRECCIONAR A DISPOSITIVOS, PERMITIENDO CONTROLAR 4 DISPOSITIVOS DIFERENTES CON UN SOLO MAESTRO.
- *LA DISTANCIA EFECTIVA DE COMUNICACION ENTRE MICRO CONTROLADORES SE ENCUENTRA EN EL RANGO DE 0-20M.
- *EL PROTOCOLO DE COMUNICACION FABRICADO ES 666.64MS MAS RAPIDO QUE EL PROTOCOLO X10.
- *EL PROTOCOLO FABRICADO SE ENFOCA EN VELOCIDAD DE COMUNICACION.



UNIVERSIDAD DEL VALLE DE GUATEMALA, FACULTAD DE INGENIERIA, INGENIERIA MECATRONICA
PROYECTO REALIZADO POR ANDRES QUINTANILLA, 2013

VII. CONCLUSIONES

1. El tiempo necesario para transmitir un comando completo del micro controlador central según la estructura diseñada de PLC es de 116.666 ms o 14 cruces por cero de la señal AC, a una velocidad de comunicación de 120bps.
2. El protocolo de comunicación fabricado, dedica 2 bits para direccionar a dispositivos, permitiendo controlar 4 dispositivos diferentes con un solo maestro.
3. La distancia de comunicación efectiva al 100% entre micro controladores se encuentra en el rango de 0-60 metros bajo condiciones controladas.
4. El protocolo X10 necesita 783.3ms para transmitir un comando completo. El fabricado utiliza únicamente el 14.9% de dicho tiempo para su transmisión, con la desventaja de controlar 252 dispositivos menos que el protocolo X10.

VIII. RECOMENDACIONES

1. Se recomienda la utilizar filtros activos pasa banda de segundo orden en las etapas de filtrado de las señales de 120kHz, para mejorar la recepción de comandos.
2. Modificar el protocolo de comunicación para aumentar la cantidad de dispositivos a controlar con un solo maestro.
3. Se recomienda continuar la investigación enfocado a control de intensidad de luminarias con tecnología LED.
4. Estudiar la posibilidad de implementar un circuito inyector de señales con mayor robustez, que permita aumentar la distancia de comunicación entre dispositivos.

IX. BIBLIOGRAFÍA

- [1]. Balbas Contreras, Alejandro. 2004. *Control de una red de Automatización X10 a través del Correo Electrónico*. Universidad del Valle de Guatemala. Trabajo de graduación presentado para optar al grado académico de Licenciado en Ingeniería Electrónica. Guatemala. 117 páginas.
- [2]. Burmester, Juan Carlos. 2012. *Elementos Semiconductores de potencia*. Universidad del Valle de Guatemala. Curso Electrónica Industrial. 9 páginas.
- [3]. Burroughs, Jon. 2010. *X-10 Home Automation Using the PIC16F887A*. Microchip Technology Inc. 26 páginas. (En línea) Disponible y consultado Febrero de 2013.
http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1824&appnote=en012050
- [4]. Cline, Dave. 2007. *Design Guidelines for Small Signal Amplifier Circuits*. (En línea) Disponible y consultado Octubre de 2013.
<http://www.mysticmarvels.com/amplifier.html>
- [5]. Cox, Doug. 2002. *Interfacing to AC Power Lines*. Microchip Technology Inc. 4 páginas. (En Línea) Disponible y consultado Octubre de 2013.
<http://ww1.microchip.com/downloads/en/AppNotes/00521c.pdf>

- [6]. Esquit Carlos. 2011. *Curso de Electrónica del Estado Sólido*. Universidad del Valle de Guatemala. Ciclo 1.
- [7]. Fairchild Semiconductor Corporation. October 2011. *2N3904 NPN General Purpose Amplifier*. (En línea y disponible) Consultado junio de 2013. <http://www.fairchildsemi.com/ds/2N/2N3904.pdf>
- [8]. Freir, Thomas. Assistant Lecturer. *Analogue Modulation, COMM1208 Communication Systems 1*. Dublin Institute of Technology. School of Electronic and Communications Engineering. (En línea y disponible) Consultado Octubre de 2013. http://www.electronics.dit.ie/staff/tfreir/COMM1208/COMM1208_Unit3_AM.doc
- [9]. González Manuel. *Análisis de pasividad y estabilidad de un circuito detector de envolvente mediante el enfoque en variable de estado*. División académica de ingeniería y arquitectura de la UJAT, Tabasco, México. Universidad de las Américas-Puebla. (En línea) Disponible y consultado marzo de 2013. <http://www.izt.uam.mx/newpage/contactos/anterior/n65ne/pasividad.pdf>
- [10]. Microchip Technology. 2012. *PIC16F882/883/884/886/887 Data Sheet*. 236 Páginas.
- [11]. Nave Carl R. *Zener Limitador*. Hyperphysics. (En línea) Disponible y consultado marzo de 2013. <http://hyperphysics.phy-astr.gsu.edu/hbasees/Electronic/limiter.html>

- [12]. Ptak, Eric. *Webiopi*. Raspberry Pi internet of Things framework. (En línea) Disponible y consultado marzo de 2013. <https://code.google.com/p/webiopi/>
- [13]. Regulus by Binary Moon. PCB Trace Width Calculator. (En Línea) Disponible y consultado noviembre de 2013. <http://circuitcalculator.com/wordpress/2006/01/31/pcb-trace-width-calculator/>
- [14]. Reston, Condit. 2010. *Transformerless Power Supplies: Resistive and Capacitive*. Microchip Technology Inc. (En línea) Disponible y consultado febrero de 2013. <http://ww1.microchip.com/downloads/en/AppNotes/00954A.pdf>
- [15]. Taylor & Francis Group. 2013. *Analysis of Small-signal Transistor Amplifiers*. (En Línea) Disponible y consultado junio de 2013. http://cw.routledge.com/textbooks/eresources/9780750687379/appendices/Small-signal_Transistor_Amplifiers.pdf
- [16]. W3C Org. *HTML & CSS*. (En línea) Disponible y consultado marzo de 2013. <http://www.w3.org/standards/webdesign/htmlcss>

X. APÉNDICE

1. Código de micro controlador central

```
/*
Universidad Del Valle de Guatemala
Facultad de Ingenieria
Departamento de Ingeniera Mecatronica
Trabajo de graduacion
Codigo micro controlador central inyector de senales
Andres Quintanilla
*****
*****Declaracion de Variables*****
*****/

int INICIO_CMD[]   ={0,1,1};
int D1_T_ON[]      ={0,1,1,0,0,1};
int D1_T_OFF[]     ={0,1,1,0,1,0};
int D1_L_INC[]     ={0,1,0,0,1,1};
int D1_L_DEC[]     ={0,1,0,1,0,0};
int D2_T_ON[]      ={1,0,1,0,0,1};
int D2_T_OFF[]     ={1,0,1,0,1,0};
int D2_L_INC[]     ={1,0,0,0,1,1};
int D2_L_DEC[]     ={1,0,0,1,0,0};
int D_ALL_L_50[]   ={1,1,0,1,0,1};
int D_ALL_L_25[]   ={1,1,0,1,1,0};
int D_ALL_T_ON[]   ={1,1,1,0,0,1};
int D_ALL_T_OFF[]  ={1,1,1,0,1,0};
int ii;
int jj;
int LEER;
bit cruce_cero;
bit enviar_inicio;
bit enviar_matriz;
bit enviar_inverso;
bit COMANDO_ENVIADO;
int CICLOS;
bit PUEDE_ENVIAR;
bit PUEDE_LEER;
bit PROBAR_FOCO1;
bit FLANCO_ZERO;
int CONTADOR_FOCO1;
int DIMMER;
int INDICADOR;
```

```

bit bandera_boton_d1;
bit bandera_boton_d2;
bit oldstate_rb3;           // Old state flag
bit oldstate_rb4;           // Old state flag
bit oldstate_rb5;           // Old state flag
bit oldstate_rb6;           // Old state flag
/*****
*****Declaracion de metodos*****
*****/
void iniciar_interrupciones() {
    OPTION_REG = 0x82;    // Assign prescaler to TMR0, 1:8
    TMR0 = 78;           // Timer0 initial value
    ANSEL = 0X00;        // Configure AN pins as digital I/O
    ANSELH = 0X00;
    C1ON_bit = 0;        // Disable comparators
    C2ON_bit = 0;
    INTCON = 0xE8;        // Enable TMRO, EUSART, PORTB CHANGE ON interrupt
    IOCB=0X80;           //IOCB BIT 7 ENABLE
    cruce_cero=0;
}
/*****
void iniciar_puertos(){
    TRISA=0x00;           //PORTA OUTPUT
    TRISB = 0xF8;        // PORTB B7-B3 INPUT, B2-B0 OUTPUT
    TRISC=0xF7;          // PORTC ES INPUT, RC3 OUTPUT
    PORTA=0x00;           // LIMPIA PUERTO
    PORTB=0x00;           // LIMPIA PUERTO
    PORTC=0X00;           // LIMPIA PUERTO
    LEER=0;
    COMANDO_ENVIADO=1;    // INICIA VARIABLES
    CICLOS=0;             // CONTROLA LOS CICLOS DE ESPERA
    PUEDE_ENVIAR=0;      // DEFINE SI PUEDE ENVIAR OTRO MENSAJE, =1 PUEDE
    PUEDE_LEER=1;
    enviar_inicio=1;
    enviar_matriz=0;
    enviar_inverso=0;
    ii=0;
    jj=0;
    CONTADOR_FOCO1=1;
    PROBAR_FOCO1=0;
    DIMMER=1;
    FLANCO_ZERO=0;
    INDICADOR=0;
    PORTB.B0=1;
    bandera_boton_d1=0;

```

```

bandera_boton_d2=0;
oldstate_rb3 = 0;
oldstate_rb4 = 0;
oldstate_rb5 = 0;
oldstate_rb6 = 0;
}
/*****/
void iniciar_pwm(){
    PWM2_Init(120000);          // Initialize PWM1 module at 120KHz
    PWM2_Set_Duty(125);        //DUTY AL 50%
    PIE1.RCIE=1;               //HABILITA EUSART RECEIVE INTERRUPT
    UART1_Init(9600);          // Initialize UART module at 9600 bps
    Delay_ms(100);
}
/*****/
void ENVIAR_X10(int MATRIZ_ENTRADA[]){
    if(enviar_matriz==0){
        if(INICIO_CMD[jj]==1){
            PWM2_Start();      //ENVIA EL PULSO DE 120kHz POR 1MS
            Delay_ms(1);
            PWM2_STOP();
        }
        jj++;
        if(jj==3){
            enviar_matriz=1;
            jj=0;
        }
    }
    else{
        if(enviar_matriz==1){
            if(enviar_inverso==0){
                if(MATRIZ_ENTRADA[ii]==1){
                    PWM2_Start();    //ENVIA EL PULSO DE 120kHz POR 1MS
                    Delay_ms(1);
                    PWM2_STOP();
                }
                ii++;
                if(ii==6){
                    enviar_inverso=1;
                    ii=0;
                }
            }
        }
        else{
            if(MATRIZ_ENTRADA[ii]!=1){
                PWM2_Start();    //ENVIA EL PULSO DE 120kHz POR 1MS
            }
        }
    }
}

```

```

    Delay_ms(1);
    PWM2_STOP();
  }
  ii++;
  if(ii==6){
    ii=0;
    COMANDO_ENVIADO=1;
    PUEDE_ENVIAR=0;
    LEER=0X00;
    enviar_matriz=0;
    enviar_inverso=0;
  }
  else{
    COMANDO_ENVIADO=0;
  }
}

}
}
}
/*****/
void TOMA_ON_OFF() {
    PORTA.B4=~PORTA.B4;
    LEER=0;
}
/*****/
void LUZ_INC() {
    LEER=0;
    if(CONTADOR_FOCO1>1){
        CONTADOR_FOCO1--;
    }
    switch(CONTADOR_FOCO1){
        case 4: PORTA.B0=1; break; //INTENSIDAD AL 25
        case 3: PORTA.B1=1; break; //INTENSIDAD AL 50
        case 2: PORTA.B2=1; break; //INTENSIDAD AL 75
        case 1: PORTA.B3=1; break; //INTENSIDAD AL 100
    }
}
/*****/
void LUZ_DEC() {
    LEER=0;
    if(CONTADOR_FOCO1<5){
        CONTADOR_FOCO1++;
    }
    switch(CONTADOR_FOCO1){

```

```

case 2: PORTA.B3=0, PORTA.B2=1, PORTA.B1=1, PORTA.B0=1 ; break; //INTENSIDAD AL
25
case 3: PORTA.B2=0; break; //INTENSIDAD AL 50
case 4: PORTA.B1=0; break; //INTENSIDAD AL 75
case 5: PORTA.B0=0; break; //INTENSIDAD AL 100
}
}
/*****/
void comando_boton(){
    if (Button(&PORTB, 4, 1, 1)) { // BOTON DECREMENTAR
        oldstate_rb4 = 1;
    }
    if (oldstate_rb4 && Button(&PORTB, 4, 1, 0)) { // Detect one-to-zero transition
        oldstate_rb4 = 0;
        switch(INDICADOR){
            case 0: LUZ_DEC() ; break; //D0
            case 1: LEER=0X34; break; //D1
            case 2: LEER=0X38; break; //D2
        }
    }
    if (Button(&PORTB, 5, 1, 1)) { // BOTON INCREMENTAR
        oldstate_rb5 = 1;
    }
    if (oldstate_rb5 && Button(&PORTB, 5, 1, 0)) { // Detect one-to-zero transition
        oldstate_rb5 = 0;
        switch(INDICADOR){
            case 0: LUZ_INC() ; break; //D0
            case 1: LEER=0X33; break; //D1
            case 2: LEER=0X41; break; //D2
        }
    }
    if (Button(&PORTB, 6, 1, 1)) { // BOTON ON OFF
        oldstate_rb6 = 1; // Update flag
    }
    if (oldstate_rb6 && Button(&PORTB, 6, 1, 0)) { // Detect one-to-zero transition
        oldstate_rb6 = 0;
        if(INDICADOR==0){
            TOMA_ON_OFF() ;
        }
        if(INDICADOR==1){
            if(bandera_boton_d1==0){
                LEER=0X31;
                bandera_boton_d1=1;
            }
            else{

```



```

    case 0X34: ENVIAR_X10(D1_L_DEC); break;
    case 0X47: ENVIAR_X10(D2_T_ON); break;
    case 0X30: ENVIAR_X10(D2_T_OFF); break;
    case 0X41: ENVIAR_X10(D2_L_INC); break;
    case 0X38: ENVIAR_X10(D2_L_DEC); break;
    case 0X35: ENVIAR_X10(D_ALL_L_50), CONTADOR_FOCO1=3, PORTA.B3=0, PORTA.B2=0,
PORTA.B1=1, PORTA.B0=1; break;
    case 0X36: ENVIAR_X10(D_ALL_L_25), CONTADOR_FOCO1=4, PORTA.B3=0, PORTA.B2=0,
PORTA.B1=0, PORTA.B0=1; break;
    case 0X43: ENVIAR_X10(D_ALL_T_ON), PORTA.B4=1; break;
    case 0X56: ENVIAR_X10(D_ALL_T_OFF), PORTA.B4=0; break;
    }
}
/*****
*****INTERRUPCIONES*****
***
*****/
void interrupt() {
    if (TMROIF_bit) {          //INTERRUPCION TIMER0
    TMROIF_bit = 0;          // clear TMROIF
    TMRO = 78;              // ASIGNA EL AJUSTE PARA EL OVERFLOW
    PORTA.B5=0;
    if(DIMMER!=5){          //desactiva gate de triacs
    if(FLANCO_ZERO==1){
        if(DIMMER>0){ //ESPERA LOS CICLOS DEL TIMER INDICADOS POR DIMMER
            DIMMER--;
        }
        if(DIMMER==0){
            PORTA.B5=1;
            Delay_us(100); // DELAY DE APAGADO DE GATE
            PORTA.B5=0;
            FLANCO_ZERO=0;
        }
    }
}
}
/*****
if (INTCON.RBIF) {          //INTERRUPCION CRUCE POR CERO
    ver_botones();
    INTCON.RBIF=0;          //limpia bandera de interrupcion
    TMROIF_bit = 0;          // clear TMROIF
    DIMMER=CONTADOR_FOCO1;
    FLANCO_ZERO=1;
    TMRO=78;                //AJUSTE DEL TIMER
    if(COMANDO_ENVIADO==1){

```

```

        CICLOS++;
    }
    if(CICLOS==6){
        PUEDE_ENVIAR=1;
        CICLOS=0;
    }
    if(PUEDE_ENVIAR){
        cruce_cero=1;
    }
}
/*****
if(PIR1.RCIF){          //Interrupcion de eusart
    LEER=UART1_Read();
    PIR1.RCIF=0;
}
}
/*****
*****Ciclo Principal*****
*****/
void main() {
    iniciar_interrupciones();
    iniciar_puertos();
    iniciar_pwm();
    LUZ_DEC();
    TOMA_ON_OFF();
    while(1) {
        if(cruce_cero==1 && LEER!=0X00){
            cruce_cero=0;
            ver_serial();
        }

    }
    //quedarse en el ciclo do
}

```



```

/*****
*****Declaracion de metodos*****
*****/
void INICIAR_PUERTOS(){
    TRISA=0x00;        //PORTA INPUT
    TRISB = 0xB8;      // PORTB ES OUTPUT, RB7 INPUT
    TRISC3_bit =1;    //RC3 INPUT PARA PWM
    PORTA=0x00;        //LIMPIA PUERTOS
    PORTB =0x00;       //LIMPIA PUERTOS
    PORTC = 0X00;

    CAMBIAR_LED=1;
    CONTADOR_FOCO1=1;
    PROBAR_FOCO1=0;
    DIMMER=1;
    FLANCO_ZERO=0;
    leer_comando=0;
    leer_otro=0;
    i=0;
    j=0;
    CHECK_INICIO=0;
    CHECK_D1_T_ON=0;
    CHECK_D1_T_OFF=0;
    CHECK_D1_L_INC=0;
    CHECK_D1_L_DEC=0;
    CHECK_D_ALL_L_50=0;
    CHECK_D_ALL_L_25=0;
    CHECK_D_ALL_T_ON=0;
    CHECK_D_ALL_T_OFF=0;
    oldstate_rb3 = 0;
    oldstate_rb4 = 0;
    oldstate_rb5 = 0;

}
/*****
void INICIAR_INTERRUPCIONES(){
    OPTION_REG = 0x82;    // Assign prescaler to TMR0, 1:8
    ANSEL = 0X04;        // Configure AN pins as digital
    ANSELH = 0X00;
    C1ON_bit = 0;        // Disable comparators
    C2ON_bit = 0;
    TMR0 = 78;           // Timer0 initial value
    INTCON = 0xA8;       // Enable TMRO interrupt
    IOCB=0X80;           //IOCB BIT 7 ENABLE PARA RB7 INTERRUPT ON CHANGE

```

```

}
/*****/
void VER_LEDS(){
  CAMBIAR_LED=0;
  switch(CONTADOR_FOCO1){
    case 1: PORTA.B3=1, PORTA.B1=1, PORTA.B0=1; break; //INTENSIDAD AL 100
    case 2: PORTA.B3=0, PORTA.B1=1, PORTA.B0=1; break; //INTENSIDAD AL 75
    case 3: PORTA.B3=0, PORTA.B1=1, PORTA.B0=1; break; //INTENSIDAD AL 50
    case 4: PORTA.B3=0, PORTA.B1=0, PORTA.B0=1; break; //INTENSIDAD AL 25
    case 6: PORTA.B3=0, PORTA.B1=0, PORTA.B0=0; break; //INTENSIDAD AL 0
  }
  for(j=0; j<12; j++){
    LEER_X10[j]=0;
  }
}
/*****/
void LUZ_INC() {
  CAMBIAR_LED=1;
  if(CONTADOR_FOCO1>1 && CONTADOR_FOCO1!=6){
    CONTADOR_FOCO1--;
  }
  if(CONTADOR_FOCO1==6){
    CONTADOR_FOCO1=4; //INTENSIDAD AL 25
  }
}
/*****/
void LUZ_DEC() {
  CAMBIAR_LED=1;
  if(CONTADOR_FOCO1<5){
    CONTADOR_FOCO1++;
  }
  if(CONTADOR_FOCO1==5){
    CONTADOR_FOCO1=6;
  }
}
/*****/
void RECIBIR_CMD(){
  LEER_X10[i]=PORTC.B3;
  i++;
  if(i==12){ //CICLO DE COMPARACION DE ARRAYS
    i=0;
    leer_comando=0;
    for(j=0; j<6; j++){
      if(LEER_X10[j]==D1_T_ON[j] && D1_T_ON[j]!=LEER_X10[j+6]){

```

```

    CHECK_D1_T_ON++;
  }
  if((LEER_X10[j]==D1_T_OFF[j] && D1_T_OFF[j]!=LEER_X10[j+6]){
    CHECK_D1_T_OFF++;
  }
  if((LEER_X10[j]==D1_L_INC[j] && D1_L_INC[j]!=LEER_X10[j+6]){
    CHECK_D1_L_INC++;
  }
  if((LEER_X10[j]==D1_L_DEC[j] && D1_L_DEC[j]!=LEER_X10[j+6]){
    CHECK_D1_L_DEC++;
  }
  if((LEER_X10[j]==D_ALL_L_50[j] && D_ALL_L_50[j]!=LEER_X10[j+6]){
    CHECK_D_ALL_L_50++;
  }
  if((LEER_X10[j]==D_ALL_L_25[j] && D_ALL_L_25[j]!=LEER_X10[j+6]){
    CHECK_D_ALL_L_25++;
  }
  if((LEER_X10[j]==D_ALL_T_ON[j] && D_ALL_T_ON[j]!=LEER_X10[j+6]){
    CHECK_D_ALL_T_ON++;
  }
  if((LEER_X10[j]==D_ALL_T_OFF[j] && D_ALL_T_OFF[j]!=LEER_X10[j+6]){
    CHECK_D_ALL_T_OFF++;
  }
}
if(CHECK_D1_T_ON==6 || CHECK_D_ALL_T_ON==6){
  PORTA.B4 = 1;    // Toggle PORTB LEDs
}
if(CHECK_D1_T_OFF==6 || CHECK_D_ALL_T_OFF==6){
  PORTA.B4 = 0;
}
if(CHECK_D1_L_INC==6){
  LUZ_INC();
}
if(CHECK_D1_L_DEC==6){
  LUZ_DEC();
}
if(CHECK_D_ALL_L_50==6){
  CAMBIAR_LED=1;
  CONTADOR_FOCO1=3;
}
if(CHECK_D_ALL_L_25==6){
  CAMBIAR_LED=1;
  CONTADOR_FOCO1=4;
}
CHECK_D1_T_ON=0;

```

```

CHECK_D1_T_OFF=0;
CHECK_D1_L_INC=0;
CHECK_D1_L_DEC=0;
CHECK_D_ALL_L_50=0;
CHECK_D_ALL_L_25=0;
CHECK_D_ALL_T_ON=0;
CHECK_D_ALL_T_OFF=0;
}
}
/*****/
void LEER_INICIO(){
LEER_X10[i]=PORTC.B3;
i++;
if(i==2){
i=0;
for(j=0; j<2; j++){
if(LEER_X10[j]==1){
CHECK_INICIO++;
}
}
if(CHECK_INICIO==2){
leer_comando=1;
}
CHECK_INICIO=0;
}
}
/*****/
void ver_botones(){
if (Button(&PORTB, 3, 1, 1)) { // Detect logical one
oldstate_rb3 = 1; // Update flag
}
if (oldstate_rb3 && Button(&PORTB, 3, 1, 0)) { // Detect one-to-zero transition
oldstate_rb3 = 0;
LUZ_INC();
}

if (Button(&PORTB, 4, 1, 1)) { // Detect logical one
oldstate_rb4 = 1; // Update flag
}
if (oldstate_rb4 && Button(&PORTB, 4, 1, 0)) { // Detect one-to-zero transition
oldstate_rb4 = 0;
LUZ_DEC();
}

if (Button(&PORTB, 5, 1, 1)) { // Detect logical one

```

```

    oldstate_rb5 = 1;          // Update flag
    }
    if (oldstate_rb5 && Button(&PORTB, 5, 1, 0)) { // Detect one-to-zero transition
        oldstate_rb5 = 0;
        PORTA.B4=~PORTA.B4;
    }
}
/*****
*****INTERRUPCIONES*****
*
*****/
void interrupt() {
    if (TMROIF_bit) {          //INTERRUPCION TIMER0
        TMRO = 78;            // ASIGNA EL AJUSTE PARA EL OVERFLOW
        PORTA.B5=0;
        if (FLANCO_ZERO==1){
            if (DIMMER>0){ //ESPERA LOS CICLOS DEL TIMER INDICADOS POR DIMMER
                DIMMER--;
            }
            if (DIMMER==0){
                PORTA.B5=1;
                Delay_us(100); // DELAY DE 0.5MS PARA LECTURA CORRECTA
                PORTA.B5=0;
                FLANCO_ZERO=0;
            }
        }
        if (CONTADOR_FOCO1==1 || CONTADOR_FOCO1==2){
            PORTA.B2=1;
        }
        else{
            PORTA.B2=0;
        }
        TMROIF_bit = 0;      // clear TMROIF
    }
/*****
if (INTCON.RBIF) {          //INTERRUPCION CRUCE POR CERO
    DIMMER=CONTADOR_FOCO1;
    FLANCO_ZERO=1;
    TMRO=78;                //AJUSTE DEL TIMER
    Delay_us(800);          // DELAY DE 0.5MS PARA LECTURA CORRECTA
    PORTB.B2 = ~PORTB.B2;   // Toggle PORTB LEDs

    if (leer_comando==1){
        RECIBIR_CMD();
    }
}

```

```
    }
    if(PORTC.B3==1 && leer_comando==0){
        LEER_INICIO();
    }
    INTCON.RBIF=0;
    TMR0IF_bit = 0;          // clear TMR0IF
}
}
/*****
*****CICLO PRINCIPAL*****
*****/
void main() {
    INICIAR_PUERTOS();
    INICIAR_INTERRUPCIONES();
    PORTA.B4=~PORTA.B4;
    do {
        ver_botones();
        if (CAMBIAR_LED){
            VER_LEDS();
        }
    } while(1);
}
```

3. Código Website

1.Pantalla de Comandos

```

<html><head><title>Tesis</title>
<meta http-equiv="content-type" content="text/html; charset=utf-8">
<link rel="stylesheet" type="text/css" media="screen" href="estilo.css">
<script type="text/javascript" src="/webiopi.js"></script>
<script type="text/javascript">
webiopi().ready(function() {
$("#inc0_25").click(function(event) {
var data= "H";
webiopi().Serial("serial0").write(data);
}); $("#dec0_25").click(function(event) {
var data= "J";
webiopi().Serial("serial0").write(data);
});
$("#inc1_25").click(function(event) {
var data= "3";
webiopi().Serial("serial0").write(data);
}); $("#dec1_25").click(function(event) {
var data= "4";
webiopi().Serial("serial0").write(data);
});
$("#inc2_25").click(function(event) {
var data= "A";
webiopi().Serial("serial0").write(data);
}); $("#dec2_25").click(function(event) {
var data= "8";
webiopi().Serial("serial0").write(data);
});
$("#todas_50").click(function(event) {
var data= "30";
webiopi().Serial("serial0").write(data);
});
$("#todas_25").click(function(event) {
var data= "20";
webiopi().Serial("serial0").write(data);
});
});

```

```
$("#inputText").keyup(function(event){
if(event.keyCode == 13){
sendData();
}
});
$("#toma0_on").click(function(event) {
var data= "P";
webiopi().Serial("serial0").write(data);
});
$("#toma0_off").click(function(event) {
var data= "I";
webiopi().Serial("serial0").write(data);
});
$("#toma1_on").click(function(event) {
var data= "1";
webiopi().Serial("serial0").write(data);
});
$("#toma1_off").click(function(event) {
var data= "2";
webiopi().Serial("serial0").write(data);
});
$("#toma2_on").click(function(event) {
var data= "G";
webiopi().Serial("serial0").write(data);
});
$("#toma2_off").click(function(event) {
var data= "10";
webiopi().Serial("serial0").write(data);
});
$("#todos_on").click(function(event) {
var data= "9";
webiopi().Serial("serial0").write(data);
});
$("#todos_off").click(function(event) {
var data= "40";
webiopi().Serial("serial0").write(data);
});
});
});
</script></head>
```

```

<body id="bmission">
<div id="container">
<div id="header">
<h1><a href="index.html">Trabajo de Graduacion</a></h1>
<div id="logo">
</div>
</div>
<div id="navigation">
<ul>
<li id="lhome"><a href="index.html">Inicio</a></li>
<li id="labout"><a href="info.html">Informacion</a></li>
<li id="lmedia"><a href="media.html">Media</a></li>
</ul>
</div>
<div id="page-header"> <div id="logo">
</div>
<div id="wrapper">
<div id="dispositivos" class="col">
<h3>Dispositivo 1</h3>
<ul>
<li>Luminarias</li>
<input id="inc0_25" value="+ 25%" type="button">
<input id="dec0_25" value="- 25%" type="button">
<li>Tomacorrientes</li>
<input id="toma0_on" value="Activar" type="button">
<input id="toma0_off" value="Desactivar" type="button">
</ul>
</div>
<div id="dispositivos" class="col">
<h3>Dispositivo 2</h3>
<ul>
<li>Luminarias</li>
<input id="inc1_25" value="+ 25%" type="button">
<input id="dec1_25" value="- 25%" type="button">
<li>Tomacorrientes</li>
<input id="toma1_on" value="Activar" type="button">
<input id="toma1_off" value="Desactivar" type="button">
</ul>
</div>

```

```

<div id="dispositivos" class="col">
<h3>Dispositivo 3</h3>
<ul>
<li>Luminarias</li>
<input id="inc2_25" value="+ 25%" type="button">
<input id="dec2_25" value="- 25%" type="button">
<li>Tomacorrientes</li>
<input id="toma2_on" value="Activar" type="button">
<input id="toma2_off" value="Desactivar" type="button">
</ul>
</div>
<div id="dispositivos" class="col">
<h3>Todos las Luminarias</h3>
<ul>
<input id="todas_25" value="Set a 25%" type="button">
<input id="todas_50" value="Set a 50%" type="button">
</ul>
</div>
<div id="dispositivos" class="col">
<p></p>
</div>
<div id="dispositivos" class="col">
<h3>Todos los Tomacorrientes</h3>
<ul>
<input id="todos_on" value="Activar" type="button">
<input id="todos_off" value="Desactivar" type="button">
</ul>
</div>

</div>

<div id="footer">
<br>
<div id="logo">
</div>
<p> Universidad Del Valle de Guatemala. Facultad de Ingenieria.
Ingenieria Mecatronica </p>
<p> Proyecto realizado por Andres Quintanilla. 2013</p>
</div>

```

```

</div>
</div>
</body></html>

```

2. Pantalla de Información

```

<html><head><title>Tesis-Info</title>
<meta http-equiv="content-type" content="text/html; charset=utf-8">
<link rel="stylesheet" type="text/css" media="screen" href="estilo.css"></head>
<body id="bmission">
<div id="container">
<div id="header">
<h1><a href="index.html">Trabajo de Graduacion</a></h1>
<div id="logo">
</div>
</div>
<div id="navigation">
<ul>
<li id="lhome"><a href="index.html">Inicio</a></li>
<li id="labout"><a href="info.html">Informacion</a></li>
<li id="lmedia"><a href="media.html">Media</a></li>
</ul>
</div>
<div id="page-header"> 
<div id="wrapper">
<span style="font-weight: bold;"><br>
Trabajo de Graduacion Presentado
por<br>
Jose Andres Quintanilla Lopez<br>
Para optar al grado academico de <br>
Licenciado en Ingenieria Mecatronica<br>
<br>
</span>
<div style="text-align: justify;">El
objetivo general de este proyecto es desarrollar un sistema de
automatizacion de tomacorrientes y luminarias, en sistemas electricos

```



```
border-bottom-width: 0;
}
h1 { font-size: 28px; }

h2 { font-size: 26px; }

h3 { font-size: 22px; }

h4 { font-size: 20px; }

h5 { font-size: 18px; }

h6 { font-size: 16px; }

ul
{
  color: #555555;
  padding: 10px 30px;
  line-height: 2em;
}

/* Structure */

body
{
  padding: 0;
  width: 850px;
  margin: 0 auto;
  background-repeat: repeat-y;
  background-position: top left;
  /*background-color: #ffffff; /*Blanco*/
  background-image: url(body_bg.png);
}

#container
{
  width: 700px;
  margin: 0 auto;
}

#header { float: left; width: 100%; clear: both; margin-top: 1px; }
#navigation { float: left; width: 100%; clear: both; }
#page-header { clear: both; position: relative; margin-bottom: 2px; }
#wrapper { text-align: center; }
```

```
#footer
{
  clear: both;
  color: #666666;
  padding: 20px 0 15px;
  text-align: center;
}

/* Header */
#header h1
{
  padding: 0;
  float: left;
  width: 140px;
  margin: 3px 0;
  font-size: 14px;
  text-transform: uppercase;
  letter-spacing: 0.1em;
  border-right: 1px solid #999999;
}

#header h1 a:hover,
#header h1 a:active
{
  border-bottom-width: 0;
}

/* Navigation */
#navigation ul
{
  padding: 0;
  width: 100%;
  float: left;
  margin: 0 0 5px;
  list-style-type: none;
  background-color: #0E4D5C; /*Verde oscuro*/
}
#navigation li { float: left; }
#navigation a:link,
#navigation a:visited
{
  float: left;
  display: block;
  color: #eff8ff; /*Blanco palido*/
  padding: 5px 15px;
```

```

border-bottom-width: 0;
}

#navigation a:hover,
#navigation a:active
{
color: #ffffff;
background-color: #135D6E; /*verde menos oscuro*/
}

body#bmission li#lmission a
{
color: #eff8ff;
background-color: #6F8AF7; /*374462*/
}

/* Wrapper */
.col { width: 200px; float: left; margin: 0 15px; text-align: left; }

.col ul { margin: 0 0 0 10px; padding: 5; line-height: 1em; }

.col li { padding-bottom: 0px; padding-top: 3px; }

.col h3 { font-size: 13px; color: #eff8ff; padding: 3px 10px; margin: 20 0 0px; }

#dispositivos h3
{
background-color: #0E4D5C; /*Verde oscuro*/
}

/* Footer */
#footer p { margin: 5px 0; }

```

XI. GLOSARIO

1. CCP: *Capture/Compare/PWM es un módulo periférico del micro controlador PIC16F887 que permite al usuario controlar diferentes eventos. En modo captura permite la sincronización de eventos. En modo comparador permite activar eventos externos cuando cierta cantidad de tiempo ha transcurrido. En modo PWM permite generar pulsos modulados con frecuencia y periodo variable.*
2. PWM: *Pulse-Width Modulated. Indica que se generan pulsos con frecuencia y periodo variable.*
3. Interrupt On Change: *Se refiere a que interrumpe el flujo del programa principal cada vez que existe un cambio de estado en determinados pines del PIC16F887.*
4. GPIO: *General Purpose Input Output. Son pines que permiten al Raspberry Pi interactuar con otros circuitos o computadoras.*
5. LAN: *Local Area Network. Es una red de computadoras que se interconectan en áreas limitadas.*