

UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería



RED DE DOMÓTICA INTEGRADA (RDI)

Trabajo de graduación presentado por: Eduardo Leroy Drummond Juárez, Javier Antonio García Perdomo, Juan Manuel Maldonado García, Franco Alessandro Noack Herrera y Diego Saravia Fernández para optar al grado académico de Licenciados en Ingeniería Mecatrónica

Guatemala,

2013

**RED DE DOMÓTICA INTEGRADA
(RDI)**

UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería



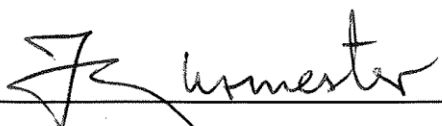
RED DE DOMÓTICA INTEGRADA (RDI)

Trabajo de graduación presentado por: Eduardo Leroy Drummond Juárez, Javier Antonio García Perdomo, Juan Manuel Maldonado García, Franco Alessandro Noack Herrera y Diego Saravia Fernández para optar al grado académico de Licenciados en Ingeniería Mecatrónica


Guatemala,

2013

Vo.Bo.:

(f) 
Ing. Juancarlos Burmester

Tribunal:

(f) 
MAEB. Roberto Delgado

Fecha de aprobación: Guatemala, 25 de noviembre de 2013

PREFACIO

El megaproyecto RDI Red de Domótica Integrada surge como opción para el manejo sencillo de variables domésticas de uso diario. Este se dividió en seis módulos, los cuales se encargan de distintas tareas para permitir que el usuario pueda modificar las variables de su interés desde un dispositivo móvil.

En la actualidad no se comercializa ningún sistema de este tipo desarrollado en Guatemala, las opciones existentes son sistemas diseñados en el extranjero, por este motivo esperamos que los resultados obtenidos de este trabajo sirvan como punto de inicio para futuras investigaciones y posible distribución comercial.

El trabajo realizado requirió compromiso, esfuerzo y dedicación por parte de los integrantes del grupo. Los resultados obtenidos cumplen con los planteamientos iniciales de diseño, por lo cual consideramos que el proceso de desarrollo se llevó a cabo con éxito.

Agradecemos a todas las personas que contribuyeron en las distintas etapas de ejecución del proyecto. A nuestras familias, por el apoyo y paciencia mostrados durante el desarrollo del mismo. Agradecemos también a nuestros asesores Ing. Juancarlos Burmester y MAEB. Roberto Delgado por sus aportes y guía, así como a todos aquellos que de manera directa o indirecta se involucraron con el proyecto.

ÍNDICE

Página

PREFACIO.....	VI
ÍNDICE DE FIGURAS.....	XII
ÍNDICE DE TABLAS.....	XVI
RESUMEN.....	XVII
I. INTRODUCCIÓN.....	1
II. OBJETIVOS.....	4
A. Objetivo general.....	4
B. Objetivos específicos.....	4
III. JUSTIFICACIÓN.....	5
IV. MARCO TEÓRICO.....	6
A. Domótica.....	6
1. Aplicación.....	6
2. Topología.....	7
3. Medio de comunicación.....	8
B. Interfaz gráfica de usuario.....	9
1. Aplicaciones móviles.....	9
4. Conectividad en dispositivos IOS.....	10
5. Xcode.....	11
6. Objective C.....	11
7. Cococa & Cocoa Touch.....	12
8. Arquitectura MVC.....	12
9. Arquitectura MVC en Xcode.....	13
10. Protocolos en Objective C.....	14
11. Wi-Fi.....	15
C. Módulo central y servidor.....	16
1. Internet protocol suite.....	16
2. Base de datos.....	20

3.	Raspberry Pi	21
4.	Software	24
D.	Infraestructura de comunicación	26
1.	Energía eléctrica	26
2.	Ruido eléctrico	27
3.	Corriente alterna	28
4.	Protocolo X10	28
5.	Protocolo de comunicación KNX.....	32
6.	Protocolo de comunicación Zigbee	32
7.	Modulación ASK	32
8.	Circuito LC.....	34
9.	Circuito LC en serie	35
10.	Circuito LC en paralelo.....	35
E.	Comunicación y seguridad.....	36
1.	Red de comunicación de datos	36
2.	Checksum.....	38
3.	Lenguaje de programación MikroC	39
4.	EUSART	40
5.	Baud rate	40
F.	Periféricos de ambiente	41
1.	Iluminación	41
2.	Tipos de lámparas utilizadas para iluminación	43
3.	Control remoto por infrarrojo.....	48
4.	Comunicación por infrarrojo.....	48
5.	Protocolos de comunicación por infrarrojo	49
G.	Periféricos de potencia	52
1.	Sistemas de riego automáticos	52
2.	Sistemas de riego para jardines	53
II.	ANTECEDENTES.....	55
III.	METODOLOGÍA	57
IV.	INTERFÁZ GRÁFICA DE USUARIO	58
A.	Selección de dispositivo para GUI.....	58
1.	Diseño experimental.....	58
2.	Discusión.....	58
B.	Selección de plataforma para desarrollo	59
1.	Diseño experimental.....	59
2.	Discusión.....	59
C.	Estructura y diseño de aplicación.....	61

1.	Diseño experimental.....	61
2.	Discusión.....	62
D.	Método de comunicación.....	64
1.	Diseño experimental.....	64
2.	Discusión.....	65
E.	Resultados generales.....	68
V.	MÓDULO CENTRAL Y SERVIDOR.....	73
A.	Configuración <i>Raspberry Pi</i>	73
1.	Diseño experimental.....	73
2.	Discusión.....	73
B.	Configuración y programación servidor	74
1.	Diseño experimental.....	74
2.	Discusión.....	74
C.	Base de datos	77
1.	Diseño experimental.....	77
2.	Discusión.....	77
D.	Comunicación serial	77
1.	Diseño experimental.....	77
2.	Discusión.....	78
E.	Multi-tareas.....	79
1.	Diseño experimental.....	79
2.	Discusión.....	79
F.	Resultados generales.....	79
VI.	INFRAESTRUCTURA DE COMUNICACIÓN	86
A.	Transmisión por corrientes portadoras.....	86
1.	Diseño experimental.....	86
2.	Resultados	87
3.	Discusión.....	89
B.	Modem de comunicación bidireccional.....	90
1.	Diseño experimental.....	91
2.	Resultados	93
3.	Discusión.....	98
C.	Fuente de alimentación	102
1.	Diseño experimental.....	103
2.	Resultados	104
3.	Discusión.....	105

VII. COMUNICACIÓN Y SEGURIDAD	107
A. Delimitación	107
1. Diseño experimental.....	107
2. Resultados	108
3. Discusión.....	109
B. Definición y programación.....	110
1. Diseño experimental.....	110
2. Resultados	112
3. Discusión.....	117
VIII. PERIFÉRICOS DE AMBIENTE.....	121
A. Control de dispositivo IR	121
1. Diseño experimental.....	121
2. Resultados	127
3. Discusión.....	139
B. Iluminación LED RGB.....	141
1. Diseño experimental.....	141
2. Resultados	142
3. Discusión.....	142
C. Control de encendido, apagado y atenuación de lámparas	144
1. Diseño experimental.....	144
2. Resultados	144
3. Discusión.....	148
IX. PERIFÉRICOS DE POTENCIA.....	151
A. Sistema de riego	151
1. Diseño experimental.....	151
2. Resultados	152
3. Discusión.....	153
B. Ventana motorizada	153
1. Diseño experimental.....	154
2. Resultados	154
3. Discusión.....	156
C. Cortina motorizada	156
1. Diseño experimental.....	156
2. Resultados	157
3. Discusión.....	158

XIII.	CONCLUSIONES.....	159
XIV.	BIBLIOGRAFÍA.....	161
XV.	APÉNDICE	165
XVI.	GLOSARIO	167

ÍNDICE DE FIGURAS

	Página
Figura No. 1. Segmentación en Módulos del Megaproyecto RDI	2
Figura No. 2. Distribución de mercado de Smartphones en EE.UU (Q2 2013)	9
Figura No. 3. Esquema de comunicación de un MVC	13
Figura No. 4. UITableView: Protocolo utilizado para desplegar tablas	14
Figura No. 5. Estándar Wi-Fi.....	15
Figura No. 6. Configuración de subredes	17
Figura No. 7. Sintaxis de objetos en formato JSON	20
Figura No. 8. Valores de objetos en formato JSON	20
Figura No. 9. Esquema Raspberry Pi.....	22
Figura No. 10. Pines de entrada y salida para propósito general de Raspberry Pi	23
Figura No. 11. Diagrama de funcionamiento DJANGO.....	25
Figura No. 12. Frecuencias y voltajes en el mundo.....	27
Figura No. 13. Transmisión de datos en X10.	29
Figura No. 14. Protocolo de transmisión de datos en X10.	31
Figura No. 15. Señal de modulación ASK.....	33
Figura No. 16. Componentes de un sistema de transmisión de datos.....	38
Figura No. 17. Envío y recepción de datos aplicando cheksum	39
Figura No. 18. Comparación de datos enviados utilizando el mismo <i>baud rate</i>	40
Figura No. 19. Región del espectro electromagnético correspondiente a la luz visible ..	41
Figura No. 20. Temperatura correspondiente al color emitido.....	42
Figura No. 21. Componente de una lámpara incandescente	43
Figura No. 22. Modulación de una señal IR	49
Figura No. 23. Bits protocolo Sony Sirc.....	50
Figura No. 24. Ejemplo de código protocolo Sony sirc.....	50
Figura No. 25. Bits protocolo NEC	51
Figura No. 26. Ejemplo de código protocolo NEC.....	51
Figura No. 27. Bits protocolo RC5 (Oisterwijk, 2011).....	52
Figura No. 28. Ejemplo de código protocolo RC5 (Oisterwijk, 2011).....	52
Figura No. 29. Diseño correcto <i>head to head</i>	53
Figura No. 30. Diseño incorrecto <i>head to head</i>	54
Figura No. 31. Sectorización de líneas para riego	54
Figura No. 32. Diagrama de bloques modulares del megaproyecto RDI	57
Figura No. 33. Arquitectura <i>MVC</i> con desplazamiento serial	61
Figura No. 34. Esquema general de la interfaz gráfica de usuario	63

Figura No. 35. Estructura de almacenamiento de datos para transmisión	66
Figura No. 36. Pantalla de inicio de sesión.....	68
Figura No. 37. Pantalla De Selección De Módulos	68
Figura No. 38. Control de módulo de iluminación RGB.....	69
Figura No. 39. Control de módulo de iluminación incandescente.....	69
Figura No. 40. Control de módulo de accesos: ventana	70
Figura No. 41. Control De Módulos De Acceso: Cortina.....	70
Figura No. 42. Control de módulo para televisores	71
Figura No. 43. Control de módulo para riego.....	71
Figura No. 44. Pantalla inicial sistema operativo Raspbian	79
Figura No. 45. Escritorio LXDE Raspbian.....	80
Figura No. 46. Archivo network modificado para IP estática	80
Figura No. 47. Servidor simple de Python ejecutándose (lado servidor)	80
Figura No. 48. Servidor simple de Python ejecutándose (lado cliente).....	81
Figura No. 49. Configuración archivo settings.py (base de datos).....	81
Figura No. 50. Configuración archivo settings.py (aplicaciones)	82
Figura No. 51. Archivo URLS.py.....	82
Figura No. 52. Diagrama de flujo archivo views.py.....	83
Figura No. 53. Servidor Django ejecutándose (lado servidor)	84
Figura No. 54. Servidor Django ejecutándose (aplicación administrador).....	84
Figura No. 55. Diagrama de flujo administración del sistema	85
Figura No. 56. Diagrama de flujo módem bidireccional	90
Figura No. 57. Filtro pasabandas LC.....	91
Figura No. 58. Circuito de operación del reloj interno.....	92
Figura No. 59. Conexión con el microcontrolador.....	92
Figura No. 60. Circuito utilizado para el optoacoplador	93
Figura No. 61. Funcionamiento optoacoplador a 400 Hz.....	95
Figura No. 62. Funcionamiento optoacoplador a 1.2 kHz.....	95
Figura No. 63. Funcionamiento optoacoplador a 10 kHz.....	95
Figura No. 64. Funcionamiento optoacoplador a 100 kHz.....	96
Figura No. 65. Filtro LC.....	96
Figura No. 66. Comparación TX pic (CH1) y DATA_IN módem (CH2).....	97
Figura No. 67. Comparación DATA_IN módem (CH1) y TX_OUT módem (CH2).	97
Figura No. 68. Comparación RX_IN módem (CH1) y DATA_OUT módem (CH2).	97
Figura No. 69. Comparación TX_OUT emisor (CH1) y RX_IN receptor (CH2).....	98
Figura No. 70. Comparación TX emisor (CH1) y RX receptor (CH2).	98
Figura No. 71. Diseño fuente de alimentación sin transformador capacitiva.....	103
Figura No. 72. fuente capacitiva con regulador de voltaje.....	104
Figura No. 73. Voltaje de salida fuente capacitiva con regulador de voltaje.	105
Figura No. 74. Intercambio de datos entre módulos.....	108

Figura No. 75. Diagrama de flujo de elemento emisor.....	112
Figura No. 76. Diagrama de flujo de función enviar.	113
Figura No. 77. Diagrama de flujo de proceso de direccionamiento.	114
Figura No. 78. Diagrama de flujo de elemento receptor.....	115
Figura No. 79. Diagrama de flujo de función enviar.	116
Figura No. 80. Pines del circuito integrado TSOP1738.....	124
Figura No. 81. Diagrama de bloques de los integrados familia TSOPXX	124
Figura No. 82. Respuesta relativa del TSOP1738 dependiendo de la frecuencia recibida	125
Figura No. 83. Sensibilidad del receptor vs longitud de onda recibida por el TSOP1738	126
Figura No. 84. Intencidad de radiación vs longitud de onda emitida por el led IR seleccionado.....	127
Figura No. 85. Definición de marca y espacio.....	128
Figura No. 86. Diagrama de flujo protocolo Samsung.	128
Figura No. 87. Diagrama de flujo protocolo Sony.....	128
Figura No. 88. Diagrama de flujo protocolo NEC.....	130
Figura No. 89. Diagrama de flujo protocolo RC5.	131
Figura No. 90. Diagrama de flujo protocolo Panasonic.....	131
Figura No. 91. Código ON/OFF para una televisión Sony, control original	134
Figura No. 92. Código ON/OFF para una televisión Sony, control desarrollado.....	134
Figura No. 93. Código ON/OFF para una televisión PHILIPS, control original.....	134
Figura No. 94. Código ON/OFF para una televisión PHILIPS, control desarrollado. ..	135
Figura No. 95. Código ON/OFF para una televisión PANASONIC, control original. ..	135
Figura No. 96. Código ON/OFF para una televisión PANASONIC, control desarrollado	135
Figura No. 97. Código ON/OFF para una televisión TOSHIBA, control original.	136
Figura No. 98. Código ON/OFF para una televisión TOSHIBA, control desarrollado.	136
Figura No. 99. Código ON/OFF para una televisión LG, control original.	136
Figura No. 100. Código ON/OFF para una televisión LG, control desarrollado.....	137
Figura No. 101. Código ON/OFF de diferentes televisiones enviados uno tras de otro.	137
Figura No. 102. Circuito utilizado para el control de dispositivos IR.	138
Figura No. 103. Circuito emisor IR modificado.	138
Figura No. 104. Diferentes colores obtenidos con el módulo de iluminación RGB.	142
Figura No. 105. Diagrama de flujo control de iluminación.	145
Figura No. 106. Detección del cruce por cero de la señal de 120V AC.	146
Figura No. 107. Detección del cruce por cero de la señal de 120V AC.	146
Figura No. 108. Atenuación de la señal 120V AC al 98% CH1 señal 120V AC, CH2 señal atenuada.	146

Figura No. 109. Atenuación de la señal 120V AC al 80% CH1 señal 120V AC, CH2 señal atenuada.	147
Figura No. 110. Atenuación de la señal 120V AC al 60% CH1 señal 120V AC, CH2 señal atenuada.	147
Figura No. 111. Atenuación de la señal 120V AC al 40% CH1 señal 120V AC, CH2 señal atenuada.	147
Figura No. 112. Atenuación de la señal 120V AC al 20% CH1 señal 120V AC, CH2 señal atenuada.	148
Figura No. 113. Atenuación de la señal 120V AC al 2% CH1 señal 120V AC, CH2 señal atenuada.	148
Figura No. 114. Protección en los puertos de entrada del PIC16F886.	149
Figura No. 115. Compartimiento y accesorios módulo de riego.	152
Figura No. 116. Diseño e instalación ventana con motor.	155
Figura No. 117. Voltaje vs tiempo de apertura de ventana.	155
Figura No. 118. Diseño e instalación cortina con motor.	157
Figura No. 118. Esquemático módulo de comunicación.	165
Figura No. 119. Esquemático fuente de alimentación.	165
Figura No. 120. PCB Módulo de comunicación con Raspberry Pi.	166
Figura No. 121. PCB Módulo de comunicación con dispositivos.	166

ÍNDICE DE TABLAS

	Página
Tabla No. 1. Estándares de protocolo 802.11	15
Tabla No. 2. Especificaciones y características RPI.....	22
Tabla No. 3. Códigos para direccionamiento de casa X10.....	29
Tabla No. 4. Códigos para direccionamiento de unidad X10.....	30
Tabla No. 5. Listado de variables de control de interfaz gráfica.....	72
Tabla No. 6. Comparación de características protocolos de comunicación.....	87
Tabla No. 7. Tiempos de ejecución instrucciones X10.....	88
Tabla No. 8. Tiempo total de envío protocolo X10.....	88
Tabla No. 9. Frecuencia de la onda portadora.....	91
Tabla No. 10. Frecuencia de la onda portadora para la transmisión de datos.....	91
Tabla No. 11. Selección de frecuencias.....	93
Tabla No. 12. Valores escogidos capacitores de desacople.....	93
Tabla No. 13. Valores escogidos filtro pasabanda.....	94
Tabla No. 14. Valores escogidos componentes externos del reloj.....	94
Tabla No. 15. Elementos utilizados fuente sin transformador capacitiva.....	103
Tabla No. 16. Elementos utilizados fuente capacitiva con regulador de voltaje.....	104
Tabla No. 17. Información de envío y recepción por módulos.....	107
Tabla No. 18. Baud rate definidos para elementos emisor y receptor.....	116
Tabla No. 19. Tiempos para ciclo de comunicación completa.....	116
Tabla No. 20. Receptores infrarrojo con frecuencia central de 38 KHz.....	125
Tabla No. 21. Comandos del control de dispositivos.....	132
Tabla No. 22. Costo de materiales y construcción módulo de riego.....	152
Tabla No. 23. Equivalencia potencia de motor vs metro cuadrado de ventana.....	154
Tabla No. 24. Relación voltaje vs tiempo de apertura de ventana.....	155
Tabla No. 25. Costo de materiales y construcción módulo de cortina.....	157

RESUMEN

El sistema realizado busca ser una opción accesible para la introducción de domótica en domicilios donde actualmente el control de variables domésticas se realiza de forma manual. Algunas de estas variables son: iluminación, ventanas, cortinas, riego y electrodomésticos de control infrarrojo. Todas estas variables podrán ser controladas de manera centralizada a través de un dispositivo móvil, facilitando así el manejo general de las variables de interés desde un Módulo de Interfaz Gráfica para el Usuario.

El sistema aprovecha los recursos existentes en el domicilio, específicamente el cableado de alimentación AC, el cual se utiliza como medio físico para la comunicación entre el Módulo Central y Servidor y los Módulos Periféricos a través de la conexión de estos con los Módulos de Infraestructura de Comunicación.

El Módulo de Comunicación y Seguridad será el encargado de regular la comunicación de dos vías efectuada entre dispositivos, facilitando la identificación individual de Módulos Periféricos y transmitiendo los datos necesarios para la modificación del estado de estos.

La integración de los distintos módulos permite implementar el Megaproyecto RDI como un sistema domótico funcional.

I. INTRODUCCIÓN

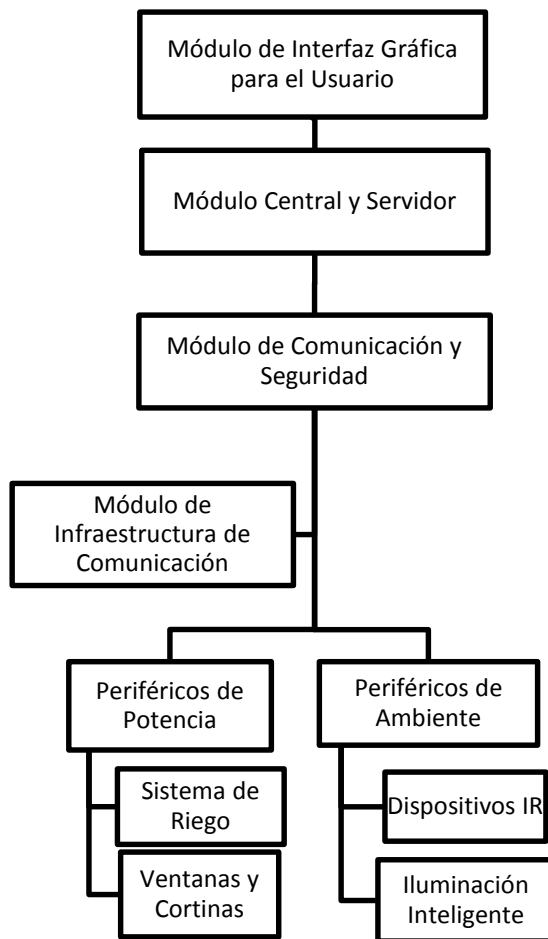
El término Domótica proviene de los vocablos en latín *domus* y *tica* que significan “casa” y “automática” respectivamente. La domótica reúne las tecnologías que funcionan en la automatización de ambientes en viviendas. Sus objetivos y funciones varían desde confort y comodidad hasta seguridad y ahorro energético en el hogar. Un sistema domótico posee la capacidad de recopilar información del ambiente a través de sensores, procesar dicha información y, en base a ésta, realizar acciones mediante actuadores. Algunos sistemas tienen también la capacidad de acceder a redes externas de comunicación o información.

Aunque el término todavía es desconocido por la mayoría de personas, el concepto ha estado presente en la mente del público desde hace más de medio siglo. Películas y series animadas de ciencia ficción crearon la idea de ambientes domésticos que fueran capaces de realizar funciones por sí mismos. Con el tiempo estas ideas comenzaron a hacerse realidad y se podían apreciar sistemas de este tipo en algunos centros recreativos. Fue a mediados de los años 70s que la domótica inició como tal, gracias a la creación del protocolo X10, basado en la transmisión de datos a través de corrientes portadoras. Muchos otros protocolos para el control domótico vinieron después de éste, con la diferencia que la mayoría de estos dejaron de utilizar la línea de corriente para la transmisión de datos, utilizando en cambio sus propios buses.

El Megaproyecto Red de Domótica Integrada (RDI) consta de seis módulos que individualmente atienden cada una de las tareas tecnológicas necesarias para desarrollar un sistema integral que permite el control de variables domésticas a través de dispositivos móviles. Entre estas variables figuran la iluminación, el riego, sistemas de televisión, ventanas y cortinas. En la Figura No. 1 se observa cada uno de estos módulos y la relación que tienen con los demás.

La interacción entre módulos permite que las variables domésticas de interés sean modificadas desde un dispositivo móvil, para esto se utilizarán distintos medios que permitan la transferencia de datos entre el sistema de manera efectiva.

FIGURA NO. 1. SEGMENTACIÓN EN MÓDULOS DEL MEGRAPROYECTO RDI



El módulo de Interfaz Gráfica para el Usuario funcionará como medio de entrada de datos, comunicando al sistema las modificaciones de distintas variables domésticas que el usuario desea efectuar desde un dispositivo móvil.

El Módulo Central y Servidor debe tener la capacidad de organizar y gestionar todos los eventos relacionados con la modificación de variables. Para esto debe cumplir las funciones tanto de Módulo Central como de Servidor. Funciona como módulo central debido a que se encarga de manejar el flujo de información entre los elementos conectados en la red. Los dispositivos que posean la aplicación móvil se conectan con el módulo central a través de una red local *LAN*, accediendo y modificando una base de datos, lo cual convierte al dispositivo central en un módulo servidor.

El Módulo de Infraestructura de Comunicación es el encargado de proveer hardware que cumpla la función de módems bidireccionales, permitiendo la comunicación entre el Módulo Central y Servidor y los actuadores (Periféricos de Potencia y Periféricos de Ambiente). Estos utilizan la línea de alimentación como medio de transferencia de datos, por lo cual el Módulo de Comunicación y Seguridad se encarga de proveer un protocolo eficiente que permita el envío y recepción de datos entre los distintos elementos conectados en la red.

Los Módulos Periféricos integran actuadores capaces de controlar la iluminación de la vivienda, así como el accionamiento de motores eléctricos dedicados al control de ventanas y cortinas. Para electrodoméstico de tipo *ON/OFF*, como ventiladores o lámparas, los módulos permiten graduar la intensidad de funcionamiento, además de permitir la comunicación con aparatos que funcionen con un control infrarrojo, tales como televisores y aparatos de sonido. Por último, los módulos brindan la capacidad de controlar las válvulas de sistemas de riego para jardines.

La interconexión de los seis módulos desarrollados permite la implementación del Megaproyecto RDI como un sistema de domótica funcional y efectivo.

II. OBJETIVOS

A. OBJETIVO GENERAL

1. Desarrollar un sistema integral para el control electrónico de variables domésticas a través de dispositivos móviles.

B. OBJETIVOS ESPECÍFICOS

1. Implementar módulos individuales encargados de las variables domésticas cotidianas como iluminación, apertura de cortinas y operación de radio y televisión.
2. Desarrollar una aplicación gráfica para dispositivos móviles que permita controlar los módulos individuales.
3. Implementar un servidor local para enlazar el módulo central con los dispositivos móviles de control.
4. Desarrollar una red de comunicación para la transmisión bidireccional de datos.

III. JUSTIFICACIÓN

El objetivo general del proyecto consiste en diseñar y elaborar un sistema capaz de manipular variables domésticas de uso diario como las luces, el control de televisiones, la apertura de cortinas, entre otros. El usuario será capaz de tomar decisiones sobre las variables mediante una interfaz gráfica móvil, permitiendo integrar éstas en un solo dispositivo mediante una aplicación (App).

En Guatemala las opciones de sistemas domóticos disponibles son limitadas pues solo permiten controlar ciertas variables domésticas sin integrarlas en un solo dispositivo, esto hace que la implementación de domótica se torne poco atractiva y que pocas personas decidan incluirla en su entorno.

La Interfaz Gráfica de Usuario debe brindar a este la capacidad de controlar todas las variables y eventos del sistema, desde un dispositivo móvil. El propósito es eliminar controles de uso individual y reunir estos comandos en una sola interfaz amigable mientras realiza sus tareas efectivamente.

El sistema debe ser capaz de analizar los datos obtenidos y dirigir la comunicación entre dispositivos. También se debe almacenar una base de datos de todos los estados y configuraciones del hogar, así como mantener una conexión intranet. Dado que es poco eficiente hacer que cada módulo cumpla todas estas funciones, se empleará un Módulo Central y Servidor.

La importancia del Módulo de Infraestructura de Comunicación es permitir el envío de paquetes de información a través de la línea de poder de corriente alterna (AC). Junto con el Módulo de Comunicación y Seguridad se permitirá la conexión entre el módulo central y los dispositivos periféricos, asegurando un envío de datos seguro y rápido de manera bidireccional.

Los Módulos Periféricos proveen al sistema de actuadores y controladores necesarios para la automatización de variables domésticas.

IV. MARCO TEÓRICO

A. DOMÓTICA

La domótica como tal, se refiere a un conjunto de soluciones tecnológicas que tienen como fin alcanzar el control y la automatización de diferentes dispositivos del hogar. Un solo sistema domótico es capaz de realizar diversas funciones y alcanzar distintos objetivos. El campo de aplicación de la domótica, aunque solo se limite a instalaciones dentro de viviendas, es muy extenso. Por esta razón las aplicaciones domóticas pueden subdividirse según distintos aspectos. (Novel)

1. Aplicación. Se basa en los objetivos del sistema domótico, ¿qué desea el usuario alcanzar a través de la instalación del sistema? Los tipos de aplicación suelen dividirse en cuatro grupos.

- Confort: No busca necesariamente crear nuevos dispositivos, sino más bien encontrar formas más fáciles para controlarlos. Sus funciones se basan en el día a día de las personas; en las actividades que realizan a diario. Por ejemplo: el control de iluminación, cortinas, sistemas de riego, etc. (Novel)
- Seguridad: El objetivo principal es hacer de la vivienda un lugar más seguro. Para esto hace uso de sensores de presencia en los diferentes accesos (como puertas o ventanas), y es capaz de habilitar el paso a través de los mismos. En algunos casos también se conecta a sistemas de alarma. (Novel)
- Gestión de la energía: Cumpliendo tanto fines económicos como de cuidado del planeta, el objetivo de esta aplicación es el ahorro de energía. Utiliza diferentes tipos de sensores para el análisis de dispositivos que no estén siendo utilizados, o que pueden ser utilizados consumiendo menor potencia. Por ejemplo: la utilización de sensores de presencia para

verificar si hay individuos en una habitación y de acuerdo a esto desactivar los dispositivos que en esta se encuentran. (Novel)

- Comunicación: Su función principal es realizar la conexión entre diferentes dispositivos que se encuentren instalados en el hogar. (Novel)

2. Topología. La topología se interesa en características como proximidad entre módulos y cantidad de conexiones entre los mismos. En domótica suelen utilizarse únicamente tres formas de conexión:

- Estrella: Es un tipo de red centralizada, es decir que existe un dispositivo conocido como “maestro” o “servidor” encargado de gestionar al resto (“esclavos” o “clientes”). La conexión estrella se identifica porque todos los dispositivos se encuentran directamente conectados con el dispositivo central, y además porque esta conexión es individual, existiendo para cada dispositivo una línea de comunicación con el módulo central. Tiene la característica de ser fácilmente expandible, pueden agregarse y removerse dispositivos sin afectar la conexión de los demás. (Novel)
- Anillo: Red descentralizada. Todos los dispositivos en este tipo de conexión poseen las mismas capacidades y prioridades. Cada uno de los dispositivos se encuentra conectado únicamente con dos más (suponiendo que el sistema cuenta con más de dos) hasta que la red quede “cerrada”. Reduce el cableado pero su costo tiende a ser elevado. (Novel)
- Bus: Es otra forma de red centralizada. Se diferencia con la conexión de tipo estrella en que, a pesar de que todos los dispositivos tienen una conexión directa con el módulo central, esta conexión no es individual. Los dispositivos comparten la línea de comunicación y es el trabajo del módulo central no solo gestionar a los demás dispositivos sino también ordenar la comunicación con ellos. (Novel)

3. Medio de comunicación. Una de las principales características que diferencia a los sistemas domóticos disponibles hoy en día es el medio de comunicación por el cual se conectan y el protocolo o estándar que utilizan. Según el medio de comunicación podemos encontrar tres tipos:

- Transmisión por corrientes portadoras: Utilizado por los sistemas domóticos más antiguos, son aquellos que utilizan las líneas de corriente como medio para transmitir los datos. Tienen la ventaja de no tener que agregar ningún cableado adicional para realizar la instalación, sino únicamente conectar los dispositivos a la alimentación. La comunicación realizada a través de estas líneas es susceptible a una gran cantidad de ruido. También depende de la calidad con la que se haya realizado la instalación eléctrica. Como ejemplos de estándares que utilizan este medio encontramos X10 y UPB. (Novel)
- Transmisión cableada: Se refiere a los sistemas que adicional a la instalación de módulos dentro el hogar, realizan un nuevo cableado dedicado únicamente al control del sistema domótico. Estos sistemas consiguen reducir casi totalmente el ruido eléctrico producido por electrodomésticos a cambio de la instalación del cableado. Incluye diferentes tipos de buses y en algunos casos también la implementación de fibra óptica. Los estándares pioneros de este medio de comunicación con KNX y LON. (Novel)
- Transmisión inalámbrica: El medio de comunicación más reciente. Al igual que la transmisión por corrientes portadoras, los dispositivos carecen de la necesidad de ser cableados. En lugar, cada uno de estos deben tener dentro de ellos módulos emisores y/o receptores del tipo de transmisión inalámbrica utilizada. Esta incluye desde sistemas que funcionan con radio frecuencia, hasta aquellos que utilizan Bluetooth o WiFi. Algunos protocolos que se reconocen con este tipo de transmisión son Zigbee y Zwave. (Novel)

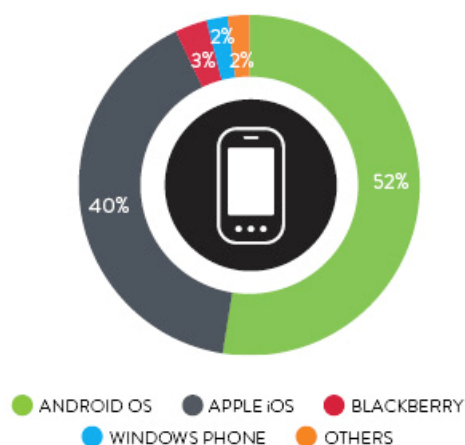
B. INTERFAZ GRÁFICA DE USUARIO

1. Aplicaciones móviles. Una aplicación móvil o App está diseñada para ser ejecutada en teléfonos inteligentes, tabletas y otros dispositivos móviles. Por lo general se encuentran disponibles a través de plataformas de distribución, operadas por las compañías propietarias de los sistemas operativos móviles como Android de Google, iOS de Apple, BlackBerry OS, entre otros. (Mikusiak)

Durante varios años, la demanda de teléfonos inteligentes ha superado otros productos en el mercado de la telefonía móvil. Los nombres más grandes hoy en día, son Apple y Android. Mientras que el primero abarca todo el proceso de diseño (hardware y software), manufactura y distribución, el segundo desarrolla únicamente el software y distribuye licencias a otras compañías que manufacturan dispositivos inteligentes como Samsung, HP, Sony, etc. (Nielsen)

El mercado tiene distribuciones muy diferentes dependiendo del continente o país en el que se estén enfocando las estadísticas. A continuación se presentan los resultados (Q2, 2013) de la distribución de *smartphones* con base en su sistema operativo en los Estados Unidos.

FIGURA NO. 2. DISTRIBUCIÓN DE MERCADO DE SMARTPHONES EN EE.UU (Q2 2013)



(Fuente: Nielsen)

2. Conectividad en dispositivos IOS. El iPhone 5, es un Smartphone con pantalla táctil desarrollado por Apple Inc. Es la 6ta generación de la línea de teléfonos iPhone y sucesor del iPhone 4S. Este teléfono es un modelo más delgado y ligero que introduce la pantalla de 4 pulgadas de alta resolución o Retina Display. En cuanto a sus atributos de procesamiento, este teléfono cuenta con un CPU propio de Apple llamado A6. (Cox)

El iPhone 5, específicamente el modelo A1428 (diseñado para ser compatible con redes de comunicación de EEUU y Canadá), está equipado con una antena dual Wi-Fi que soporta conectividad selectiva para los protocolos 802.11 a/b/g/n y n a 5GHz. (Mikusiak)

Hoy en día, la mayoría de dispositivos Wi-Fi utilizan la banda de 2.4 GHz para comunicarse entre sí, como teléfonos y otras redes Wi-Fi cercanas. Al igual que lo hacen otras tecnologías no Wi-Fi como el Bluetooth, walkmans e incluso el horno microondas. Esto hace que el ambiente este saturado de señales dentro de esta frecuencia. (Cox)

En contraste, la banda de 5GHz es también utilizada por dispositivos Wi-Fi, mediante los protocolos 802.11 a y 802.11 n (5GHz), para comunicarse, únicamente que esta frecuencia tiene considerablemente menos clientes y un mayor ancho de banda (40 MHz). (Wi-Fi)

Como resultado de esta nueva tecnología, es posible determinar cuál de las dos bandas presenta una mejor conectividad, e implementarla para la transmisión de datos.

3. Xcode. Potente entorno de desarrollo integrado (IDE) de Apple para la creación de aplicaciones para Mac, iPhone y iPad. Xcode incluye herramientas de análisis de instrumentos, simulador de iOS y SDK para iOS y OS X. (Apple Developer)

La interfaz de Xcode integra la edición de código, diseño de interfaz de usuario, pruebas y depuración, todo dentro de una sola ventana. El compilador es capaz de destacar los errores de codificación a medida que se escribe, y es lo suficientemente inteligente como para solucionar los problemas de forma automática. (Cox)

Xcode 5 es la versión actual de esta IDE, siendo liberada al público el 18 de septiembre de 2013 para el desarrollo de aplicación en iOS 7 u OS X Mavericks

4. Objective C. Sub-conjunto del lenguaje de programación C que proporciona capacidades orientadas a objetos y una ejecución dinámica. Objective C hereda la sintaxis, tipos primitivos y las sentencias de control de flujo de C y añade la sintaxis para la definición de clases y métodos, al igual que ofrecer soporte a nivel de lenguaje para la gestión de gráfico de objetos y objetos literales mientras que proporciona escritura dinámica y binding, difiriendo muchas responsabilidades hasta el tiempo de ejecución. (Otierney)

Este lenguaje de programación de propósito general, es utilizado por Apple para el desarrollo de sus sistemas operativos OS X y iOS, sus respectivas APIs, Cocoa y Cocoa Touch y Xcode. (Otierney)

5. Cocoa & Cocoa Touch. *Frameworks* estrechamente integrados en la experiencia de desarrollo de Xcode. Los APIs de alto nivel de Cocoa permiten la creación de redes, implementar animaciones y controlar el comportamiento y la apariencia de la plataforma de la aplicación con sólo unas cuantas líneas de código. (Wikipedia)

Al desarrollar con Cocoa, se están creando aplicaciones de la misma forma que iOS se crea. Haciendo que la aplicación herede automáticamente el comportamiento y apariencia de iOS, con acceso completo a la red subyacente del sistema operativo UNIX. Haciendo el conjunto de Cocoa y el IDE de Xcode la manera de crear aplicaciones nativas de iOS. (Apple Developer)

6. Arquitectura MVC. Cocoa utiliza la arquitectura *Model-View-Controller* (MVC) a través de toda su interfaz. Los modelos encapsulan los datos de la aplicación, las vistas muestran y editan los datos y el controlador supervisa la lógica entre ambos. Al separar las responsabilidades de esta manera, se termina con una aplicación fácil de diseñar, implementar y mantener. (iOS Developer Library)

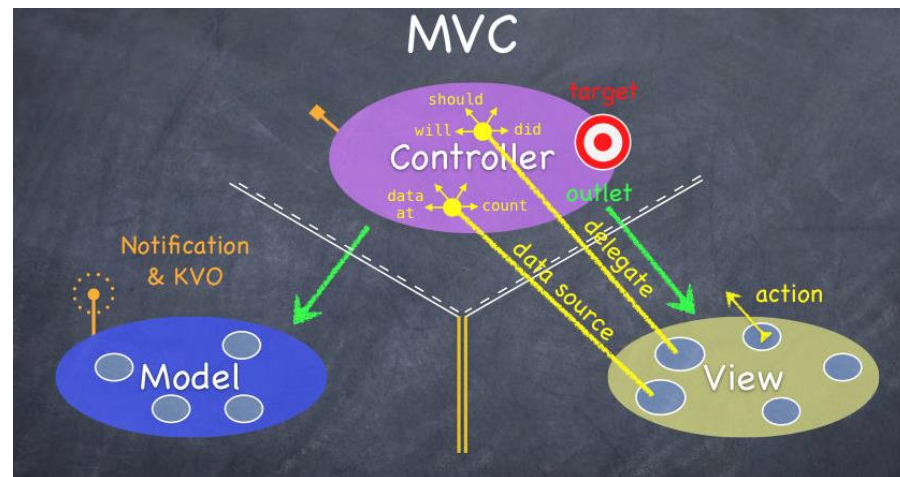
Los programas orientados a objetos se benefician de varias maneras mediante la adaptación del patrón de diseño MVC, ya que muchos objetos en estos programas tienden a ser altamente reutilizables. Los programas, en general son adaptables a las necesidades cambiantes, en otras palabras, son más fácilmente extensibles que los programas que no están basadas en MVC. (Cocoa Lab)

Por otra parte, muchas de las tecnologías y arquitecturas de cacao tales como enlaces, diseño del programa y guionización están basados en MVC y exige que los objetos personalizados desempeñen una de las funciones definidas por el MVC. (Hegarty)

7. Arquitectura MVC en Xcode. La estrategia de diseño MVC es bastante antigua, sin embargo es altamente eficiente para la creación de aplicaciones. Esto basado en el nivel de independencia que tiene cada uno de sus elementos al ser definidas y comunicarse entre ellas. (Hegarty)

En Xcode, la comunicación entre Modelo, Vista y Controlador están restringidas y por lo tanto, Apple ha implementado mecanismos de comunicación entre objetos específicos en función del emisor y receptor. (iOS Developer Library)

FIGURA NO. 3. ESQUEMA DE COMUNICACIÓN DE UN MVC



(Fuente: Hegarty)

- Controlador / Modelo: El controlador debe conocer absolutamente todo sobre el modelo, por lo que la comunicación es directa y no está restringida.
- Modelo / Controlador: El modelo no puede comunicarse directamente con el controlador, para esto, utiliza un mecanismo de Notificaciones & KVOs para alertar al controlador sobre un cambio en el modelo. (iOS Developer Library)
- Controlador / Vista: El controlador es el encargado de reflejar el modelo en la vista, por lo que la comunicación es directa y sin restricciones. Para esto, Xcode utiliza un mecanismo de salidas o "outlets" para comunicarse con los objetos de la GUI.

- Vista / Controlador: La vista debe comunicar al controlador sobre acciones que el usuario esté realizando sobre la GUI. Por lo que sí existe este tipo de comunicación, pero debe ser ciega y estructurada.

8. Protocolos en Objective C. En el mundo de la programación orientada a objetos, es muy importante tener la capacidad de definir una serie de comportamientos que se esperan que un objeto siga en cualquier situación dada.

Objective C permite definir protocolos, los cuales declaran los métodos que se esperan ser utilizados para alguna situación en particular. Estos métodos pueden ser obligatorios u opcionales, solo es necesario que la clase que implemente a dicho protocolo utilice todos los protocolos marcados como obligatorios. (Apple Developer)

Así como uno puede crear sus propios protocolos, ya existen algunos objetos genéricos que implementan los suyos. Como por ejemplo un *UITableView*, una vista que presenta información en forma de tabla. La cual espera tener la habilidad de comunicarse con un objeto que cumpla con la función de fuente de información para poder saber qué es lo que se necesita desplegar. Esto significa que la fuente de información debe responder a una serie específica de mensajes que la tabla pueda enviar. (Hegarty)

FIGURA NO. 4. ULTABLEVIEW: PROTOCOLO UTILIZADO PARA DESPLEGAR TABLAS



9. Wi-Fi. *Wireless Fidelity*, más conocido por su abreviación, Wi-Fi, es un protocolo de comunicación digital, a través de la cual dispositivos electrónicos pueden comunicarse entre ellos mediante transmisión inalámbrica. (Wi-Fi)

Wi-Fi es un término de marketing inicialmente aplicado al estándar de la IEEE 802.11b, ahora utilizado de forma ubicua para nombrar a todos los estándares que caen dentro de la categoría 802.11 de redes LAN inalámbricas. De manera que el término Wi-Fi encapsula los estándares 802.11 x, donde x es la respectiva versión Wi-Fi, entre las cuales se encuentran las versiones a, b, g y n. (Wi-Fi)

TABLA NO. 1. ESTÁNDARES DE PROTOCOLO 802.11

802.11 Protocolo	Liberado	Frecuencia. GHz	Ancho de Banda MHz	Tasa de Transferencia Máxima Mbps	Canales de Flujo permitidos MIMO	Modulación	Alcance en Interiores (m)	Alcance en Exteriores (m)
	1997	2.4	20	2	1	DSSS, FHSS	20	100
a	1999	5	20	54	1	OFDM	35	120
		3.7					N/A	5000
b	1999	2.4	20	11	1	DSSS	35	140
g	2003	2.4	20	54	1	OFDM, DSSS	38	140
n	2009	2.4/5	20	72.2	4	OFDM	70	250
			40	150			70	250
ac	2012	5	20/40/80/160	87/200/433/866	8	OFDM	Indefinido	Indefinido

(Fuente: 21)

Siendo un protocolo inalámbrico, el estándar Wi-Fi utiliza la banda de frecuencia ISM (Industrial, Científico y Médico), la cual está definida por el rango de frecuencias entre 2.4 GHz y 5 GHz para la implementación específica del protocolo IEEE 802.11. (Wi-Fi)

FIGURA NO. 5. ESTÁNDAR WI-FI



(Fuente: 21)

C. MÓDULO CENTRAL Y SERVIDOR

1. Internet protocol suite. Los protocolos de Internet (IP) y de Control de transmisión (TCP) fueron adoptados como estándar militar de Estado Unidos en 1983, teniendo la ventaja de ser independiente del hardware utilizado para transmitirlo. Hoy en día TCP/IP no hace referencia únicamente a estos dos sino a una familia de diversos protocolos dedicados a la comunicación y envío de información a través del Internet. Ejemplos de estos son HTTP (*HyperText Transfer Protocol*), FTP (*File Transfer Protocol*), SMTP (*Simple Mail Transfer Protocol*), ARP (*Address Resolution Protocol*), entre otros. (Cheng & Malaika, 1997)

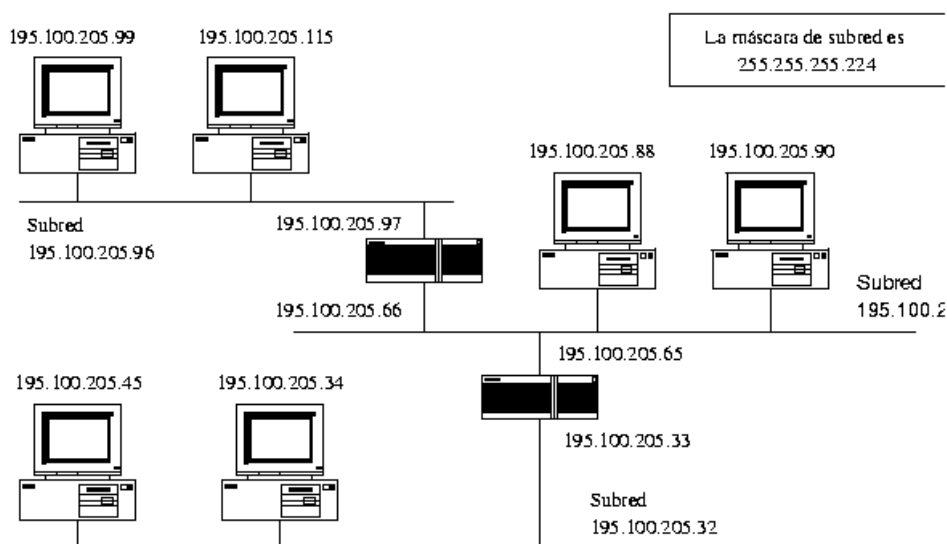
Los protocolos se basan en el envío de datos a través de paquetes. La unidad de transferencia utilizada se denomina datagramas IP. Cada uno de estos datagramas contiene una cabecera con información acerca del paquete como las direcciones de origen y destino, protocolo utilizado y otras opciones. Seguido a la cabecera se envían los datos. Las direcciones IP son números de 32 bits que identifican a todos los dispositivos conectados a una red que se comunica vía Internet. (Frystyk, 1994)

a. Subred. Dado el crecimiento exponencial de Internet y las aplicaciones desarrolladas para la misma, fue necesario comenzar a buscar métodos que ayudaran a optimizar el direccionamiento IP. Las subredes entonces logran entonces mejorar el tráfico en la red mayor y hacer la comunicación más rápida y organizada. Esto se logra utilizando *routers*, repetidoras o puertas de enlace, dependiendo del nivel en el que se haga la división.

Cualquier dispositivo conectado debe estar identificado con una dirección IP única. Una dirección IP que no pertenezca a una subred contiene en ella una porción que identifica la red y otra que identifica al host. Por otra parte, los

dispositivos presentes en subredes contienen un identificador de red, un identificador de subred y el identificador de host. El identificador de subred se crea utilizando una máscara de subred. (Frystyk, 1994)

FIGURA NO. 6. CONFIGURACIÓN DE SUBREDES



(Fuente: Hernando, 2002)

b. HTTP. Protocolo que se encuentra en el nivel o capa de aplicación

Es un protocolo de tipo solicitud/respuesta cuya función principal es recuperar documentos en el Internet con respuestas rápidas. Cada uno de los mensajes de HTTP enviados es una solicitud o una respuesta. Los mensajes de solicitud pueden ser de diferentes tipos dependiendo el método que se desee realizar, y a todos se les regresa un mensaje de respuesta. Los métodos utilizados son: (Fielding & Reschke, *Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content*, 2013)

- **GET:** Este método es el principal mecanismo para recuperar información. Se utiliza para solicitar datos de una fuente específica. Generalmente cualquier mensaje en el que se desee obtener información realiza una solicitud **GET**. Como respuesta a esta solicitud se envía un mensaje que contiene la cabecera – con información de control – y los datos solicitados.

- *HEAD*: El método es similar a *GET*, difiriendo únicamente en la respuesta. El mensaje respondido a contiene solamente la cabecera. Se utiliza para obtener información de la fuente.
- *POST*: A cambio de los métodos anteriores, este método en lugar de solicitar información, la envía. El mensaje solicita que la fuente procese la información enviada.
- *PUT*: Este método solicita crear o modificar el estado de la fuente. La información que se envía con este método busca ser almacenada en la fuente, mientras que la información enviado con el método *POST* busca ser procesada por la misma para realizar diferentes posibles acciones.
- *DELETE*: El método solicita, en lugar de borrar la información, borrar la asociación de mapeo que se tiene a hacia dicha información.
- *CONNECT*: Se utiliza para conexiones realizadas a través de proxys. Solicita establecer un túnel abierto para el envío a ciegas con de paquetes.
- *OPTIONS*: Este método solicita información acerca de las opciones de comunicación. Por ejemplo, la información acerca de que métodos son aceptados y que otros no.
- *TRACE*: Solicita iniciar un ciclo de depuración en el cual el destinatario envía de regreso el mismo mensaje recibido.

c. Cliente – Servidor. Hace referencia a un tipo de arquitectura de redes en las que los servidores son los encargados de proveer recursos, mientras que los clientes hacen uso de ellos. Esta arquitectura es la utilizada a través de la comunicación HTTP. **(Fielding & Reschke, Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content, 2013)**

La diferencia entre clientes y servidores radica únicamente en sus funciones dentro de la red en particular. Un mismo computador puede programarse para cumplir funciones de servidor en una red, mientras que realiza funciones de cliente en otra.

- Pull vrs. Push. La tecnología *Pull* es la comúnmente utilizada en la comunicación HTTP. En esta son los clientes son los encargados de establecer una conexión con el servidor y mandar solicitudes a este, y los servidores aceptan las conexiones, reciben y procesan las solicitudes y mandar respuestas a los clientes. Es la utilizada por cualquier navegador al momento de entrar a un sitio web o realizar una descarga.

Por otra parte la tecnología *Push* intenta cambiar la necesidad de que el cliente inicie la comunicación. En este tipo de comunicación la solicitud es realizada por el servidor. Se utiliza para aplicaciones en donde el cliente quiere ser informado por cualquier cambio en el servidor.

d. DHCP. De las siglas de *Dynamic Host Configuration Protocol*, es el protocolo encargado de proveer datos de configuración a cada dispositivo conectado a la red. Consta de dos componentes: el primero es un protocolo encargada de proveer a cada dispositivo de los parámetros de configuración necesarios y el segundo que es un mecanismo que asigna a cada uno una dirección IP dentro de la red en la que se encuentra conectado.

La arquitectura del protocolo es también cliente-servidor, en donde el servidor trabaja como administrador del sistema y es el encargada de proveer los datos y las direcciones; y los clientes son los dispositivos solicitando la información. (Droms, 1997)

e. JSON. Siglas de *JavaScript Object Notation*, representan un formato ligero para el intercambio de datos. Tiene los beneficios de ser amigable para los usuarios al leerlos o escribirlos, y ser sencillo para las computadoras al generar y analizar. JSON, aunque está basado en la programación *JavaScript*, es un formato independiente del lenguaje utilizado en la aplicación.

Posee una estructura de pares nombre-valor. Cada uno de estos pares – o el conjunto desordenado de los mismos - se le conoce como objetos. Es posible ordenar varios objetos haciendo arreglos o listas de los mismos. **(JSON Project)**

FIGURA NO. 7. SINTAXIS DE OBJETOS EN FORMATO JSON

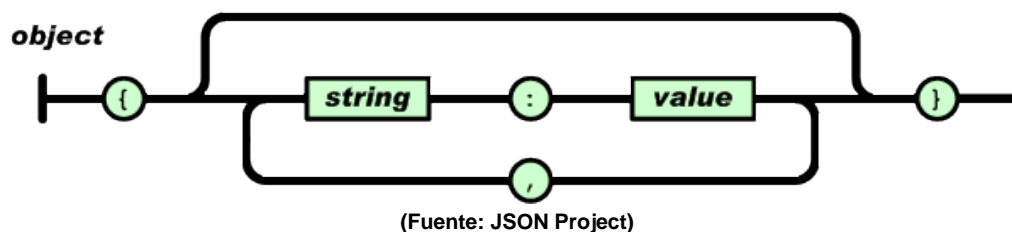
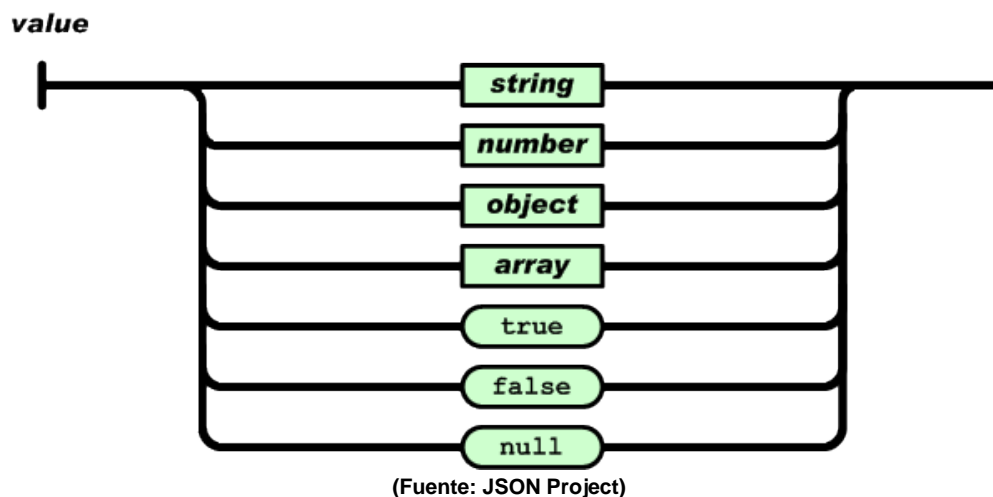


FIGURA NO. 8. VALORES DE OBJETOS EN FORMATO JSON



2. Base de datos. Un dato es cualquier hecho o información conocida, que se puede registrar. En términos generales, una base de datos hace referencia a una colección de datos que tienen una relación entre sí. Específicamente se entiende por base de datos una colección coherente de datos con significados inherentes. Es decir que una colección de datos aleatorios no es considerada una base de datos. Las bases de datos son diseñadas y construidas para propósitos específicos, representando aspectos del mundo real.

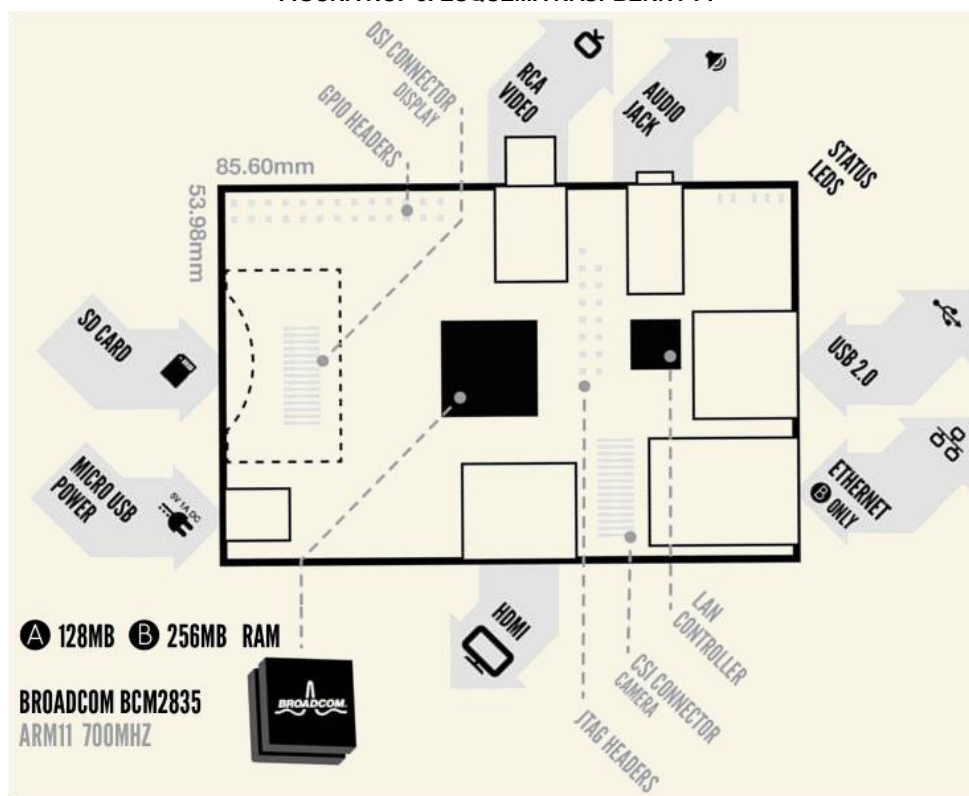
a. Modelo relacional. Los modelos conceptuales (o de alto nivel) son modelos que se basan en cómo el usuario percibe la base de datos. Por otro lado los modelos físicos (de bajo nivel) proporcionan información acerca de cómo los datos se encuentran almacenados en el computador. Entre estos dos conjuntos se encuentran los modelos más utilizados actualmente, los modelos de representación. Estos tienen conceptos que no se encuentran ni muy alejados de la percepción de usuarios ni de la forma en cómo se almacenan los datos. Dentro de los modelos de representación se encuentra el Modelo Relacional.

El modelo relacional representa la base de datos como una colección de relaciones, donde cada relación se representa mediante una tabla de valores. Cada relación (tabla) está compuesta de de tuplas (filas) y atributos (columnas). (Elmasri & Navathe, 2002)

- SQL: El nombre proviene del *inglés Structured Query Language*. En un inicio fue diseñado e implementado por IBM, pero hoy en día el lenguaje se considera entre las principales razones de la popularidad de los modelos relacionales. SQL proporciona una interfaz de lenguaje declarativo de alto nivel; es decir que el usuario solo debe especificar el resultado deseado, y el sistema se encarga de tomar el camino más eficiente para mostrarlo. Este es un lenguaje de base de datos que cuenta con enunciados de definición, consulta y actualización de datos. También cuando con funciones de seguridad, autorización y definición de vistas de la base de datos.

3. Raspberry Pi. *Raspberry Pi Foundation* es una fundación registrada en Reino Unido en mayo del 2009. El objetivo principal era crear un producto que permitiera a estudiantes escolares una mejor y más accesible educación en el área de computación. A principios del 2012 fueron lanzadas oficialmente los modelos A y B del computador en placa *Raspberry Pi*. (Mullins, 2011)

FIGURA NO. 9. ESQUEMA RASPBERRY PI



(Fuente: Element 14 community, 2013)

TABLA NO. 2. ESPECIFICACIONES Y CARACTERÍSTICAS RPI.

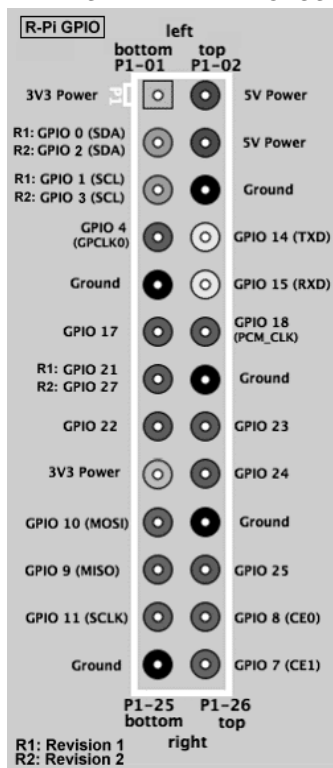
Característica	Modelo A	Modelo B
SoC	Broadcom BCM2835	
CPU	ARM1176JZF-S (700MHz)	
GPU	Broadcom Core IV	
Memoria RAM	256MB	512MB
USB 2.0 (No. Puertos)	1	2
Salida Video	Conector RCA, HDMI	
Salida Audio	Conector Jack 3.5mm, HDMI	
Almacenamiento	Ranuras SDIO	
Conexión a Red	-	10/100 Ethernet
Periféricos (bajo nivel)	GPIO(x8), UART, SPI, I2C, 3.3V, 5V, GND	
Consumo energético	500mA (2.5W)	700mA(3.5W)
Alimentación	5V (ViaMicroUSB o pines GPIO)	
Dimensiones	85.00mm x 53.98mm	
Sistemas Operativos	Debian GNU/Linux, Fedora, Arch Linux	

(Fuente: Element 14 community, 2013)

La *Raspberry Pi* da un fácil acceso de periféricos hacia el CPU a través de sus pines de entrada y salida. La computadora en placa es producida con un conector de 26 pines (2.54mm), arreglados en 2 columnas de 13 pines cada una. Se proporcionan 8 pines de propósito general de entrada y salida (GPIO) y además pines dedicados a los protocolos UART, I2C y SPI. También cuenta con pines de alimentación de 5V, 3.3V y tierra. Uno de los pines de propósito general soporta PWM.

El nivel alto de voltaje en todos los pines es de 3.3V. Se diseñó de esta manera ya que la *Raspberry Pi* no cuenta con protección contra sobre-voltaje. Teniendo salidas de 3.3V hace que los usuarios utilicen módulos extras para el control de dispositivos que requieran mayor potencia. El máximo de corriente permitida suministrar por cada pin es de 50mA, con excepción del pin de alimentación de 5V. La corriente suministrada por este pin depende de obtenida de la alimentación vía USB. (Element14 Community, 2012)

FIGURA NO. 10. PINES DE ENTRADA Y SALIDA PARA PROPÓSITO GENERAL DE RASPBERRY PI



(Fuente: GPIOs, 2013)

4. Software.

a. RASPBIAN. En ámbitos muy generales, un sistema operativo es el encargado de administrar los recursos físicos del computador. Es la conexión entre las aplicaciones y el hardware. El sistema operativo *Debian* es un sistema libre basado en el núcleo Linux. Por lo tanto *Raspbian*, es un sistema operativo (aún no oficial) de *Debian*, creado específicamente para el procesador utilizado por la *Raspberry Pi*. Utiliza el entorno de escritorio conocido como LXDE (*Lightweight X11 Desktop Environment*) y tiene previamente instalados aplicaciones como navegador web, Python y miles de paquetes que optimizan el funcionamiento del dispositivo. (Raspbian Project, 2012)

Los programas *shell* son aquellos que permiten al usuario acceder a todas las funciones del sistema operativo. Algunos poseen interfaz gráfica mientras que otros no. Bash (*BornAgain Shell*) es un programa libre basado en la *shell* de Unix pero diseñada para GNU. Permite ejecutar comandos desde consola y un historial ilimitado de las mismas, entre otras funciones. (GNU Project, 2011)

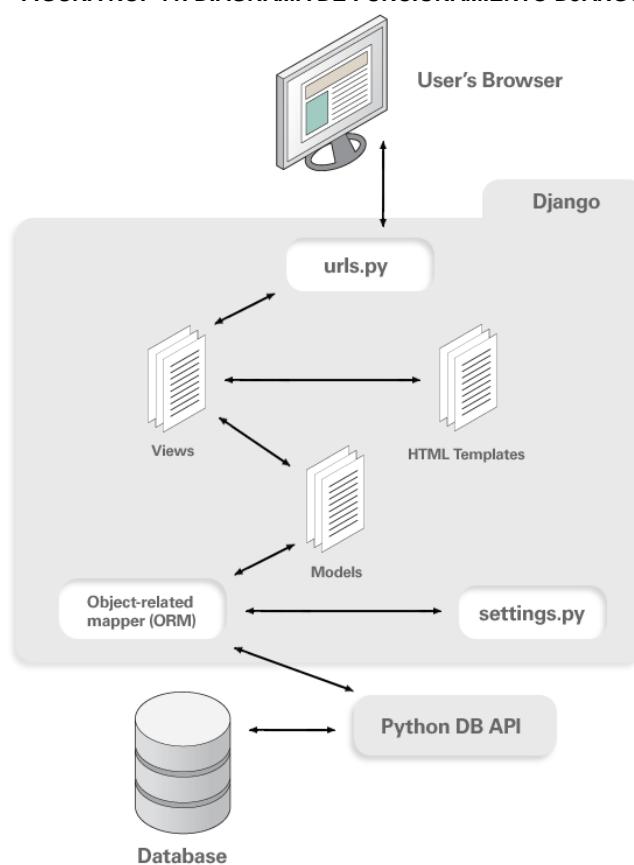
b. DJANGO. Las personas desarrolladoras de software constantemente se encuentran con la rudimentaria tarea de realizar proyectos similares una y otra vez. Por esta razón surgieron los marcos de trabajo (o *frameworks*). Un marco de trabajo tiene predefinido los términos generales de un determinado proyecto a realizar. Django es un marco de trabajo basado en *Python* utilizado para aplicaciones Web. Tiene como objetivo el generar aplicaciones web de alto nivel rápidamente. **(Django Project, s.f.)**

Django posee un ORM (*Object-RelationalMapper*) que es el centro de todo el marco de trabajo. Como su nombre lo indica, el ORM es el encargo de realizar el mapeo entre los modelos de la aplicación de Django (objetos de *Python*) y la base de datos sobre la que se trabaja (base de datos relacional).

Esto quiere decir que después de definir los modelos en la aplicación, es posible crear, borrar, recuperar y actualizar la base de datos a través de los mismos.

El procedimiento que sigue el programa una vez se hace una solicitud al servidor empieza por buscar dentro del archivo “urls.py” una URL que concuerde con la especificada en la solicitud. Una vez encontrada, invoca el archivo “views.py” asociado con la URL. El archivo *view* hace uso de modelos creados en la aplicación, y dependiendo cual sea el tipo de solicitud, pueda desplegar una página en formato HTML. Por ejemplo en el caso que se ha realizado el método *GET* desde un ordenador. La respuesta HTTP que contiene la página HTML o datos planos HTTP (como JSON) es enviada a la dirección que origino la solicitud. El archivo “settings.py” contiene las configuraciones que regulan la aplicación.

FIGURA NO. 11. DIAGRAMA DE FUNCIONAMIENTO DJANGO



(Fuente: Vasiliev, 2009)

c. **SQLITE**. SQLite es una librería ligera y compacta que implementa un motor de base de datos SQL. Su funcionamiento se basa en la lectura y escritura directa de archivos de disco. Es un software libre que no necesita configuración previa para funcionar. (**SQLite, s.f.**)

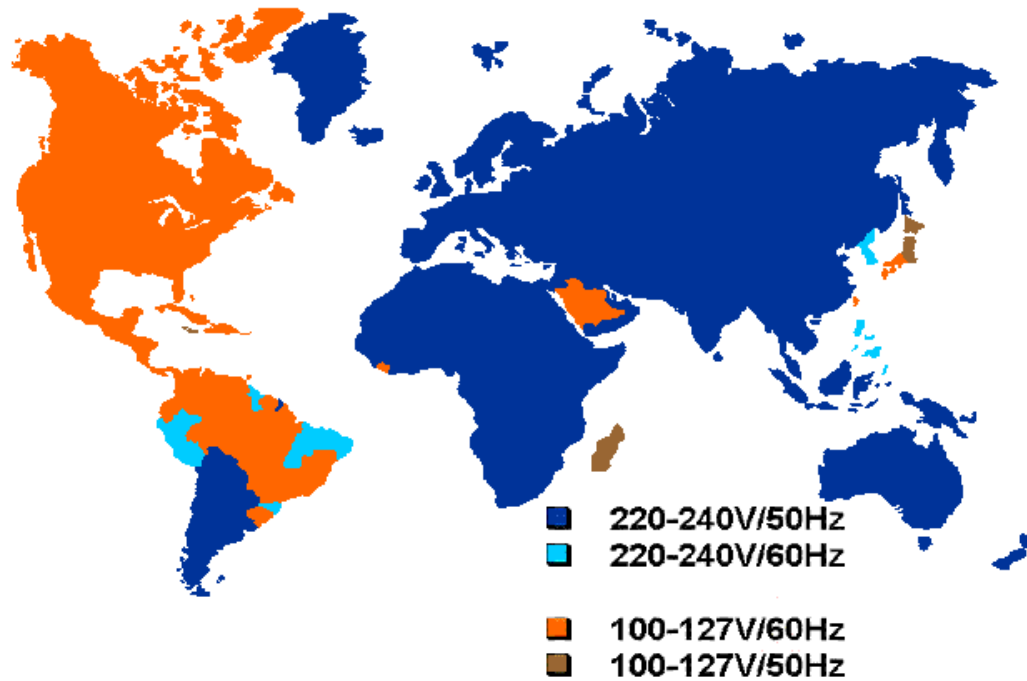
D. INFRAESTRUCTURA DE COMUNICACIÓN

1. **Energía eléctrica**. Se denomina energía eléctrica a la forma de energía que resulta de la existencia de una diferencia de potencial entre dos puntos, lo que permite establecer una corriente eléctrica entre ambos cuando se les pone en contacto por medio de un conductor eléctrico. (IES Cristobal de Monroy)

El sistema de generación y distribución eléctrica de la corriente alterna fue inventado en el siglo XIX por el físico austriaco Nikola Tesla. Realizando experimentos encontró que 60 Hz era la mejor frecuencia para la generación de la energía de la corriente alterna (CA), utilizando 240 voltios como tensión. Sin embargo estos datos no son estándar en todo el mundo.

En toda Europa, Groenlandia y la mayor parte de América del Sur, junto a casi la totalidad de África, Asia y Oceanía se utiliza un voltaje de 230V ($\pm 10\%$), a una frecuencia de 50 Hz (se tienen excepciones, como algunos países de América del Sur que utilizan 60 Hz). Los demás países, principalmente Japón y el resto de América usan de 100 a 127V, con una frecuencia de 60 Hz. En Guatemala la alimentación es de 120V con una frecuencia de 60 Hz. (Otae, 2013)

FIGURA NO. 12. FRECUENCIAS Y VOLTAJES EN EL MUNDO



(Fuente: Otae, 2013)

2. Ruido eléctrico. Se denomina ruido eléctrico (también conocido como interferencia o parásitos) a todas aquellas señales, de origen eléctrico, no deseadas que se encuentran unidas a la señal principal, de manera que la pueden alterar produciendo efectos que pueden ser perjudiciales. (Miyara & Lahoz, 2003)

Las fuentes típicas que producen interferencias son aparatos eléctricos como la televisión, radio, DVD, computadoras, transformadores, etc. Para que estos aparatos generen ruido no necesitan estar encendidos, ya que algunos de ellos siguen encendidos en “*stand by*” cuando se apagan. Cuando el ruido eléctrico se encuentra sobre la red eléctrica puede ocasionar atenuación o bloqueo de las señales transmitidas o recibidas a través de este medio. (Miyara & Lahoz, 2003)

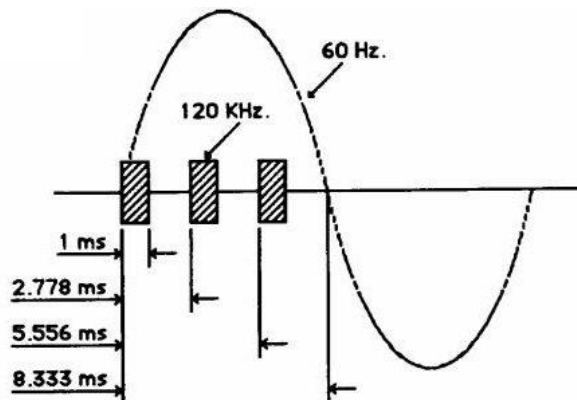
3. Corriente alterna. Se denomina corriente alterna (CA por sus siglas en español y AC por sus siglas en inglés, de *alternating current*) a la corriente eléctrica en la que la magnitud y el sentido varían cíclicamente. La forma de oscilación de la corriente alterna más comúnmente utilizada es la oscilación sinusoidal, ya que consigue una transmisión más eficiente de la energía. Genéricamente la corriente alterna se refiere a la forma en la cual la electricidad llega a los hogares y empresas. (American, 2013)

- Valor instantáneo: valor que toma la corriente en un instante determinado.
- Valor pico: valor máximo que toma la oscilación sinusoidal.
- Valor pico a pico: diferencia entre su máximo positivo y su pico negativo.
- Valor eficaz: valor de corriente continua que produce la misma disipación de potencia en una resistencia que su equivalente de corriente alterna. Por sus siglas en inglés es conocido como el valor RMS (*root mean square*). (American, 2013)

4. Protocolo X10. Este es un protocolo de comunicación desarrollado en Escocia, en 1978, por *Pico Electronics*, que permite el control de dispositivos eléctricos utilizando la línea de corriente alterna (AC) para transmitir señales de control entre equipos de automatización. X10 surgió de una familia de chips denominada los proyectos X. Su idea era obtener un circuito que pudiera ser insertado en un sistema mayor y que pudiera ser controlado remotamente. (Infantes Díaz, 2009)

a. Método físico. La transmisión se encuentra sincronizada con el cruce por cero de la señal de poder AC. Un bit de información puede ser transmitido en cada cruce por cero. Un 1 digital es representando por un tren de pulsos de 1 ms, a una frecuencia de 120 KHz, mientras que el 0 digital se representa por la ausencia de estos pulsos cuando se produce el cruce por cero. (**Powerline Control Systems, 2003**)

FIGURA NO. 13. TRANSMISIÓN DE DATOS EN X10.



(Fuente: Powerline control systems, 2003)

b. Direccionamiento. A cada uno de los dispositivos que utilizan este protocolo se le puede asignar uno de los dieciséis posibles “códigos de casa” (nombrados desde “A” hasta “P”) y una de las dieciséis posibles unidades de direccionamiento (desde 1 hasta 16), dando un total de 256 dispositivos que pueden ser direccionados en una sola línea de transmisión. El “código de casa” es utilizado para separar un grupo de dispositivos que se encuentran en una casa con los de otra vecina. La recepción de mensajes depende de la correcta asignación de código de casa y dirección. **(Powerline Control Systems, 2003)**

TABLA NO. 3. CÓDIGOS PARA DIRECCIONAMIENTO DE CASA X10.

Dirección de casa	H1	H2	H3	H4
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

(Fuente: Powerline control systems, 2003)

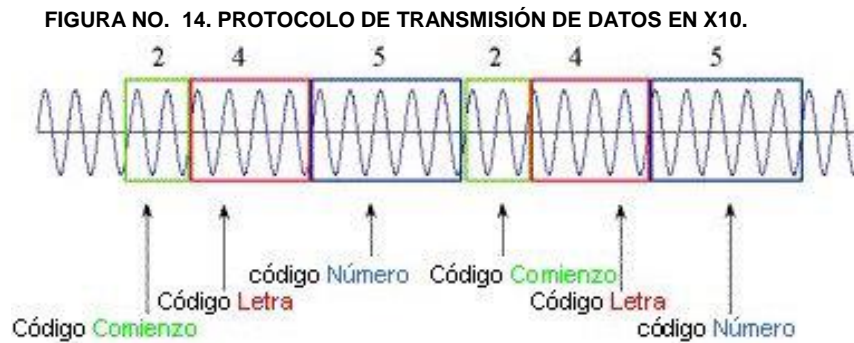
TABLA NO. 4. CÓDIGOS PARA DIRECCIONAMIENTO DE UNIDAD X10.

Dirección de unidad	D1	D2	D3	D4
1	0	1	1	0
2	1	1	1	0
3	0	0	1	0
4	1	0	1	0
5	0	0	0	1
6	1	0	0	1
7	0	1	0	1
8	1	1	0	1
9	0	1	1	1
10	1	1	1	1
11	0	0	1	1
12	1	0	1	1
13	0	0	0	0
14	1	0	0	0
15	0	1	0	0
16	1	1	0	0

(Fuente: Powerline control systems, 2003)

c. Estructura del mensaje. Un bloque completo de datos en X10 se encuentra compuesto por un código de inicio (secuencia especial de 4 bits, 1110), seguido por el código de casa, y finalmente por el código de llave. El código de casa tiene una longitud de 8 bits, formado por cuatro pares de bits complementados, mientras que el código de llave tiene una longitud de 10 bits formado por cinco pares complementados de bits. El código de llave puede ser tanto un código de unidad, como un código de función, dependiendo si el mensaje es una dirección o un comando. **(Powerline Control Systems, 2003)**

Para la transmisión se debe de enviar dos veces el paquete de datos a través de la línea eléctrica. Una transmisión de código completa abarca once ciclos de la línea eléctrica, los primeros dos ciclos representa el código de inicio, los siguientes cuatro representan el código de casa y los últimos cinco representa el código de unidad. Este bloque debe ser siempre transmitido teniendo tres ciclos de la línea eléctrica entre cada grupo de dos códigos. (Cuevas, Martínez, & Merino, 2002)



(Fuente: Infantes Díaz, 2009)

d. Tipos de dispositivos. Se distinguen distintos tipos de dispositivos X10:

- Transmisores: encargados de enviar una señal especialmente codificada de bajo voltaje que es superpuesta sobre el voltaje cableado. Es capaz de enviar información hasta a 256 dispositivos sobre el cableado eléctrico. Múltiples transmisores pueden enviar señales al mismo módulo (Cuevas, Martínez, & Merino, 2002)
- Receptores: al igual que los transmisores pueden comunicarse con 256 direcciones distintas. Generalmente tienen un código establecido por el usuario para indicar la dirección del dispositivo. (Cuevas, Martínez, & Merino, 2002)
- Bidireccionales: estos dispositivos tienen la capacidad de responder y confirmar la correcta realización de la orden, lo cual puede ser muy útil cuando el sistema está conectado a un programa ordenador que muestra los estados en que se encuentra la instalación domótica de la vivienda. (Cuevas, Martínez, & Merino, 2002)
- Inalámbricos: unidades que permiten conectarse a través de una antena y enviar señales de radio desde una unidad inalámbrica e inyectar la señal X10 en el cableado eléctrico. (Cuevas, Martínez, & Merino, 2002)

5. Protocolo de comunicación KNX. KNX es la abreviatura de Konex, asociación que nace en 1999 y que desarrolla un protocolo certificado libre especializado en la domótica. El sistema realiza la transmisión de datos a través de un cable o bus al que están conectados todos los dispositivos. El bus permite que todos los componentes de las instalaciones domóticas estén intercomunicados entre sí, de esta forma se logra que cualquier componente de órdenes a otro sin importar la distancia entre ellos. (Peña Alcaráz, 2012)

6. Protocolo de comunicación Zigbee. Conjunto de protocolos de alto nivel de comunicación inalámbrica. Su objetivo son aplicaciones que requieren comunicaciones seguras con baja tasa de envío. A pesar de que su objetivo es la domótica, en la actualidad el mayor problema que presenta es que no está optimizado para transmitir gran cantidad de datos. Gracias a su bajo consumo, fácil instalación y su topología en red de mallas, hace que cada vez cobre más fuerza en el ámbito domótico. (Peña Alcaráz, 2012)

7. Modulación ASK. Modulación es el proceso en el que se convierte una señal digital en una señal analógica. El transmisor emite una señal denominada portadora, siendo generalmente una señal eléctrica sinusoidal de mucha mayor frecuencia que otra señal denominada moduladora. Esta constituye la información que se prepara para la transmisión. Su objetivo es modificar alguna característica de la onda portadora, de manera que se pueda obtener una señal que incluya la información de la moduladora. De esta manera el demodulador puede recuperar la señal original, quitando la portadora. La modulación por desplazamiento de amplitud (*Amplitud Shift Keying*, por sus siglas en inglés), también conocida como OOK (*On-off-keying*), es una técnica de modulación digital en la que la información se encuentra modulada en amplitud. (Luque Rodríguez & Clavijo Suero, 1995)

Para el análisis se considera a la señal moduladora (datos) como una señal binaria, que guarda la información a transmitir, con niveles de tensión de 0 a A voltios.

La función de la señal moduladora es la siguiente:

$$v_m(t) = \begin{cases} A & \text{para un 1 binario} \\ 0 & \text{para un 0 binario} \end{cases}$$

La función de la señal portadora está dada por:

$$v_c(t) = V_c \text{sen}(2\pi * f_c * t)$$

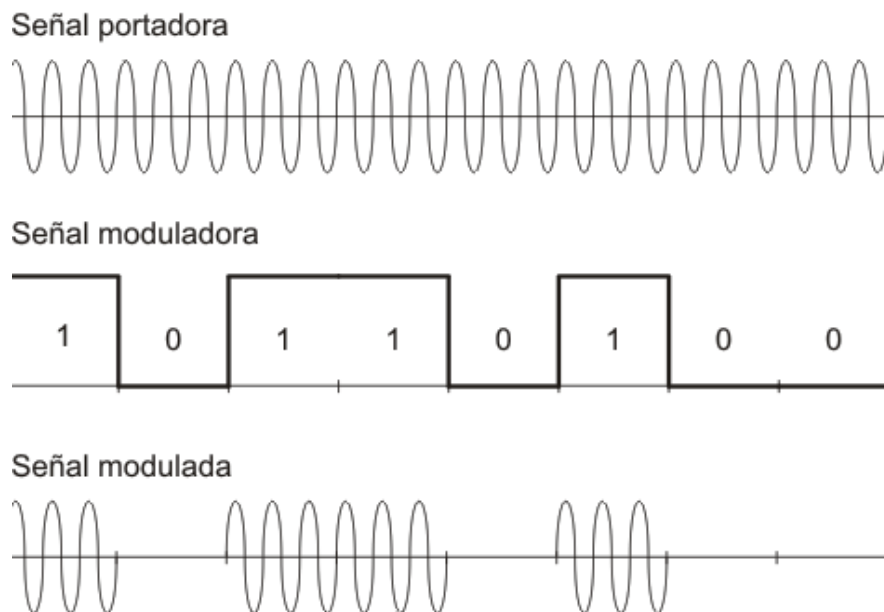
Donde V_c es el valor pico de la señal portadora y f_c es la frecuencia de la señal portadora.

La función de la señal modulada ASK es:

$$v(t) = V_m V_c \text{sen}(2\pi * f_c * t)$$

Si $v_m = 0$ entonces $v(t) = 0$ y si $v_m = A$ entonces $v(t) = AV_c \text{sen}(2\pi * f_c * t)$, tal como se muestra en la Figura 15.

FIGURA NO. 15. SEÑAL DE MODULACIÓN ASK.



(Fuente: Luque Rodríguez & Clavijo Suero, 1995)

La velocidad de transmisión de datos digitales tiene como dimensional el baudio (símbolos por segundo). En la modulación ASK un símbolo de la señal modulada porta únicamente un bit de información, por lo que para este caso coincide la velocidad de transmisión de datos baudios y en bits por segundo. (Luque Rodríguez & Clavijo Suero, 1995)

8. Circuito LC. Un circuito LC, también conocido como circuito resonante, se encuentra formado por un inductor y un capacitor. En dicho circuito existe una frecuencia para la cual se produce un fenómeno de resonancia eléctrica llamada frecuencia de resonancia, para la cual la reactancia inductiva es igual a la reactancia capacitiva ($X_c = X_L$). Por lo tanto la impedancia será mínima e igual a la resistencia óhmica, lo que equivale a decir que el circuito estará en fase. (Gómez Tejedor & Olmos Sanchís, 1999)

Una de las características de los circuitos LC es que la energía liberada por un elemento reactivo (inductor o capacitor) es exactamente igual a la absorbida por el otro. Es decir, durante la primera mitad de un ciclo de entrada el inductor absorbe toda energía liberada por el condensador, y durante la segunda mitad del ciclo el condensador vuelve a capturar la energía proveniente del inductor. Es esta condición oscilatoria la que se conoce como resonancia, y la frecuencia en la que esta condición se produce es llamada frecuencia resonante. (Gómez Tejedor & Olmos Sanchís, 1999)

Los circuitos resonantes son especialmente útiles cuando se desea hacer sintonizadores en los cuales se quiere dar suficiente potencia a solamente una frecuencia (o un rango de frecuencias muy reducido) dentro de un espectro. (Gómez Tejedor & Olmos Sanchís, 1999)

9. Circuito LC en serie. La impedancia de un circuito LC en serie es:

$$Z = jL\omega - j\frac{1}{\omega C} = j\left(L\omega - \frac{1}{\omega C}\right) = jX_s$$

Donde X_s es:

$$X_s = L\omega - \frac{1}{\omega C}$$

Debe existir un valor ω tal que haga nulo el valor de X_s , este valor será la pulsación de resonancia del circuito a la que se le denomina ω_0 .

$$L\omega_0 = \frac{1}{\omega_0 C}$$

De donde despejando ω_0 se encuentra:

$$\omega_0 = \sqrt{\frac{1}{LC}}$$

Teniendo en cuenta que $\omega_0 = 2\pi f_0$ la frecuencia de resonancia será:

$$f_0 = \frac{1}{2\pi\sqrt{LC}}$$

10. Circuito LC en paralelo. La impedancia de un circuito LC en paralelo es:

$$Z = \frac{jL\omega \cdot 1/j\omega C}{jL\omega + 1/j\omega C} = \frac{jL\omega}{1 - \omega^2 LC} = jX_p$$

Donde X_p es:

$$X_p = \frac{L\omega}{1 - \omega^2 LC}$$

Analizando el comportamiento del conjunto para distintos valores de ω se tiene:

$$\omega = 0 \text{ entonces } X_p = 0$$

$\omega < \omega_0$ entonces $X_p > 0$ se tiene un comportamiento inductivo

$\omega > \omega_0$ entonces $X_p < 0$ se tiene un comportamiento capacitivo

$\omega^2 LC = 1$ entonces $X_p = \infty$

$\omega = \infty$ entonces $X_p = 0$

Despejando ω_0 se encuentra:

$$\omega_0 = \sqrt{\frac{1}{LC}}$$

Teniendo en cuenta que $\omega_0 = 2\pi f_0$ la frecuencia de resonancia será:

$$f_0 = \frac{1}{2\pi\sqrt{LC}}$$

Siendo esta denominada frecuencia de anti resonancia, frecuencia a la cual la impedancia se hace infinita. (Gómez Tejedor & Olmos Sanchís, 1999)

E. COMUNICACIÓN Y SEGURIDAD

1. Red de comunicación de datos. Una red es un conjunto de dispositivos (nodos) conectados por enlaces de un medio físico. Un nodo puede ser cualquier dispositivo capaz de enviar y/o recibir datos generados por otros nodos de la red. Los enlaces conectados con los dispositivos se denominan canales de comunicación. (Behrouz A., 2001)

La palabra datos se refiere a hechos, conceptos e instrucciones presentados en cualquier formato acordado entre las partes que crean y utilizan dichos datos. En el contexto de los sistemas de información basados en computadora, los datos se representan con unidades de información binaria producidas y consumidas en forma de ceros y unos. (Behrouz A., 2001)

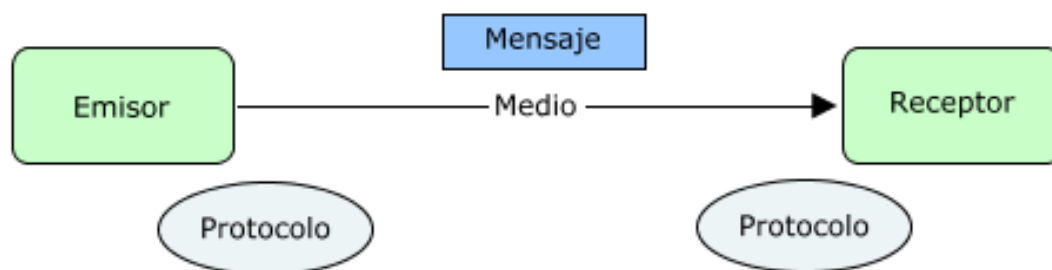
La transmisión de datos es el intercambio de datos entre dos dispositivos a través de alguna forma de medio de transmisión. La transmisión de datos se considera local si los dispositivos de comunicación están en el mismo lugar físico o en un área geográfica restringida, se considera remota si estos están separados por una distancia considerable. Para que la transmisión de datos sea posible, los dispositivos de comunicación deben ser parte de un sistema conformado por hardware y software. La efectividad del sistema de comunicación depende de tres características esenciales: (Behrouz A., 2001)

- Entrega: El sistema debe entregar los datos en el destino correcto. Los datos deben ser recibidos por el dispositivo o usuario adecuado y solamente por ese dispositivo o usuario.
- Exactitud: El sistema debe entregar los datos con exactitud. Los datos que se alteren en la transmisión son incorrectos y no se pueden utilizar.
- Puntualidad: El sistema debe entregar los datos con puntualidad. Los datos deben entregarse a medida que se producen, en el mismo orden en que se producen y sin un retraso significativo.

Un sistema de transmisión de datos está formado por cinco componentes: (Behrouz A., 2001)

- Mensaje: Es la información (datos) a comunicar.
- Emisor: Es el dispositivo que envía los datos del mensaje.
- Receptor: Es el dispositivo que recibe el mensaje.
- Medio: Es el camino físico por el cual viaja el mensaje del emisor al receptor.
- Protocolo: Es un conjunto de reglas que gobiernan la transmisión de datos.

FIGURA NO. 16. COMPONENTES DE UN SISTEMA DE TRANSMISIÓN DE DATOS



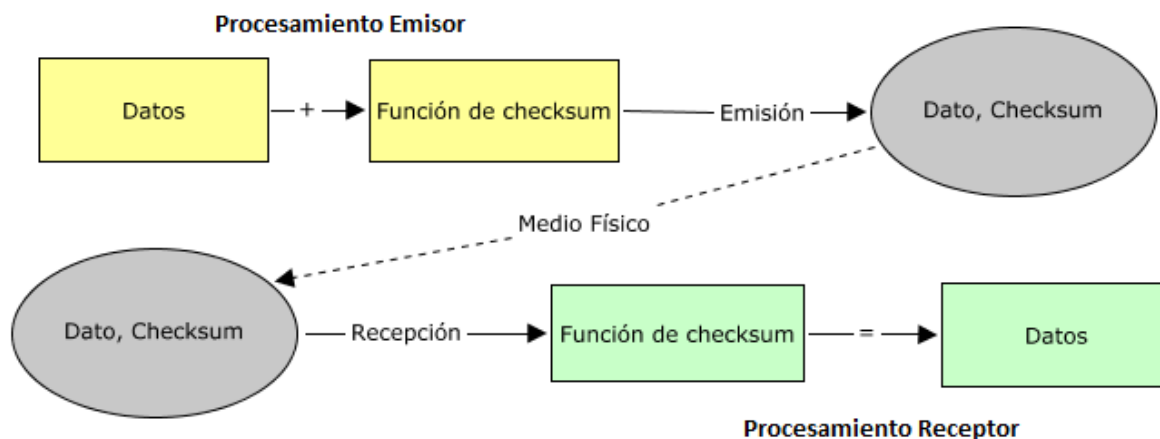
2. Checksum. Los *checksums* son utilizados para asegurar la integridad de porciones de datos que se desean transmitir. Un *checksum* se puede definir como un número específico obtenido a partir de la porción de datos a transmitir. (Wireshark, 2004-2013)

La transmisión de datos en la mayoría de redes produce errores tales como cambiar el valor, anular y duplicar bits. Como resultado de esto los datos recibidos pueden no ser idénticos a los transmitidos, lo cual resulta en información no válida. (Wireshark, 2004-2013)

Debido a estos errores de transmisión, los protocolos de redes usualmente implementan algún tipo de *checksum* para detectar dichos errores. El transmisor calcula un *checksum* del dato y luego lo envía junto con el dato. El receptor calcula el *checksum* del dato recibido con el mismo algoritmo que el utilizado por el emisor. Si el *checksum* recibido y el calculado son iguales el receptor asume que el dato es válido, de lo contrario, el dato es desechado. (Wireshark, 2004-2013)

Existen distintos tipos de *checksums*, la elección o diseño de un *checksum* para un protocolo en específico depende del error esperado en la red, de la capacidad de procesamiento del sistema y del desempeño esperado. (Wireshark, 2004-2013)

FIGURA NO. 17. ENVÍO Y RECEPCIÓN DE DATOS APLICANDO CHEKSUM



3. Lenguaje de programación mikroC. *mikroC PRO for PIC* es un software propiedad de mikroElektronika, el cual está diseñado como herramienta de programación de alto nivel para el desarrollo de aplicaciones con microcontroladores PIC de Microchip. (mikroElektronika, 2009)

Los microcontroladores PIC son los chips de 8-bits más populares del mundo, utilizados en gran variedad de aplicaciones, mientras que el lenguaje de programación de alto nivel C es reconocido en el entorno del software por su alta eficiencia. La combinación de estas dos herramientas tecnológicas permite el desarrollo sencillo de aplicaciones complejas. (mikroElektronika, 2009)

La principal ventaja de este lenguaje es la posibilidad de utilizar las librerías compatibles con los distintos módulos de los microcontroladores, facilitando así su implementación, en específico para este proyecto con los módulos de comunicación por hardware y software *EUSART*. (mikroElektronika, 2009)

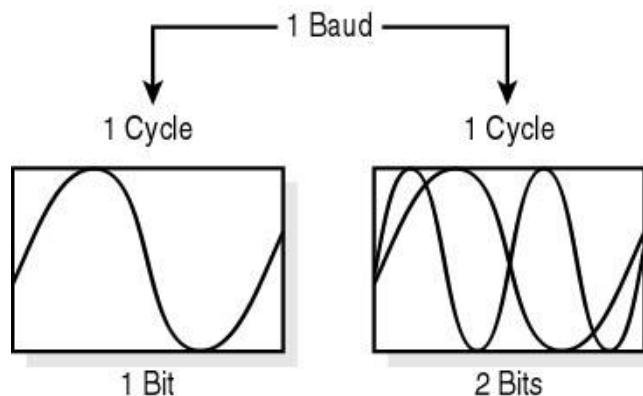
4. EUSART. EUSART son las siglas de “*Enhanced Universal Synchronous Asynchronous Receiver Transmitter*”. Éste es un módulo integrado en muchos de los microcontroladores PIC, el cual es un periférico de comunicación de entrada/salida de datos. También se le conoce como Interfaz de Comunicación Serial (SCI). (Incorporated, 2009, págs. 155 - 170)

Se puede implementar en configuración “*full-duplex* asíncrona” o “*half-duplex* síncrona”. El primer término se refiere a la capacidad de enviar y recibir datos al mismo tiempo, mientras que el segundo se refiere a la necesidad de implementar una sincronización por medio de un pulso de reloj. (Incorporated, 2009, págs. 155 - 170)

5. Baud rate. También conocido como tasa de baudios, el *baud rate* puede interpretarse como el número de cambios de voltaje o frecuencia en un segundo. Un sistema operando a “n” baudios por segundo significa que su señal portadora básica es de “n” ciclos por segundo.

En el caso en que cada ciclo sea un bit, la velocidad máxima de transferencia de datos estará restringida por el número de baudios por segundo. Sin embargo, utilizando distintas técnicas de modulación, se puede transmitir más de un bit por cada ciclo. (Brainbell, s.f.)

FIGURA NO. 18. COMPARACIÓN DE DATOS ENVIADOS UTILIZANDO EL MISMO *BAUD RATE*



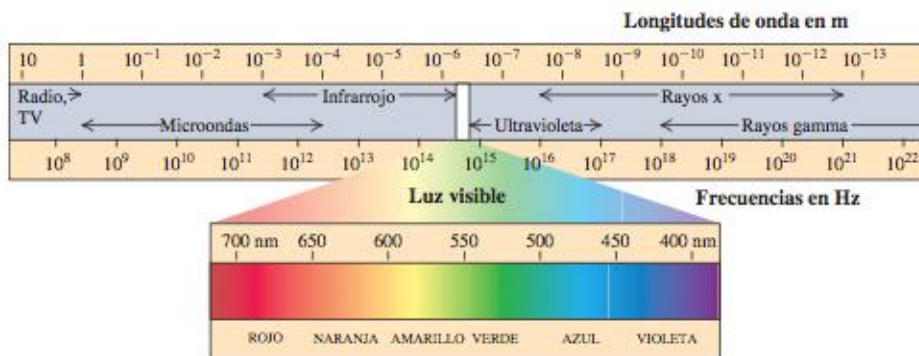
(Fuente: Brainbell)

F. PERIFÉRICOS DE AMBIENTE

1. Iluminación. En los sistemas de iluminación es necesario el empleo de una serie de conceptos básicos, que principalmente son:

- Luz visible. Se refiere a la radiación correspondiente a la región del espectro electromagnético que el ojo humano es capaz de percibir. Su intervalo de longitud de onda va de 400 a 700 nm, con frecuencias correspondientes de 750 a 430 THz aproximadamente. Las distintas partes del espectro visible evocan en los humanos las sensaciones de los diferentes colores. En la Figura No. 2 se presentan las longitudes de onda de los colores en la parte visible del espectro. (Foster)

FIGURA NO. 19. REGIÓN DEL ESPECTRO ELECTROMAGNÉTICO CORRESPONDIENTE A LA LUZ VISIBLE



(Fuente: Foster)

- Flujo luminoso: Es la cantidad de luz emitida por una fuente luminosa. Es un factor que depende únicamente de las propiedades intrínsecas de la fuente. Su unidad en el SI es el lumen (lm). (Finol)
- Rendimiento luminoso: Es la cantidad de energía que se convierte en luz visible con relación a la energía total consumida. Permite medir la eficiencia luminosa de una fuente. En el SI se mide en flujo luminoso por unidad de energía consumida (lm/J). (Finol)

- **Intensidad luminosa:** Flujo luminoso emitido en un ángulo y en una dirección. Se utiliza para expresar cómo se reparte la luz de una fuente en las distintas direcciones, ya que las fuentes luminosas normalmente no emiten el mismo flujo luminoso en todas las direcciones. La intensidad se mide en candelas (cd). (Finol)
- **Iluminancia:** Cantidad de luz que llega a una superficie determinada. Su unidad en el SI, lux, equivale al flujo luminoso de un lumen que incide homogéneamente sobre una superficie de un metro cuadrado. (Finol)
- **Brillantez fotométrica:** Se define para una superficie en una dirección determinada, y es la relación entre la intensidad luminosa y la superficie vista por un observador situado en la misma dirección. Su unidad de medida en el SI es cd/m². (Finol)
- **Temperatura del color:** las fuentes de luz pueden crear atmósferas cálidas o frías en su apariencia. La temperatura de color, expresada en Kelvin (K), es una forma de describir esta tonalidad. Cuanto mayor sea la temperatura de color, la luz será más fría y azulosa. (Foster)

FIGURA NO. 20. TEMPERATURA CORRESPONDIENTE AL COLOR EMITIDO

Temperatura	Fuentes típicas
1000K	Velas, lámparas de aceite
2000K	Amanecer muy temprano, lámparas de tungsteno de bajo efecto
2500K	Bombillas caseras
3000K	Luz de estudio (continua), "photo floods"
4000K	Lámparas de magnesio claras (hoy en día obsoletas)
5000K	Luz día normal, flash electrónico
5500K	El sol de mediodía
6000K	Día muy soleado con cielo despejado
7000K	Cielo ligeramente nublado
8000K	Cielo brumoso
9000K	Sombra amplia en un día despejado
10,000K	Cielo muy brumoso
11,000K	Cielos azules sin sol
20,000+K	Sombra amplia en montañas o en un día muy despejado

(Fuente: Young & Freedman, 2009)

- Reflectancia: Proporción de luz reflejada por una superficie. (Finol)
- Contraste: Diferencia de luminancia entre un objeto y su entorno o entre diferentes partes de un objeto. (Finol)

2. Tipos de lámparas utilizadas para iluminación

a. Lámparas incandescentes. Estas utilizan un filamento de tungsteno dentro de un globo de vidrio al vacío o lleno de un gas inerte que evite la evaporación del tungsteno y reduzca el ennegrecimiento del globo. Existen lámparas de muy diversas formas, que pueden resultar muy decorativas. En la Figura No. 4 se muestran los componentes de una lámpara típica de iluminación general. **(Orion)**

FIGURA NO. 21. COMPONENTE DE UNA LÁMPARA INCANDESCENTE



(Fuente: Orion)

Las lámparas incandescentes también se presentan en una amplia gama de colores y acabados. Se trata de unas lámparas que siguen teniendo aceptación en la iluminación doméstica debido a su bajo coste y pequeño tamaño.

Con todo, su baja eficiencia genera costes de explotación muy altos en la iluminación comercial e industrial, por lo que normalmente se prefieren las lámparas de descarga. Una lámpara de 100 W tiene una eficiencia típica de 14 lúmenes/vatio en comparación con los 96 lúmenes/vatio de una lámpara fluorescente de 36 W. (Orion)

Las lámparas incandescentes todavía se utilizan cuando la atenuación de la luz es una característica de control conveniente, ya que resulta fácil atenuarlas reduciendo la tensión de alimentación. (Orion)

b. Lámparas halógenas de tungsteno. Son parecidas a las lámparas incandescentes y producen luz de la misma manera, a partir de un filamento de tungsteno. Ahora bien, el globo contiene gas halógeno (bromo o yodo) que actúa controlando la evaporación del tungsteno. La mayoría de las lámparas halógenas de tungsteno duran más tiempo que sus equivalentes incandescentes y el filamento alcanza una temperatura más alta, creando más luz y un color más blanco. **(Orion)**

Las lámparas halógenas de tungsteno han encontrado aceptación en situaciones cuyos principales requisitos son un tamaño reducido y un alto rendimiento. Como ejemplo típico cabe citar la iluminación de escenarios, incluyendo el cine y la televisión, donde el control direccional y la atenuación son requisitos habituales. (Orion)

c. Lámparas fluorescentes. Son lámparas de mercurio de baja presión. Para el cebado de la lámpara requieren de un precalentamiento de los electrodos para que la ionización del gas y del vapor de mercurio sea suficiente para realizar la descarga.

Necesitan equipo de control externo para efectuar el cebado y para regular la corriente de la lámpara. Además de la pequeña cantidad de vapor de mercurio, hay un gas de cebado (argón o criptón). (Orion)

La baja presión del mercurio genera una descarga de luz de color azul pálido. La mayor parte de la radiación está en la región ultravioleta a 254 nm, una frecuencia de radiación característica del mercurio. En el interior de la pared del tubo hay un fino revestimiento fosfórico, que absorbe los rayos ultravioleta e irradia la energía en forma de luz visible. Existe toda una gama de materiales fosfóricos con diversas características de coloración y reproducción del color. Los modernos materiales fosfóricos más duraderos, mejoran la constancia del flujo luminoso y aumentan la vida útil de la lámpara. Estas lámparas no son adecuadas para el efecto de atenuación. (Orion)

d. Lámparas fluorescentes compactas (LFC). Aprovecha la tecnología de los tradicionales tubos fluorescentes para hacer lámparas de menor tamaño que puedan sustituir a las lámparas incandescentes con pocos cambios en la armadura de instalación y con menor consumo. **(Orion)**

En comparación con las lámparas incandescentes, las LFC tienen una vida útil más larga y consumen menos energía eléctrica para producir la misma cantidad de luz. Como desventajas, muchas de ellas no alcanzan su máximo brillo de forma inmediata y es más problemático deshacerse de ellas ya que contienen residuos tóxicos. Además, no son adecuadas para el efecto de atenuación. (Orion)

e. Lámparas de inducción. Son lámparas de mercurio de baja presión con revestimientos trifosfóricos y cuya producción de luz es similar a la de las lámparas fluorescentes. La energía se transmite a la lámpara por radiación de alta frecuencia, aproximadamente a 2,5 MHz, desde una antena situada en el centro de la lámpara. No existe conexión física entre la bombilla y la bobina. Sin electrodos u otras conexiones alámbricas, la construcción del recipiente de descarga es más sencilla y duradera. La vida útil de la lámpara se determina principalmente por la fiabilidad de los componentes electrónicos y la constancia del flujo luminoso del revestimiento fosfórico. **(Orion)**

f. Iluminación basada en LEDs. Un LED (*Lighting Emitting Diode*) es un diodo semiconductor capaz de emitir luz. Desde hace muchos años se ha venido usando en diversos dispositivos, sobre todo en botones para indicar estados como por ejemplo si el aparato está encendido o apagado.

El primer LED se desarrolló en 1927 por Oleg Vladimírovich Lósev pero no fue hasta los sesenta cuando comenzó a usarse en la industria. Desde entonces los avances se han sucedido y desde los primeros LEDs que sólo se podían construir en color rojo, verde o amarillo y con una intensidad de luz baja hemos pasado a nuevos que pueden ofrecer una luz blanca y suficiente para iluminar una habitación. (Oisterwijk, 2011)

La luz de LED es monocromática y depende del material utilizada en el semiconductor. Según éste conseguimos que la luz emitida sea roja, azul, ultravioleta, entre otros. Para conseguir diferentes tonos de color, teniendo en cuenta ese detalle de ser luz monocromática, lo que hacemos es recurrir la combinación entre ellos y el uso de diferentes intensidades. De esta forma para conseguir luz blanca podemos mezclar la luz de tres LEDs, uno azul, otro rojo y por último verde, llamados LEDs RGB. (Oisterwijk, 2011)

La iluminación basada en LED tiene grandes ventajas respecto a otros sistemas de iluminación. Entre estas ventajas se puede mencionar:

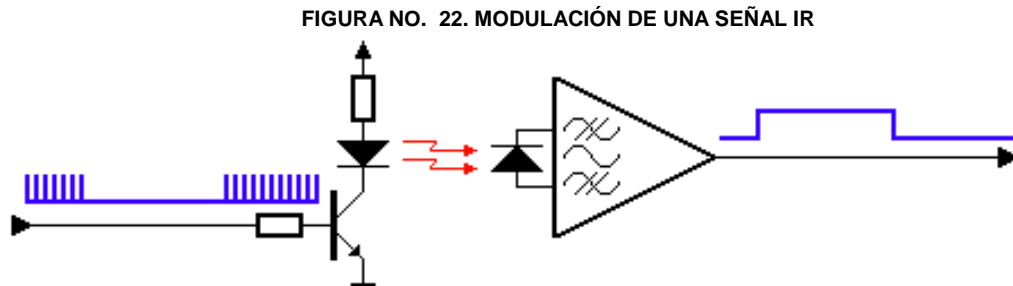
- **Ahorro energético:** Los sistemas de iluminación LED se alimentan directamente conectados a la línea eléctrica (220V) o a baja tensión (12/24V). Su bajo consumo permite que incluso se alimenten con paneles solares, utilizando una energía renovable, limpia e independiente. En cualquier caso, para una misma iluminación, conseguimos reducir el consumo hasta 8 veces.
- **Menor mantenimiento:** Mientras las lámparas de bajo consumo duran unas 10000 horas, los LED llegan hasta las 50000, siendo además más robustos y más difíciles de romper.
- **Luz más brillante:** En las mismas condiciones que otras fuentes de luz, la luz que emite el LED es mucho más nítida y brillante.
- **Mayor rapidez:** El LED se enciende de forma casi instantánea, mucho más rápido que el halógeno y el fluorescente, del orden de algunos microsegundos.
- **Baja temperatura:** El reducido consumo del LED emite muy poco calor.
- **Sin interrupciones de iluminación:** El LED absorbe las vibraciones sin producir fallos ni variaciones de iluminación.
- **Amplio catálogo de color:** Existen en el mercado una gran gama de LED's que nos permitirán iluminar con un determinado color.
- **Portabilidad:** Su bajo consumo permite la fabricación de sistemas portátiles con alimentación autónoma.
- **Medio ambiente:** Para su fabricación no se utiliza ningún producto contaminante, y durante su uso se consume muy poca energía, con lo que no genera tanto CO₂ como los focos convencionales. Tampoco contienen mercurio. (Santamaría)

3. Control remoto por infrarrojo. Un control remoto o mando a distancia es un dispositivo electrónico usado para realizar una operación remota sobre una máquina o dispositivo como el televisor u otro tipo de aparato electrónico casero, como DVD, computadoras, y para encender y apagar un interruptor, la alarma, o abrir la puerta del estacionamiento. Los controles remotos para esos aparatos son normalmente pequeños objetos (fácilmente manipulables con una mano) con una matriz de botones para ajustar los distintos valores, como por ejemplo, el canal de televisión, el número de canción y el volumen. De hecho, en la mayoría de dispositivos modernos el control contiene todas las funciones de control, mientras que el propio aparato controlado sólo dispone de los controles más primarios. La mayoría de estos controles remotos se comunican con sus respectivos aparatos vía señales de infrarrojo (IR) y sólo unos pocos utilizan señales de radio. (Orion)

4. Comunicación por infrarrojo. Infrarrojo (IR) es un tipo de radiación electromagnética, de mayor longitud de onda que la luz visible, pero menor que la de las microondas. No es visible para el ojo humano, lo cual es una de las principales razones por la que se utiliza IR para los propósitos del control remoto. Otra razón es que los LEDs IR son relativamente fáciles de elaborar y por lo tanto tienen un precio muy accesible. Para los controles IR se utiliza principalmente IR con una longitud de onda de 950 nm. (Oisterwijk, 2011)

Una de las desventajas del IR es que existen muchas fuentes que emiten radiación de este tipo. El Sol, por ejemplo, es la fuente que más radiación IR emite. Además, hay otras fuentes que pueden afectar como las lámparas de iluminación, las candelas, los sistemas de calefacción, e incluso el cuerpo humano. Por lo tanto se deben tomar ciertas precauciones para garantizar que los mensajes por IR lleguen al receptor sin errores. (Oisterwijk, 2011)

- Modulación: La modulación es la respuesta para que la señal IR pueda llegar al receptor libre de ruido. El principio fundamental es hacer parpadear la fuente de IR a una frecuencia particular, y sintonizar el receptor IR a la misma frecuencia, para que de esta manera el receptor ignore todas las demás fuentes de IR.

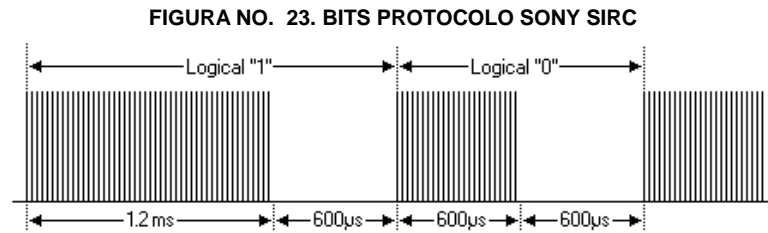


(Fuente: Oisterwijk, 2011)

En la figura se puede observar una señal IR siendo modulada y enviada utilizando un LED IR, y luego siendo detectada esta señal del otro lado con el receptor. En este tipo de comunicación, las frecuencias utilizadas más comúnmente se encuentran entre los 30 y los 60 kHz. Para los electrónicos de consumo las frecuencias más utilizadas se encuentran entre los 36 y 40 kHz. (Oisterwijk, 2011)

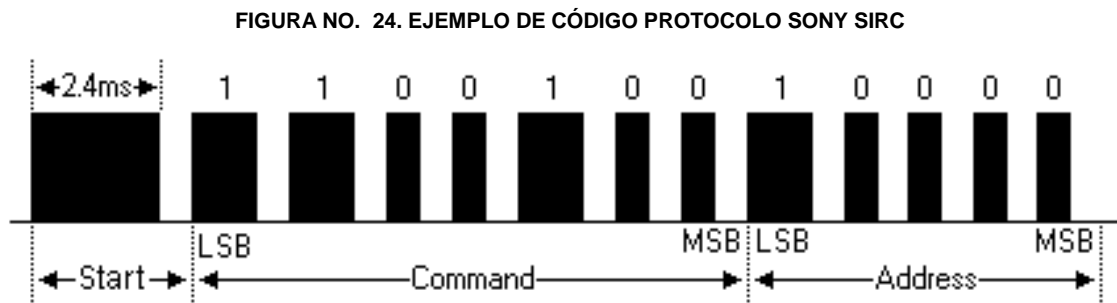
5. Protocolos de comunicación por infrarrojo. Entre los protocolos más utilizados para la comunicación IR en electrónicos de consumo se encuentran los siguientes:

a. Protocolo SONY SIRC. Utiliza una codificación por ancho de pulso. Un pulso de 1.2 ms emitido con una frecuencia portadora a 40 kHz representa un "1" lógico, mientras que para un "0" lógico se emite un pulso de 0.6 ms. Todos los pulsos están separados por un espacio sin emisión de 0.6 ms. Se recomienda un ciclo de trabajo de 25 a 35 % para la señal portadora.



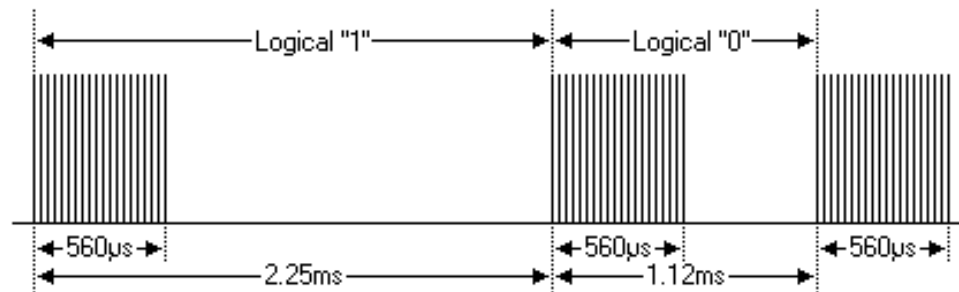
A continuación se presentan algunas de sus características:

- Utiliza modulación por ancho de pulso
- Frecuencia portadora de 40 kHz
- Bit time de 1.2 ms o 0.6 ms
- Existen versiones de 12-bits, 15-bits y 20-bits
- Para la versión de 12-bits el comando es de 7-bits y la dirección de 5-bits.



b. Protocolo NEC. Utiliza una codificación por distancia de pulso. Cada pulso emitido con una frecuencia portadora a 38 kHz dura 0.56 ms, y para representar un "1" lógico se requiere de 2.25 ms, mientras que para un "0" lógico se requiere 1.125 ms. Se recomienda un ciclo de trabajo de 25 a 33% para la señal portadora.

FIGURA NO. 25. BITS PROTOCOLO NEC



(Fuente: Oisterwijk, 2011)

A continuación se presentan algunas de sus características:

- Utiliza modulación por distancia de pulso
- Frecuencia portadora de 38 kHz
- Bit time de 1.125 ms o 2.25 ms
- 8-bits de dirección y 8-bits de comando.
- El comando y la dirección se transmiten dos veces para verificación (una de ellas negada).
- Finaliza con una marca de 560 μ s.

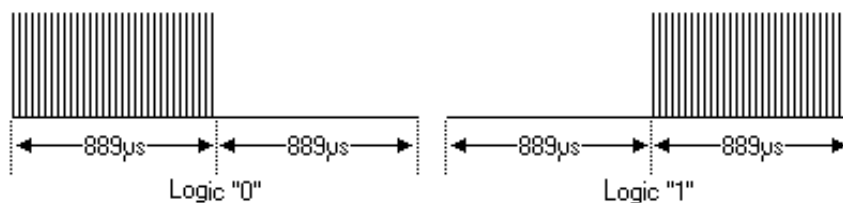
FIGURA NO. 26. EJEMPLO DE CÓDIGO PROTOCOLO NEC



(Fuente: Oisterwijk, 2011)

c. Protocolo RC5 de Philips. Utiliza una modulación bifásica, también conocida como codificación Manchester, y una frecuencia portadora de 36 kHz. Cada bit transmitido dura 1.778 ms. Se recomienda un ciclo de trabajo de 25 a 33% para la señal portadora para reducir el consumo de energía.

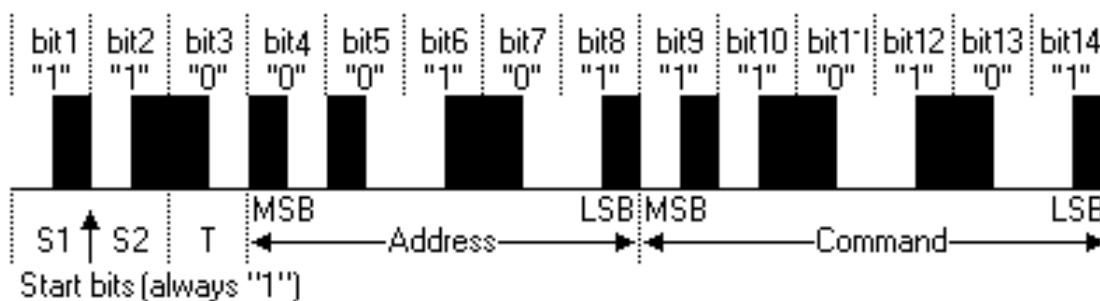
FIGURA NO. 27. BITS PROTOCOLO RC5 (OISTERWIJK, 2011)



A continuación se presentan algunas de sus características:

- Modulación bifásica (codificación Manchester)
- Frecuencia portadora de 36 kHz
- Bit time de 1.778 ms
- 5-bits de dirección y 6-bits de comando.

FIGURA NO. 28. EJEMPLO DE CÓDIGO PROTOCOLO RC5 (OISTERWIJK, 2011)



G. PERIFÉRICOS DE POTENCIA

1. Sistemas de riego automáticos. Los sistemas de riego para viviendas han tenido gran auge en los Estados Unidos, ya que el costo de mantenimiento se reduce en gran parte al implementar un sistema de este tipo. Existen controles de tiempo que permiten coordinar el tiempo y la hora de riego de un jardín, además existen ciertos actuadores extras como sensores de humedad y precipitación para brindar al controlador la única información que puede recibir

del ambiente exterior. Dentro de las marcas sobresalen: RainBird, Hunter, Toro, Orbit, Netafim, etc. Las mencionadas son las innovadoras en el uso adecuado del agua para los jardines. Las tres formas de riego más utilizadas en jardines son: (Rain Bird Inc.)

- **Aspersión:** Es la más utilizada ya que permite regar las áreas extensas como reducidas. Se caracteriza por ser un riego de distribución de agua superficial pero con su línea de distribución debajo de la tierra.
- **Micro aspersión:** Es la menos común, ya que se utiliza más en invernaderos y no tanto casa, permite distribuir el agua de una forma tipo brisa es adecuada para cuando se tiene plantas que no se deben de golpear con gotas de agua sólidas.
- **Goteo:** Es las más eficiente de las tres, ya que permite distribuir el agua únicamente en la raíz de la planta disminuye el uso del agua y es un riego que se coloca de forma superficial.

2. **Sistemas de riego para jardines.** Un sistema de riego es funcional solo si se logra hacer un buen diseño para abarcar únicamente las áreas deseadas de un jardín con la cantidad de agua adecuada. Es indispensable para eso sectorizar el riego por etapas para diferenciar entre área de grama, arbusto y jardineras. La parte primordial del riego es la colocación de los aspersores donde lo que se debe de conocer es el diseño HEAD TO HEAD, donde explica que los aspersores deben de ir colocados para que uno riegue al otro y así sucesivamente hasta terminar la línea principal. (Hunter Industries)

FIGURA NO. 29. DISEÑO CORRECTO HEAD TO HEAD

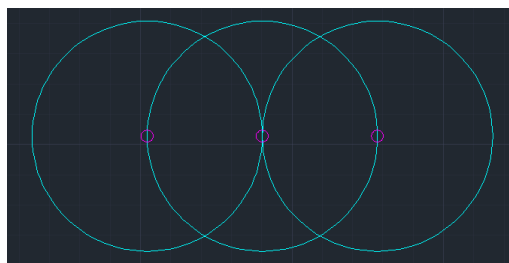
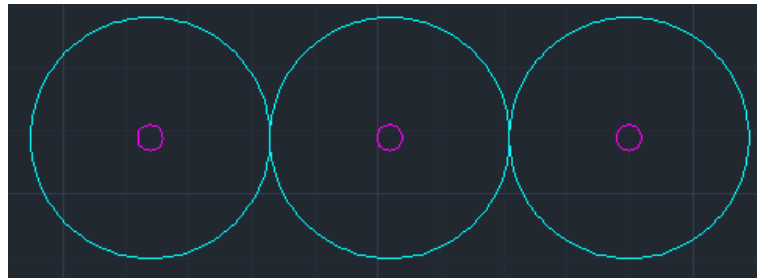
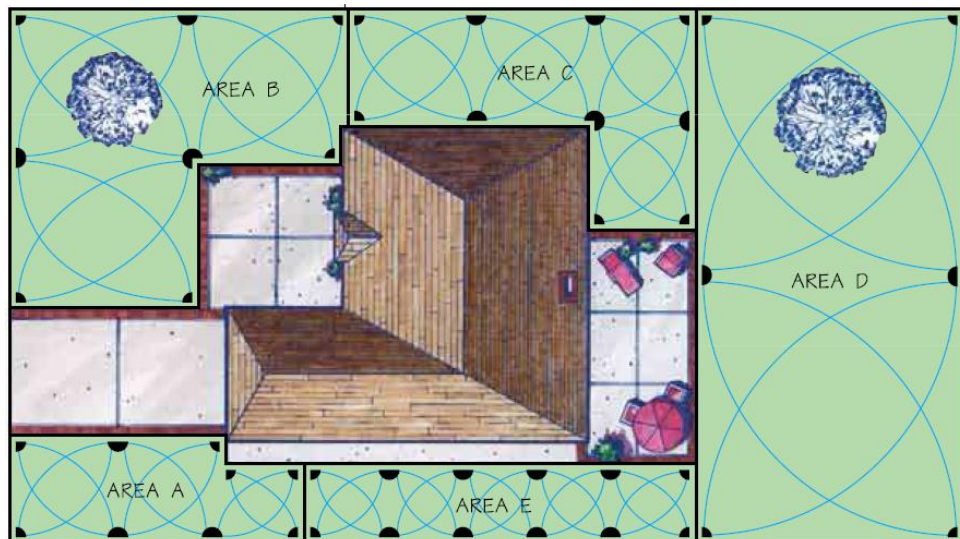


FIGURA NO. 30. DISEÑO INCORRECTO HEAD TO HEAD



Se debe conocer el suministro de donde se tomará el agua para poder sectorizar las áreas. Si tenemos tuberías reducidas vamos a necesitar sectorizar por más etapas las áreas para poder mantener la presión en las tuberías y así poder cumplir con la función de los aspersores. En la siguiente figura se observa una vivienda con un diseño preliminar.

FIGURA NO. 31. SECTORIZACIÓN DE LÍNEAS PARA RIEGO



II. ANTECEDENTES

La aparición de sistemas de automatización de variables físicas en ambientes públicos, como centros comerciales, hospitales e incluso oficinas, inició antes que en los entornos domésticos, esto se debe a que en dichos espacios públicos suelen utilizarse soluciones de tipo industrial. El desarrollo de la domótica como tal comienza en los años 70's, década en la cual fue creado el protocolo X10. Esto dio al público la capacidad de instalar sistemas que permitieran el control de dispositivos en cualquier hogar que poseyera una línea de alimentación de corriente alterna sin necesidad de instalar buses de comunicación específicos.

La popularidad y sencillez del protocolo X10 permitieron que a finales de los años 80's y principios de los 90's se desarrollaran muchos dispositivos de fácil acceso al público. Sin embargo, la comunicación basada en X10 contaba con muchas desventajas tales como problemas con el envío y recepción de datos debido al ruido en el medio de transmisión, la vulnerabilidad a posibles intrusos y el tiempo prolongado para realizar las distintas operaciones del sistema. Por estas razones, los protocolos sucesores optaron por utilizar buses propios diferentes a la línea de alimentación.

En Guatemala, el campo de comercialización de sistemas domóticos pertenece a empresas multinacionales que importan los sistemas desarrollados en otras partes del mundo. A pesar de esto, distintas universidades del país han publicado investigaciones relacionadas con el diseño de sistemas domóticos. Cabe mencionar que ninguno de los sistemas propuestos en estas investigaciones fue comercializado.

En el año 2004 se presentó en la Universidad del Valle de Guatemala un trabajo de graduación cuyo objetivo fue desarrollar un sistema encargado de controlar módulos X10 a través del correo electrónico. Este consistía en

controlar los distintos módulos desde cualquier lugar con conexión a Internet, sin necesidad de adquirir una IP privada. A pesar que el proyecto se llevó a cabo exitosamente, el sistema en sí poseía las desventajas del protocolo X10. (Balbás Contreras, 2004)

En la Universidad San Carlos de Guatemala se presentó en el 2005 un trabajo de investigación que proponía la implementación de un sistema domótico utilizando nuevamente las líneas de alimentación, en este se utilizó el estándar CEBus en vez del protocolo X10. La utilización de este estándar permitió obtener retroalimentación de los módulos conectados en la red, habilitando el flujo de información bidireccional. A pesar de utilizar un estándar distinto, las desventajas de seguridad y tiempo de respuesta fueron similares a las del protocolo antes mencionado. (González Galindo, 2005)

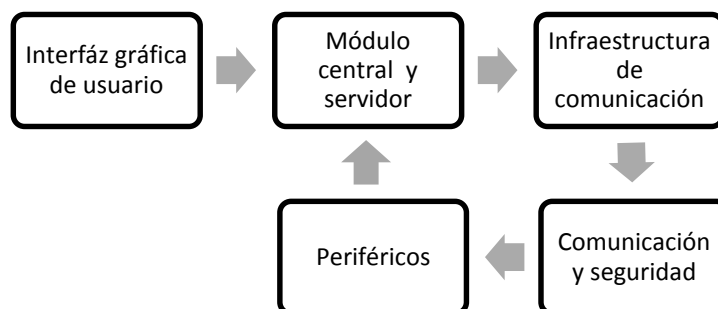
En el año 2012 se presentó en la Universidad del Valle de Guatemala un trabajo de graduación relacionado a la inmótica. El objetivo de este fue hacer uso del protocolo X10 para automatizar un aula dentro de la universidad. Se lograron implementar funciones capaces no solo del control de dispositivos (como iluminación, proyección de vídeo y sistema de persianas), sino también de ayuda para el aprendizaje de los alumnos. El proyecto contaba con un servidor web propio, habilitando una página web con una biblioteca virtual en la cual los estudiantes podían acceder a material de estudio y grabaciones de clases previas. (Sánchez Estrada, Mazariegos Soto, & Jordán García, 2012)

Los proyectos investigados mostraron la ventaja de aportar soluciones económicas y de fácil instalación frente a las alternativas existentes en el mercado. Al utilizar la red de alimentación eléctrica AC como medio de transferencia de datos, se aprovechan recursos físicos ya existentes en el hogar, permitiendo que la implementación de los sistemas se lleve a cabo de manera sencilla, sin embargo, ninguno de los sistemas llegó al punto de un producto final comercializable.

III. METODOLOGÍA

El megaproyecto RDI se trabajó de manera seccionada, definiendo 6 módulos que al integrarse cumplen con los requerimientos de diseño planteados en los objetivos de este trabajo. Los distintos módulos interactúan de manera específica entre sí, dependiendo del flujo de información entre estos para cumplir con su función individual. En la Figura 32 se observa la interconexión de los módulos según su dependencia respectiva.

FIGURA NO. 32. DIAGRAMA DE BLOQUES MODULARES DEL MEGAPROYECTO RDI



Los módulos específicos se desarrollaron utilizando su propia metodología, esta se detalla de manera individual en los capítulos correspondientes a cada uno.

IV. INTERFÁZ GRÁFICA DE USUARIO

A. SELECCIÓN DE DISPOSITIVO PARA GUI

1. Diseño experimental. En la actualidad el Smartphone es el dispositivo que las personas utilizan durante la mayor parte del día, teniéndolo consigo más que tiempo que cualquier otro tipo de dispositivo electrónico. Este cuenta con las capacidades de conectividad y procesamiento suficiente como para reemplazar a la mayoría de controles que se encuentran normalmente en un hogar. Es por esto que se seleccionó como el dispositivo primario en el que se estará implementando la interfaz gráfica de control del sistema domótico RDI.

2. Discusión. RDI busca que el usuario tenga acceso a una amplia variedad de dispositivos que se encuentran normalmente en una vivienda, haciendo esto de forma rápida, efectiva y agradable desde cualquier lugar dentro del hogar. Todo esto encapsulado en un dispositivo móvil que ya se encuentre en el mercado.

Hoy en día existen dos dispositivos que cumplen con estos requisitos: el *smartphone* y la tablet, pero no hace falta hacer mucho análisis para inclinarse por el primero, ya que es el dispositivo que la mayoría de usuarios ya tienen a su disposición y el que regularmente cargamos siempre en el bolsillo. Haciéndolo una plataforma ideal para el desarrollo de una aplicación de control para el sistema domótico RDI. Es por esto que se eligió para esta primera fase la implementación única del interfaz gráfico de usuario en *smartphones*.

Específicamente, se estará desarrollando para los teléfonos inteligentes de *Apple* de 6^{ta} y 7^{ta} generación. Debido a que se busca tener el radio mayor posible de comunicación inalámbrica con el servidor, esto gracias al sistema dual de antenas que está integrado en estas dos últimas líneas de *smartphones*. Contando también con la ventaja de que estos tienen características idénticas de pantalla y resolución, un atributo que facilita considerablemente el trabajo del desarrollador, teniendo este únicamente que enfocarse en la diferencia de procesamiento de 32 y 64 bits con la que estos dispositivos cuentan.

B. SELECCIÓN DE PLATAFORMA PARA DESARROLLO

1. Diseño experimental. Sabiendo para qué tipo de dispositivo se estaría desarrollando, se definió que la interfaz de control únicamente estaría disponible para la plataforma *iOS* de *Apple*. Las herramientas que este sistema operativo utiliza para su desarrollo son accesibles para cualquier usuario de forma gratuita desde el *App Store*, siempre y cuando se tenga un computador Mac. Aunque cabe mencionar que para realizar pruebas en dispositivos reales y poder publicar aplicaciones para que sean colocadas para descarga masiva, es necesario estar registrado en el programa de desarrolladores para *iOS* de *Apple*, el cual tiene un costo (al momento de realizar este trabajo de investigación), de \$99 por una suscripción anual.

2. Discusión. En el mercado actual, hay dos nombres que dominan el mundo de los dispositivos móviles: *Android* y *Apple*. Estas dos compañías se basan en conceptos muy diferentes para el desarrollo de aplicaciones, conceptos que se refleja notablemente en las herramientas nativas que ofrecen para el desarrollo de aplicaciones. Lo cual genera una pregunta muy importante para cualquier persona que desea desarrollar aplicaciones, ¿*Apple* o *Android*?

En este proyecto, se decidió utilizar el sistema operativo *iOS* originario de *Apple*, por lo que ahora se darán algunas razones por las cuales no se optó por un sistema operativo más popular, con menos restricciones y menor costo, como lo es *Android*.

Fragmentación Hardware: esto se refiere a la gran diversidad de dispositivos que corren hoy en día bajo el sistema operativo *Android*. Un aspecto que probablemente beneficia al consumidor que tiene una mayor variedad de dispositivos a su disposición para elegir en base a sus necesidades, pero es un verdadero dolor de cabeza para el desarrollador. Esto debido a que este debe adaptarse a cada uno de los requisitos específicos de cada teléfono, tamaño de pantalla, resolución de pantalla, conectividad, librerías gráficas, etc. Una tarea para nada sencillo tomando en cuenta las herramientas que ofrece *Android*.

Fragmentación Software: De igual manera que la fragmentación hardware, el sistema operativo *Android* tiene una amplia variedad de versiones activas en el mercado. Entre las cuales se encuentran las más populares: *Jelly Bean*, *Ice Cream Sandwich* y *Gingerbread*, todos con una distribución muy similar en el mercado. Dificultando considerablemente el trabajo de un desarrollador, dado que es necesario considerar las capacidades y restricciones con las que cuenta cada uno de estos elementos *software*.

Calidad de producto: Uno de los objetivos del proyecto es desarrollar un sistema domótico accesible para un sector de mercado significativo, manteniendo la calidad del mismo. Es por esto que es necesario dar al usuario un ambiente de control que refleje calidad y eficiencia, lo cual es evaluado por *Apple* antes de publicar cualquier aplicación en el *App Store*.

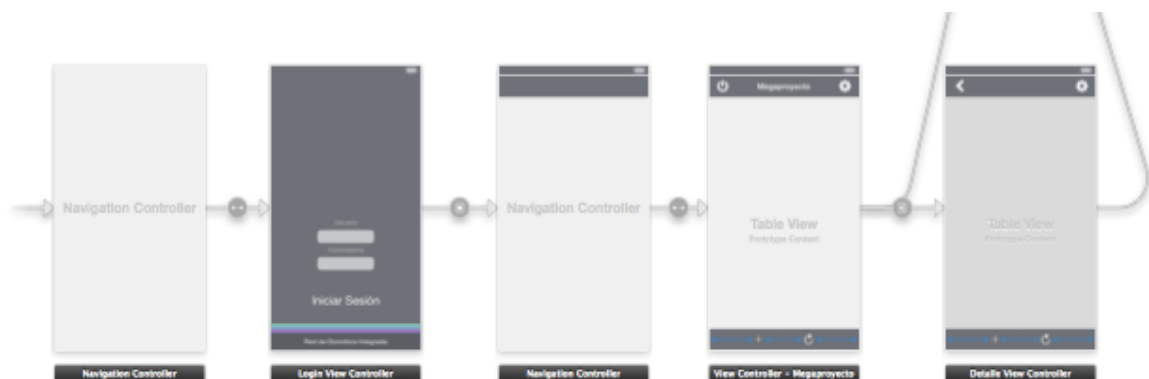
Virus: Un problema que afecta casi exclusivamente a los dispositivos *Android*, es la inmensa cantidad de virus que hay en el sistema operativo desarrollado por los de *Mountain View*. Este es el único sistema que tiene malware que además de corromper aplicaciones, es capaz de acabar completamente con el teléfono inteligente.

C. ESTRUCTURA Y DISEÑO DE APLICACIÓN

1. Diseño experimental. Existen muchas formas de construir una aplicación, tanto estructural como gráficamente. Campos que se definen de dos formas muy diferentes en el *IDE* nativo de *Apple*, *xCode*.

Estructuralmente, se define la lógica interna entre controlador-modelo-vista mediante la arquitectura *MVC*, la cual tiene como objetivo independizar al modelo de la *GUI*, agregándole mucha más flexibilidad a la aplicación a la hora de desarrollar para diferentes dispositivos. Particularmente en este proyecto se utilizó la arquitectura *MVC* de forma serial.

FIGURA NO. 33. ARQUITECTURA MVC CON DESPLAZAMIENTO SERIAL



Gráficamente se buscó tener una interfaz con elementos simplistas que promovieran la intuitividad del usuario, aunque también fue importante que se diera la información suficiente para que el usuario se sintiera cómodo con la interfaz y al mismo tiempo en control total de todos sus dispositivos conectados al sistema. Una estrategia que va muy bien con el diseño mismo del reciente sistema operativo *iOS 7* y cuya implementación se facilita dadas las recientes actualizaciones realizadas a las librerías de objetos en *xCode 5*.

Independientemente del diseño y estructura, esta aplicación es fundamentalmente un intermediario entre el usuario y el módulo de control, debido a que esta no es propietaria de la información que despliega, y por lo tanto, depende totalmente de la relación que exista vía Wi-Fi entre ella y un servidor que sea capaz de otorgarle la información necesaria para la completa implementación de la *GUI*.

2. Discusión. La estructura de una aplicación móvil normalmente depende de muchos factores. Algunos de estos son la plataforma o *IDE* en la que se estará desarrollando, si la aplicación va a ser dueña de su propia información o la va a obtener de un servidor, la función que esta va a cumplir, el público a quien está dirigida, si va a utilizar librerías externas o no, etc. Todos estos factores influyen directamente en la estructura que el diseñador debe crear para una aplicación, asumiendo claramente que este implementando de forma correcta la arquitectura *MVC*.

La aplicación de control para el sistema RDI no fue la excepción a esta regla, por lo que se establecen los factores principales que definen a la aplicación como tal:

- Se utilizó el *IDE* nativo de *Apple*, *xCode*.
- El modelo de la aplicación es brindada por un servidor local.
- La aplicación cumple la función de interfaz gráfica de control para usuario.
- Está dirigida a usuarios que tengan la RDI instalada en su hogar.

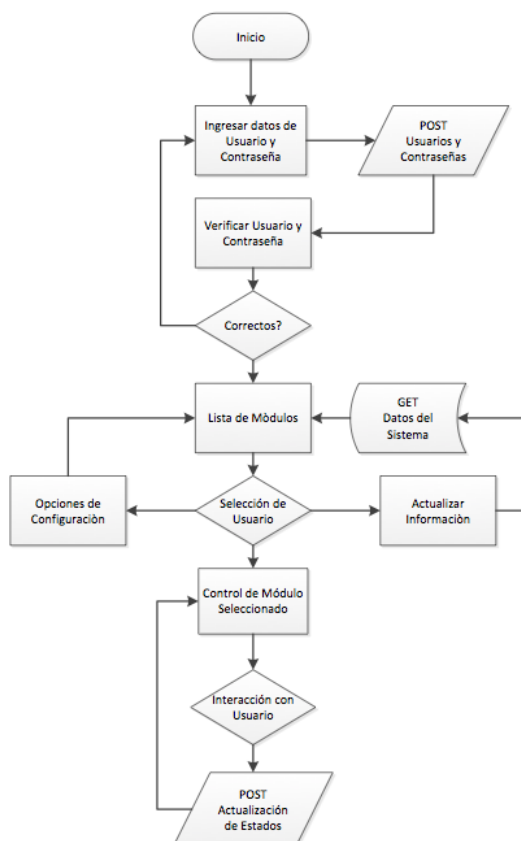
Habiendo definido los factores fundamentales de la aplicación, se continuó a establecer requisitos con los que debe cumplir el interfaz gráfica, con el objetivo de brindar al usuario con una experiencia amigable e intuitiva de uso.

- Medidas de seguridad para el usuario.
- Interfaz de control propio para cada tipo de módulo.
- Flexibilidad en configuración de redes.
- Adaptabilidad para cualquier tipo de configuración de sistema.

Con todas las prioridades establecidas, es posible iniciar el proceso de diseño del interfaz gráfico. Para lo cual, se eligió una estructura serial de movimiento, es decir una secuencia de vistas que únicamente pueden acceder a su antecesor o invocar a una nueva vista a través de un *segue* o transición de dirección única. Haciendo que la experiencia del usuario sea fluida, dejando ningún lugar a que este se pierda dentro del interfaz.

A continuación se presenta el esquema general de la interfaz gráfica de usuario, en donde es posible observar el flujo serial que se planteó:

FIGURA NO. 34. ESQUEMA GENERAL DE LA INTERFAZ GRÁFICA DE USUARIO



Este diseño se implementó con el constructor de interfaces y librerías gráficas que provee *xCode*, como se observa en la Figura 33, por medio de vistas individuales con controladores independientes entre sí, con el objetivo de que la aplicación sea lo más flexible posible y adaptable a cualquier tipo de configuración de la red.

D. MÉTODO DE COMUNICACIÓN

1. Diseño experimental. Dada la naturaleza de la aplicación, la comunicación dispositivo-servidor es esencial para el funcionamiento correcto del sistema, pero dado que el servidor es un recurso que no está exclusivamente dedicado a dicha comunicación, se implementó comunicación indirecta. Esto significa que el dispositivo envía información de eventos encapsulados en un formato que se caracterice por ser rápido y ligero en cuanto a transmisión de datos. El servidor recibe esta información y actualiza una base de datos, específicamente las variables del dispositivo que activó el evento. Para que luego, cuando el ordenador este “desocupado”, pueda mandar dichas actualizaciones a través de la línea de alimentación hasta el módulo indicado, para que este establezca los nuevos estados.

Específicamente, para implementar esta comunicación se utilizó el protocolo HTTP y sus métodos *GET* y *POST*. Siendo el primero utilizado para solicitar al servidor toda la información de la base de datos, con el propósito de actualizar la *GUI* con eventos que pudieron haber sucedido externos al sistema, y deban ser reflejados al usuario, mientras que el segundo tiene la función de transmitir los eventos solicitados por el usuario hacia el servidor de forma eficiente y concisa.

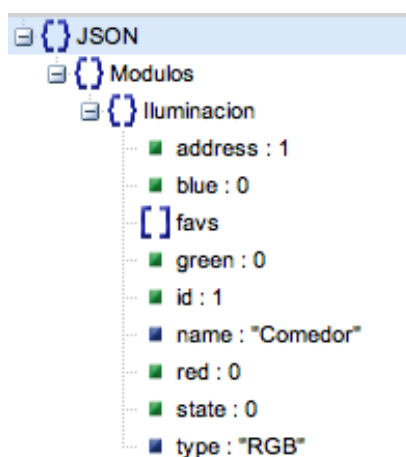
2. Discusión. La comunicación entre el dispositivo móvil y módulo central, debía tener dos características fundamentales: Lapsos de comunicación cortos (menores a 100 ms) y alcance considerable de comunicación (mayor a 20 metros). Estos parámetros asegurarían que el usuario tuviera una experiencia agradable de uso de la Red de Domótica Integrada (RDI).

Para el desarrollo e implementación de la aplicación se utilizaron *smartphones Apple* de 5^{ta} y 6^{ta} generación, los cuales tienen el hardware necesario para comunicarse con redes celulares, redes locales y/o *Bluetooth*. Por lo que fue necesario, seleccionar la opción más adecuada para la comunicación con el servidor local. Inicialmente, podemos descartar la conexión celular, ya que nuestro servidor no es accesible desde el internet al no estar publicado. Utilizar el estándar *Bluetooth* podría ser una opción viable por su alcance y velocidad de transmisión, pero la computadora que se utilizó no tenía el hardware necesario para implementar este protocolo, lo cual representaría un costo adicional en cada sistema. Por lo que se eligió la comunicación local, ya que un *Router Wi-Fi* está presente hoy en día en la mayoría de hogares, y cumple con las características de velocidad de transmisión y alcance.

Conociendo el protocolo que utilizaremos para la transmisión de datos, es necesario ahora definir el formato en que este será enviado. Dada la naturaleza del sistema, no es necesario el manejo de archivos grandes, únicamente texto que encapsule las variables y direcciones que el sistema debe utilizar para su uso correcto, por lo que se busca un formato ligero y eficiente. Las dos opciones más utilizadas son *JSON* y *XML*, dos formatos que tienen tanto sus diferencias como similitudes. *XML* se enfoca más en ser compatible con cualquier tipo de formato de archivo, mientras que *JSON* únicamente utiliza texto y otros tipos de datos tradicionales, haciendo a este último mucho más sencillo de compilar y ser transmitido. Adicionalmente, *JSON* está orientado a ser compatible con varios lenguajes de programación, entre los cuales está *Python*, lenguaje utilizado para la programación del servidor local. Estos fueron argumentos suficientes para inclinarnos más por el formato de transmisión orientado a objetos de *JavaScript*.

Como se mencionó con anterioridad, el formato *JSON* únicamente soporta datos tradicionales, entre los cuales podemos encontrar listas y diccionarios, objetos que al ser combinados entre sí, tienen la capacidad de transportar grandes cantidades de información de forma ordenada y estructurada. Por lo tanto, se utilizó esta herramienta para encapsular todas las variables del sistema en un mismo objeto, que tuviera un peso considerablemente pequeño para ser transmitido con mayor eficiencia. A continuación se presenta la estructura con que se almacenó la información del sistema:

FIGURA NO. 35. ESTRUCTURA DE ALMACENAMIENTO DE DATOS PARA TRANSMISIÓN



Como se observa en la Figura 35, existen 4 niveles de abstracción implementadas en el argumento que se transmite desde la aplicación al servidor:

Para la comunicación utilizando *JSON* se definió la siguiente estructura:

- Encabezado: Únicamente funciona como una capsula que contiene todos los datos estructuradamente.

{JSON : Diccionario de módulos}

- Diccionario de tipos de módulos: Contiene un diccionario que define todos los tipos de módulos como llaves, y sus respectivos valores son listas de todos los dispositivos en el sistema.

{Módulos : {Iluminación : Lista de dispositivos de Iluminación}}

- Lista de diccionarios de variables para cada dispositivo: Es una lista donde cada objeto contenido es un diccionario de variables que definen los estados de cada variable de un módulo específico del sistema.

```
[[{address : 1, name : "Comedor", id : 1, red : 34, green : 255, blue: 190, state : 0, type : "RGB"}, {address : 2, name : "Cocina", id : 2, red : 50, green : 100, blue: 0, state : 1, type : "LED"}, {address : 3, ...}]]
```

Trabajar con una estructura como la anteriormente presentada hace que la búsqueda, despliegue, actualización y envío de dicho *JSON* sea considerablemente sencilla tanto para la aplicación como para el servidor. Aunque cabe mencionar que esta estructura es utilizada únicamente al realizar una petición de tipo *GET* al servidor. Cuando el usuario evoca un evento en la *GUI*, y la aplicación debe actualizar esta información en el servidor, se implementa una petición de tipo *POST*, el cual únicamente envía la información específica del módulo en el que se realizó el cambio de estado, como se muestra a continuación:

```
{address : 1, name : "Comedor", id : 1, red : 255, green : 255, blue: 255, state : 1, type : "RGB"}
```

Utilizar una estructura de datos tan flexible, hace que la misma aplicación sea altamente adaptable a cambios en el sistema y diversas configuraciones que muy probablemente vaya a tener este en diferentes escenarios.

El método utilizado para la comunicación servidor-aplicación fue indirecta. Es decir, al activarse un comando o cambio de estado por parte del usuario, el dispositivo móvil conectado a la red local, envía por medio del protocolo HTTP un requisito de actualización que el servidor únicamente acepta y actualiza en la base de datos del sistema. Un programa secundario en el módulo central es alertado por el servidor, y este transmite al módulo periférico específico los nuevos datos.

E. RESULTADOS GENERALES

FIGURA NO. 36. PANTALLA DE INICIO DE SESIÓN



FIGURA NO. 37. PANTALLA DE SELECCIÓN DE MÓDULOS



FIGURA NO. 38. CONTROL DE MÓDULO DE ILUMINACIÓN RGB



FIGURA NO. 39. CONTROL DE MÓDULO DE ILUMINACIÓN INCANDESCENTE



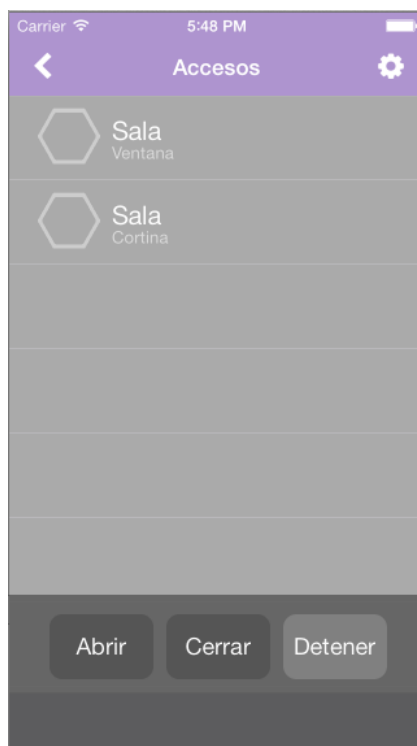
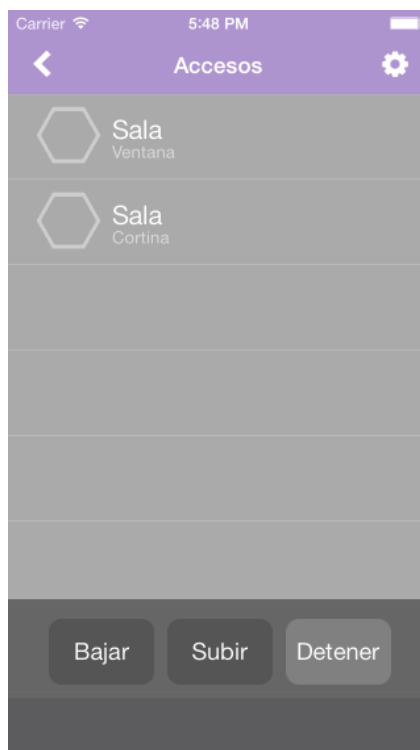
FIGURA NO. 40. CONTROL DE MÓDULO DE ACCESOS: VENTANA**FIGURA NO. 41. CONTROL DE MÓDULOS DE ACCESO: CORTINA**

FIGURA NO. 42. CONTROL DE MÓDULO PARA TELEVISORES



FIGURA NO. 43. CONTROL DE MÓDULO PARA RIEGO

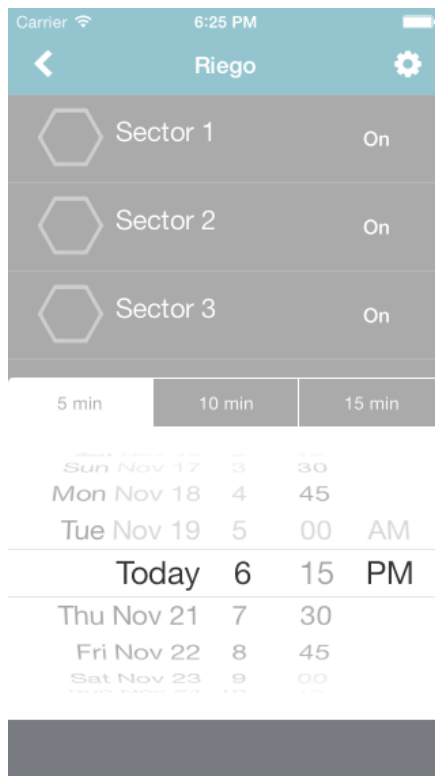


TABLA NO. 5. LISTADO DE VARIABLES DE CONTROL DE INTERFAZ GRÁFICA.

Interfaz Gráfico de Usuario				
Módulo	Tipo de actuador	Variables de control	Tipo de variable	Rango
Iluminación	Ahorradoras	Estado	Booleano	0/1
	Incandescentes	Estado Intensidad	Booleano Entero	0/1 0-8
	LED RGB	Estado Rojo Azul Verde	Booleano Entero Entero Entero	0/1 0-8 0-8 0-8
Accesos	Ventanas	Estado Detener	Booleano Booleano	Abrir/Cerrar 0/1
	Cortinas	Estado Detener	Booleano Booleano	Subir/Bajar 0/1
Riego	Válvula	Estado Temporizador Fecha	Booleano Entero NSDate	0/1 300/600/900s Fecha > Actual
TV	Emisor infrarrojo	Estado Volumen Canal Mute Input Memory Jump	Booleano Triestado Triestado Booleano Booleano Booleano Booleano	0/1 -1/+ -1/+ 0/1 0/1 0/1 0/1

V. MÓDULO CENTRAL Y SERVIDOR

A. CONFIGURACIÓN *RASPBERRY PI*

1. Diseño experimental. Para la realización del módulo central y servidor se utilizó el computador en placa *Raspberry Pi*, Modelo B. La *Raspberry Pi* cuenta con pines configurables para comunicarse a través del protocolo *UART* con la red domótica. También posee un puerto Ethernet para conectarse al enrutador del hogar y así poder funcionar como servidor local.

2. Discusión. Se eligió utilizar como módulo central y servidor una *Raspberry Pi* (Modelo B), debido a su alta capacidad de procesamiento en comparación de otros dispositivos de tamaño similar. También se aprovechó el hecho que el modelo adquirido contaba con una conexión a red vía *Ethernet*, y pines de entrada y salida para realizar la comunicación serial.

Dado que la *Raspberry Pi* no posee un disco duro, lo primero por hacer fue adquirir una tarjeta SD de 4GB o más para montar el sistema operativo a utilizar. La opción más recomendable es *Raspbian*, debido a su alta compatibilidad no solo por el procesador, sino también por la cantidad de paquetes que el sistema tiene preinstalados con funciones específicas para la *Raspberry Pi*. Una vez grabada la imagen del sistema operativo en la tarjeta SD y después de configurar unos aspectos básicos, el dispositivo está listo para utilizarse. Por último se recomienda realizar una actualización de los paquetes instalados ejecutando desde consola el comando:

```
>_ sudo apt-get update
```

B. CONFIGURACIÓN Y PROGRAMACIÓN SERVIDOR

1. Diseño experimental. Se grabó una imagen de disco del sistema operativo Raspbian, de Linux, en una tarjeta SD para utilizar la Raspberry Pi. La programación fue realizada utilizando el lenguaje Python. Tanto la lógica de programación del sistema domótico como la función del servidor. Para la función de servidor se utilizó el marco de trabajo Django. Este marco de trabajo fue diseñado para Python y utiliza un servidor Apache.

2. Discusión. Se buscó que todos los dispositivos móviles conectados al sistema tuvieran la información del mismo actualizada a todo momento. Descartando la idea que cada uno de estos dispositivos se viera obligado a almacenar cantidades grandes de datos, se recurrió a la arquitectura cliente-servidor. No fue necesario el envío de archivos o imágenes sino únicamente datos en formato *JSON*. Por esta razón contar con un servidor HTTP es suficiente.

Para el envío de solicitudes HTTP desde los dispositivos móviles es necesario contar con una IP estática. En una red mayor, este servicio tiene un costo, por lo que se decidió trabajar en subredes. Así entonces se asignó a la *Raspberry Pi* una dirección estática dentro de la red local sin necesidad de elevar el costo de operación del sistema.

Para hacer el cambio de IP dinámica a IP estática se localizó el archivo "interfaces" guardado en el directorio "/etc/network" de nuestra *Raspberry Pi*. Al editarlo notamos que nuestra red predefinida (llamada "eth0") utiliza el protocolo DHCP. Se realizó el cambio entre el protocolo por la configuración de dirección IP, máscara de subred, dirección de subred, dirección de *broadcast* y puerta de enlace que deseada.

Para verificar que toda configuración (software como hardware) esté funcionando correctamente, se ejecutó una versión de un servidor utilizando *Python*. Tanto *Python* como su librería *Simple HTTP Server* se encuentran instalados en el sistema *Raspbian*, por lo que solo es necesario ejecutar el siguiente comando dentro del directorio que se desee montar en el servidor:

```
>_ python -m SimpleHTTPServer
```

a. Django. Una vez descargado e instalado el marco de trabajo en la *Raspberry Pi*, se ejecuta dentro de un directorio específico el comando:

```
>_ django-admin.py startproject NombreProyecto
```

Con esto, se crean los directorios y archivos necesarios para configurar y ejecutar el servidor. Para la configuración es necesario editar el archivo “*settings.py*” localizado en el directorio principal nombrado como nuestro proyecto. Dentro de este archivo se configuran opciones como la zona horaria, lenguaje código de nuestro servidor, configuración de administradores y uso de parámetros. En este archivo se debe especificar la dirección y el motor de la base de datos que se utilizará. Se recomienda habilitar la aplicación de Administrador para gestionar la base de datos desde otro ordenador vía web.

Seguido a esto se prosiguió a editar el archivo “*urls.py*”. Aquí se deben de colocar todos los archivos URL de cada uno de las aplicaciones creadas dentro del proyecto. Nuevamente, para poder contar con la aplicación de Administrador, se deben habilitar las URLs que pertenecen a esta.

Para que el ORM funcione correctamente, cada vez que se realiza un cambio en la base de datos, esta debe sincronizarse utilizando el comando:

```
>_ python manage.py syncdb
```

Por último desde el directorio principal del proyecto, se ejecuta el comando:

```
>_ python manage.py runserver DirecciónIP
```

b. Aplicaciones web. Se conoce como una aplicación web a todas las funciones que realiza un servidor web que están a disposición de los clientes conectados a la misma. Con el objetivo de tener mayor facilidad con el manejo de clases y objetos de diferentes programas, todas las aplicaciones deben de estar almacenadas en directorios de paquetes Python. Para esto, se elige el directorio que guardará la aplicación y dentro de este creamos un documento de nombre “__init__.py”. Con esto, el motor de búsqueda de *Python* agrega el directorio a su lista de importación de módulos. Para crear una nueva aplicación, dentro de la carpeta ejecutamos el comando:

```
> _ django-admin.py startapp NombreAplicación
```

Después de esto, dentro de la carpeta se almacenan todos los archivos necesarios para configurar y programar las funciones de la aplicación. A este grupo de archivos se agrega uno de nombre “admin.py” para habilitar la administración de la aplicación.

Dentro del proyecto fueron creadas dos aplicaciones: la primera encargada del manejo de métodos *GET* y la segunda de métodos *POST*. Cada una tiene su propia configuración y URLs. Esto se hizo por razones únicamente de organización dentro del proyecto, ya que sería posible diseñar únicamente una aplicación capaz de manejar todos los métodos realizados por los clientes.

Debido a la sencillez y facilidad de uso de y estructura, se decidió utilizar *JSON* como el formato de intercambio de datos entre el servidor y los clientes. Se utilizaron únicamente métodos *PULL*. Es decir que cada vez que un cliente desee recibir una actualización de la información en la base de datos, este debía realizar una solicitud al servidor y esperar una respuesta del mismo. No se utilizó la tecnología *PUSH* debido a que no todos los dispositivos están configurados para aceptarla. También se tomó en cuenta que las solicitudes pueden tardar más tiempo en llegar a su destino.

C. BASE DE DATOS

1. Diseño experimental. La base de datos fue almacenada utilizando la librería *SQLite*. Esta base de datos es modificada tanto por la red domótica como también por el marco de trabajo del servidor. Django utiliza un mapeador para transformar objetos entre clases *Python* y la base de datos relacional utilizada por *SQLite*.

2. Discusión. Se utilizó una base de datos con un modelo relacional de lenguaje *SQL*. Debido a su ligereza y a la compatibilidad con *Python* se utilizó la librería *SQLite3* para la gestión de la base de datos.

D. COMUNICACIÓN SERIAL

1. Diseño experimental. Una vez que se logró ejecutar el servidor correctamente y que cada uno de los clientes fue capaz de acceder a la base de datos y modificar la información que en ella se encuentra, se prosiguió a diseñar el programa que gestionaría la comunicación con todos los módulos conectados al sistema domótico. Desde el punto de vista del módulo central el envío y recepción de datos se realiza de manera serial utilizando el protocolo *UART* a una velocidad de 1200 baudios por segundo. En realidad *la Raspberry Pi* se comunica vía *UART* con un módulo externo y este es el encargado de transmitirlo a los demás dispositivos utilizando otro protocolo de comunicación. Este otro protocolo utiliza la modulación *ASK* para transmitir datos a través de la línea de alimentación. La velocidad de transmisión (1200 baud/seg) está limitada por el tiempo de respuesta del modulador de señales.

2. **Discusión.** Todos los dispositivos se encuentran físicamente conectados a la única línea de transmisión del sistema, por lo que es necesario que cada uno tenga una dirección y así saber que datos están dirigidos hacia él y que datos no. El protocolo implementado utiliza 7 bits para el direccionamiento, dando la capacidad de conectar 128 dispositivos a la red. Inicialmente todos los dispositivos tienen una dirección inicial predefinida (126). El módulo central se encarga de revisar periódicamente esta dirección en busca de nuevos dispositivos conectados a la red. Una vez encontrado uno, se le asigna una dirección diferente y se guarda en la base de datos la información del nuevo dispositivo.

Es una arquitectura centralizada donde ningún otro dispositivo, aparte del módulo central, puede iniciar la comunicación. El módulo central envía datos a los dispositivos e inmediatamente después espera una respuesta de los mismos.

La transmisión de datos se realiza a través de paquetes. El paquete emitido por el módulo central consta de 3 *bytes*: El primer byte de encabezado (255) da a conocer el inicio de un nuevo paquete. El segundo byte de dirección indica a que dispositivo se le está transmitiendo información. El último byte de datos indica la información que se desea transmitir. El paquete recibido como respuesta consta de 2 *bytes* únicamente, debido a que la respuesta siempre está dirigida al módulo central por lo cual no se necesita especificar una dirección. El primer byte sigue siendo el encabezado (255) y el segundo byte es el de datos.

E. MULTI-TAREAS

1. Diseño experimental. Se programaron archivos Bash en la consola del sistema operativo para poder ejecutar ambas aplicaciones en paralelo. También para poder configurar que las aplicaciones iniciaran cada vez que se iniciara el dispositivo.

2. Discusión. Teniendo el programa de servidor y el programa de módulo central terminados, el último paso fue programarlos para que ambos se ejecutaran en paralelo. Aunque la *Raspberry Pi* es dispositivo con un solo CPU, el sistema operativo *Raspbian* es capaz de emular el paralelismo. Para esto se creó un programa de Shell que ejecutara ambas aplicaciones. El programa ejecuta primero la aplicación de módulo central, deja esta corriendo en el fondo y luego ejecuta la aplicación de servidor.

F. RESULTADOS GENERALES

FIGURA NO. 44. PANTALLA INICIAL SISTEMA OPERATIVO RASPBIAN

```
The programs included with the Debian GNU/Linux system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent  
permitted by applicable law.  
Last login: Sun Aug 25 16:07:27 2013 from 192.168.1.106  
pi@raspberrypi ~ $ █
```

FIGURA NO. 45. ESCRITORIO LXDE RASPBIAN

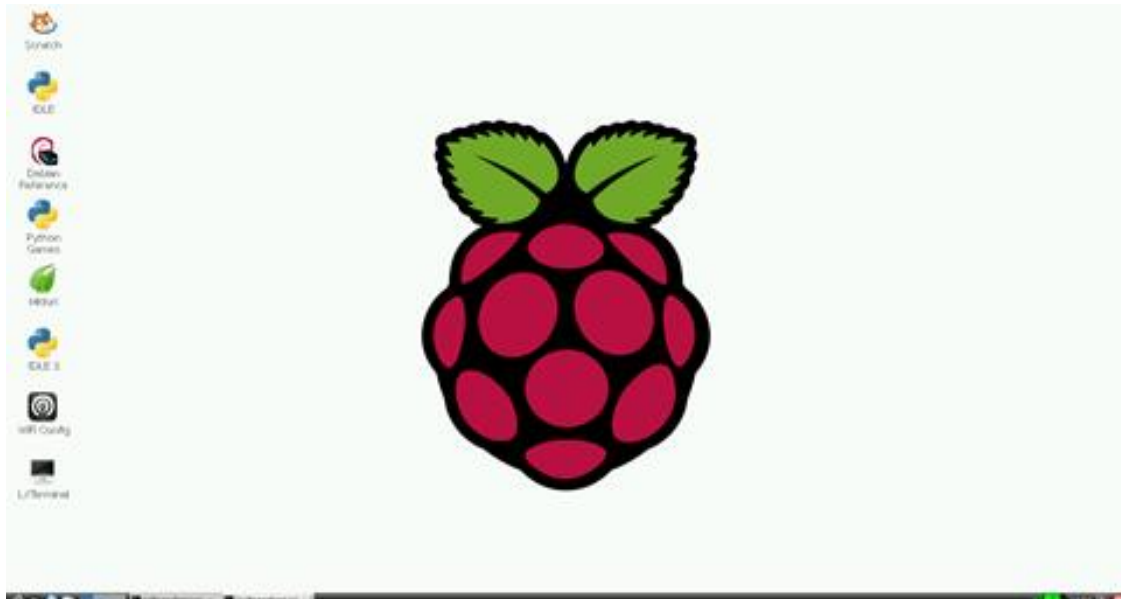


FIGURA NO. 46. ARCHIVO NETWORK MODIFICADO PARA IP ESTÁTICA

```

iface lo inet loopback
iface eth0 inet static
    address 192.168.1.110
    netmask 255.255.255.0
    network 192.168.1.0
    broadcast 192.168.1.255
    gateway 192.268.1.1

```

FIGURA NO. 47. SERVIDOR SIMPLE DE PYTHON EJECUTÁNDOSE (LADO SERVIDOR)

```

pi@raspberrypi ~ $ python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
192.168.1.100 - - [26/Aug/2013 06:49:41] "GET / HTTP/1.1" 200 -
192.168.1.100 - - [26/Aug/2013 06:49:41] code 404, message File not found
192.168.1.100 - - [26/Aug/2013 06:49:41] "GET /admin HTTP/1.1" 404 -
192.168.1.100 - - [26/Aug/2013 06:49:41] code 404, message File not found
192.168.1.100 - - [26/Aug/2013 06:49:41] "GET /favicon.ico HTTP/1.1" 404 -
192.168.1.100 - - [26/Aug/2013 06:49:41] code 404, message File not found
192.168.1.100 - - [26/Aug/2013 06:49:41] "GET /favicon.ico HTTP/1.1" 404 -

```

FIGURA NO. 48. SERVIDOR SIMPLE DE PYTHON EJECUTÁNDOSE (LADO CLIENTE)

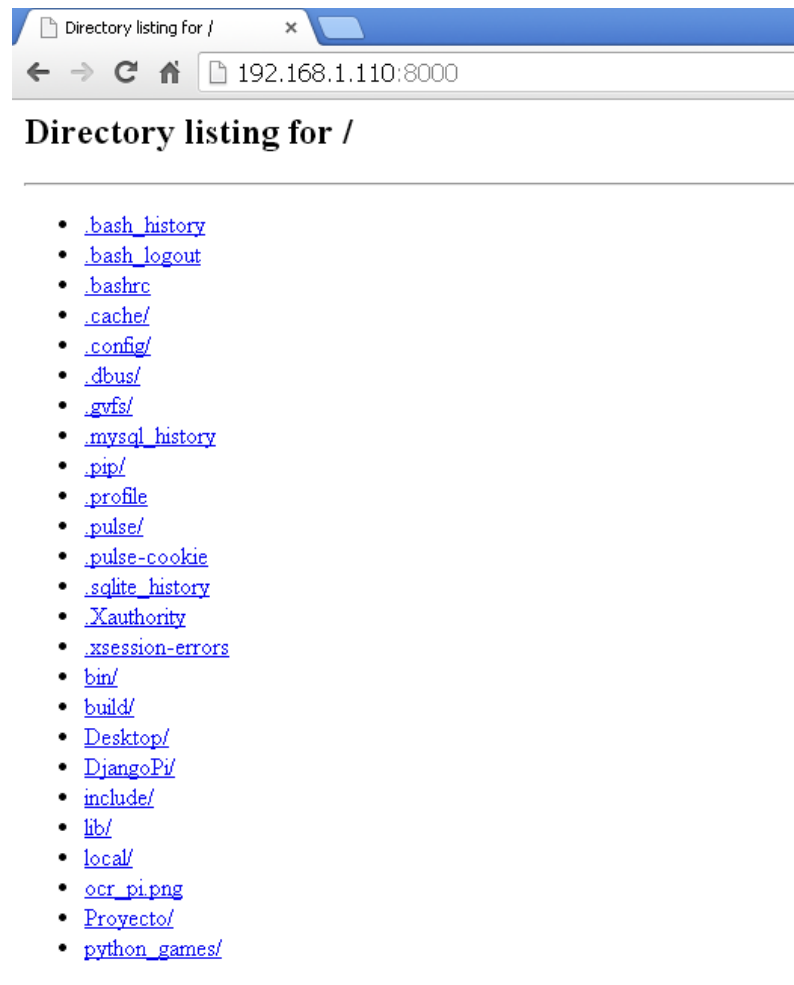


FIGURA NO. 49. CONFIGURACIÓN ARCHIVO SETTINGS.PY (BASE DE DATOS)

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3', # Add 'postgresql_psycopg2', 'mysql', 'sqlite3' or 'oracle'.
        'NAME': 'DatosProyecto.db',          # Or path to database file if using sqlite3.
        'USER': '',                          # Not used with sqlite3.
        'PASSWORD': '',                      # Not used with sqlite3.
        'HOST': '',                          # Set to empty string for localhost. Not used with sqlite3.
        'PORT': '',                          # Set to empty string for default. Not used with sqlite3.
    }
}
```

FIGURA NO. 50. CONFIGURACIÓN ARCHIVO SETTINGS.PY (APLICACIONES)

```

INSTALLED_APPS = (
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.sites',
    'django.contrib.messages',
    'django.contrib.staticfiles',

    # New Apps:
    'ModuloCentral.apps.webServices.wsDatos',
    'ModuloCentral.apps.webServices.wsUsuarios',
    'ModuloCentral.apps.home',

    # Uncomment the next line to enable the admin:
    'django.contrib.admin',
    # Uncomment the next line to enable admin documentation:
    'django.contrib.admindocs',
)

```

FIGURA NO. 51. ARCHIVO URLS.PY

```

GNU nano 2.2.6                                     File: urls.py

from django.conf.urls import patterns, include, url

# Uncomment the next two lines to enable the admin:
from django.contrib import admin
admin.autodiscover()

urlpatterns = patterns('',
    # Examples:
    # url(r'^$', 'ModuloCentral.views.home', name='home'),
    # url(r'^ModuloCentral/', include('ModuloCentral.foo.urls')),

    # New URLs
    url(r'^$', include('ModuloCentral.apps.home.urls')),
    url(r'^$', include('ModuloCentral.apps.webServices.wsDatos.urls')),
    url(r'^$', include('ModuloCentral.apps.webServices.wsUsuarios.urls')),

    # Uncomment the admin/doc line below to enable admin documentation:
    url(r'^admin/doc/', include('django.contrib.admindocs.urls')),
    # Uncomment the next line to enable the admin:
    url(r'^admin/', include(admin.site.urls)),
)

```

FIGURA NO. 52. DIAGRAMA DE FLUJO ARCHIVO VIEWS.PY

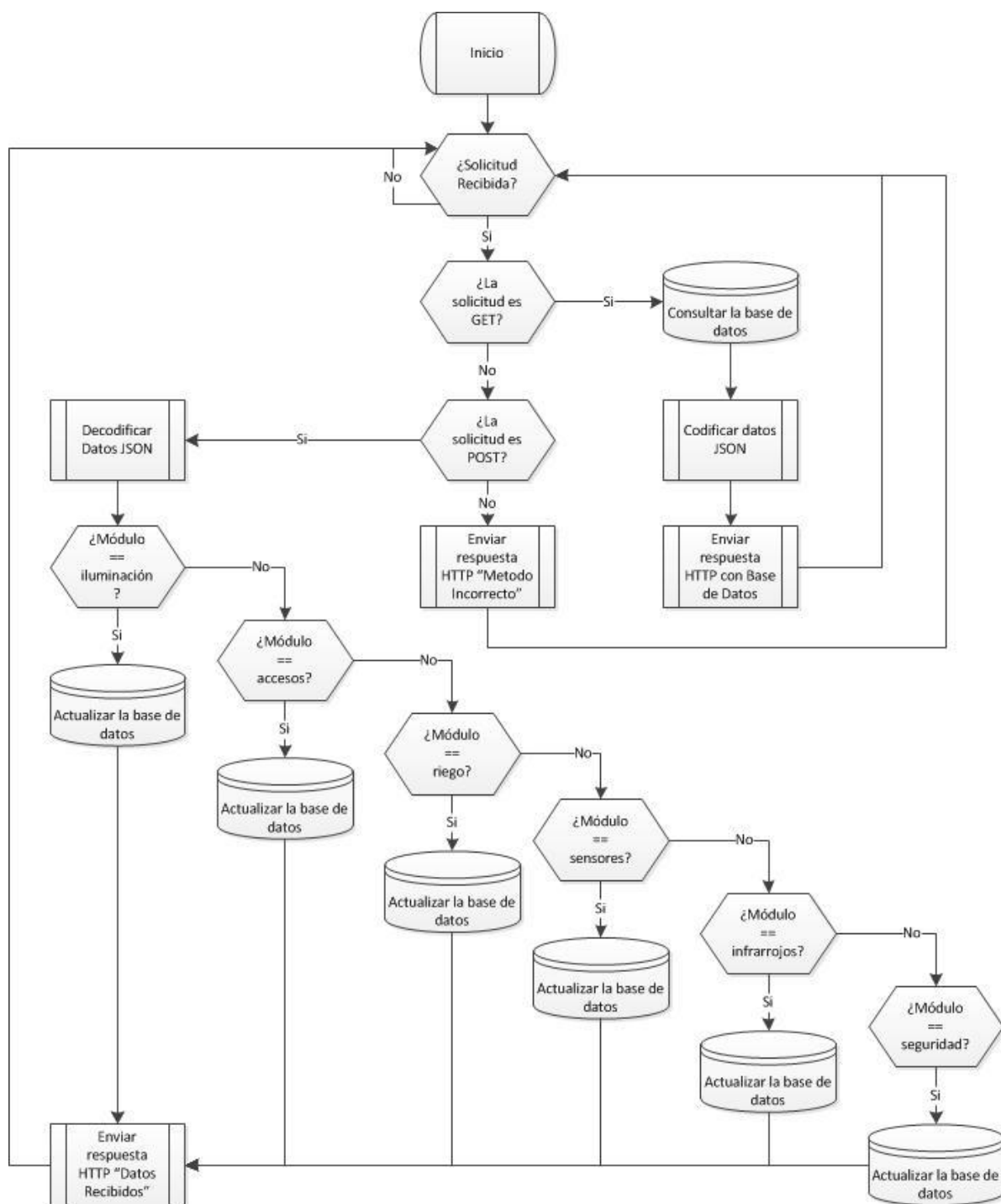


FIGURA NO. 53. SERVIDOR DJANGO EJECUTÁNDOSE (LADO SERVIDOR)

```

pi@raspberrypi ~/Proyecto/ModuloCentral $ bash main.sh
.....
.....: RDI :.....
.....
Validating models...

0 errors found
Django version 1.4.1, using settings 'ModuloCentral.settings'
Development server is running at http://192.168.1.110:8000/
Quit the server with CONTROL-C.
[26/Aug/2013 12:58:27] "GET /admin HTTP/1.1" 301 0
[26/Aug/2013 12:58:28] "GET /admin/ HTTP/1.1" 200 8805
[26/Aug/2013 12:59:30] "GET /admin/home/luz/ HTTP/1.1" 200 4514
[26/Aug/2013 12:59:30] "GET /admin/jsi18n/ HTTP/1.1" 200 2158
[26/Aug/2013 12:59:33] "GET /admin/home/luz/4/ HTTP/1.1" 200 6656
[26/Aug/2013 12:59:33] "GET /admin/jsi18n/ HTTP/1.1" 200 2158
[26/Aug/2013 12:59:35] "POST /admin/home/luz/4/ HTTP/1.1" 302 0
[26/Aug/2013 12:59:36] "GET /admin/home/luz/ HTTP/1.1" 200 4657
[26/Aug/2013 12:59:37] "GET /admin/jsi18n/ HTTP/1.1" 200 2158

```

FIGURA NO. 54. SERVIDOR DJANGO EJECUTÁNDOSE (APLICACIÓN ADMINISTRADOR)

Site administration | Django: x

192.168.1.110:8000/admin/

Django administration Welcome, drummond. Documentation / Change password / Log out

Site administration

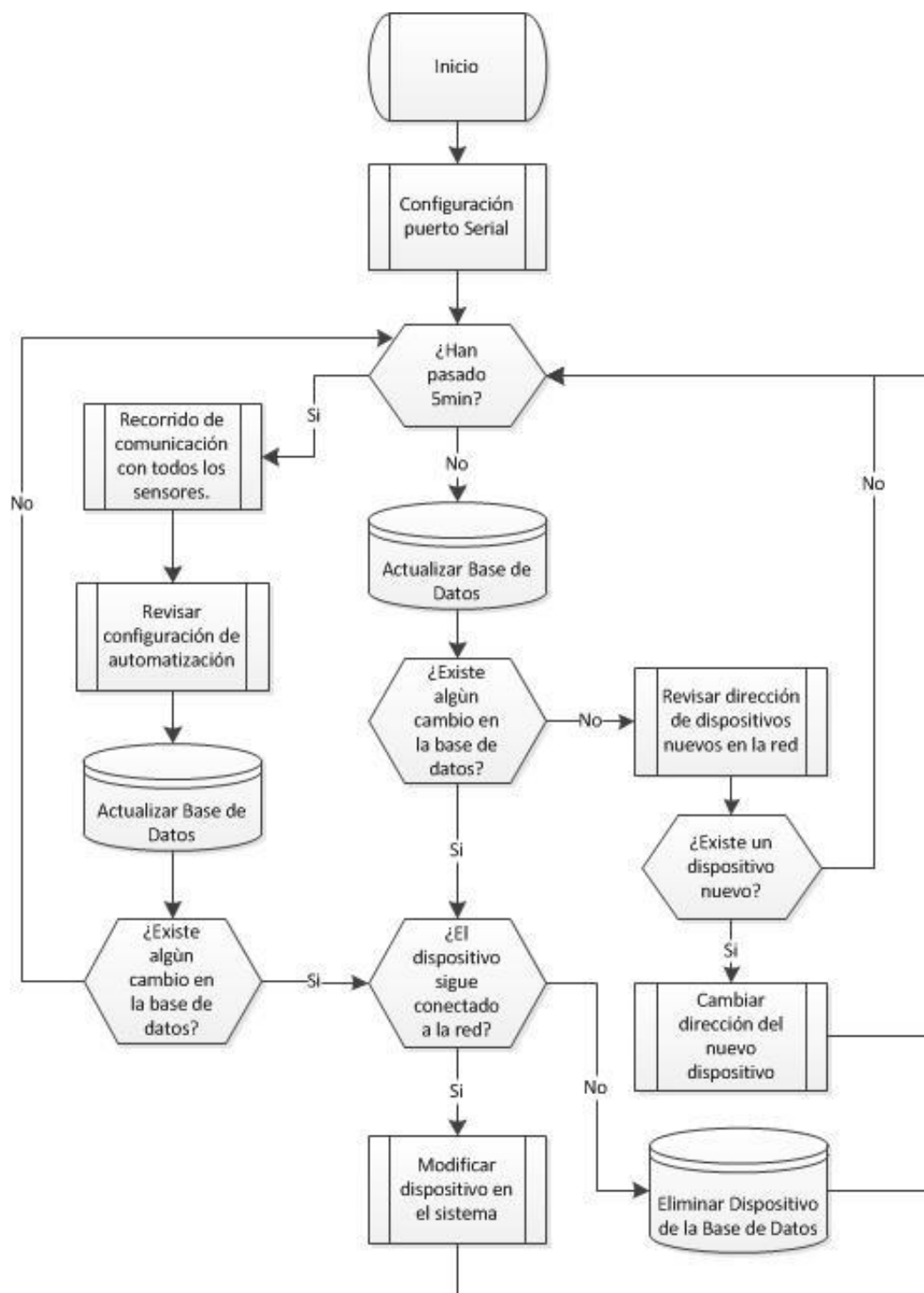
Model	Add	Change
Auth		
Groups	➕ Add	✏ Change
Users	➕ Add	✏ Change
Home		
Accesos	➕ Add	✏ Change
Luzs	➕ Add	✏ Change
Riegos	➕ Add	✏ Change
Seguridades	➕ Add	✏ Change
Sensoress	➕ Add	✏ Change
Tvs	➕ Add	✏ Change
Usuarios	➕ Add	✏ Change
Sites		
Sites	➕ Add	✏ Change

Recent Actions

My Actions

- ➕ Sensor No.1
- ➕ Sensores
- ➕ Camara No.1
- ➕ Seguridad
- ➕ Riego No.2
- ➕ Riego
- ➕ Riego No.1
- ➕ Riego
- ➕ Luz No.2
- ➕ Luz
- ➕ Luz No.4
- ➕ Luz
- ➕ Luz No.3
- ➕ Luz
- ➕ Luz No.2
- ➕ Luz
- ➕ Luz No.1
- ➕ Luz
- ➕ mback
- ➕ Usuario

FIGURA NO. 55. DIAGRAMA DE FLUJO ADMINISTRACIÓN DEL SISTEMA



VI. INFRAESTRUCTURA DE COMUNICACIÓN

A. TRANSMISIÓN POR CORRIENTES PORTADORAS

1. Diseño experimental. Se analizaron las ventajas y desventajas de cada uno de los tres tipos de comunicación que puede presentar un sistema domótico, además se buscaron los protocolos actuales existentes y que utilizan esos tipos de comunicación.

Las características que se analizaron fueron las siguientes:

- Confiabilidad de la comunicación: confiabilidad de la comunicación entre controlador y módulos. Se busca que el sistema asegure la llegada del mensaje y que pueda ser capaz de confirmar la correcta recepción.
- Seguridad de la comunicación: el sistema debe de garantizar que no pueda ser operado cuando se tiene un acceso no autorizado.
- Precio: se busca una solución económica, ya que el principal problema que presentan los sistemas domóticos es el alto costo que estos requieren.
- Protección de la inversión: asegura que el controlador instalado y los distintos módulos puedan ser reemplazados fácilmente en el futuro.
- Modularidad: se necesita que el sistema pueda utilizar módulos de distintos fabricantes.

Luego de analizadas esas características se determinó que en RDI se iba a utilizar la transmisión por corrientes portadoras. El protocolo más utilizado para este tipo de comunicación es X10. Se realizó un análisis de las ventajas y desventajas de este protocolo y se determinó que no era un método confiable para la transmisión de datos, además de que presenta una tasa de transmisión baja, por lo que se trabajó junto con el módulo de seguridad y comunicación para desarrollar un protocolo robusto que asegure la correcta transmisión y

recepción de la información; y que sea capaz de mejorar la velocidad de transmisión.

TABLA NO. 6. COMPARACIÓN DE CARACTERÍSTICAS PROTOCOLOS DE COMUNICACIÓN.

Protocolo	Medio de comunicación	Confiable	Seguro	Buen Precio	Protege la inversión	Modular
KNX	Cableada	Sí	Sí	No	Si	Sí
X10	Corrientes portadoras	No	No	Sí	Sí	Sí
Zigbee	Inalámbrica	Sí	No	No	Si	No

2. Resultados. Se analizó cada una de las características, mencionadas anteriormente, que deben de presentar los sistemas domóticos para los protocolos KNX, X10 y Zigbee.

Luego de haber determinado que el medio de comunicación a utilizar era el de corrientes portadoras se procedió a analizar las características del protocolo X10.

El protocolo X10 permite el envío de datos cada cruce por cero de la señal eléctrica, la cual tiene una frecuencia de 60 Hz.

El período es el inverso de la frecuencia:

$$T = \frac{1}{f} = \frac{1}{60} = 16.67 \text{ ms}$$

Nos interesa saber cada cuánto tiempo la señal cruza por cero, esto se obtiene dividiendo el período dentro de 2, ya que en un ciclo la señal sinusoidal cruza dos veces por cero:

$$t_{\text{cruce por cero}} = \frac{T}{2} = 8.33 \text{ ms}$$

Para la transmisión de un paquete de datos se necesitan 11 ciclos de la señal sinusoidal, 2 para enviar el código de inicio, 4 para el código de casa y 5 para el código de unidad.

Conociendo el tiempo que transcurre entre cada cruce por cero se logró determinar el tiempo que requiere el protocolo para el envío de los códigos mencionados anteriormente.

TABLA NO. 7. TIEMPOS DE EJECUCIÓN INSTRUCCIONES X10.

	Cantidad de bits	Ciclos necesarios	Tiempo de ejecución
Código de inicio	4	2	0.033333333
Código de casa	8	4	0.066666667
Código de unidad	10	5	0.083333333
Espera	0	3	0.05
		Tiempo total (s)	0.23333333

Como se observa en la Tabla 7, el tiempo requerido para enviar un paquete de datos completos es de aproximadamente 233 ms. Sin embargo el protocolo requiere que los datos sean enviados dos veces teniendo tres ciclos de silencio entre cada par, por lo que el tiempo total resulta ser de 417 ms. Este tiempo sería el necesario para enviar a través de la línea eléctrica la dirección del dispositivo que se desea controlar, por lo que todavía hace falta enviar el código de la instrucción que se desea que realice. Es por esto que se vuelven a enviar los mismos comandos, cambiando únicamente el código de unidad. Como la cantidad de bits que se envían es la misma el tiempo requerido resulta ser nuevamente de 417 ms, por lo que el tiempo total requerido para que un dispositivo X10 realice una acción es de aproximadamente 833 ms.

TABLA NO. 8. TIEMPO TOTAL DE ENVÍO PROTOCOLO X10.

Comando	Tiempo (S)
Dirección	0.416666667
Función	0.416666667
Total	0.833333333

3. Discusión. Como se puede observar en la Tabla 6, el único protocolo entre los analizados que presenta un buen costo es X10, además de ser más popular en el mundo, por lo que en un principio se decidió trabajar con este protocolo.

La instalación de cableado dedicado específicamente para el envío de datos puede ser complicada y costosa, es por eso que el protocolo X10 toma como medio de transmisión el cableado de alimentación de poder existente ya en los hogares. Con este método cualquier persona está en la capacidad de poder automatizar dispositivos de su hogar sin la necesidad de tener que hacer cambios en la estructura de la misma o sin tener que comprar una casa nueva para disfrutar de estos beneficios.

Como se observa en la Tabla 6 una de las principales debilidades de X10 es su baja confiabilidad, ya que la mayoría de sus módulos no son bidireccionales, lo que impide conocer los estados del sistema ocasionando que no se pueda determinar si los datos enviados fueron recibidos correctamente. Además se encontró que su velocidad de transmisión es muy baja, lo que ocasiona que se requiera de un tiempo elevado para que los distintos comandos puedan ser enviados y recibidos. Como se puede observar en la Tabla 8 el tiempo requerido para enviar dirección y comando de función a un dispositivo es de aproximadamente 833 ms, tiempo que se obtiene en el mejor de los casos. Incluso este tiempo puede llegar a ser mayor dependiendo de las veces que sea necesario enviar los comandos para que sean recibidos correctamente.

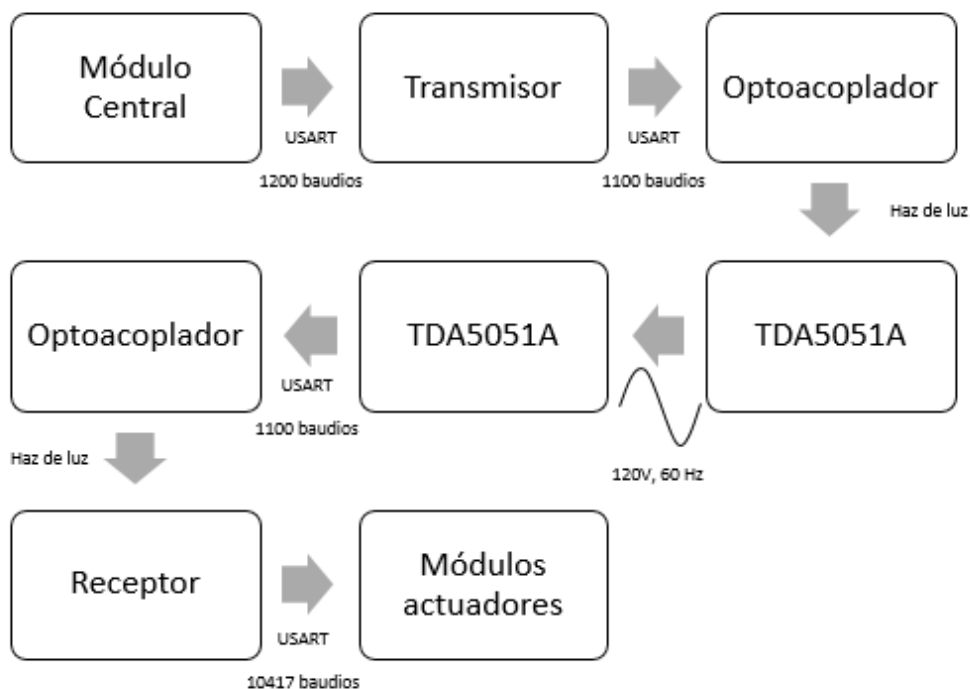
Se investigó sobre otros métodos posibles para enviar información a través de la línea eléctrica, fue así como se encontró el circuito integrado TDA5051A, el cual es un módem utilizado para realizar automatización en el hogar. Se utilizó este integrado para la modulación de los datos y se trabajó junto con el módulo de comunicación y seguridad para la creación de un protocolo de comunicación que asegurara el correcto envío de los datos, mejorando de esta manera las debilidades que presenta el protocolo X10.

B. MODEM DE COMUNICACIÓN BIDIRECCIONAL

Se utilizó el circuito integrado TDA5051A, ya que este que utiliza la modulación ASK para la transmisión y recepción de señales digitales a través de la línea de corriente AC. Se decidió escoger este integrado ya que es una solución efectiva y económica capaz de transmitir a una tasa de hasta 1200 baudios, lo que mejora considerablemente la velocidad de transmisión en comparación con el protocolo X10. Además presenta un módulo USART que permite una fácil conexión con distintos microcontroladores.

A continuación se analizarán y discutirán los requerimientos necesarios que se deben de cumplir para poder obtener una operación confiable y robusta en la línea eléctrica, logrando de esta manera la comunicación entre módulo central y módulos actuadores, siguiendo el proceso mostrado en la Figura 56.

FIGURA NO. 56. DIAGRAMA DE FLUJO MÓDEM BIDIRECCIONAL



1. Diseño experimental. El primer paso consistió en determinar la frecuencia de operación del módem. Esta se encuentra únicamente definida por la elección del cristal o por la frecuencia de la señal aplicada a la entrada del oscilador, como se puede ver en la Tabla 9.

TABLA NO. 9. FRECUENCIA DE LA ONDA PORTADORA.

Frecuencia del oscilador	Frecuencia de salida del reloj	Frecuencia de la portadora
f_{osc}	$f_{osc}/2$	$f_{osc}/64$

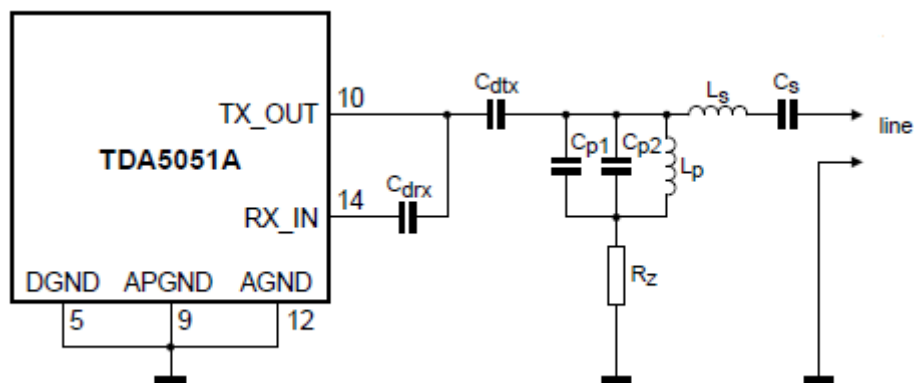
Sin embargo para cumplir con los requisitos de la norma EN50065-1 la frecuencia de la onda portadora debe de estar entre los 95 kHz y los 148.5 kHz como puede verse en la Tabla 10.

TABLA NO. 10. FRECUENCIA DE LA ONDA PORTADORA PARA LA TRANSMISIÓN DE DATOS

9 kHz a 95 kHz	95 kHz a 125 kHz	125 kHz a 140 kHz	140 kHz a 148.5 kHz
Restringida al uso	Libre para consumidores	Libre para consumidores	Libre para consumidores

El siguiente paso consistió en lograr el acople de la señal modulada con la línea eléctrica. Para esto se realizó el filtro pasa banda, con elementos pasivos, que se observa en la Figura 57.

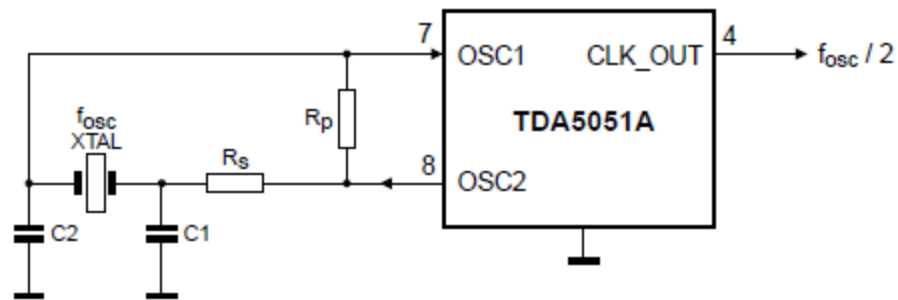
FIGURA NO. 57. FILTRO PASABANDAS LC.



(Fuente: 14)

Luego de haber diseñado el filtro se realizó la conexión con el microcontrolador. Para esto el primer paso fue determinar la configuración de reloj que utilizaría el integrado. Se tomó la decisión de utilizar el reloj interno del integrado, utilizando el circuito de la Figura 58.

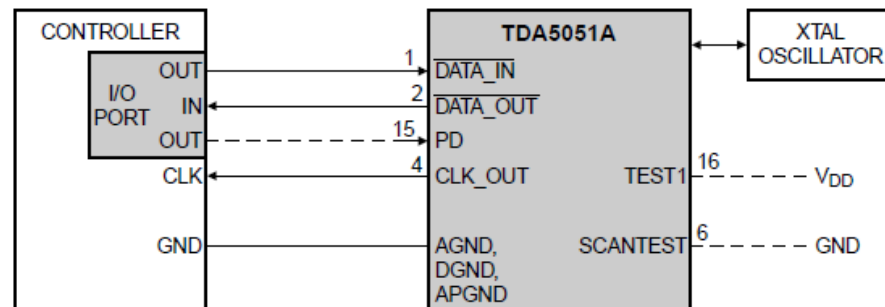
FIGURA NO. 58. CIRCUITO DE OPERACIÓN DEL RELOJ INTERNO.



(Fuente: 14)

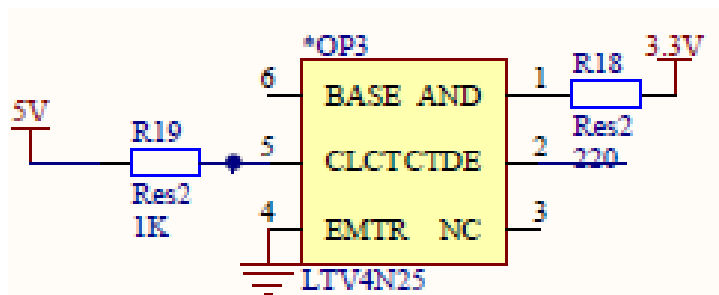
El siguiente paso consistió en determinar la forma en que los datos y las señales se transmitirían del integrado al microcontrolador. Como se puede ver en la Figura 59, la información que sale del integrado puede conectarse a cualquier microcontrolador con puertos de entrada y salida que pueden ser tanto TTL como CMOS. Para lograr un aislamiento eléctrico en esta conexión se diseñó el circuito de la Figura 11 utilizando el optoacoplador LTV4N25.

FIGURA NO. 59. CONECCIÓN CON EL MICROCONTROLADOR.



(Fuente: 14)

FIGURA NO. 60. CIRCUITO UTILIZADO PARA EL OPTOACOPLADOR



Ya teniendo seleccionado el valor de todos los componentes a utilizar en el módem de comunicación se realizaron las pruebas de funcionamiento.

2. Resultados. Se determinó que el oscilador que se utilizaría tendría una frecuencia de 8 MHz. Los valores de la frecuencia de salida del reloj, así como la frecuencia portadora se pueden ver en la Tabla 11.

TABLA NO. 11. SELECCIÓN DE FRECUENCIAS.

Frecuencia del oscilador	Frecuencia de salida del reloj	Frecuencia de la portadora
8 MHz	4 MHz	125 kHz

Los capacitores C_{dtx} y C_{dtx} mostrados en el circuito de la figura 9 tienen la función de realizar el desacoplamiento DC de los pines TX_OUT y RX_IN del módem.

TABLA NO. 12. VALORES ESCOGIDOS CAPACITORES DE DESACOPLE

Capacitor	Valor utilizado
C_{dtx}	1 uF electrolítico no polarizado (16V)
C_{dtx}	10 nF cerámico

El filtro LC debe ser sintonizada para la frecuencia portadora elegida. Para ello se utilizó la siguiente fórmula:

$$f_c = \frac{1}{2\pi\sqrt{L_s * C_s}} = \frac{1}{2\pi\sqrt{L_p * (C_{p1} + C_{p2})}} = \frac{f_{osc}}{64} = 125 \text{ kHz}$$

Con f_c y f_{osc} en hertz, L en henrys y C en faradios.

Sabiendo que la frecuencia portadora que se utiliza es de 125 KHz, se escogió uno de los dos parámetros desconocidos, en este caso el valor del inductor L_s (47 μH) y se despejó para el valor del capacitor C_s :

$$C_s = \frac{1}{4\pi^2 * L_s * f_c^2} = \frac{1}{4\pi^2 * (47\mu\text{H}) * (125\text{KHz})^2} = 34.5\text{nF}$$

El resultado indica que el capacitor debe de ser de 34.5 nF, sin embargo este no es un valor comercial, por lo que se decidió utilizar el valor más cercano que es de 33 nF. Con este nuevo valor de capacitancia la frecuencia a la que se encuentra sintonizado el filtro es:

$$f_c = \frac{1}{2\pi\sqrt{L_s * C_s}} = \frac{1}{2\pi\sqrt{(47\mu\text{H}) * (33\text{nF})}} = 127.8\text{kHz}$$

TABLA NO. 13. VALORES ESCOGIDOS FILTRO PASABANDA

L_s (μH)	C_s (250 V AC) (nF)	L_p (μH)	C_{p1} (nF)	C_{p2} (nF)
47	33	47	33	-----

Los últimos elementos que quedaban por escogerse eran los utilizados para el circuito de la Figura 58. Se decidió utilizar el reloj interno del módem por lo que se necesitaron dos capacitores, una resistencia y un cristal. El valor escogido para estos puede verse en la Tabla 11.

TABLA NO. 14. VALORES ESCOGIDOS COMPONENTES EXTERNOS DEL RELOJ.

Componente	Valor
C1 y C2	Capacitores cerámicos de 27 pF
R_p	2.2 M Ω
R_s	0 Ω
Oscilador	8 MHz

Se realizaron pruebas para determinar el rango de frecuencias a las que la optocopla funcionaba correctamente. Los resultados pueden observarse en las figuras 61 a 64.

FIGURA NO. 61. FUNCIONAMIENTO OPTOACOPLADOR A 400 HZ.

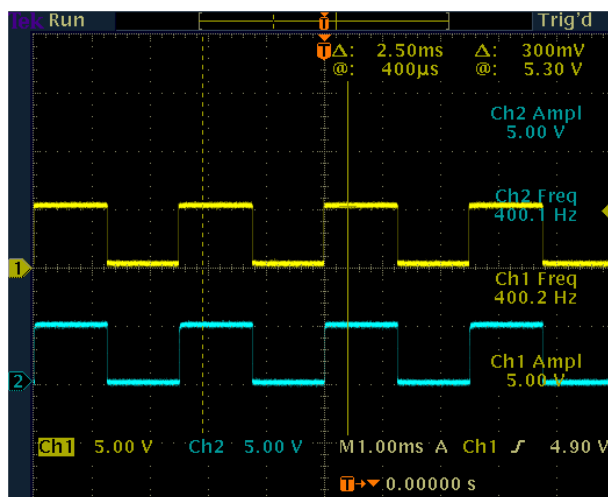


FIGURA NO. 62. FUNCIONAMIENTO OPTOACOPLADOR A 1.2 KHZ.

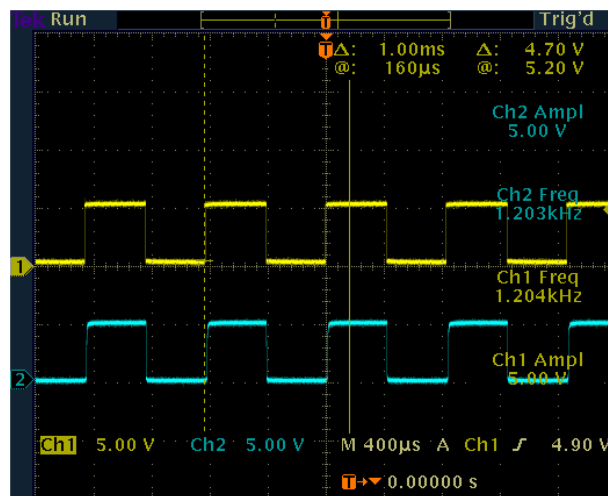


FIGURA NO. 63. FUNCIONAMIENTO OPTOACOPLADOR A 10 KHZ.

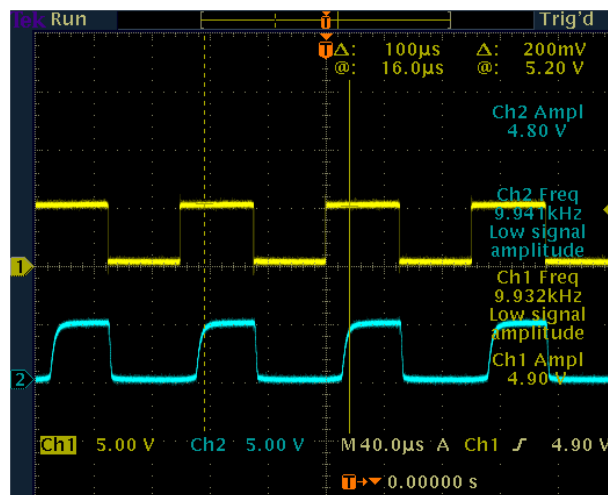
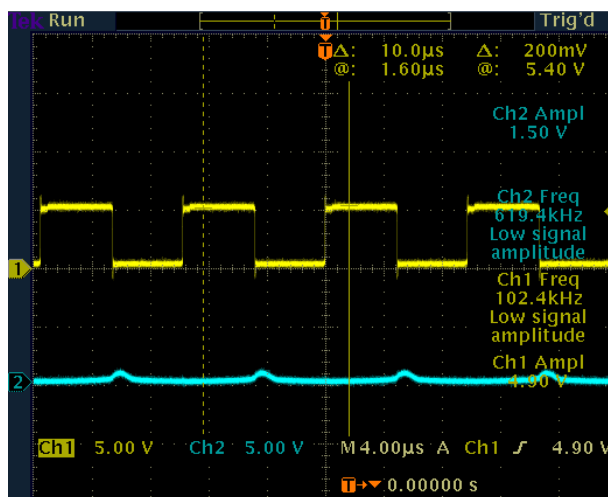
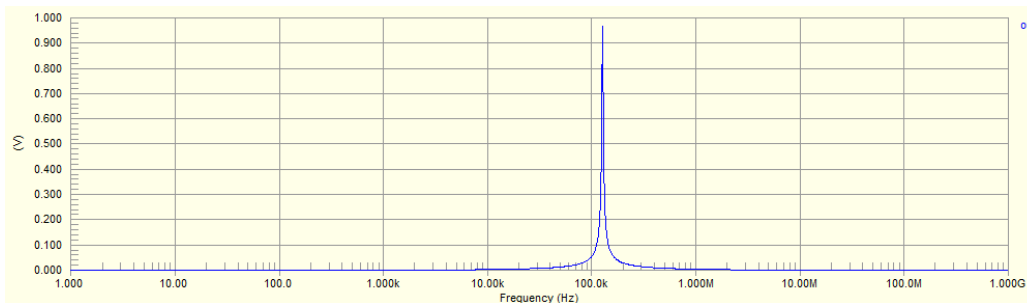


FIGURA NO. 64. FUNCIONAMIENTO OPTOACOPLADOR A 100 KHZ.



Se simuló el circuito de la Figura 57 para determinar el correcto funcionamiento el filtro LC. Como se puede observar en la figura 65 el filtro se encuentra sintonizado a una frecuencia de aproximadamente 128 kHz.

FIGURA NO. 65. FILTRO LC.



El siguiente paso consistió en analizar la señal modulada a través de todo el proceso de comunicación. El paquete de datos sale del microcontrolador, pasa por el optoacoplador, llega al módem, es modulado y enviado a través de la línea de corriente AC por medio de TX_OUT, luego es recibido por RX_IN, demodulado y enviado por DATA_OUT hacia el receptor. El proceso puede observarse más detalladamente en la figura 7. Los resultados de este proceso se encuentran en las figuras 66 a 70.

FIGURA NO. 66. COMPARACIÓN TX PIC (CH1) Y DATA_IN MÓDEM (CH2).

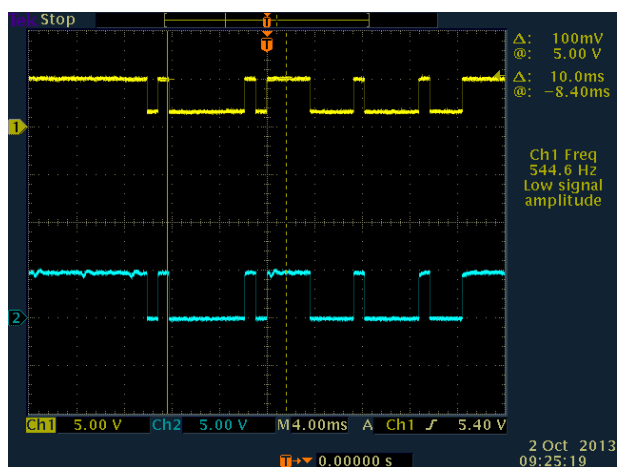


FIGURA NO. 67. COMPARACIÓN DATA_IN MÓDEM (CH1) Y TX_OUT MÓDEM (CH2).

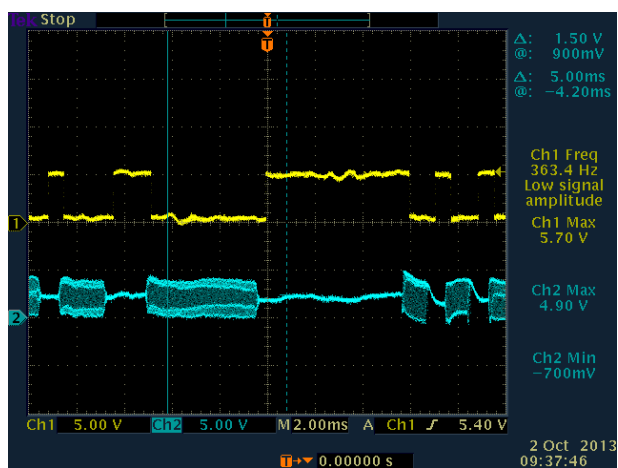


FIGURA NO. 68. COMPARACIÓN RX_IN MÓDEM (CH1) Y DATA_OUT MÓDEM (CH2).

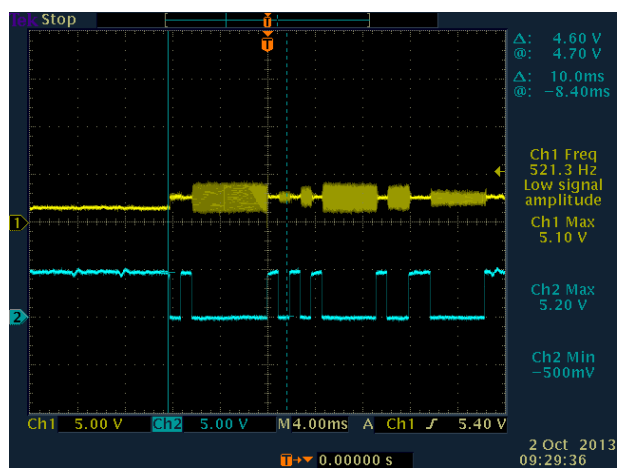


FIGURA NO. 69. COMPARACIÓN TX_OUT EMISOR (CH1) Y RX_IN RECEPTOR (CH2).

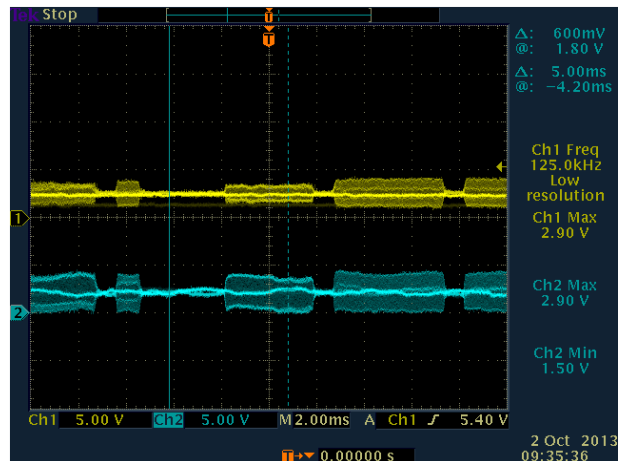
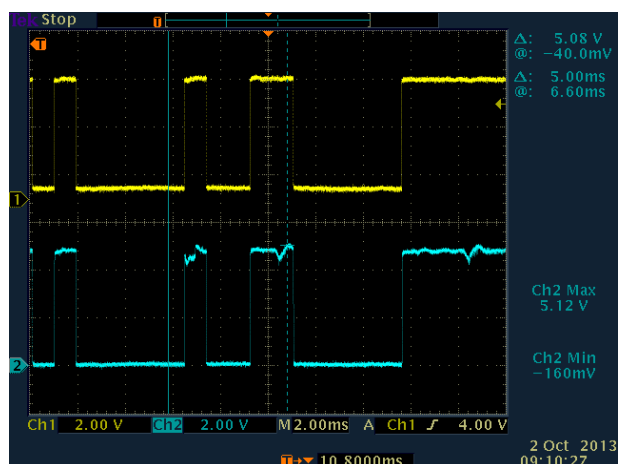


FIGURA NO. 70. COMPARACIÓN TX EMISOR (CH1) Y RX RECEPTOR (CH2).



3. Discusión. El primer paso en el desarrollo del módem de comunicación bidireccional fue la delimitación de todos los elementos necesarios que serían utilizados. Se comenzó determinando la frecuencia de operación que tendría el sistema. Como se puede ver en la Tabla 9 este valor únicamente depende de la frecuencia del oscilador que se quiera utilizar, sin embargo se debía de tomar en cuenta que no se puede escoger cualquier valor de frecuencia, ya que esta se encuentra delimitada según la norma EN50065-1, la cual indica que el rango de frecuencias permitido para la transmisión de señales se debe de encontrar entre los 95 kHz y los 148.5 kHz.

La frecuencia del oscilador es 64 veces más grande que estos dos valores, lo que indica que se debía de escoger un oscilador entre los 6.272 MHz y 9.312 MHz. Es por esto que se tomó la decisión de utilizar un oscilador de 8 MHz, ya que este era un valor comercial. Como se puede ver en la Tabla 11, con este valor de oscilador se obtuvo que la frecuencia de la señal portadora sería de 125 kHz.

Existen dos formas de acoplar las señales moduladas con la línea de alimentación AC: la primera, que es la más económica, es utilizando dos filtros LC (este método tiene la desventaja de que no brinda aislamiento eléctrico). El segundo método es utilizando un transformador de altas frecuencias. Siguiendo con la metodología utilizada de buscar realizar un sistema doméstico económicamente accesible se tomó la decisión de realizar el acople utilizando dos filtros LC. El filtro, que se observa en la Figura 65, cumple dos funciones: en la recepción rechaza la señal de 60 Hz y sirve como pasa bajos para el filtro digital; mientras que en la transmisión rechaza armónicos inesperados.

Para la selección de los elementos utilizados en los filtros se tomó la decisión de escoger el valor del inductor y de acuerdo a este determinar el valor que tendría que tener el capacitor. Habiendo escogido una inductancia de $47\mu\text{H}$, se obtuvo una capacitancia de 34.5nF , sin embargo este no es un valor comercial, por lo que se decidió utilizar un capacitor de 33 nF . Este cambio ocasiono que los filtros se sintonizaran a una frecuencia de 128 kHz , un valor que no se encuentra muy alejado de los 125 kHz que presenta la señal portadora. A pesar de este cambio los filtros continuaron funcionando correctamente, como se puede observar en la Figura 65.

Los pines del modulador *TX_OUT* y *RX_IN* presentan un DC offset de 2.5V , por lo que era necesario colocarles capacitores que cumplieran con la función de realizar un desacoplamiento DC. El valor de estos capacitores no era crítico, lo único que debían de cumplir es que permitieran una baja impedancia, por lo que, como se puede ver en la Tabla 12, se decidió utilizar un capacitor electrolítico de $1\mu\text{F}$ para el C_{dtx} y un capacitor cerámico de 10 nF para C_{drx} .

Habiendo escogido un oscilador de 8 MHz para su utilización en el circuito se debía de definir en cuál de las dos configuraciones de reloj iba a ser utilizado el módem: teniendo una señal externa de reloj o utilizando su reloj interno. La mejor opción era utilizar el reloj interno del oscilador y que este brindara la señal de reloj necesaria para el módem. Esto hubiera provocado que no fuera necesario utilizar los elementos del circuito de la Figura 58. Sin embargo la familia de microcontroladores PIC16F tienen un oscilador interno con frecuencia máxima de 8 MHz, lo que le permite tener una salida de reloj máxima de 2 MHz, valor que no cumple con lo establecido anteriormente de utilizar un oscilador de 8 MHz. Fue por esto que se decidió utilizar un cristal externo junto con los elementos de la Tabla 14.

Para realizar la conexión entre el microcontrolador y el módem se debía de escoger alguno de los puertos de entrada y salida que el primero presenta. Se escogió el módulo *EUSART*, y se decidió que la comunicación fuera asíncrona, de manera que no se tuviera la necesidad de enviar la señal de reloj junto con los datos o de necesitar alguna otra línea para la comunicación. En un principio se estaba utilizando el PIC16F887, el cual presenta un único módulo *EUSART*, pero es de tamaño grande en comparación con otros ya que es de empaquetado de 40 patas, fue por esto que se decidió cambiar al PIC16F886, un microcontrolador de 28 patas. Este PIC también cuenta con un único módulo *EUSART*, por lo que fue necesario que se desarrolla el otro módulo en software, para poder de esta manera tener el *USART* físico para la comunicación con el módem y el *USART* desarrollado en software para la comunicación con los módulos periféricos.

Otro de los aspectos importante que se decidió realizar fue la utilización de opto acopladores para aislar al módem del microcontrolador, ya que este no presenta aislamiento eléctrico con respecto a la línea de alimentación AC. Se realizaron pruebas para determinar el rango de frecuencias en el cual el optoacoplador funcionaba correctamente. Como se observa en las figuras 61 y 62, cuando estos trabajan con señales que presentan una frecuencia baja la

salida es casi idéntica a la entrada, pero mientras mayor sea la frecuencia de la señal de entrada la señal de salida empieza a perder sus características. Esto puede observarse en la Figura 63, en donde se utilizó una señal de entrada con una frecuencia de 10kHz que presentó una salida que ya no era cuadrada. Caso más crítico fue el que se obtuvo al trabajar a una frecuencia de 100 kHz, ya que en este punto, como se puede ver en la Figura 64, la señal de salida se encuentra totalmente atenuada. Fue por esta limitante y por la velocidad de transmisión máxima que puede soportar el módem que se escogieron las velocidades de transmisión mostradas en la Figura 56.

El proceso por el que pasa un paquete de datos que viaja a través de la infraestructura de comunicación es el siguiente: el módulo central envía el comando a realizarse por medio de *USART*, a una velocidad de 1200 baudios. Estos datos llegan al receptor, el cual a una velocidad de 1100 baudios envía la señal hacia el optoacoplador. El optoacoplador envía a la misma velocidad los datos hacia el modulador. Como se puede observar en la Figura 66 el paquete de datos que sale del *TX* del emisor llega al *DATA_IN* del módem sin sufrir mayor modificación. En el momento que se tiene un flanco negativo en ese pin la señal ya modulada es enviada a la línea eléctrica a través del pin *TX_OUT* tal y como se puede observar en la Figura 66. Esta señal llega entra al módem por medio del pin *RX_IN* para poder ser demodulada. Concluido este proceso los datos son colocados en *DATA_OUT* para ser enviados vía *USART* al optoacoplador, que a su vez los envía hacia el pin *RX* del microcontrolador. Este proceso puede observarse en la Figura 67, mientras que en las Figuras 68 y 69 se puede observar la comparación entre la señal modulada enviada y la recibida, así como los datos, pudiendo notar que la transmisión fue realizada con éxito.

C. FUENTE DE ALIMENTACIÓN

En las tecnologías actuales se busca que los productos sean cada vez más pequeños y ligeros. Por lo tanto las fuentes de alimentación que utilicen deben de ser diseñados para cumplir estos requisitos sin olvidar parámetros importantes tales como la eficiencia y costo.

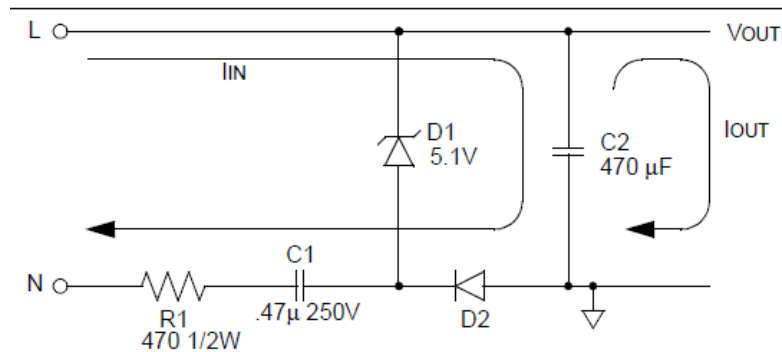
Existen varias maneras de convertir un voltaje AC en un voltaje DC para la utilización en circuitos electrónicos. La forma tradicional de realizar esta conversión es utilizando una fuente de alimentación lineal, la cual utiliza para la conversión un transformador y un circuito rectificador, para luego pasar al proceso de filtrado y de regulación. También se pueden utilizar fuentes de alimentación conmutadas. Sin embargo existen aplicaciones en las que únicamente se necesita alimentar un microcontrolador y otros pocos dispositivos de bajo consumo de corriente. Es por esto que las fuentes que utilizan transformador o que son conmutadas no son soluciones óptimas económicamente hablando.

Las fuentes de voltaje sin transformador son una alternativa a bajo costo, que tienen la ventaja de ser de menor tamaño en comparación con los demás tipos de fuente. Es por esto que se decidió implementar este tipo de fuente de alimentación, buscando que la misma cumpliera con los requisitos de corriente que presentan los elementos utilizados en el diseño del módem de comunicación bidireccional.

Los dos tipos básicos de fuente de alimentación sin transformador son las resistivas y las capacitivas. Se hizo el análisis de ambos circuitos, determinando la corriente máxima que podían suministrar, así como su voltaje de salida. Se determinaron sus ventajas y desventajas, además de que se realizaron consideraciones adicionales de seguridad, ya que no existe aislamiento con el voltaje de la línea AC.

1. Diseño experimental. Se construyó una fuente de alimentación sin transformador capacitiva como la de la Figura 71, utilizando los elementos de la tabla 12. El voltaje permanece constante en la carga siempre y cuando la corriente de salida (I_{out}) sea igual o menor a la corriente de entrada (I_{in}), que se encuentra limitada por la resistencia R_1 y por la reactancia de C_1 .

FIGURA NO. 71. DISEÑO FUENTE DE ALIMENTACIÓN SIN TRANSFORMADOR CAPACITIVA.



(Fuente: 3)

TABLA NO. 15. ELEMENTOS UTILIZADOS FUENTE SIN TRANSFORMADOR CAPACITIVA

Elemento	Valor	Dimensionamiento
Resistencia (R1)	470Ω	0.5 watts
Capacitor (C1)	0.47μF	250V
Diodo zener (D1)	5.1V	0.5 watts
Diodo (D2)	0.6V	0.25 watts
Capacitor (C2)	470μF	16V

Luego a la fuente de alimentación sin transformador capacitiva se le agregaron dos elementos para mejorar su funcionamiento. Primero se agregó un inductor en serie con el capacitor y la resistencia que se encuentran a la entrada. Esto se realizó para limitar los picos de corriente que pueden llegar a existir en algún momento. También se agregó el integrado LT7805 para regular de mejor manera el voltaje de salida. El circuito puede observarse en la Figura 72.

FIGURA NO. 72. FUENTE CAPACITIVA CON REGULADOR DE VOLTAJE.

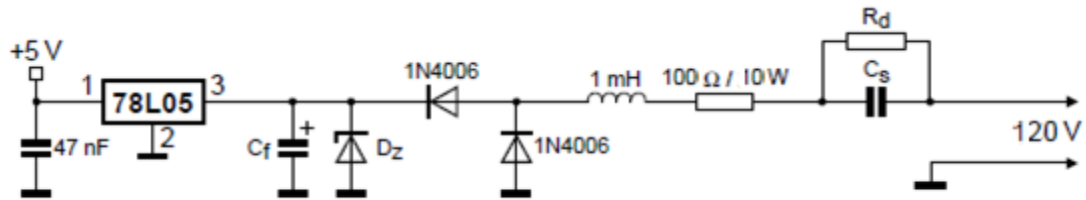


TABLA NO. 16. ELEMENTOS UTILIZADOS FUENTE CAPACITIVA CON REGULADOR DE VOLTAJE

Elemento	Valor	Dimensionamiento
Resistencia (R1)	100Ω	10 watts
Diodo zener (D1)	7.5V	1 watt
2 Diodos	0.7V	0.125 watts
Capacitor (Cs)	470μF	250V
Resistencia (Rd)	1MΩ	0.5 watts
Regulador de voltaje	5V	-----
Inductor	1mH	1 watt

2. Resultados. Para el cálculo de la corriente de entrada se utilizaron las ecuaciones para una fuente capacitiva. La ecuación para la corriente resulta ser:

$$I_{in} = \frac{\sqrt{2}V_{RMS} - V_z}{2\left(\frac{1}{2\pi f C_1} + R_1\right)}$$

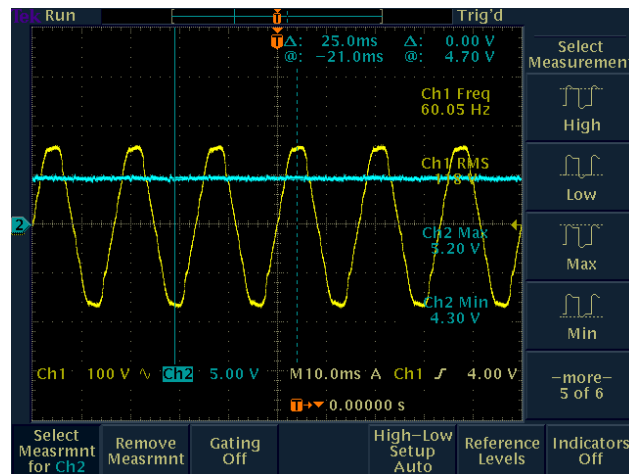
Sabiendo que el voltaje RMS es de 120V y que la frecuencia de la línea eléctrica es de 60 Hz se calculó la corriente máxima de entrada que puede obtenerse con la fuente:

$$I_{in} = \frac{\sqrt{2}(120V) - (5.1V)}{2\left(\frac{1}{2\pi(60Hz)(4.7\mu F)} + 100\Omega\right)} = 125mA$$

Con esta corriente se calculó la potencia de la resistencia:

$$P = VI = I^2R = (125mA)^2(100\Omega) = 1.56 \text{ watts}$$

FIGURA NO. 73. VOLTAJE DE SALIDA FUENTE CAPACITIVA CON REGULADOR DE VOLTAJE.



3. Discusión. Luego de analizar las principales ventajas y desventajas de las fuentes de alimentación sin transformador resistivas y capacitivas, se determinó que la solución más eficiente era utilizar la fuente capacitiva, ya que a pesar de presentar un costo más alto son energéticamente más eficientes, ya que es menor la cantidad de energía que se disipa en forma de calor a través de la resistencia.

Analizando el circuito, se puede observar que mientras mayor sea la capacitancia del capacitor que se utilice su reactancia será menor, lo que provocará que fluya una mayor corriente a través de la resistencia. Fue por esto que la primera modificación que se realizó fue el cambiar el capacitor de tantalio que originalmente era de $2.2\mu\text{F}$ al valor comercial más alto que encontramos disponible para 250V, que fue de $4.7\mu\text{F}$. Otro de las modificaciones que se hizo fue con respecto a la potencia de la resistencia, ya que luego de dejar los circuitos conectados una cantidad de tiempo considerable se pudo observar que la resistencia se calentaba bastante. Esto no solo puede ocasionar que la resistencia se dañe, sino que también elementos que se encuentren muy cercanos a ella, incluso la misma placa impresa puede dañarse. Por esto se decidió aumentar la potencia de la resistencia de 2watts a 10 watts. Por último

se agregó un regulador de voltaje LT7805 con el objetivo de que el voltaje de salida fuera más constante.

Se tomaron medidas de seguridad ya que la fuente no se encuentra aislada de la línea de alimentación. Se utilizó un fusible, para dar protección contra cortocircuitos, mientras que para los picos de voltaje se utilizó un varistor. Por último se utilizaron resistencias en paralelo a los capacitores que se encontraban conectados directamente a los 120V de la línea AC, de manera que sirvieran como método de descarga al momento de la desconexión del circuito, ya que el capacitor podía seguir cargado, luego de haberse desconectado.

VII. COMUNICACIÓN Y SEGURIDAD

A. DELIMITACIÓN

El megaproyecto RDI consta de distintos módulos específicos para cumplir con los objetivos planteados, estos módulos están diseñados para interactuar entre sí, por lo cual es necesario que exista comunicación entre los mismos.

El protocolo debe comunicarse con el módulo central, el módulo de infraestructura de comunicación y con los distintos módulos periféricos. Cada dispositivo envía y recibe distinto tipo de datos, por lo cual es necesario adecuar el protocolo para cada uno de estos módulos.

1. Diseño experimental. El primer paso para definir las reglas que implementará el protocolo es conocer que datos necesita cada módulo, por lo cual se elaboró un listado con la información de envío y recepción para cada uno. Esto se puede observar en la Tabla 17.

TABLA NO. 17. INFORMACIÓN DE ENVÍO Y RECEPCIÓN POR MÓDULOS

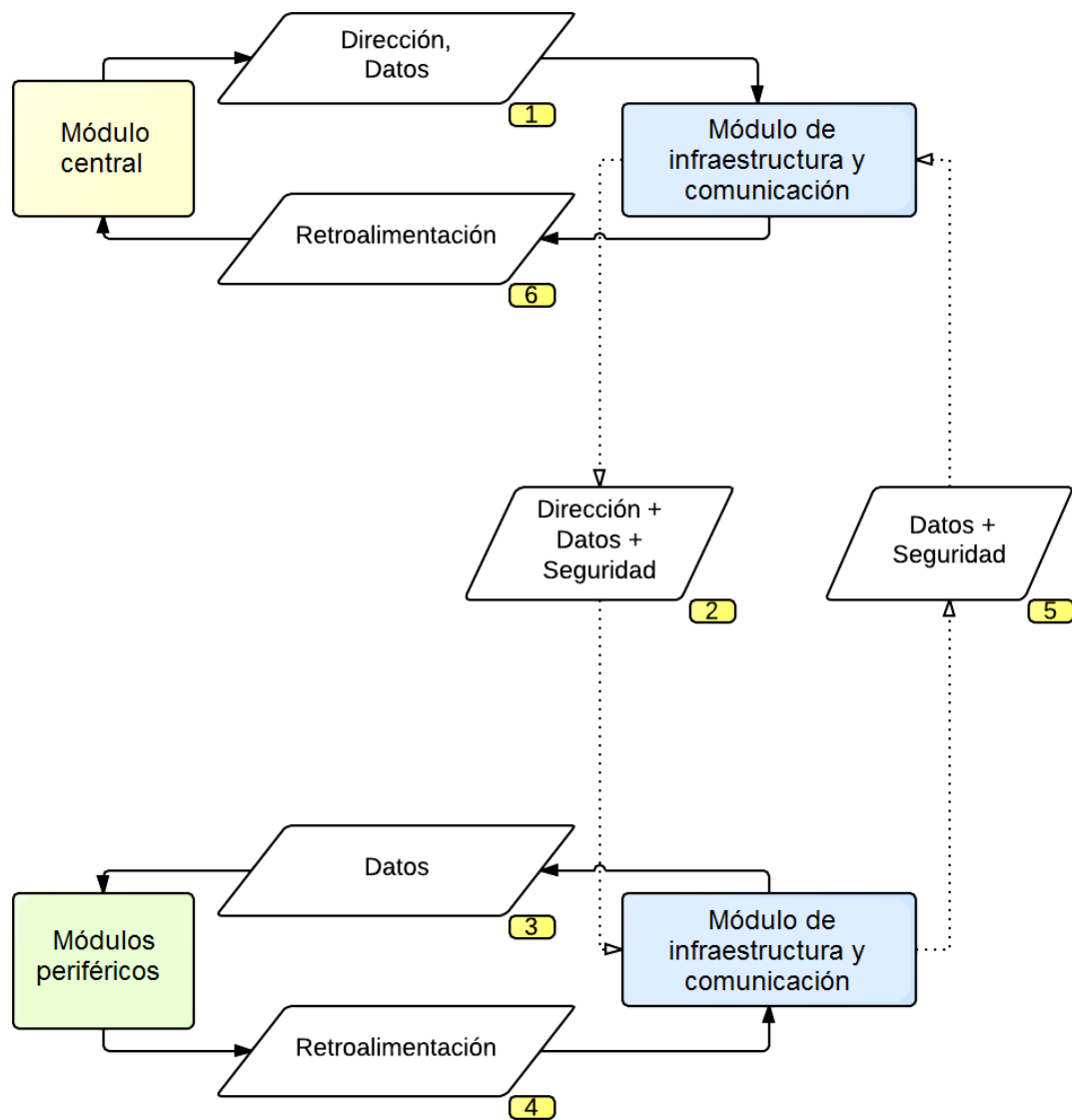
Módulo Central		Módulos periféricos	
Enviar	Recibir	Recibir	Enviar
<i>Dirección</i>	<i>Retroalimentación</i>	<i>Comando</i>	<i>Retroalimentación</i>
<i>Comando</i>			

Luego se buscó definir cuál sería la secuencia de flujo para el intercambio de información entre módulos, tomando en cuenta que la comunicación por el medio físico, la línea de alimentación AC, se realizaría a través del módulo de infraestructura de comunicación.

Se realizaron distintos diagramas representando el intercambio de datos entre módulos, descartando aquellos que se consideraban demasiado extensos hasta alcanzar el diseño final, el cual se observa en los resultados de este apartado.

2. Resultados

FIGURA NO. 74. INTERCAMBIO DE DATOS ENTRE MÓDULOS.



3. Discusión. En la Figura 7 se observa el flujo de información entre módulos del megaproyecto, el protocolo debe permitir que se realice este proceso por cada comando que se desee enviar a cualquier dispositivo periférico, tomando en cuenta que el módulo central espera obtener una respuesta proveniente del dispositivo.

La secuencia completa de comunicación inicia con el módulo central (1), el cual envía la dirección del dispositivo con el cual se desea comunicar, seguido del dato o comando que vaya dirigido a este. Luego el módulo de infraestructura recibe estos dos datos, calculando un valor de seguridad específico basado en la combinación de estos dos. Estos tres datos son enviados desde un módulo de infraestructura hacia el medio físico de transporte (2), en este caso el cable de alimentación 120VAC del domicilio.

A través del medio físico de transporte, todos los módulos escuchan estos tres datos enviados por el módulo anterior, analizando en base a la información recibida si el comando iba dirigido hacia ellos o no, luego que alguno de los módulos de infraestructura reconoce que los datos iban dirigidos a él, este transfiere el dato o comando enviado desde el módulo central hacia el dispositivo periférico (3), el cual realiza alguna acción predefinida según el comando que recibió. El siguiente paso es la emisión de un dato de respuesta desde el dispositivo periférico hacia el módulo de infraestructura (4), el cual recibe este dato y calcula un valor de seguridad específico en base a este. Luego el módulo de infraestructura envía a través del medio físico el dato proveniente del dispositivo periférico y el dato de seguridad (5).

Por último, el módulo de infraestructura conectado con el módulo central recibe el dato enviado desde el dispositivo periférico junto con el dato de seguridad, analizando ambos para transferir al módulo central únicamente el dato del dispositivo periférico, terminando con un ciclo de comunicación (6).

B. DEFINICIÓN Y PROGRAMACIÓN

El intercambio de información entre dispositivos es un proceso en el cual cada elemento debe tener definida la manera de enviar y recibir datos. Esto debe establecerse para permitir el correcto flujo de la información, lo cual se lleva a cabo con la implementación de reglas específicas para cada elemento involucrado en la comunicación.

Las reglas definidas para esta comunicación deben ser capaces de permitir la emisión y recepción de información para la comunicación entre módulos. Este proceso debe llevarse a cabo de manera ordenada y adecuándose a las características de cada módulo.

1. Diseño experimental. La idea de definir un protocolo de comunicación entre módulos fue permitir el flujo de información entre estos de manera sencilla, realizándose todo el procesamiento y manejo de datos en un microcontrolador implementado en el módulo de infraestructura de datos.

Tal como se observa en la Figura No.7, la tarea del protocolo de comunicación consiste en proporcionar enlaces entre los módulos del megaproyecto. El primer enlace, denominado “Emisor”, comunica el módulo central con el módulo de infraestructura de comunicación. El segundo enlace, denominado “Receptor”, comunica el módulo de infraestructura de comunicación con los distintos módulos periféricos.

El ciclo completo de comunicación inicia en el Emisor, el cual envía los datos de interés al Receptor, este transfiere dichos datos al dispositivo deseado y luego envía un dato de retroalimentación hacia el Emisor.

Parte del proceso de diseño consistió en analizar las distintas formas de comunicación entre dispositivos que posee un microcontrolador PIC, entre las cuales destaca la comunicación *EUSART* por su simplicidad y manejo intuitivo.

Esto permitió proponer distintos algoritmos tomando en cuenta que los módulos de infraestructura ejecutarán las funciones de recibir o enviar datos de manera exclusiva, *half-duplex*, realizando únicamente una de estas dos funciones según sea requerido.

Es importante recalcar que el envío de datos a través del medio físico debe realizarse tomando en cuenta que los datos pueden corromperse, por lo cual es necesario implementar algún método que asegure la integridad de los datos al momento de su recepción, descartando los mismos en dado caso se identifique que estos fueron alterados.

El diseño de los algoritmos para los elementos Emisor y Receptor se realizaron buscando cumplir con el ciclo de comunicación planteado anteriormente, esto se llevó a cabo describiendo los algoritmos por medio de diagramas de flujo.

El primer diagrama de flujo establecido fue el del elemento Emisor ya que es en este que la comunicación inicia. Basándose en el diagrama de flujo, se escribió un programa en el lenguaje mikroC, procediendo a trasladar este al microcontrolador para probar su funcionamiento.

Luego de realizar las pruebas, se realizaron modificaciones paralelas tanto en el diagrama de flujo como en el programa del microcontrolador, llevando este proceso hasta que se obtuvieran los resultados necesarios para cumplir con la etapa del ciclo de comunicación.

El siguiente paso fue llevar a cabo el mismo proceso de diseño de algoritmos para el elemento Receptor, realizando modificaciones en la lógica del mismo hasta que su funcionamiento fuera el esperado.

Por último se buscó determinar los tiempos para un ciclo de comunicación completo en el mejor y peor de los casos de una transmisión efectiva.

2. Resultados

FIGURA NO. 75. DIAGRAMA DE FLUJO DE ELEMENTO EMISOR.

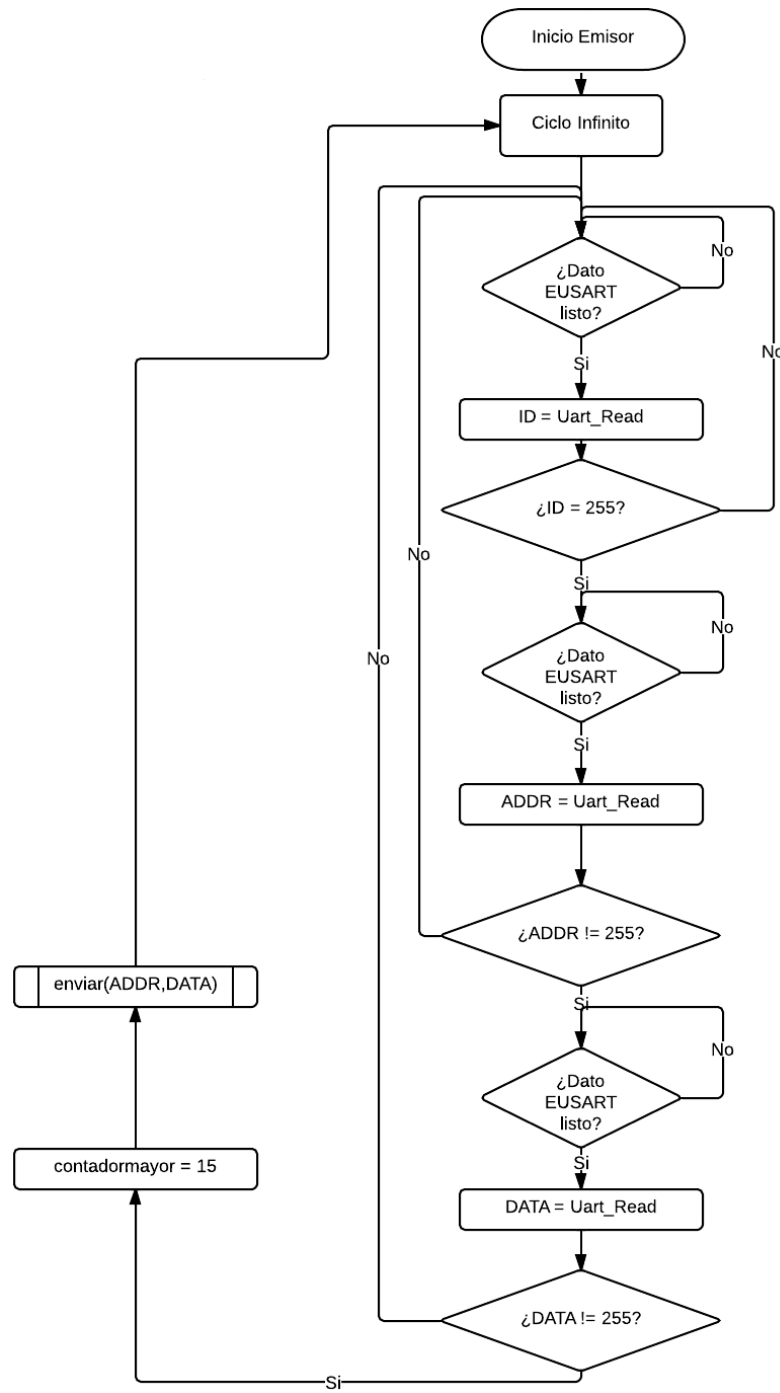


FIGURA NO. 76. DIAGRAMA DE FLUJO DE FUNCIÓN ENVIAR.

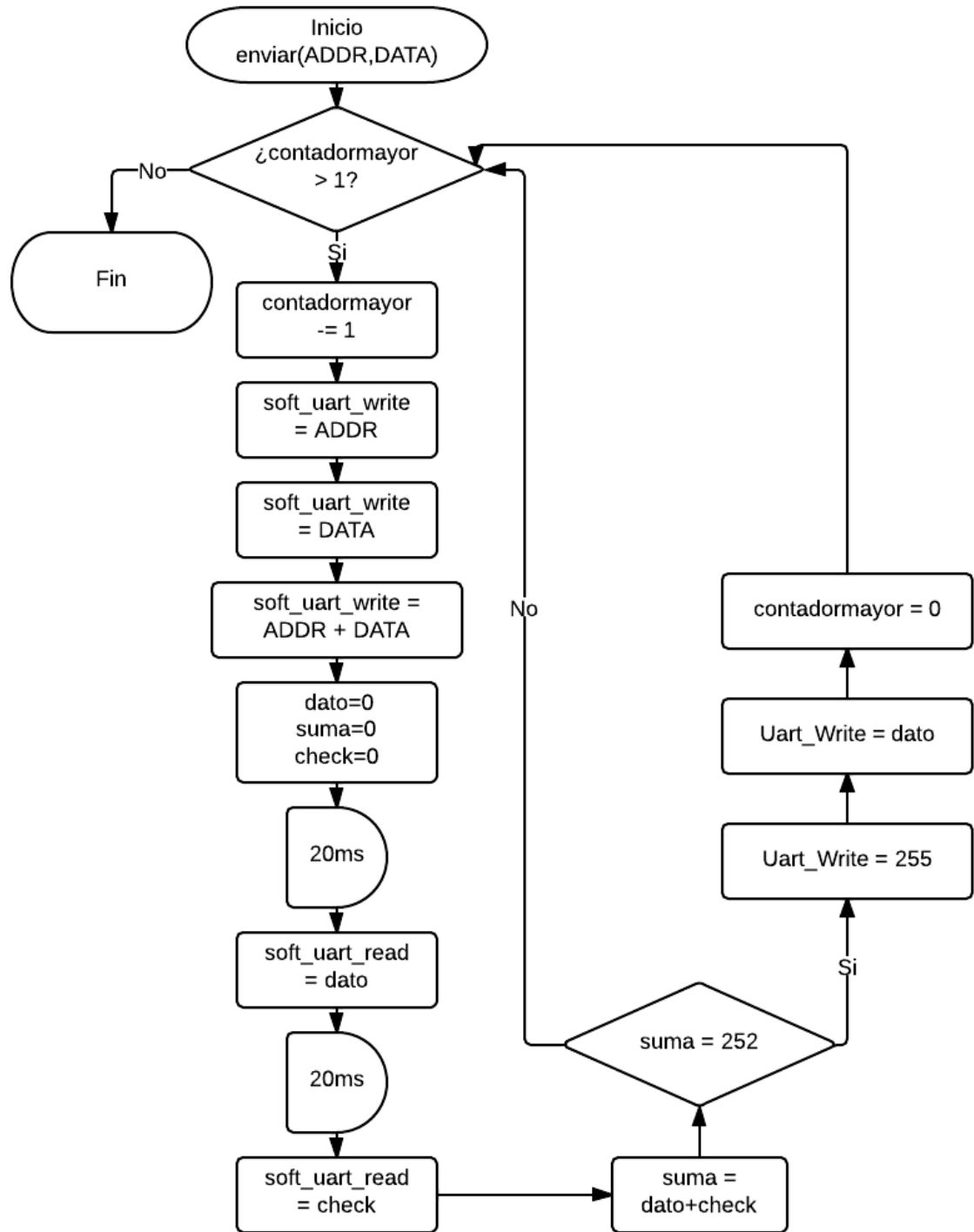


FIGURA NO. 77. DIAGRAMA DE FLUJO DE PROCESO DE DIRECCIONAMIENTO.

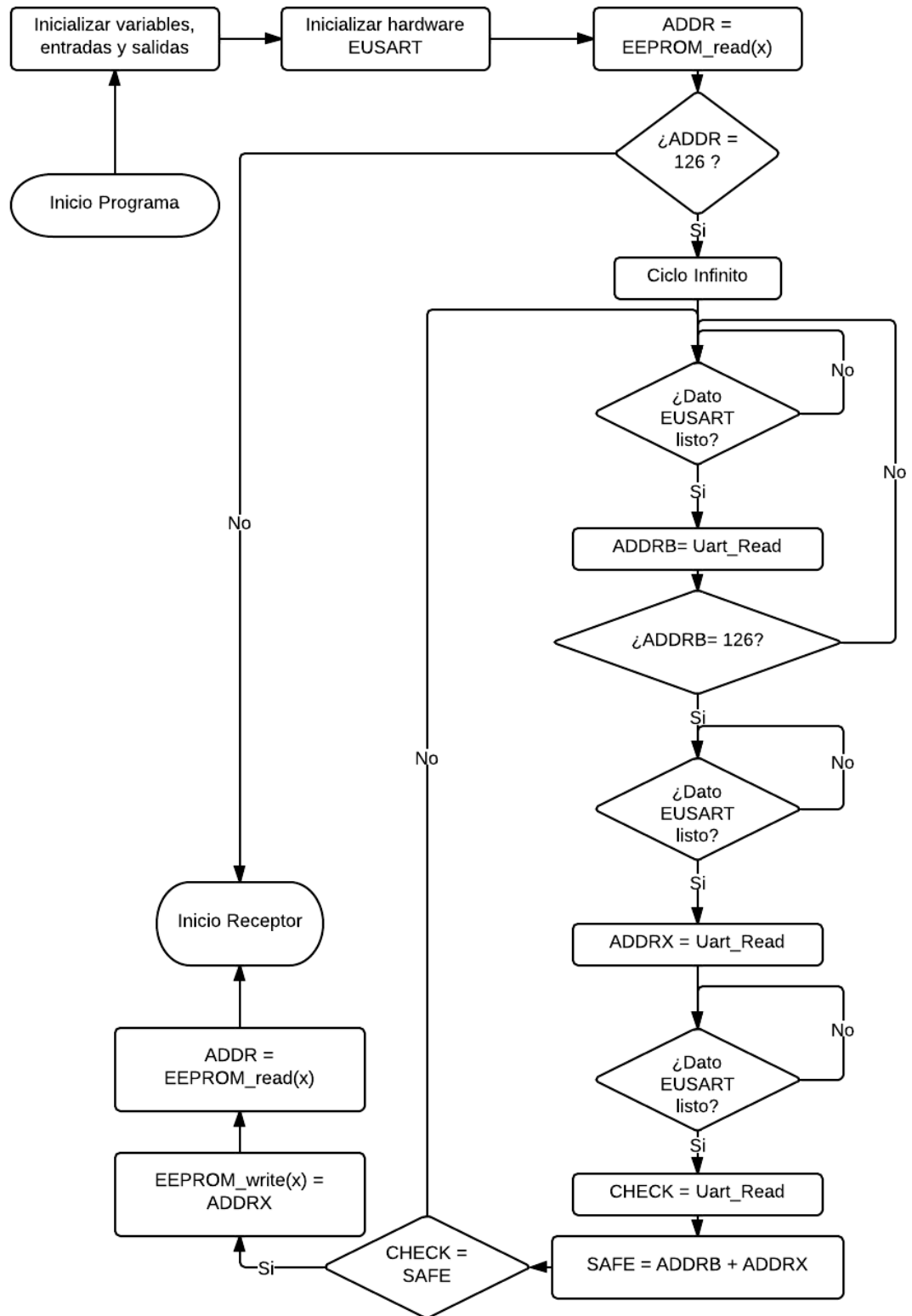


FIGURA NO. 78. DIAGRAMA DE FLUJO DE ELEMENTO RECEPTOR.

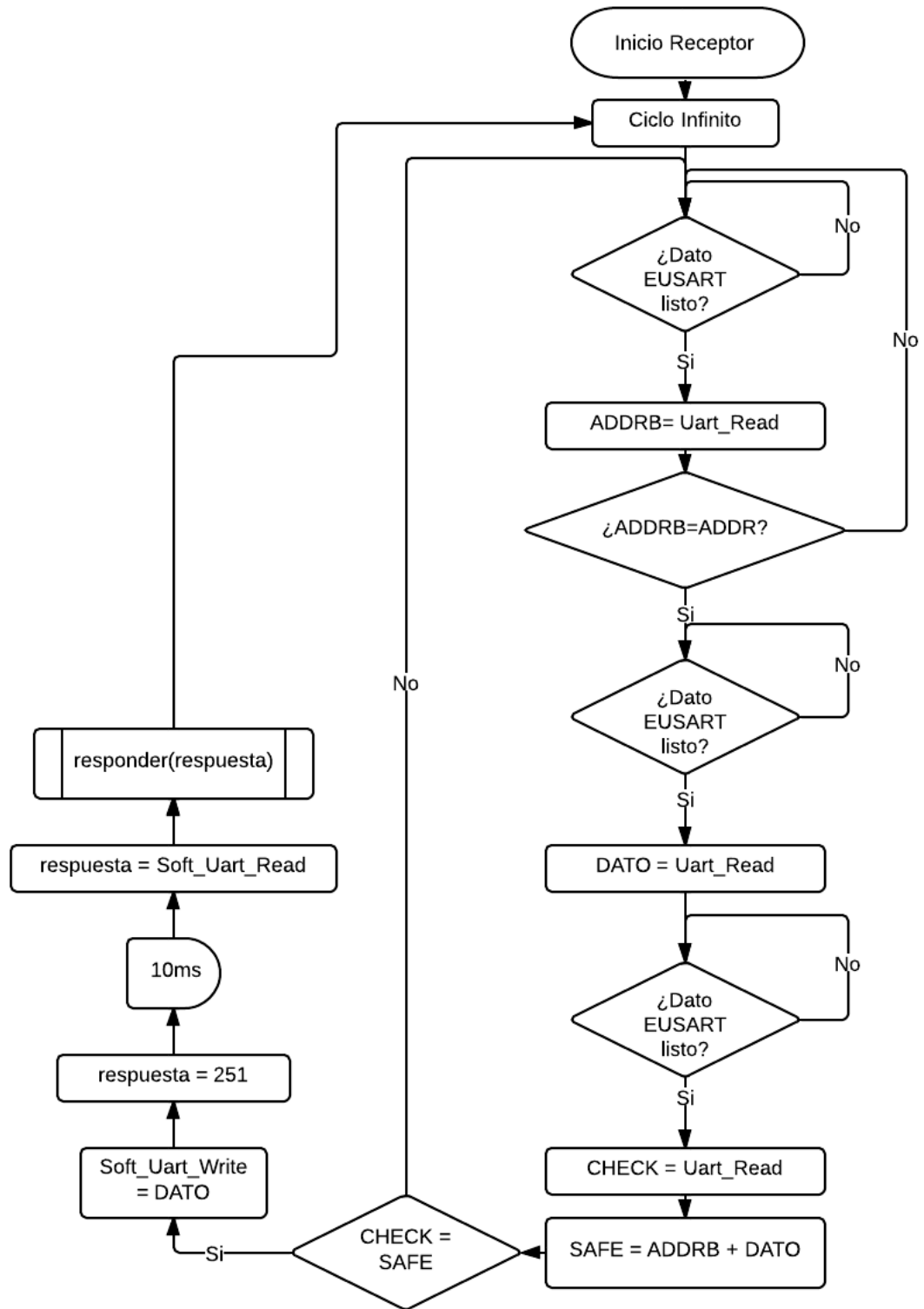


FIGURA NO. 79. DIAGRAMA DE FLUJO DE FUNCIÓN ENVIAR.

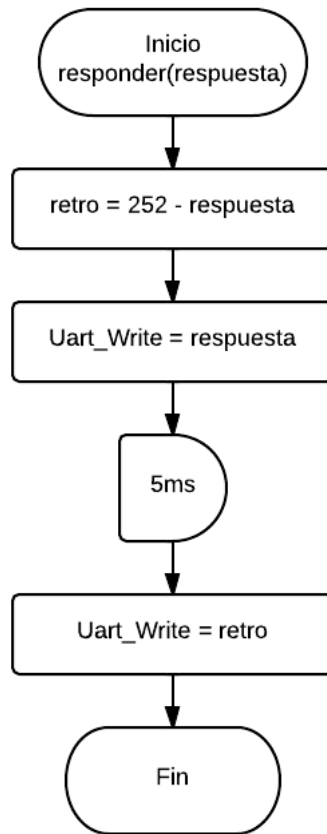


TABLA NO. 18. BAUD RATE DEFINIDOS PARA ELEMENTOS EMISOR Y RECEPTOR

	Elemento Emisor (baud rate)		Elemento Receptor (baud rate)	
	<i>Módulo central</i>	<i>Módulo infraestructura</i>	<i>Módulo periférico</i>	<i>Módulo infraestructura</i>
Microcontrolador	1200	1100	10417	1100

TABLA NO. 19. TIEMPOS PARA CICLO DE COMUNICACIÓN COMPLETA

<i>Intento de ciclos</i>	Módulo central		M. infraestructura de c.		<i>Tiempo total</i>
	<i>Emisión</i>	<i>Recepción</i>	<i>Emisión</i>	<i>Recepción</i>	
1	0.02	0.013	0.0218	0.04	0.0948
15	0.02	0.195	0.327	0.04	0.582

3. Discusión. La comunicación por EUSART puede llevarse a cabo de manera síncrona o asíncrona, dependiendo de las necesidades del sistema. Para este protocolo, se decidió realizar la comunicación de manera asíncrona, permitiendo que el dispositivo esté preparado para recibir datos en cualquier momento sin necesidad de depender de acontecimientos externos. Esto es de suma importancia ya que la transferencia de datos se dará cuando el módulo central así lo indique, por lo cual el dispositivo emisor debe adecuarse a esta necesidad.

Entre las características físicas del módulo *EUSART* se encuentra la función receptora, la cual es capaz de almacenar dos datos en un buffer de entrada hasta que éstos sean leídos. Esta función puede ser tanto beneficiosa como perjudicial ya que implica la recepción de datos aun cuando no se desee recibir ninguno.

Esta característica específica del módulo *EUSART* resultó ser un inconveniente al momento de realizar la comunicación con los módulos de infraestructura y comunicación. El motivo de esto fue que la conexión entre el microcontrolador que contiene el protocolo y el módulo de infraestructura se realiza con un chip de entrada y salida modulador de señales (TDA 5051A), el cual por sus características físicas, las cuales no son de relevancia en el análisis de esta investigación, devuelve al microcontrolador el mismo dato que éste le envía para la modulación, ocasionando que el buffer de entrada del *EUSART* se llene al momento de enviar datos a través del modulador.

Además del módulo *EUSART*, el cual es un módulo implementado en hardware, los microcontroladores son capaces de realizar tareas similares a las del hardware por medio de software, por lo cual es posible utilizar un módulo implementado en software para la comunicación serial asíncrona, denominándose a este nuevo módulo *SoftUart*.

Debido a que los módulos en software deben programarse, su utilización no es la primera opción al momento de buscar desarrollar aplicaciones de

comunicación con un microcontrolador. En el caso de un *SoftUart*, este únicamente es capaz de recibir datos cuando se le indica específicamente que lo haga, bloqueando indefinidamente el curso del programa principal hasta que este reciba un dato o una interrupción termine con dicha rutina. Esto puede considerarse como una desventaja pero para los requerimientos de este proyecto las características del *SoftUart* permiten efectuar la comunicación *half-duplex* sin presentarse el problema antes mencionado con el *EUSART* de hardware, siendo esto una ventaja.

El ciclo de comunicación inicia en el emisor, este debe ser capaz de enviar los datos solicitados por el módulo central a través del modulador y luego, después de un tiempo determinado, esperar a recibir un dato de respuesta proveniente de los dispositivos periféricos a través del modulador. Este dato será transmitido al módulo central a manera de retroalimentación, el proceso descrito se puede observar en las Figuras 75 y 76.

La siguiente etapa del ciclo de comunicación está a cargo del receptor, el cual debe estar preparado para recibir los datos provenientes del módulo central a través del modulador, luego de analizar los datos recibidos el receptor envía el dato de comando al dispositivo periférico con el cual se encuentre conectado, esperando a recibir un dato de retroalimentación del mismo para enviárselo al módulo central a través del modulador. Este proceso se puede observar en las Figuras 77 y 78.

La implementación de seguridad en el protocolo para asegurar la integridad de datos en su transferencia por el medio físico, se llevó a cabo por medio de *checksums*, implementando la misma técnica de manera distinta para los elementos emisor y receptor. En el caso del emisor, el protocolo debe recibir los datos de dirección y comando, los cuales suma para obtener un valor específico según la combinación de estos, el resultado de esta operación se denomina seguridad. Luego de realizar esto, el protocolo envía a través del medio físico la dirección del dispositivo, el dato de comando y el valor de seguridad, los cuales son interpretados en este orden al momento de la

recepción, por parte del Receptor, para comprobar que la suma de los primeros dos datos recibidos sea equivalente al valor del tercer dato. Este procedimiento se puede observar en las Figuras 76, 77 y 78.

En el caso del *checksum* implementado en el receptor, el protocolo recibe un dato de retroalimentación proveniente del dispositivo periférico luego de haber entregado al mismo un dato de comando. El protocolo utiliza este valor de retroalimentación para restarlo al número 252, obteniendo como resultado un nuevo dato de seguridad. El siguiente paso es enviar a través del medio físico el dato de retroalimentación seguido por el dato de seguridad, los cuales son interpretados por el elemento emisor para poder entregar correctamente el dato de retroalimentación al Módulo Central. Esto se puede observar en las Figuras 75 y 78.

En dado caso la recepción de datos indique que la integridad de los mismos fue comprometida, situación en la que el dato de seguridad no corresponde a los datos recibidos previamente, el ciclo de comunicación se repetirá un número limitado de veces hasta que esto se logre. Esto se puede dar en cualquiera de las etapas del ciclo de comunicación, por lo que el sistema debe estar preparado en caso esto suceda, el procedimiento se observa en las Figuras 75 y 76.

El direccionamiento de dispositivos nuevos agregados a la red se implementó asignando un valor de dirección predeterminado en memoria, la cual es la misma para todos los dispositivos “nuevos”, esta dirección es el valor 126. Cuando el dispositivo se conecta a la red, el Módulo Central envía un dato dirigido hacia la dirección 126, el cual es interpretado únicamente por el modulo nuevo, el cual luego de analizar la integridad de los datos recibidos, interpreta el valor del dato recibido cómo su nueva dirección, reescribiendo esta nueva dirección en la casilla de memoria designada a esto. El proceso de direccionamiento se puede observar en la Figura 77.

La configuración del baud rate para las distintas comunicaciones entre dispositivos se llevó a cabo según las características del Módulo de Infraestructura de Comunicación, siendo necesario limitar la velocidad según el tiempo de respuesta del modulador de señales, 1200 baudios por segundo. Los baud rates implementados en la programación de los microcontroladores se pueden observar en la Tabla 18.

Analizando la operación del protocolo desde el elemento emisor, en base a la culminación de un ciclo de comunicación exitoso, se obtuvieron los tiempos según el número de intentos necesarios para que la información llegara a su destino. Cuando el ciclo se completaba en el primer intento, mejor de los casos, el tiempo de operación fue de 0.095 segundos. Cuando la información llegaba en el intento número 15, peor de los casos, el tiempo fue de 0.58 segundos. Ambos tiempos son menores que el tiempo determinado para un ciclo de comunicación completo usando el protocolo X 10, tomando en cuenta que este protocolo de comparación únicamente considera la emisión de datos de manera unidireccional, sin implementar ningún método que asegure la integridad de los datos. Esta información se observa en la Tabla 19.

VIII. PERIFÉRICOS DE AMBIENTE

A. CONTROL DE DISPOSITIVO IR

1. Diseño experimental. Los pasos que se siguieron para realizar el control de dispositivos IR fueron los siguientes:

- Determinar los requerimientos del módulo.
- Estudiar los diferentes protocolos de comunicación utilizados en el control de dispositivos IR.
- Determinar cuáles son los protocolos, dispositivos y comandos más utilizados para el control de equipo audiovisual (que incluye televisión y equipos de sonido).
- Seleccionar los componentes que cumplieran con las necesidades del módulo.
- Diseñar el circuito para integrar estos componentes.
- Desarrollar el programa para el funcionamiento adecuado del módulo, incluida su comunicación con las diferentes partes del Megaproyecto.

El control remoto de televisión (y en general de la mayoría de electrodomésticos) se basa en la transmisión y recepción de señales IR con una longitud de onda de 950 nanómetros. El ojo humano no puede percibir esta longitud de onda de manera que las señales IR no se puede detectar visualmente. Se utiliza el mando a distancia para emitir la señal IR modulada mediante un LED infrarrojo. Este diodo emite pulsos infrarrojos en forma codificada a una frecuencia de 38 kHz generalmente. Al pulsar cada botón en el mando a distancia, se emiten señales IR en una forma codificada en particular. Esto ayuda a controlar las distintas funciones del dispositivo con precisión. El dispositivo a controlar incluye un receptor de infrarrojos, el cual es sensible sólo a señales IR moduladas a una frecuencia cercana a los 38kHz.

El control de dispositivos IR debe ser capaz de enviar los códigos IR necesarios para el control de los diferentes dispositivos electrónicos como televisión y equipo de sonido. Además, debe ser capaz de almacenar en memoria los códigos directamente de los controles originales de estos dispositivos, debido a las limitantes del proyecto que no permiten tener todos los códigos de todos los dispositivos en una base de datos. Para realizar todas sus funciones el sistema debe comunicarse adecuadamente con el Módulo de Infraestructura de Comunicación, el cual será el encargado de guiar la comunicación entre los módulos periféricos y el módulo central. La comunicación con el Módulo de Infraestructura de Comunicación se realizará utilizando comunicación serial.

Debido a la gran cantidad de protocolos utilizados para la comunicación IR de dispositivos que utilizan control remoto, se limitó a implementar únicamente los protocolos más utilizados en los dispositivos que se utilizan en Guatemala. Se buscó poder controlar dispositivos de las principales marcas, como Panasonic, LG, Toshiba, Sony, Samsung y Philips, por lo que los protocolos implementados fueron los siguientes:

- Protocolo SONY SIRC
- Protocolo NEC
- Protocolo Panasonic
- Protocolo RC5 de Philips
- Protocolo Samsung.

Además, se implementó un algoritmo para enviar códigos IR basado únicamente en un arreglo de datos numéricos que indican los tiempos correspondientes de encendido y apagado, de manera que es posible enviar una gran variedad de códigos a pesar de que el protocolo original no se encuentre implementado en el control desarrollado.

Para realizar el control de dispositivos IR, primero se procedió a seleccionar los componentes básicos que se utilizaron para lograr los objetivos del módulo:

- Selección del microcontrolador.

Se utilizó el microcontrolador ATMEGA328P de la familia ATMEL. Se seleccionó este microcontrolador debido principalmente a su capacidad de almacenamiento. Este microcontrolador cuenta con una memoria Flash de 32 kB para el almacenamiento del código, 2 kB de SRAM y 1 kB de *EEPROM*, lo cual es esencial para almacenar los códigos necesarios para el funcionamiento del control de dispositivos IR.

Adicional a esto, el microcontrolador posee suficientes entradas y salidas para los dispositivos de entrada y salida utilizados, y posee tres *timer*/contadores que permiten obtener PWMs con diferentes frecuencias, lo cual es necesario para la transmisión por IR. Se utilizó un cristal de 16MHz para generar el tren de pulsos para el reloj del microcontrolador, dado que esta es su frecuencia máxima de operación y permite obtener las frecuencias necesarias para la operación del sistema.

- Selección del receptor IR.

Para determinar qué receptor IR utilizar, primero se identificó el rango de frecuencia de las señales que se iban a recibir. Con base en los protocolos investigados, se determinó que la mayoría utilizan frecuencias portadoras entre los 36 y 40 kHz, siendo las más comunes 38 y 40 kHz. Por lo tanto se buscó un receptor cuya frecuencia central fuera de 38 kHz. Se seleccionó el TSOP1738 de la familia TSOP17XX de Vishay (XX corresponde a la frecuencia central aproximada en kHz), el cual es un receptor miniatura utilizado para sistemas de control remoto por infrarrojo y tiene las siguientes características:

- Frecuencia de paso de banda de 38kHz
- Bajo consumo de energía (menos de 50mW)
- Alta inmunidad a la luz ambiental
- La salida utiliza lógica *Active-Low* (salida negada)
- Filtro interno para demodular la señal

FIGURA NO. 80. PINES DEL CIRCUITO INTEGRADO TSOP1738

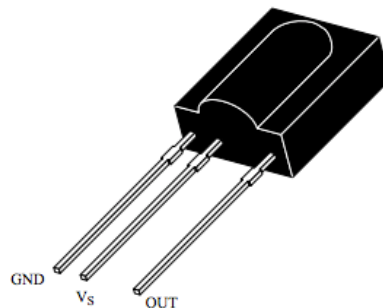
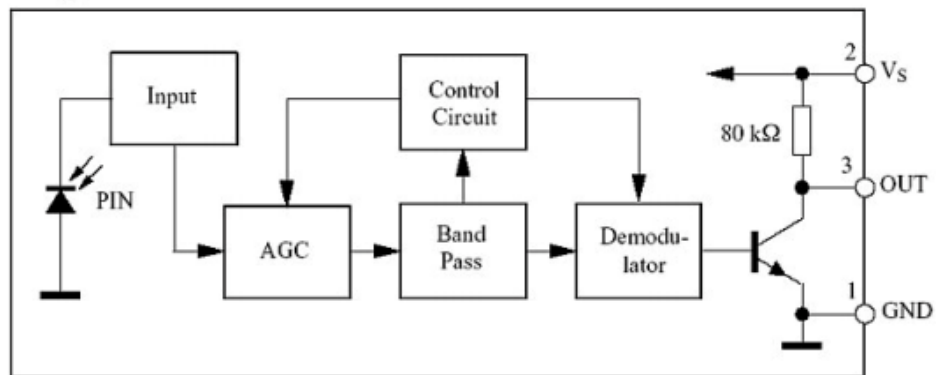


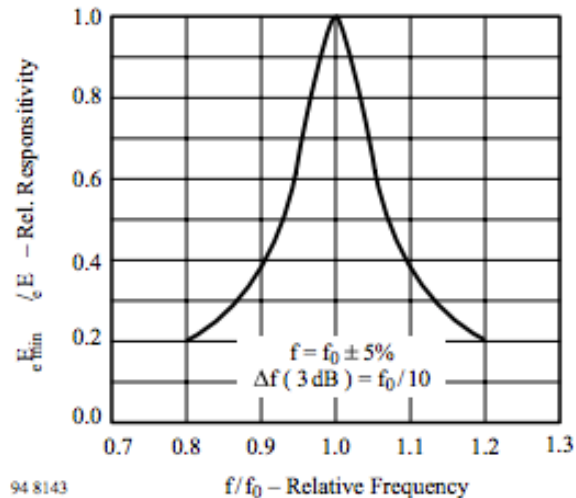
FIGURA NO. 81. DIAGRAMA DE BLOQUES DE LOS INTEGRADOS FAMILIA TSOPXX



Es importante mencionar que, dado que se busca poder recibir códigos de diferentes protocolos y no todos funcionan con la misma frecuencia portadora, el receptor seleccionado debe tener un rango amplio de frecuencia de operación. En la Figura 16 se muestra cómo se comporta el receptor TSOP1738 dependiendo de la frecuencia, siendo 38kHz la frecuencia central. Con base en

esta gráfica se puede observar que el receptor será capaz de recibir señales que tengan una frecuencia portadora de 36 y 40 kHz sin ningún problema. Esto lo hace compatible con un gran número de mandos a distancia de electrodomésticos.

FIGURA NO. 82. RESPUESTA RELATIVA DEL TSOP1738 DEPENDIENDO DE LA FRECUENCIA RECIBIDA



Además del integrado TSOP1738, existen otras alternativas de otros fabricantes que pueden ser utilizados en su defecto. En la siguiente tabla se muestra algunas de las diferentes alternativas, todas ellas con una frecuencia central de 38 kHz.

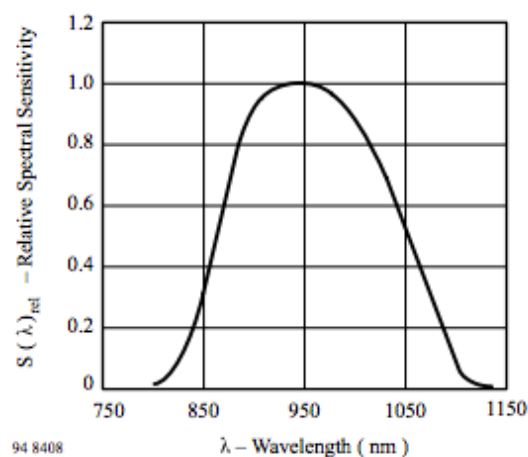
TABLA NO. 20. RECEPTORES INFRARROJO CON FRECUENCIA CENTRAL DE 38 KHZ

Nombre del fabricante	Nombre del integrado
Sharp	IS1U60
Sharp	IS1U621
Panasonic	PNA4602M
Panasonic	PNA4612M
Siemens	SFH5110-38
LiteOn	LTM-97DS-38
Temic	TFMS 5380
Kodenshi	PIC 26043SM (37.9kHz)
Vishay	TSOP1238
Vishay	TSOP1838
AZ Displays	HRM3800

- Selección del emisor IR.

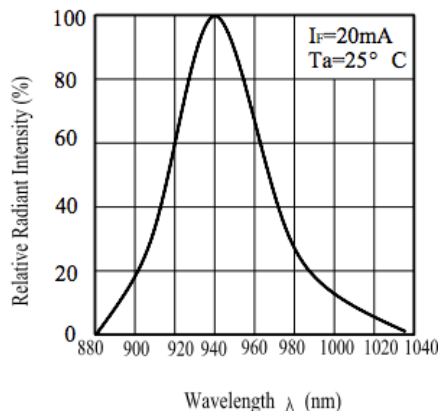
Se sabe que la radiación infrarroja abarca un rango relativamente grande de longitudes de onda en el espectro electromagnético, y los diferentes receptores infrarrojos tienen mayor sensibilidad a ciertas longitudes de onda de radiación infrarroja. Por ejemplo, en la siguiente figura se muestra una comparación entre la longitud de onda de la señal IR y la sensibilidad del receptor TSOP1738, y se puede observar que se obtiene una mayor sensibilidad para una longitud de onda aproximada de 950 nm. Por esta razón, se buscó un LED IR que tuviera su máxima emisión a una longitud de onda cercana a los 950 nm.

FIGURA NO. 83. SENSIBILIDAD DEL RECEPTOR VS LONGITUD DE ONDA RECIBIDA POR EL TSOP1738



Se utilizó un LED infrarrojo de 5 mm del fabricante Everlight Electronics, el cual es muy utilizado para mandos a distancia y cámaras de visión nocturna. El LED IR seleccionado tiene un pico de longitud de onda de 940 nm, como se muestra en la siguiente figura, lo que garantiza que es apto para que los diferentes dispositivos reciban esta señal. Este LED IR puede soportar una corriente continua de 100mA, y para aplicaciones de mandos a distancia soporta pulsos de hasta 1000mA, para lograr un mayor alcance.

FIGURA NO. 84. INTENSIDAD DE RADIACIÓN VS LONGITUD DE ONDA EMITIDA POR EL LED IR SELECCIONADO



2. Resultados. Se creó un programa para el microcontrolador ATMEGA328P con el objetivo de controlar los diferentes dispositivos IR. El programa realiza las siguientes funciones:

- Recibir datos del Módulo Central: El microcontrolador recibe información enviada por el Módulo Central y codificada por el Módulo de Comunicación de los comandos enviados por el usuario desde el teléfono móvil para saber qué código IR debe enviar. Para la recepción se verifica constantemente el puerto serial para verificar si se ha recibido un dato y luego se verifica el dato recibido para realizar la acción correspondiente.
- Enviar códigos IR: El microcontrolador envía los códigos digitales al emisor infrarrojo para la transmisión de los códigos IR. Estas señales se envían a la frecuencia modulada correspondiente dependiendo del protocolo y utilizando un ciclo de trabajo adecuado.
- Recepción y almacenamiento de códigos IR: El microcontrolador recibe, en uno de sus puertos configurado como entrada, las señales recibidas por el receptor IR utilizado y las decodifica para almacenarlas en memoria para su posterior utilización.

Además, en el programa se incluyen las funciones que permiten implementar los diferentes protocolos de comunicación IR que se decidió incluir

en el control de dispositivos IR. Se estudió cada uno de los protocolos y a continuación se muestran los diagramas de flujo simplificados para el envío de códigos en cada uno de los protocolos.

Para realizar los diagramas de flujo se denominará *Marca* al estado en el que la señal IR es modulada a la frecuencia correspondiente, es decir que se varía el estado ON y OFF del transmisor IR a cierta frecuencia, y *Espacio* al estado en el que el transmisor IR se encuentra apagado entre dos *Marcas*, es decir que no se emite luz infrarroja durante un *Espacio*, así como se muestra en la Figura 85.

FIGURA NO. 85. DEFINICIÓN DE MARCA Y ESPACIO

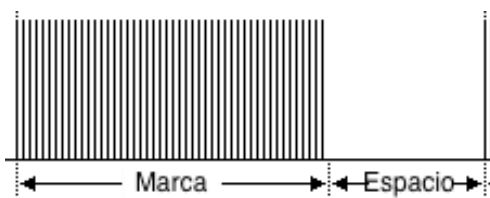


FIGURA NO. 86. DIAGRAMA DE FLUJO PROTOCOLO SAMSUNG.

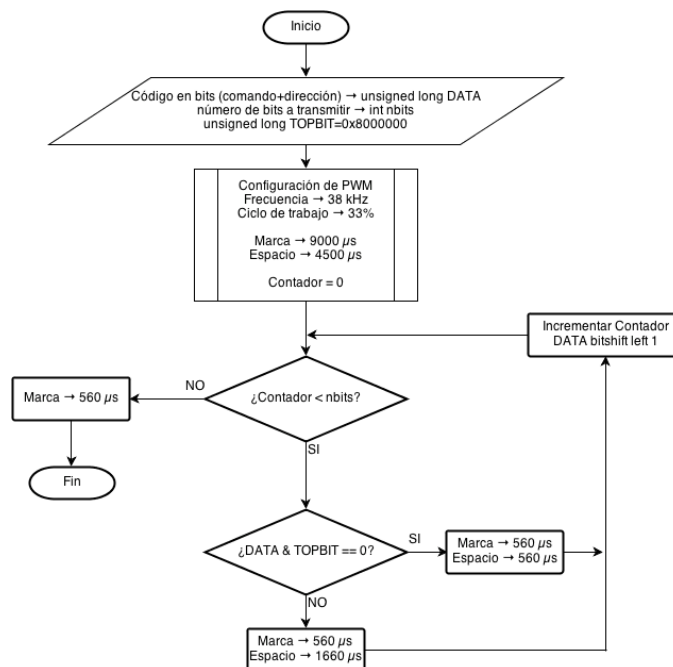


FIGURA NO. 87. DIAGRAMA DE FLUJO PROTOCOLO SONY

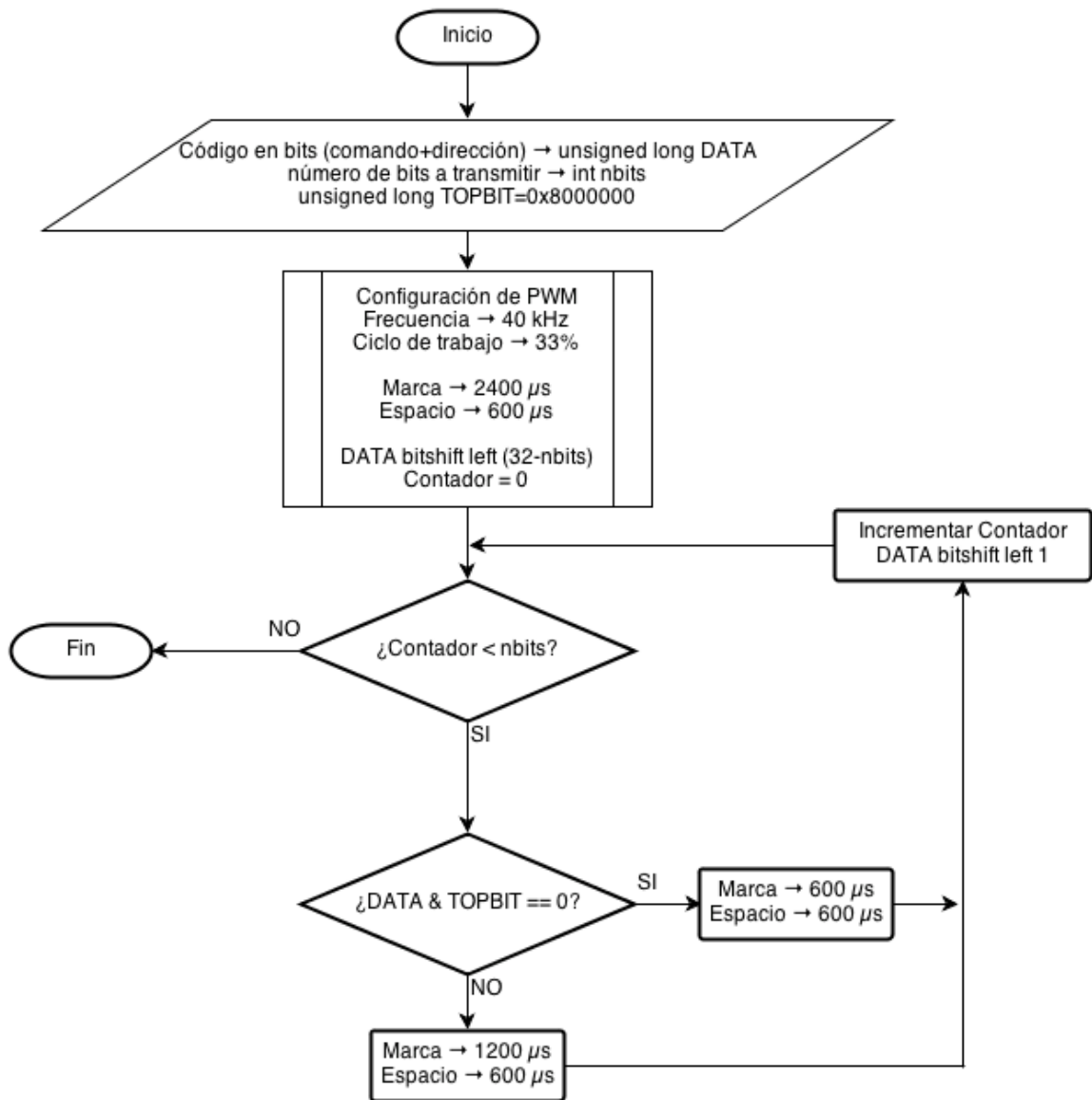


FIGURA NO. 88. DIAGRAMA DE FLUJO PROTOCOLO NEC.

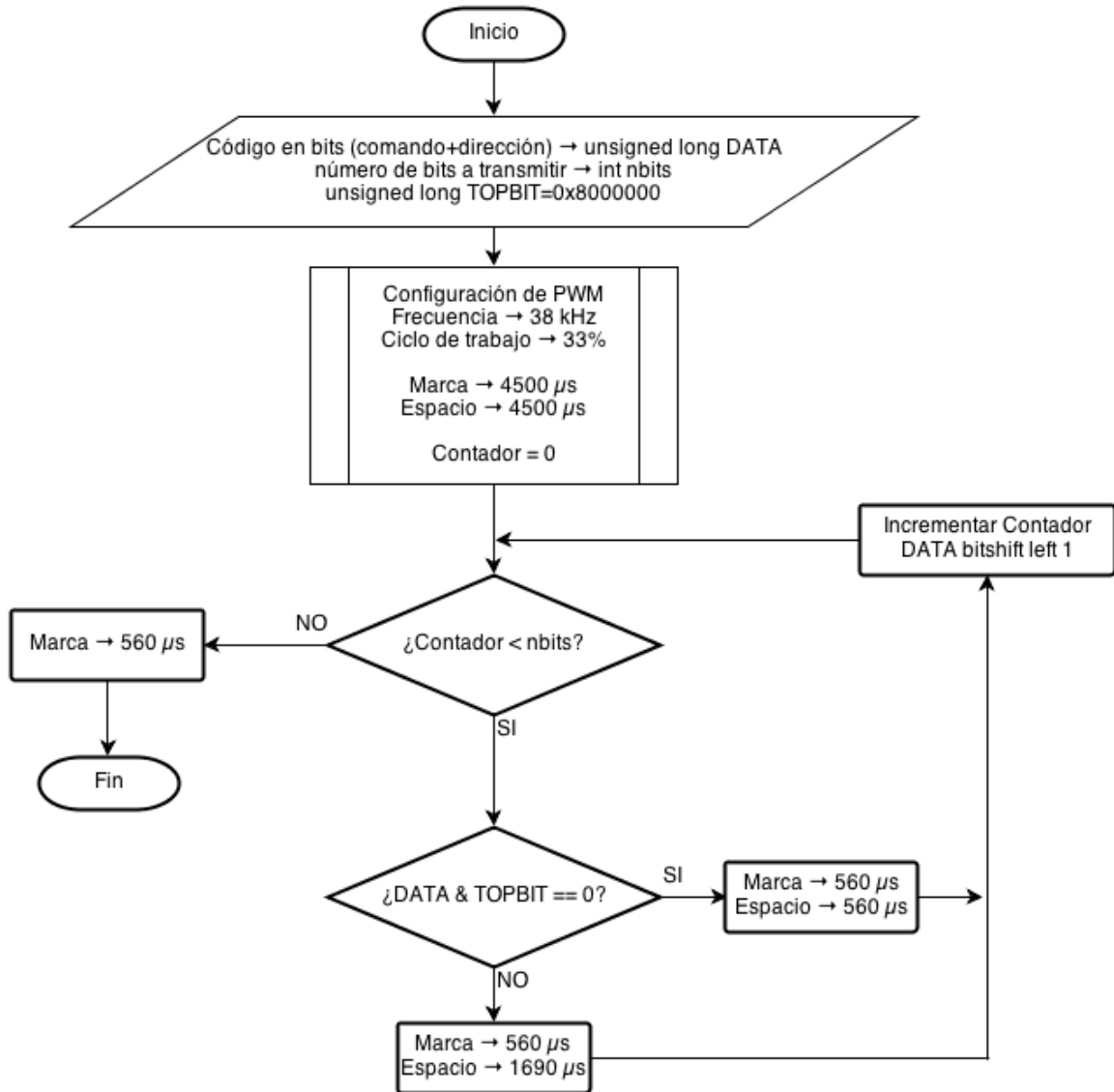


FIGURA NO. 89. DIAGRAMA DE FLUJO PROTOCOLO RC5.

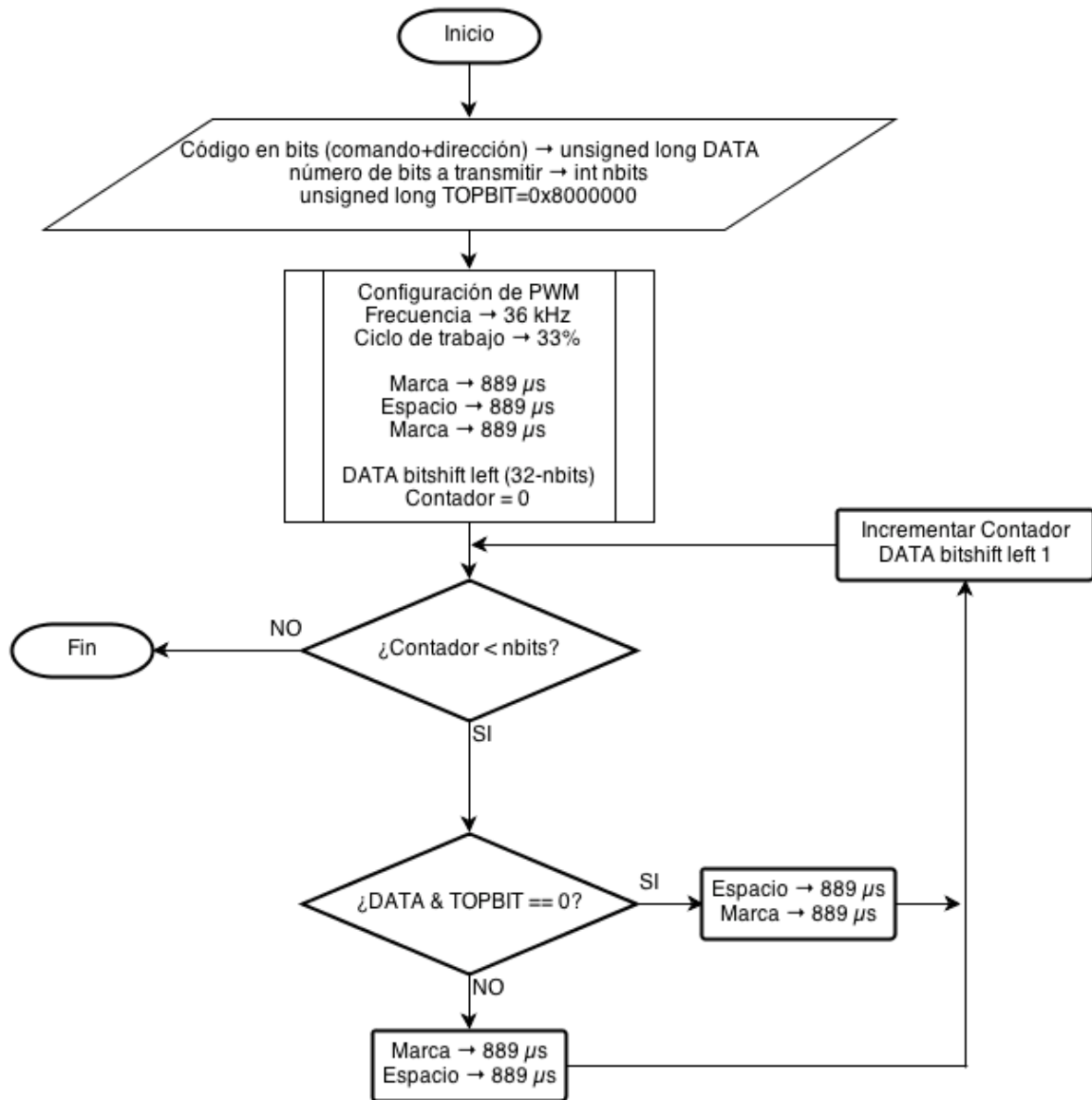


FIGURA NO. 90. DIAGRAMA DE FLUJO PROTOCOLO PANASONIC

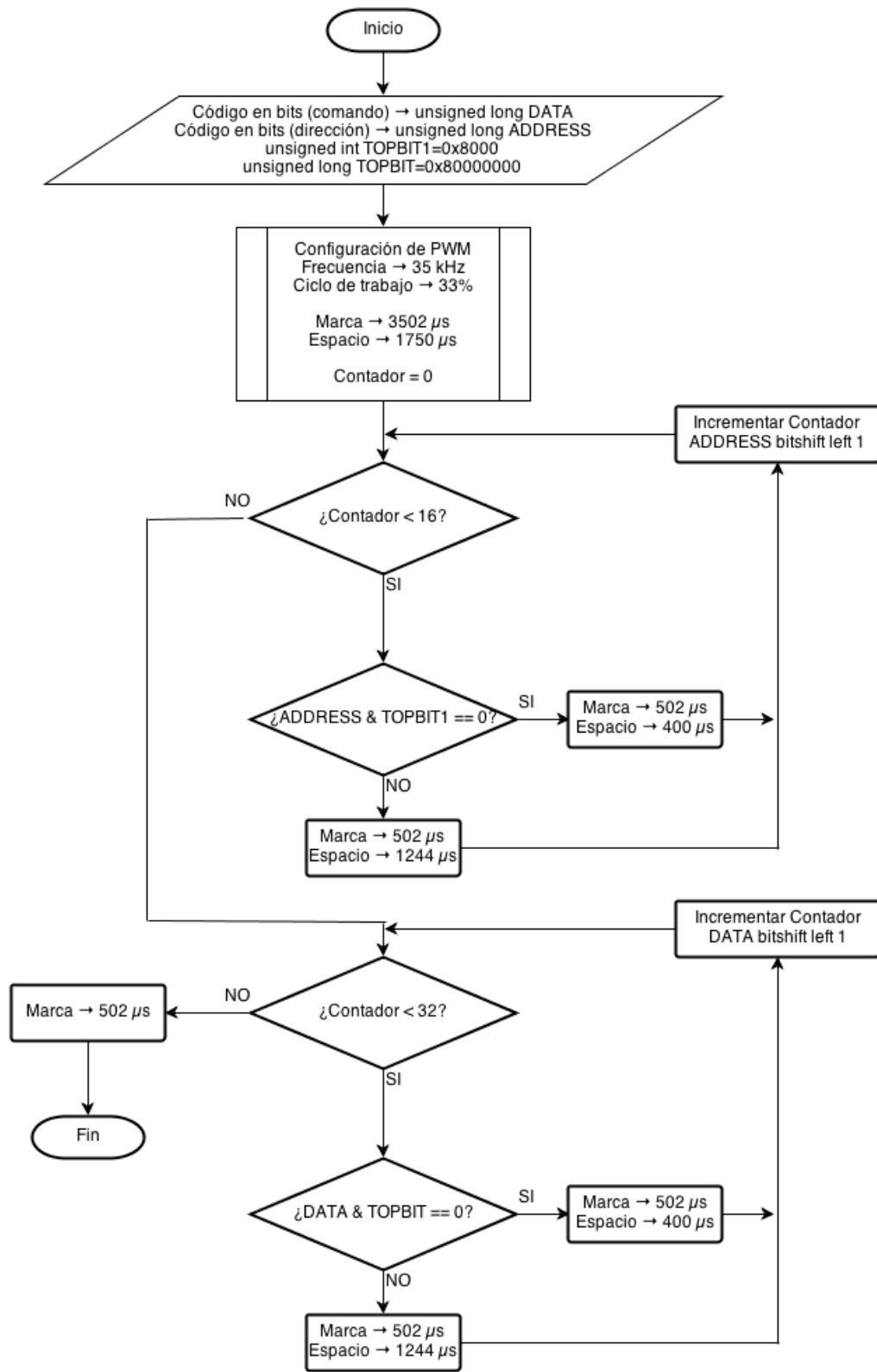


TABLA NO. 21. COMANDOS DEL CONTROL DE DISPOSITIVOS

Byte Recibido	Tipo de dispositivo	Comando	Byte Recibido	Tipo de dispositivo	Comando
0	TV	0	42	DVD	Mute
1	TV	1	43	DVD	Subtitles
2	TV	2	44	DVD	Up
3	TV	3	45	DVD	Down
4	TV	4	46	DVD	Right
5	TV	5	47	DVD	Left
6	TV	6	48	DVD	Enter
7	TV	7	64	Audio	Power, ON/OFF
8	TV	8	65	Audio	Volume +
9	TV	9	66	Audio	Volume -
10	TV	Input/ TV video	67	Audio	Mute
11	TV	Channel Up	68	Audio	Tape
12	TV	Channel Down	69	Audio	CD
13	TV	Volume +	70	Audio	Tuner
14	TV	Volume -	71	Audio	Function
15	TV	Mute	72	Audio	Play
16	TV	Power, ON/OFF	73	Audio	Stop
17	TV	Jump, Last	74	Audio	Pause
32	DVD	Power, ON/OFF	75	Audio	fwd >>
33	DVD	Volume +	76	Audio	rev <<
34	DVD	Volume -	77	Audio	next >>
35	DVD	Play	78	Audio	prev <<
36	DVD	Stop	90	Memoria	M1
37	DVD	Pause	91	Memoria	M2
38	DVD	fwd >>	92	Memoria	M3
39	DVD	rev <<	93	Memoria	M4
40	DVD	next >>	94	Memoria	M5
41	DVD	prev <<	127	Comunicacion	Tipo de dispositivo

Se realizó una comparación entre los códigos emitidos por el control desarrollado y los códigos emitidos por los controles originales de los dispositivos. Solo se incluyeron los códigos de los dispositivos en los que se pudo probar el funcionamiento del control desarrollado.

FIGURA NO. 91. CÓDIGO ON/OFF PARA UNA TELEVISIÓN SONY, CONTROL ORIGINAL

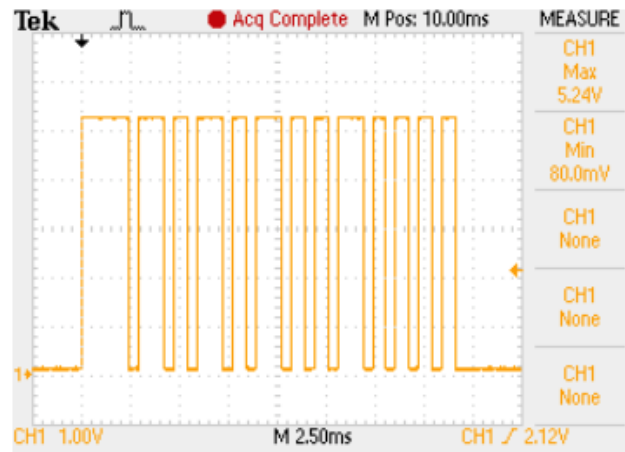


FIGURA NO. 92. CÓDIGO ON/OFF PARA UNA TELEVISIÓN SONY, CONTROL DESARROLLADO.

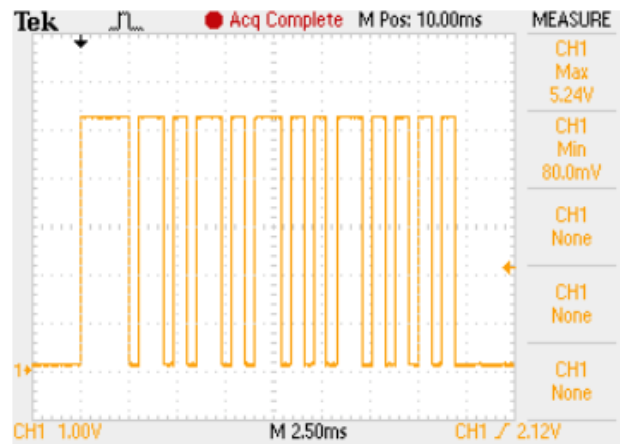


FIGURA NO. 93. CÓDIGO ON/OFF PARA UNA TELEVISIÓN PHILIPS, CONTROL ORIGINAL

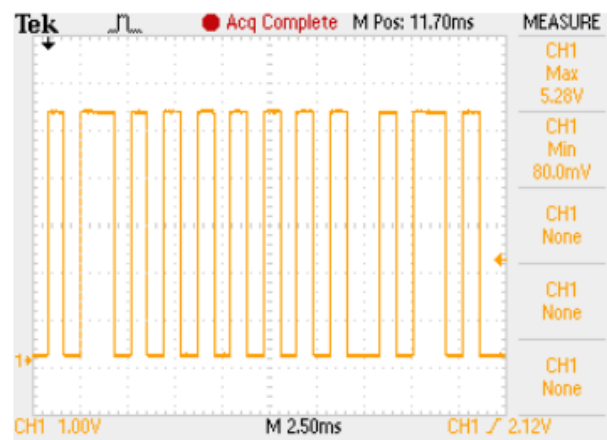


FIGURA NO. 94. CÓDIGO ON/OFF PARA UNA TELEVISIÓN PHILIPS, CONTROL DESARROLLADO.

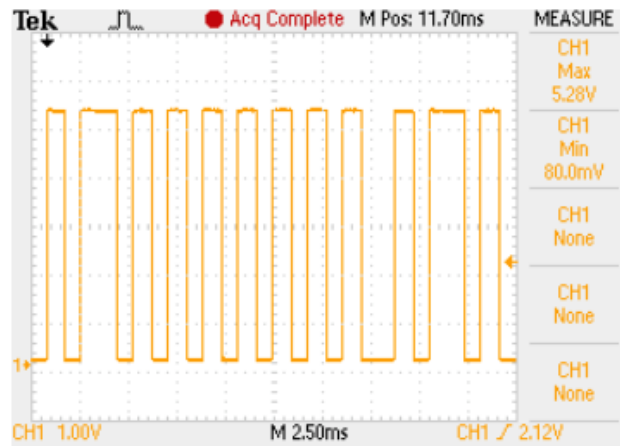


FIGURA NO. 95. CÓDIGO ON/OFF PARA UNA TELEVISIÓN PANASONIC, CONTROL ORIGINAL.

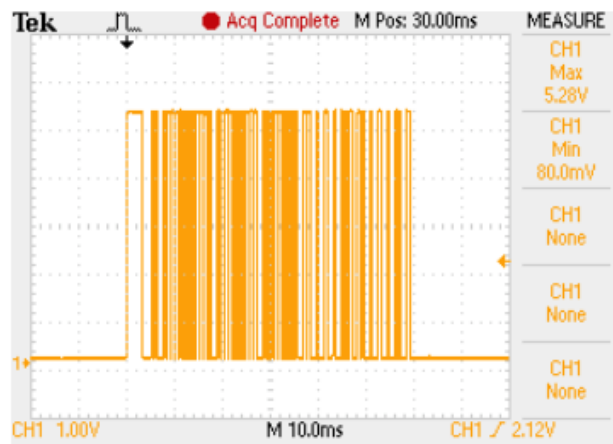


FIGURA NO. 96. CÓDIGO ON/OFF PARA UNA TELEVISIÓN PANASONIC, CONTROL DESARROLLADO

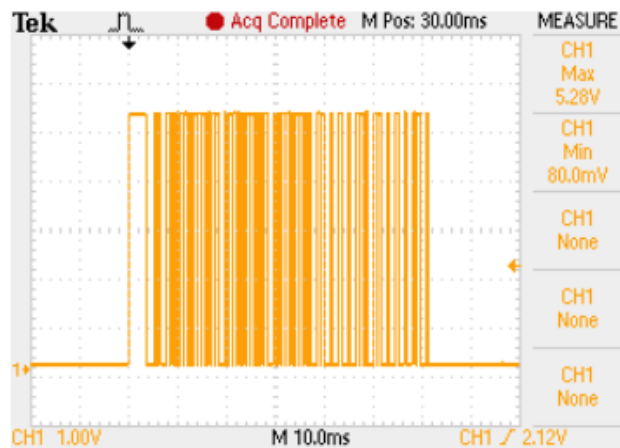


FIGURA NO. 97. CÓDIGO ON/OFF PARA UNA TELEVISIÓN TOSHIBA, CONTROL ORIGINAL.

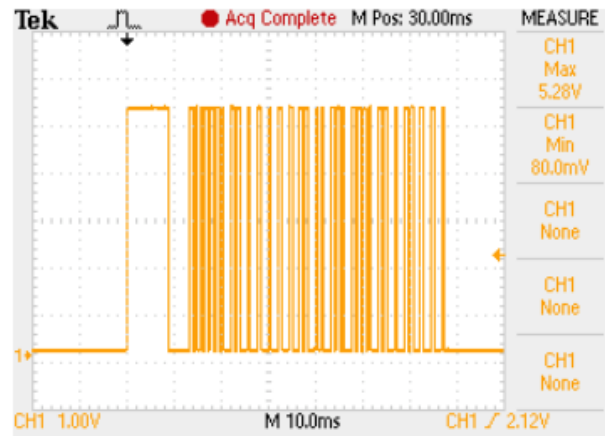


FIGURA NO. 98. CÓDIGO ON/OFF PARA UNA TELEVISIÓN TOSHIBA, CONTROL DESARROLLADO.

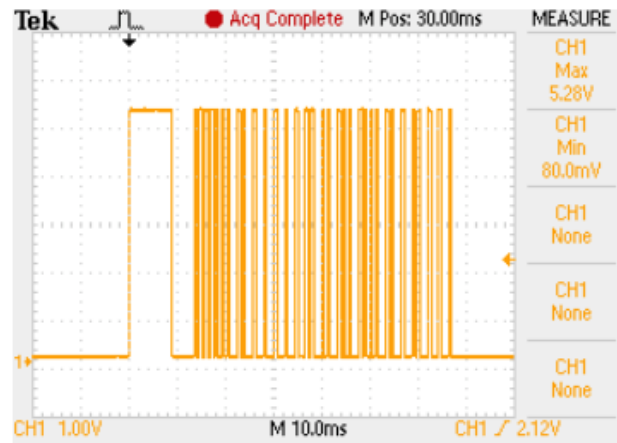


FIGURA NO. 99. CÓDIGO ON/OFF PARA UNA TELEVISIÓN LG, CONTROL ORIGINAL.

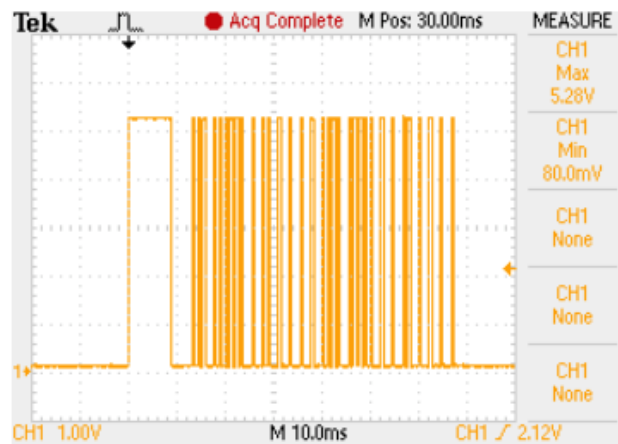


FIGURA NO. 100. CÓDIGO ON/OFF PARA UNA TELEVISIÓN LG, CONTROL DESARROLLADO.

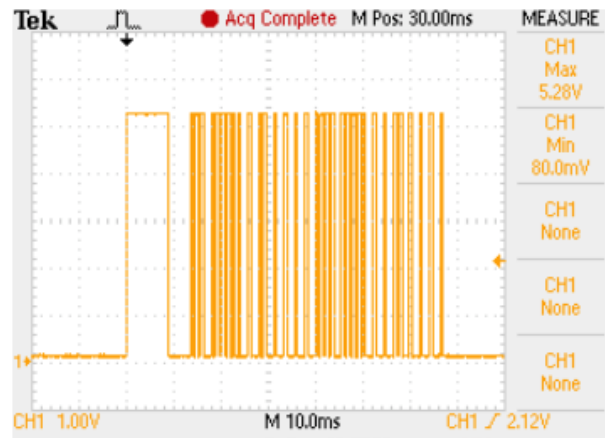


FIGURA NO. 101. CÓDIGO ON/OFF DE DIFERENTES TELEVISIONES ENVIADOS UNO TRAS DE OTRO.

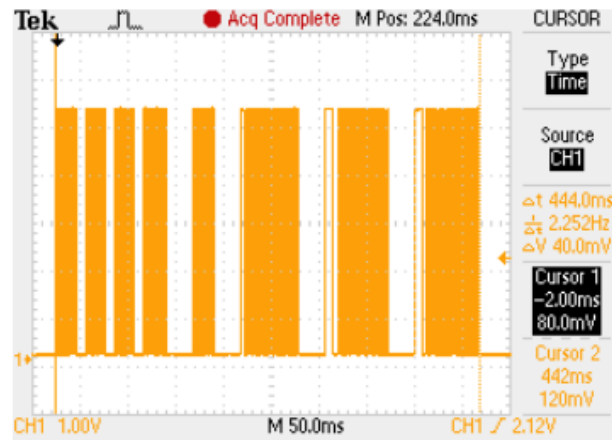


FIGURA NO. 102. CIRCUITO UTILIZADO PARA EL CONTROL DE DISPOSITIVOS IR.

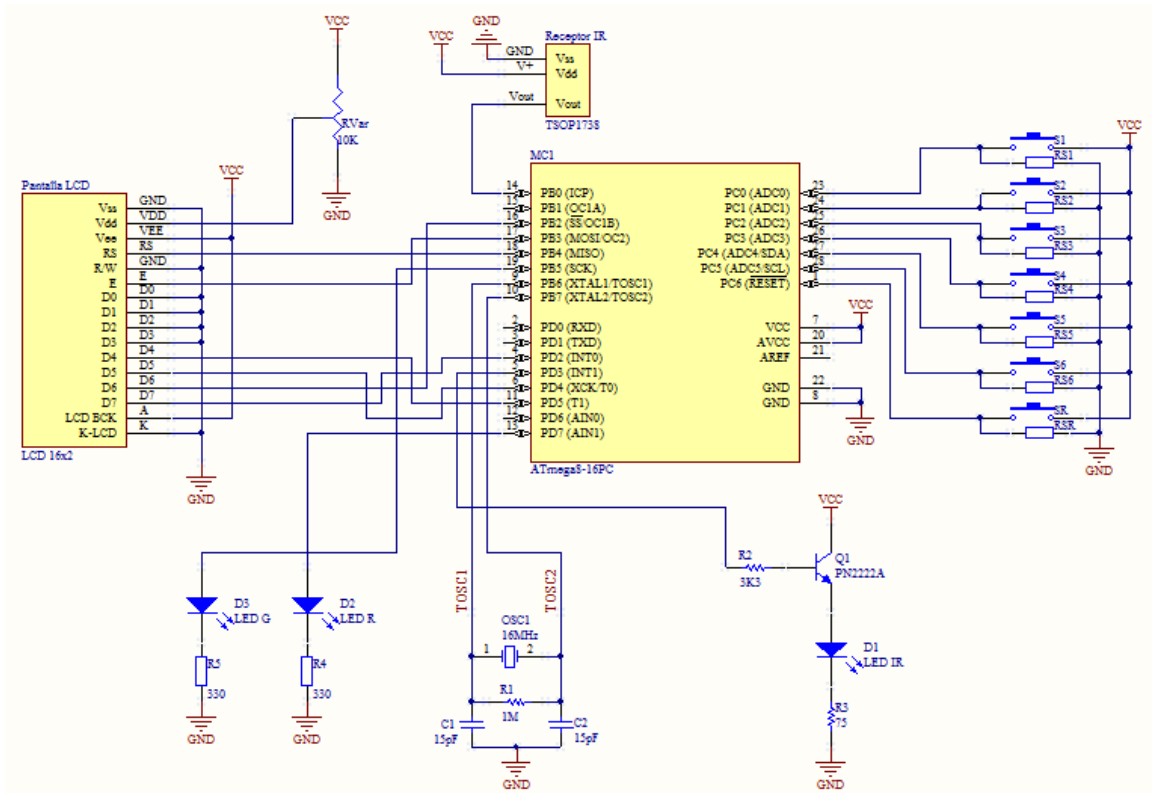
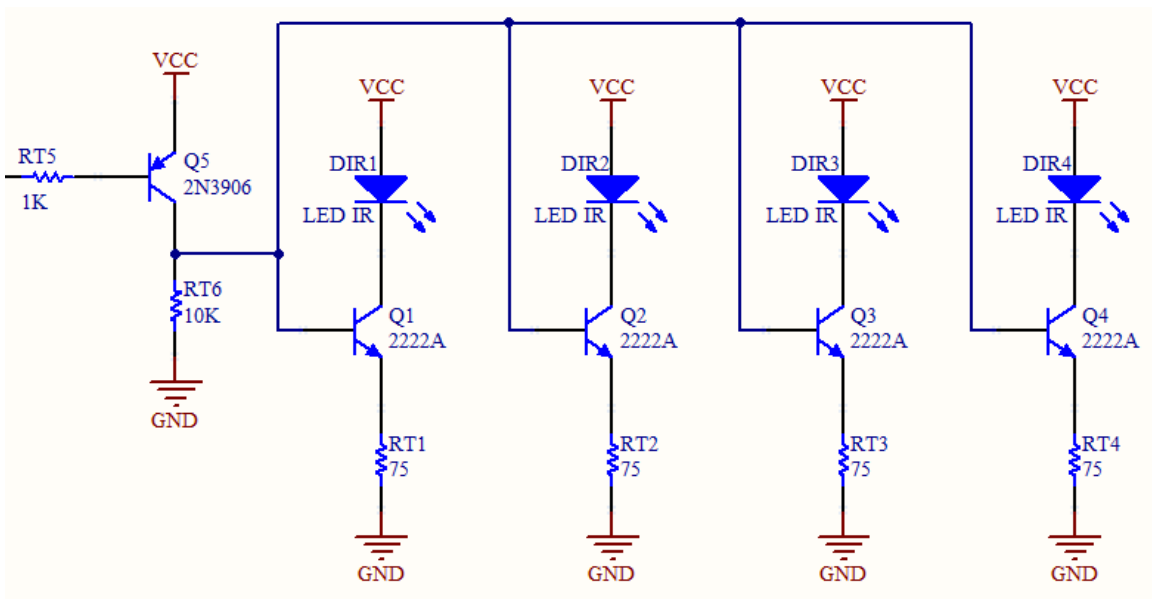


FIGURA NO. 103. CIRCUITO EMISOR IR MODIFICADO.



3. **Discusión.** Los controles universales tradicionales tienen dos desventajas. La primera es que necesitan una configuración previa de parte del usuario para poder controlar cada uno de los dispositivos. La segunda es que, si no tienen los códigos en la base de datos de un dispositivo, no son capaces de “aprender” estos códigos y por lo tanto no son capaces de controlarlo. El dispositivo desarrollado elimina estos dos problemas.

Para no tener que configurar el dispositivo, los códigos para un mismo comando de diferentes dispositivos se envían uno tras de otro. Por ejemplo, si uno desea encender o apagar una televisión, únicamente debe seleccionar la opción *ON/OFF TV*, y el control enviará los códigos de encendido y apagado almacenados en su base de datos para las diferentes televisiones uno tras de otro, y alguno de los códigos debe funcionar para la televisión que desea encender o apagar el usuario. Uno de los problemas de este sistema es que se consume energía innecesariamente debido a todos los códigos emitidos, sin embargo, debido a que el sistema necesita estar conectado a la red eléctrica para recibir los comandos, no es necesario la utilización de baterías para el control. Otro problema que se presentó con este sistema, es que si se tiene una gran cantidad de códigos enviados uno tras otro, el tiempo de envío de un solo comando puede ser demasiado grande para que el sistema funcione adecuadamente. En la Figura 101 se muestra el comando *ON/OFF* para seis diferentes televisiones enviados uno tras otro (el código para SONY se envía 3 veces por el protocolo), y el tiempo total necesario para enviar los seis códigos es de 444 ms.

Además, el sistema es capaz de “aprender” nuevos códigos directamente de los controles originales de los dispositivos, por lo que el usuario puede agregar funciones que previamente no se encontraban en la base de datos. Para obtener estos códigos se conectó el receptor TSOP1738 a microcontrolador, y con el programa desarrollado se analiza el tipo de código que se recibe, es decir se verifica si el código coincide con alguno de los protocolos incluidos en el sistema, y de esta forma almacenar los comandos requiere menos memoria. Si

el código recibido no coincide con ninguno de los protocolos incluidos en el sistema, el código será almacenado como un arreglo de valores enteros que contiene los tiempos de encendido y apagado que debe tener el emisor IR, y se utiliza una frecuencia portadora de 38 kHz, debido a que no es posible determinar con certeza la frecuencia exacta del código recibido, debido a las propiedades del receptor.

Uno de los problemas encontrados durante la elaboración del control de dispositivos IR, fue el uso optimizado de la memoria del microcontrolador, debido a que cuenta únicamente con 2 kB de memoria SRAM, y los códigos almacenados para el control de los diferentes dispositivos necesita más memoria. Para solucionar este problema se utilizó una librería que permite almacenar directamente en la memoria Flash del microcontrolador (32 kB), por lo que la memoria SRAM quedó libre para ser utilizada en otros aspectos necesarios para el funcionamiento del dispositivo. Además, se almacenó los códigos recibidos que es capaz de “aprender” el dispositivo en la memoria EEPROM del microcontrolador, para que el usuario no tenga que volver a configurar el dispositivo cada vez que se apague.

Uno de los problemas que surgieron durante las pruebas del control de dispositivos IR fue que el ángulo de envío del LED IR utilizado es relativamente pequeño (20 grados según las especificaciones del fabricante), por lo que se decidió cambiar el circuito transmisor y utilizar cuatro LEDs IR que apuntaran en diferentes direcciones, para lograr un ángulo de envío mayor. Esto es importante debido a que el sistema está diseñado para que se mantenga en un lugar fijo, y si el dispositivo debe ser capaz que controlar diferentes dispositivos sin que el usuario deba mover el control. Este problema no se da en los controles convencionales debido a que es el usuario el que apunta con el control hacia el dispositivo que se desea controlar.

B. ILUMINACIÓN LED RGB

1. Diseño experimental. Se utilizó el microcontrolador ATMEGA328P de la familia ATMEL. Se seleccionó este microcontrolador debido principalmente a que posee 6 puertos que pueden ser utilizados para el uso de PWMs, y es bajo este principio que se logra crear los diferentes colores en los LEDs RGB.

Se utilizó LEDs RGB de alta iluminación, aproximadamente 9400 milicandelas, y de baja potencia. Funcionan con 20 mA por cada LED. Es importante mencionar que este módulo no busca desarrollar un sistema de iluminación para realizar tareas, sino únicamente se utilizará para la creación de ambientes de iluminación para adornar los diferentes espacios de una vivienda.

Para realizar todas sus funciones el sistema debe comunicarse adecuadamente con el Módulo de Infraestructura de Comunicación, el cual será el encargado de guiar la comunicación entre los módulos periféricos y el módulo central. La comunicación con el Módulo de Infraestructura de Comunicación se realizará utilizando comunicación serial. Para realizar las pruebas se envió los comandos directamente de la computadora al microcontrolador utilizando la misma comunicación serial.

2. Resultados

FIGURA NO. 104. DIFERENTES COLORES OBTENIDOS CON EL MÓDULO DE ILUMINACIÓN RGB.



3. Discusión. Se utilizaron LEDs RGB de alta iluminación, 9400 milicandelas, para lograr producir la luz necesaria según los requerimientos del ambiente. Estos LEDs tiene un ángulo sólido de emisión de 45° , el cual es equivalente a 9.86 estereorradianes, por lo que cada LED RGB utilizado emite una cantidad de luz equivalente a 90 lúmenes cuando se utilizan en su máxima capacidad. Debido a que no se utilizaran todo el tiempo a su máxima capacidad se utilizó un factor de 2 para cada LED, con lo que se obtuvo 45 lúmenes por LED. Se diseñó una lámpara que pudiera dar 450 lúmenes para iluminar una habitación, por lo que se utilizó 10 LEDs RGB por unidad.

Debido a la cantidad de LEDs utilizados, se analizó tres diferentes configuraciones para el circuito encargado del encendido de los LEDs RGB. Se configuró el microcontrolador para obtener una salida PWM por cada color (rojo, verde y azul), y debido a la poca corriente que pueden dar las salidas del microcontrolador se utilizó un circuito transistores para obtener la corriente necesaria en los LEDs.

Inicialmente se pensó en conectar las tiras de LEDs en serie, para de esta manera obtener la misma corriente en los LEDs y por lo tanto todos emiten la misma intensidad luminosa. Esta topología tiene la ventaja que es más eficiente, sin embargo tiene la gran desventaja que si uno de los LEDs falla, todos los demás LEDs conectados también fallan Además de que es necesario un mayor voltaje al tener los LEDs conectados en serie. Por lo tanto, para aumentar la vida útil de la lámpara, se decidió no utilizar esta configuración.

Luego se utilizó una configuración en paralelo de los LEDs. Esta configuración tiene la ventaja que si uno de los LEDs falla, no fallan todos los demás LEDs en las otras ramas, y la vida útil de la lámpara no se ve afectada si son pocos los LEDs que fallan. Sin embargo, presenta varias desventajas. Entre ellas se puede mencionar que debido a que los LEDs tienen distintos voltajes de codo y resistencia directa, la corriente que pasa por cada LED es distinta y no se obtiene la misma intensidad luminosa en cada uno. Este problema se puede solucionar regulando la corriente en cada LED agregando resistencias de diferentes valores, sin embargo esto aumenta los costos de la lámpara.

Por lo tanto se decidió a utilizar una configuración mixta, que incluye una configuración en paralelo de varias ramas de LEDs conectados en serie, y de esta manera se mantuvieron las ventajas de ambas configuraciones sin afectar el costo de la lámpara y su vida útil.

C. CONTROL DE ENCENDIDO, APAGADO Y ATENUACIÓN DE LÁMPARAS

1. Diseño experimental. Esta parte del proyecto tiene como objetivo

objetivo desarrollar un sistema que sea capaz de controlar principalmente el encendido y apagado de las diferentes lámparas de la vivienda, como un dispositivo periférico del sistema domótico desarrollado en el Megaproyecto RDI. Se busca desarrollar un sistema económico y que pueda ser utilizado para controlar individualmente cada lámpara, o controlar un grupo de lámparas de forma simultánea. Además, se busca que el sistema pueda ser utilizado para la atenuación de las lámparas en las que sea posible, ya que no todos los tipos de lámpara permiten ajustar la intensidad.

2. Resultados. Se utilizó un microcontrolador ATMEGA328P, debido a que este microcontrolador cuenta con las entradas y salidas digitales necesarias para la realización del módulo. Además, se utilizó un circuito de seguridad en una de sus entradas digitales lo que permitió realizar fácilmente un circuito detector de cruce por cero de la señal eléctrica de 120V AC, lo cual era necesario para controlar la atenuación de las lámparas.

Se utilizó un TRIAC BTA06 para controlar la señal eléctrica de 120V AC, el cual es activado por un pulso que se genera por el microcontrolador. Además, se utilizó una optocoplá MOC3020 para separar la señal de la red eléctrica de 120V AC del circuito del microcontrolador y controlar la corriente de compuerta para la activación del TRIAC.

Para realizar todas sus funciones el sistema debe comunicarse adecuadamente con el Módulo de Infraestructura de Comunicación, el cual será el encargado de guiar la comunicación entre los módulos periféricos y el módulo central. La comunicación con el Módulo de Infraestructura de Comunicación se realizará utilizando comunicación serial. Para realizar las pruebas se envió los comandos directamente de la computadora al microcontrolador utilizando la misma comunicación serial.

FIGURA NO. 105. DIAGRAMA DE FLUJO CONTROL DE ILUMINACIÓN.

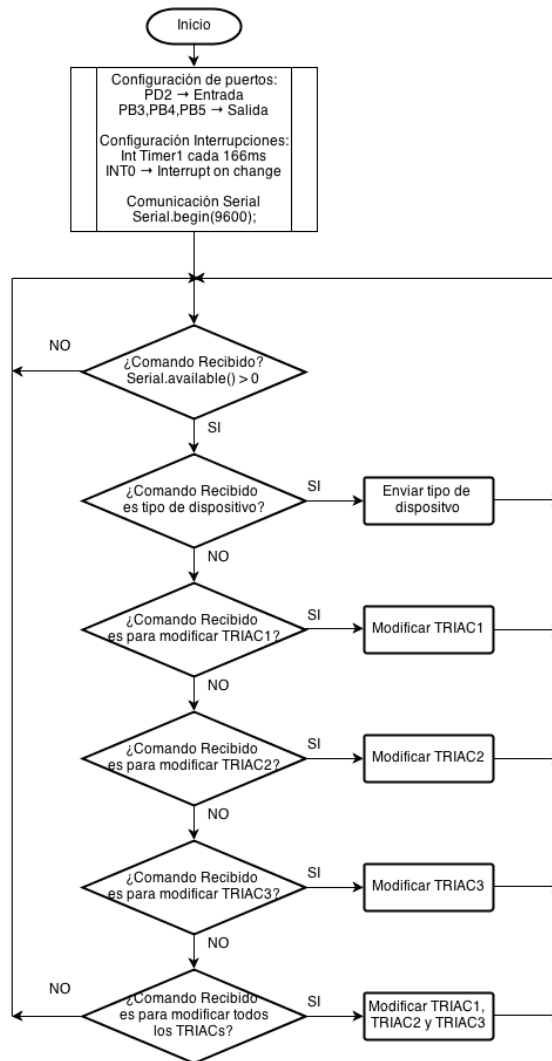


FIGURA NO. 106. DETECCIÓN DEL CRUCE POR CERO DE LA SEÑAL DE 120V AC.



FIGURA NO. 107. DETECCIÓN DEL CRUCE POR CERO DE LA SEÑAL DE 120V AC.

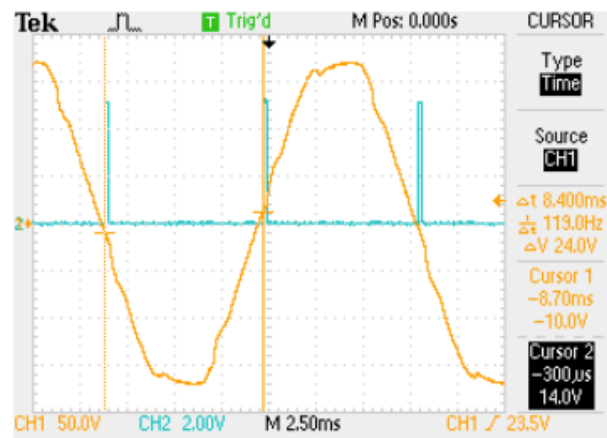


FIGURA NO. 108. ATENUACIÓN DE LA SEÑAL 120V AC AL 98% CH1 SEÑAL 120V AC, CH2 SEÑAL ATENUADA.

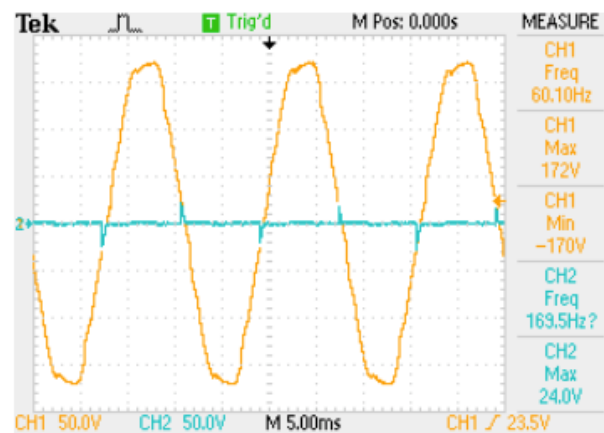


FIGURA NO. 109. ATENUACIÓN DE LA SEÑAL 120V AC AL 80% CH1 SEÑAL 120V AC, CH2 SEÑAL ATENUADA.

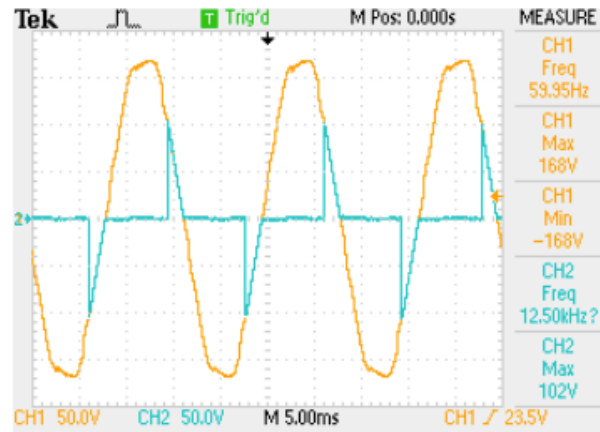


FIGURA NO. 110. ATENUACIÓN DE LA SEÑAL 120V AC AL 60% CH1 SEÑAL 120V AC, CH2 SEÑAL ATENUADA

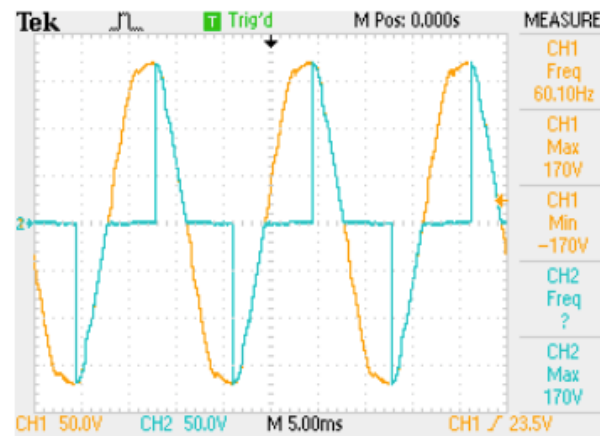


FIGURA NO. 111. ATENUACIÓN DE LA SEÑAL 120V AC AL 40% CH1 SEÑAL 120V AC, CH2 SEÑAL ATENUADA.

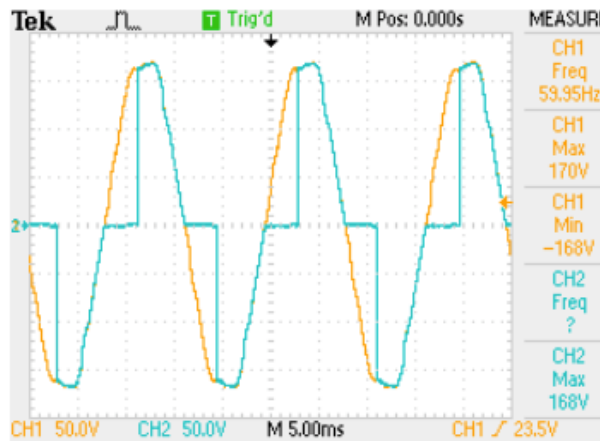


FIGURA NO. 112. ATENUACIÓN DE LA SEÑAL 120V AC AL 20% CH1 SEÑAL 120V AC, CH2 SEÑAL ATENUADA.

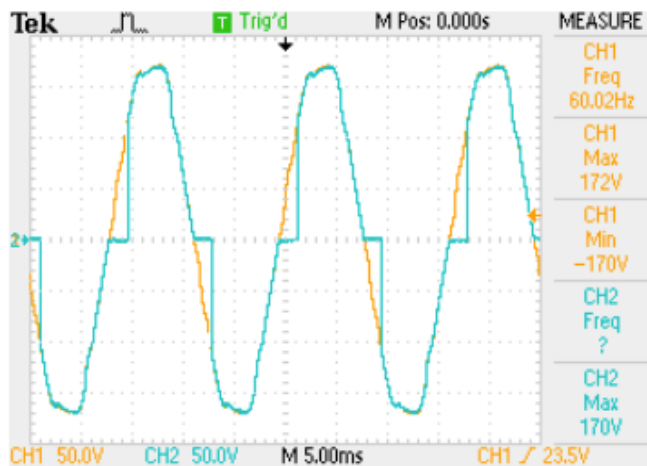
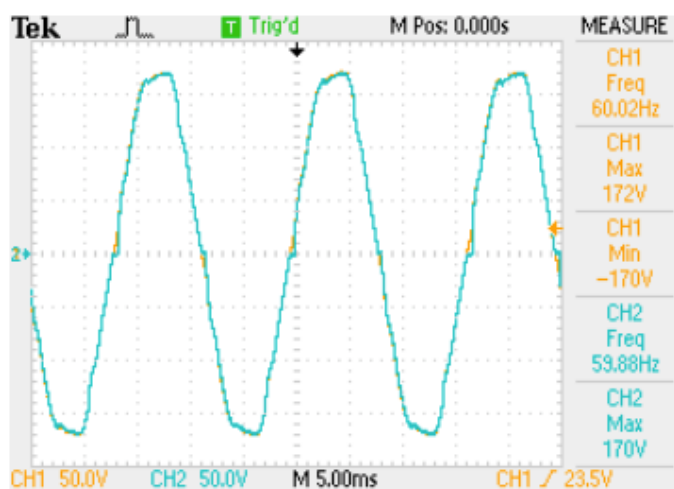
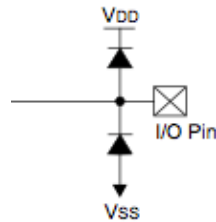


FIGURA NO. 113. ATENUACIÓN DE LA SEÑAL 120V AC AL 2% CH1 SEÑAL 120V AC, CH2 SEÑAL ATENUADA.



3. Discusión. Para poder determinar el cruce por cero de la señal de 60 Hz se emuló una de las características del microcontrolador PIC 16F886, este dispositivo trae una protección en sus puertos de entrada, la cual está implementada por diodos como se puede observar en la Figura 114.

FIGURA NO. 114. PROTECCIÓN EN LOS PUERTOS DE ENTRADA DEL PIC16F886.



Para la detección del cruce por cero se utilizó la interrupción INT0 del microcontrolador ATMEGA328P, colocando entre la entrada y la red eléctrica una resistencia adecuada para reducir la corriente en la entrada del microcontrolador.

El circuito de detección del cruce por cero desde el microcontrolador funcionó de forma adecuada, las señales son enviadas con la frecuencia específica y el tiempo específico, justo cuando la señal de corriente alterna cruza el paso por cero. Por lo que cada una de las pruebas realizadas con dicho circuito entregó un resultado satisfactorio. Los resultados del circuito implementado para la detección de cruce por cero fueron los esperados, ya que la resistencia utilizada para limitar la corriente de entrada al microcontrolador y bajar el voltaje funcionó de forma adecuada. El circuito implementado para la detección del cruce por cero de la señal de 60Hz cuenta únicamente con una resistencia, esta resistencia fue seleccionada de tal forma que restringiera el paso de la corriente al micro controlador. El microcontrolador ATMEGA328P no soporta una corriente mayor a 100mA en sus puertos de entrada, por lo que se seleccionó una resistencia de 1 M Ω , lo que limita la corriente de entrada al microcontrolador.

Cuando se requiere controlar potencias, tales como lámparas incandescentes, tubos fluorescentes, motores, electrodomésticos, etc. se requiere de interfaces de potencia basadas en dispositivos capaces de accionar potencia a partir de señales débiles. A esto debe agregarse sistemas de protección y aislamiento que permitan separar físicamente la parte lógica (en este caso el microcontrolador) de la parte de potencia.

Se utilizó la señal digital proveniente de los puertos de salida del microcontrolador, la cual es limitada en corriente. Esta señal es aplicada al cátodo del LED interno del optoacoplador. El brillo producido por el LED interno del optoacoplador acciona el TRIAC del mismo, que, a su vez, acciona el TRIAC de potencia. El optoacoplador utilizado fue un MOC3020 y un MOC3021. El TRIAC BTA06 es capaz de manejar 8A sobre 400V. Es recomendable ensamblar el TRIAC sobre un disipador térmico.

IX. PERIFÉRICOS DE POTENCIA

A. SISTEMA DE RIEGO

En el desarrollo de cualquier proyecto, los costos tienen un factor decisivo para el éxito de la idea, es por eso que para desarrollar un control para sistema de riego con la capacidad de comunicarse con el modulo central se pensó en una forma de optimizar al máximo los recurso y no invertir ni agregar cosas de más. Debido a que el megaproyecto RDI se basa en equipos integrados para casa es importante que este módulo sea igual de económico y con los mismos elementos de calidad que componen los otros equipos.

1. Diseño experimental. Se necesitaba de un sistema de riego que fuera capaz de poder recibir direcciones del módulo de Comunicación y seguridad de RDI propuesto por Diego Saravia, donde es el medio para recibir las ordenes de forma serial que envía el Modulo Central. Para esto el modulo del riego está diseñado sobre un PIC 16F886 para poder establecer la comunicación serial UART por los puertos RX y TX que se encuentran en las terminales 18-RC7 y 17-RC6. Se trabajos con la misma tasa que envían datos que es de un Baud Rate de 1200 para poder estar en sintonía con la comunicación de la red.

Con el mismo microcontrolador se implementó el control de 3 salidas independientes para la activación de las electroválvulas del sistema de riego, las cuales se encuentran en los puertos: 2-RA0, 3-RA1 y 4-RA2. Estos puertos se independizaron con un optoacoplador MOC3021 que separa entre la señal de control y la corriente de activación del siguiente elemento. Luego del Moc se conectó un TRIAC BT136 que permite hacer el encendido y apagado de señales de voltajes alterno, la señal de control para la activación de la electroválvula es de 24VAC a una frecuencia de 50/60Hz.

2. Resultados. Se utilizó un compartimiento resistente al agua del tamaño adecuado para almacenar el controlador de riego y el filtro decodificador de señales del Módulo de comunicación y seguridad. Se observa un transformador de 110VAC a 24 VAC que es la señal que se utiliza para la activación de los solenoides de las electroválvulas.

FIGURA NO. 115. COMPARTIMIENTO Y ACCESORIOS MÓDULO DE RIEGO.



TABLA NO. 22. COSTO DE MATERIALES Y CONSTRUCCIÓN MÓDULO DE RIEGO

Item	Descripcion	Cantidad	Precio Q	Subtotal
1	Caja interperie	1	Q66.20	Q66.20
2	CableTCJ 2x14	1.5	Q4.75	Q7.13
3	Espiga polarizada	1	Q7.50	Q7.50
4	Borneras	1	Q18.00	Q18.00
5	Pic 16f886	1	Q46.00	Q46.00
6	Moc 3021	4	Q5.00	Q20.00
7	Triac BT136	4	Q10.00	Q40.00
8	Transformador de 5VDC	1	Q35.00	Q35.00
9	Transformador 110/24VAC	1	Q95.00	Q95.00
10	Circuito impreso	1	Q20.00	Q20.00
11	Diseño y construccion	1	Q75.00	Q75.00
El precio total del modulo del riego es de				429.825

3. Discusión. Fue necesario buscar un compartimiento que sea resistente al agua para poder colocarlo ya sea en la parte exterior como interior de la casa. El cable con espiga polarizada permite la alimentación de los circuitos como también la lectura de datos que se transmiten en la línea AC. Se puede observar en la parte inferior derecha el transformador de 110 a 24 VAC que es la señal que activa las electroválvulas del sistema de riego. El pic permite controlar los optoacopladores de forma independiente, ya que lo principal en el riego es poder activar una sola válvula en un determinado tiempo. En este Pic se utilizan los cuatro primeros puertos de salida iniciando desde el RA0 hasta el RA3. Con la configuración elegida para configurar un pin con la señal de activación de cada válvula se logra implementar hasta 8 electroválvulas independientes por módulo de riego.

Para la alimentación del PIC se utilizó un transformador convencional de 5 voltios de corriente directa para facilitar la construcción y diseño ya que no incrementa significativamente el costo del módulo en general en más de un 8% equivalente a Q 35.00. Se tiene previsto que al momento de comercializar el modulo se puede obtener una reducción en costo de un 40% lo cual se reflejaría en dinero en doscientos cincuenta y siete quetzales. Esto es una quinta parte del precio del controlador de riego que se puede encontrar en Guatemala en la actualidad y un tercio del precio que se puede encontrar en los Estados Unidos.

B. VENTANA MOTORIZADA

El tema de ventanas automatizada aunque no es difícil de visualizar hay muy poco desarrollo sobre esta rama en Guatemala. Como el nombre lo dice se trata de una ventana con un motor o un pistón hidráulico o eléctrico el que proporciona el movimiento al sistema. Por lo general se encontrara sistemas de pistones en ventanas abatibles y sistemas con motores en ventanas corredizas por su movimiento de apertura y la adaptación con elementos estándar del mercado existente.

1. Diseño experimental. Se investigó sobre los diferentes materiales con los cuales se realizan las ventanas, donde se destacan: PVC, Aluminio, metal, madera. Es importante iniciar con los materiales, ya que el tamaño compacto de los actuadores y la potencia mínima generada requiere contemplar elemento de construcción livianos y sistemas con coeficiente de fricción reducido para facilitar la apertura y cierre. Luego de analizar ventajas y desventajas de los materiales se optó por trabajar con perfiles de aluminio. El aluminio es de lo materiales más livianos, elegantes y se les puede acoplar fácilmente los actuadores.

En cuanto al actuador se eligió un motor DC de 9 voltios, baja rotación y alto torque. El motor consume 300mA en rotación continua y 500mA en el inicio del funcionamiento. El motor se controla con el integrado L298N el cual permite una potencia máxima de 2 amperios por motor. Para el control de apertura y cierre de la ventana se utilizaron switches de fin de carrera para mandar el estado de la ventana al PIC de control. Para el procesamiento de datos y la recepción de información se utilizó el mismo PIC que en el sistema de riego, el PIC 16F886 el cual en los puertos RA0 y RA1 se envía la señal de control hacia el controlador de motores L298N y en las terminales RB0 y RB1 se encuentran los dos sensores de la ventana para identificar cuando llego al fin o inicio del movimiento.

2. Resultados

TABLA NO. 23. EQUIVALENCIA POTENCIA DE MOTOR VS METRO CUADRADO DE VENTANA

Metros² de ventana	Potencia motor (watts)
0.25	2.75
0.5	5.3
1	12.2
1.5	19.7
2	24.8
2.5	33.2
3	45.5

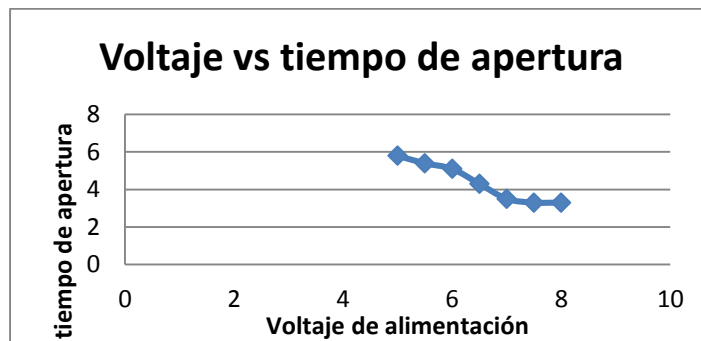
TABLA NO. 24. RELACIÓN VOLTAJE VS TIEMPO DE APERTURA DE VENTANA

Voltaje (voltio)	Tiempo de apertura (segundos)
5	5.8
5.5	5.4
6	5.1
6.5	4.3
7	3.5
7.5	3.3
8	3.3

FIGURA NO. 116. DISEÑO E INSTALACIÓN VENTANA CON MOTOR.



FIGURA NO. 117. VOLTAJE VS TIEMPO DE APERTURA DE VENTANA.



3. Discusión. La utilización de perfiles de aluminio permitió adaptar el mecanismo del motor con faja para brindar el movimiento deseado, ya que fue fácil de manipular este material para la apertura de agujeros para la colocación de sensores y el del cableado eléctrico. Los tiempos obtenidos se obtuvieron de la medición con un cronometro de tres diferente corridas con el prototipo de la ventana con motor. Se colocaba el voltaje deseado en la fuente de voltaje y con un botón de contacto se permitía el paso de corriente para iniciar la medición del tiempo.

Para poder adaptar el voltaje del motor se utilizó una señal modulada (PWM) con el PIC, donde se configuro a un 80% de ciclo de trabajo ya que el driver de motor se alimentaba con 9 voltios DC. El motor es el de una impresora Lexmark Z76 ya que cumplía con los requerimientos de la ventana, sin embargo el modulo se diseñó para poder controlar motores de 2 amperios máximo de corriente para poder abarcar ventanas convencionales de una casa.

C. CORTINA MOTORIZADA

Las cortinas automáticas han tenido un gran auge en la actualidad y Guatemala no se ha quedado atrás en ofrecerlas al mercado guatemalteco. Para este megaproyecto se utilizaron cortinas enrollables horizontales ya que con un solo motor nos permite abrirlas y cerrarlas. La elección de colores no afecta el funcionamiento ni la dimensión del motor, sin embargo el tipo de tela si afecta ambos factores ya que dependiendo de la tela va a ser el peso que va a necesitar levantar el motor para poder dejar pasar la luz.

1. Diseño experimental. Se implementó el control de dos salidas del PIC para la activación del motor de la cortina, los cuales se encuentran en los puertos: 2-RA0 3-RA1. Estos puertos se conectan a un driver de motor que permite el giro en ambos sentidos, soportando la operación desde 12 a 24 VDC. Para la cortina de muestra se utilizó una alimentación en el driver de 12VDC.

2. Resultados. Se acopló una cortina enrollable horizontal a la ventana prototipo del megaproyecto para poder mostrar el funcionamiento correcto de la cortina como se observa a continuación. Esta se realizó con una tela convencional de cortinas color negro tipo *black out* para la reflexión de luz solar.

FIGURA NO. 118. DISEÑO E INSTALACIÓN CORTINA CON MOTOR.



TABLA NO. 25. COSTO DE MATERIALES Y CONSTRUCCIÓN MÓDULO DE CORTINA

Item	Descripción	Cantidad	Precio Q	Subtotal
1	Riel 1 mt de largo	1	Q85.50	Q85.50
2	Cojinetes 1.2"	2	Q9.75	Q19.50
3	Carga cojinete	2	Q6.00	Q12.00
4	Motos para cortina 1mt2	1	Q115.00	Q115.00
5	1mt2 de tela black out	1	Q98.20	Q98.20
6	Sensores magneticos	2	Q12.00	Q24.00
7	Iman de neodimio	1	Q25.00	Q25.00
8	Transformador de 9VDC	1	Q120.00	Q120.00
9	PIC 16 f886	1	Q45.00	Q45.00
10	Circuito impreso	1	Q20.00	Q20.00
11	Diseño y construccion	1	Q100.00	Q100.00
El precio total del módulo del cortina es de				664.2

3. Discusión. Fue necesario construir una cortina motorizada ya que el costo de comprarla en Guatemala era muy elevado. La elección del motor de 9 voltios fue correcta ya que proporciona entre 15 y 20 watts de potencia para elevar la cortina. Para este sistema se utilizó el mismo driver de motores que en el de la ventana (LM298N), sin embargo a este módulo se le exige más corriente al driver si se compara con el motor de la ventana, ya que se obtuvo un consumo de 0.66 amperios que necesita suministrar el driver, esto es un 33% de la corriente máxima que puede aportar el driver seleccionado. Para este módulo es importante resaltar que este diseño dependerá mucho de las telas con las que cuente la cortina, ya que hay telas bastante pesadas que requerirán de un motor más robusto que los seleccionados para el proyecto. Siempre enfocándonos en la reducción del consumo eléctrico es importante tener en cuenta telas livianas para la fabricación de cortinas esto hará que se requiera motores pequeños y con menor consumo eléctrico.

Si la cortina es muy grande o de una tela muy pesada el sistema permite utilizar dos motores en paralelo ya que el driver permite controlar dos motores DC y solo se necesita interconectar las líneas de activación de la parte A del control de motor con la parte B para obtener una activación en conjunto de ambos motores.

XIII. CONCLUSIONES

- Se desarrolló una aplicación para dispositivos móviles Apple, capaz de comunicar al usuario con el módulo central por medio del estándar 802.11 de la IEEE a través de una red local LAN, implementando una interfaz gráfica para el usuario.
- Se diseñó una interfaz gráfica específica para cada tipo de módulo.
- Se implementó comunicación bidireccional entre la interfaz gráfica de usuario y el servidor local por medio del protocolo de internet HTTP y sus métodos GET y POST.
- El módulo central es capaz de realizar la función de servidor web dentro de una red local para comunicarse con los dispositivos móviles
- El módulo central es capaz de enviar y recibir información de cada uno de los dispositivos conectados al sistema
- Se logró la comunicación entre el módulo central con los distintos dispositivos individuales a través de la red eléctrica
- Se desarrolló un protocolo de comunicación asíncrono bidireccional para enlazar el módulo central con los dispositivos periféricos.
- El protocolo diseñado es más rápido que el protocolo del sistema de domótica X10.
- El protocolo diseñado implementa un checksum de seguridad para corroborar la integridad de los datos en el proceso de comunicación.
- Se diseñó, construyó e implementó un sistema que controla el encendido y apagado de las diferentes lámparas de una vivienda.
- Se diseñó y construyó un control universal infrarrojo para manejar los diferentes dispositivos que lo permiten, como la televisión y el equipo de sonido.
- Se diseñó y construyó un dispositivo de iluminación que es capaz de crear diferentes ambientes en la vivienda utilizando iluminación LED RGB.
- Se diseñó y construyó un módulo compacto y económico para el control de válvulas de riego que se comunica con la red y permite activar solenoides de 24 VAC de electroválvulas.

- Se diseñó y construyó un módulo compacto para el control de cortinas de tubo de despliegue vertical para adaptación en ventanas existentes.
- Se elaboró un sistema compacto que controla motores de ventanas corredizas de hasta 2 amperios.
- Se desarrolló un sistema domótico integral que permite el control de variables domésticas desde una aplicación móvil.

XIV. BIBLIOGRAFÍA

- American*. (20 de Septiembre de 2013). Obtenido de <http://www.american.edu.co/vs/webelectric/tiposdecorrienteelectric.html>
- Apple Developer*. (s.f.). Recuperado el Julio de 2013, de Xcode 5: <https://developer.apple.com/technologies/tools/whats-new.html>
- Apple Developer*. (s.f.). Recuperado el Mayo de 2013, de Working with Protocols: <https://developer.apple.com/library/ios/documentation/cocoa/conceptual/ProgrammingWithObjectiveC/WorkingwithProtocols/WorkingwithProtocols.html>
- Behrouz A., F. (2001). *Transmisión de datos y redes de comunicaciones*. Madrid: McGraw-Hill.
- Brainbell. (s.f.). *Free IT training and computer tutorials*. Recuperado el Junio de 2013, de Modem Speeds: http://www.brainbell.com/tutors/A+/Hardware/Modem_Speeds.htm
- Cheng, J., & Malaika, S. (1997). *Web Gateway Tools*. Lancaster: John Wiley & Sons Inc.
- Cocoa Lab*. (s.f.). Recuperado el Enero de 2013, de Become an Xcoder: <http://www.cocoalab.com/BecomeAnXcoder>.
- Cox, J. (s.f.). *iPhone 5 and %GHz Wi-Fi*. . Recuperado el Septiembre de 2013, de Network World: <http://www.networkworld.com/news/2012/091412-iphone5-5ghz-262466.html?page=1>
- Cuevas, J., Martínez, J., & Merino, P. (2002). *El protocolo X10: una solución antigua a problemas actuales*. Málaga.
- Django Project. (s.f.). *Django Documentation*. Recuperado el Abril de 2013
- Droms, R. (Marzo de 1997). *Dynamic Host Configuration Protocol*. Recuperado el Abril de 2013, de <http://www.ietf.org/rfc/rfc2131.txt>
- Element14 Community. (2012, Marzo 8). *Raspberry Pi - GPIO Expansion - Low Level Peripherals*. Retrieved Febrero 2013, from <http://www.element14.com/community/docs/DOC-43484.pdf>
- Element14 Community. (2013, Septiembre 10). *Raspberry Pi Single Board Computer*. Retrieved Septiembre 2013, from <http://www.element14.com/community/docs/DOC-42993.pdf>
- Fielding, R., & Reschke, J. (25 de Septiembre de 2013). *Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing*. Recuperado el Septiembre de 2013, de <http://www.ietf.org/id/draft-ietf-httpbis-p1-messaging-24.txt>

- Fielding, R., & Reschke, J. (25 de Septiembre de 2013). *Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content*. Recuperado el Septiembre de 2013, de <http://www.ietf.org/id/draft-ietf-httpbis-p2-semantics-24.txt>
- Finol, G. (s.f.). *Temperatura de color*. Recuperado el Noviembre de 2012, de <http://gelmertfinol.wordpress.com/2012/01/18/temperatura-de-color/>
- Foster, R. (s.f.). *Tipos de Lámparas e Iluminación*. Recuperado el Octubre de 2013, de Enciclopedia de Salud y Seguridad en el trabajo: <http://www.insht.es/InshtWeb/Contenidos/Documentacion/TextosOnline/EnciclopediaOIT/tomo2/46.pdf>
- Frystyk, H. (Julio de 1994). *The Internet Protocol Stack*. Recuperado el Marzo de 2013, de <http://www.w3.org/People/Frystyk/thesis/TcpIp.html>
- GNU Project. (28 de Junio de 2011). *GNU Bash*. Recuperado el Junio de 2013, de <http://www.gnu.org/software/bash/>
- Gómez Tejedor, J. A., & Olmos Sanchís, J. J. (1999). *Cuestiones y problemas de electromagnetismo y semiconductores*. Valencia: Universidad Politécnica de Valencia.
- González Galindo, E. E. (2005). *Domótica utilizando estándar CEBus sobre la red eléctrica residencial*. Guatemala.
- GPIOs*. (4 de Julio de 2013). Recuperado el Septiembre de 2013, de <http://elinux.org/File:GPIOs.png>
- Hegarty, P. (s.f.). *CS 193p iPhone Application Development*. Recuperado el Noviembre de 2012, de Stanford University: <http://www.stanford.edu/class/cs193p/cgi-bin/drupal/>
- Hernando, R. (27 de Enero de 2002). *TCP/IP*. Recuperado el Marzo de 2013, de <http://www2.rhernando.net/modules/tutorials/doc/redes/tcp.html>
- Hunter Industries*. (s.f.). Recuperado el Julio de 2013, de Guía para el diseño e instalación de riegos con normas LEED: http://www.hunterindustries.com/sites/default/files/Hunter_LEED_Guide.pdf
- IES Cristobal de Monroy. (s.f.). *La energía eléctrica*. Sevilla.
- Incorporated, M. T. (2009). *Microchip*. Recuperado el Febrero de 2013, de PIC16F882/883//884/886/887: <http://ww1.microchip.com/downloads/en/DeviceDoc/41291G.pdf>
- Infantes Díaz, J. A. (2009). *Descripción de X-10*. Málaga.
- iOS Developer Library*. (s.f.). Recuperado el Julio de 2013, de Model-View-Controller: <https://developer.apple.com/library/ios/documentation/general/conceptual/CocoaEncyclopedia/Model-View-Controller/Model-View-Controller.html>

- iOS Developer Library*. (s.f.). Recuperado el Agosto de 2013, de Key Value Observing:
<https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/KeyValueObserving/KeyValueObserving.html>
- JSON Project. (s.f.). *Introducing JSON*. Recuperado el Abril de 2013, de <http://www.json.org/>
- Luque Rodríguez, J., & Clavijo Suero, S. (1995). *Modulación de señales digitales*. Sevilla. mikroElektronika. (2009). *mikroe*. Recuperado el Febrero de 2013, de SOFTWARE AND HARDWARE SOLUTIONS FOR THE EMBEDDED WORLD:
http://www.mikroe.com/downloads/get/30/mikroc_pic_pro_manual_v101.pdf
- Mikusiak, L. (s.f.). *iOS vs Android: Development Comparison*. Recuperado el Mayo de 2013, de <http://www.passion4teq.com/articles/ios-android-development-comparison-2/>
- Miyara, F., & Lahoz, L. (2003). *Introducción al análisis frecuencial y al ruido eléctrico*. Rosario.
- Mullins, R. (2011, Noviembre). *The Raspberry Pi Foundation*. Retrieved Septiembre 2013, from <http://www.element14.com/community/servlet/JiveServlet/previewBody/43302-102-1-232041/raspi-talk.pdf>
- Nielsen. (s.f.). *Who's winning the US Smartphone Market*. Recuperado el Agosto de 2013, de <http://www.nielsen.com/us/en/newswire/2013/whos-winning-the-u-s-smartphone-market-.html>
- Novel, B. (s.f.). *Clasificación de los sistemas domóticos y normalización en el área domótica*. Barcelona: Asociación Española de Domótica.
- NXP. (2011). *TDA5051A Home Automation Modem*.
- Oisterwijk. (2011). *Oisterwijk*. Recuperado el Junio de 2013, de The Netherlands:
<http://www.sbprojects.com>
- Orion. (s.f.). *Grupo de investigación Orion*. Recuperado el Junio de 2013, de Design and construction of LED lighting systems:
http://grupoorion.unex.es/orion/index.php?option=com_content&view=article&id=57&Itemid=76
- Otae. (15 de Agosto de 2013). Obtenido de <http://www.otae.com/enchufes/mapamon.htm>
- Otierney. (s.f.). *Objective C*. Recuperado el Agosto de 2013, de <http://www.otierney.net/objective-c.html>
- Peña Alcaráz, M. (2012). *Comunicaciones en el informe doméstico (domótica)*. Comillas.
- Powerline Control Systems. (2003). *X-10 to UPB migration document*. California.
- Rain Bird Inc. (s.f.). Recuperado el Julio de 2013, de Manual de usuario para sistemas de riego:
http://rainbird.com/documents/diy/DIYCatalog2009_sp.pdf

- Raspbian Project. (2012). *About Raspbian*. Recuperado el Enero de 2013, de <http://www.raspbian.org/RaspbianAbout>
- Santamaría, P. (s.f.). *Cómo elegir la bombilla LED correcta para cada necesidad*. Recuperado el Noviembre de 2012, de Especial: Iluminación LED: <http://www.xatakahome.com/iluminacion-y-energia/que-es-la-iluminacion-led-especial-iluminacion-led>
- SQLite. (s.f.). *About SQLite*. Recuperado el Mayo de 2013, de <http://www.sqlite.org/about.html>
- Vasiliev, Y. (Agosto de 2009). *Building Oracle Database-backed Web Application in Django*. Recuperado el Abril de 2013, de <http://www.oracle.com/technetwork/articles/dsl/vasiliev-django-100257.html>
- Wi-Fi. (s.f.). Recuperado el Marzo de 2013, de Wi-Fi: Core Technologie: <http://www.wi-fi.org/knowledge-center/articles>
- Wi-Fi. (s.f.). Recuperado el Marzo de 2013, de Wi-Fi: Core Technologie: <http://www.wi-fi.org/knowledge-center/>
- Wikipedia. (s.f.). Recuperado el Agosto de 2013, de API: http://en.wikipedia.org/wiki/Application_programming_interface
- Wireshark. (2004-2013). Recuperado el Septiembre de 2013, de Checksums: http://www.wireshark.org/docs/wsug_html_chunked/ChAdvChecksums.html
- Young, H. D., & Freedman, R. A. (2009). *Física universitaria volumen 2*. México: Pearson Educación.

XV. APÉNDICE

FIGURA NO. 119. ESQUEMÁTICO MÓDULO DE COMUNICACIÓN.

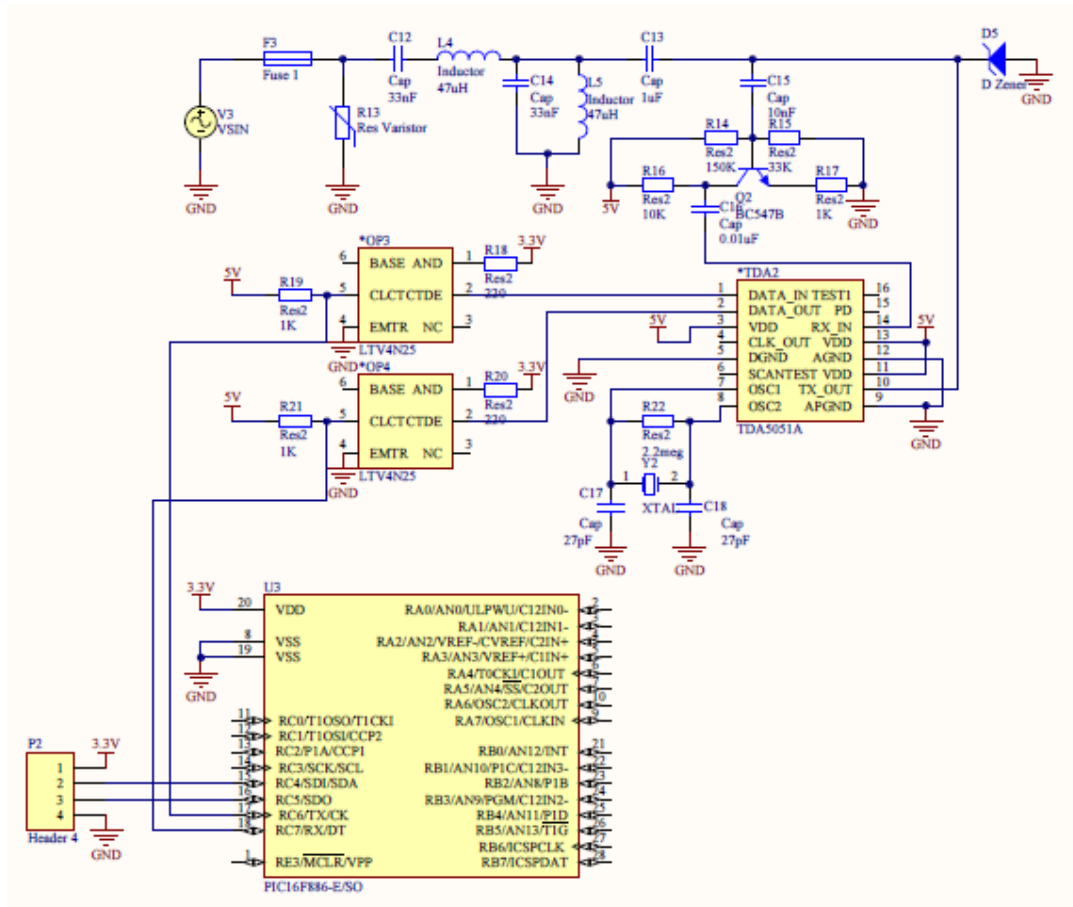


FIGURA NO. 120. ESQUEMÁTICO FUENTE DE ALIMENTACIÓN.

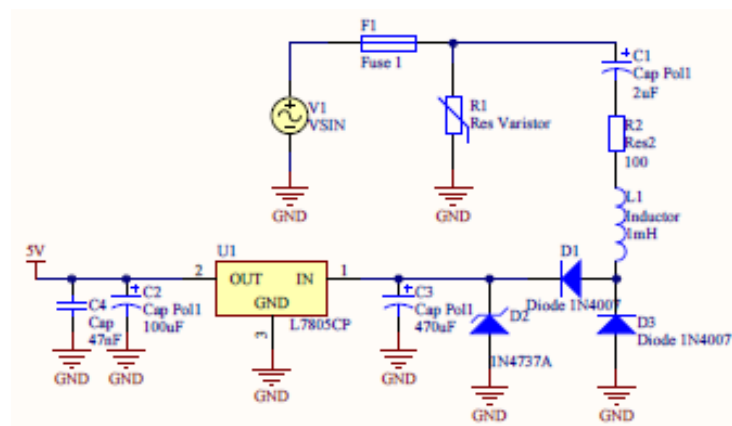


FIGURA NO. 121. PCB MÓDULO DE COMUNICACIÓN CON RASPBERRY PI.

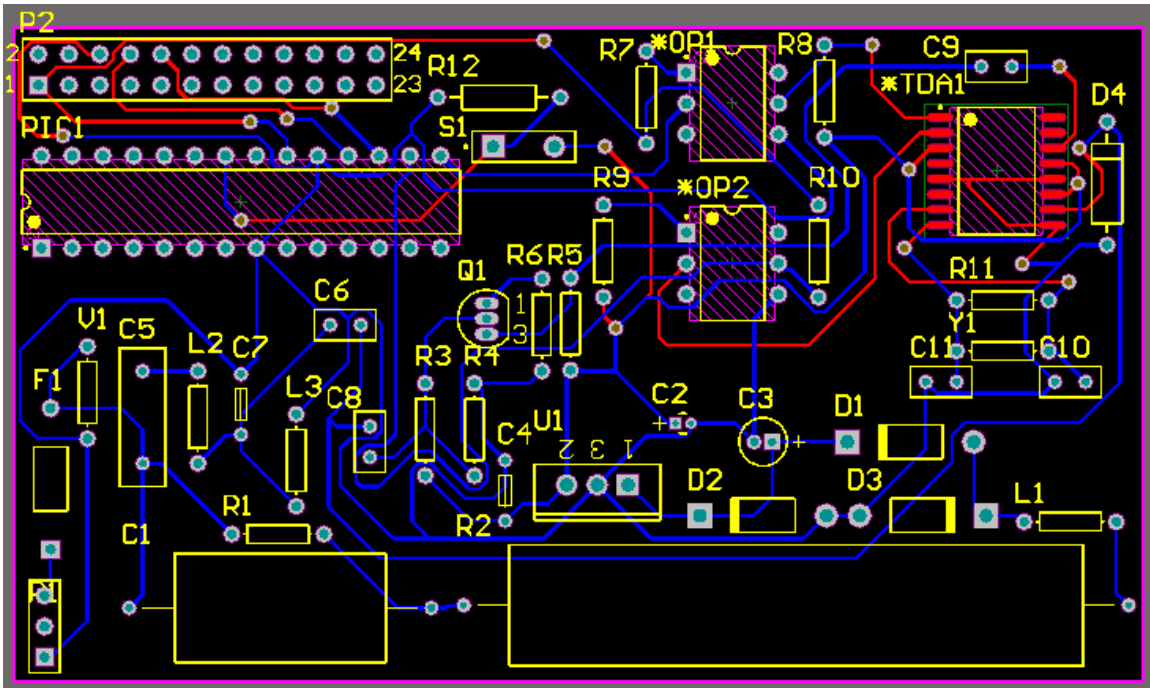
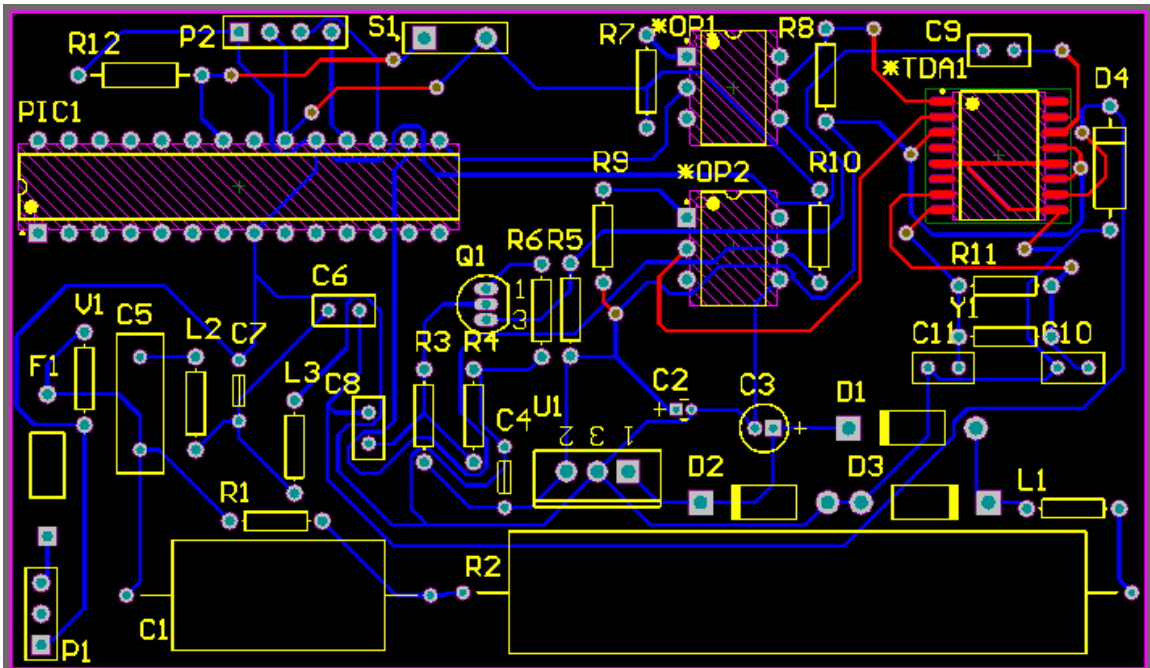


FIGURA NO. 122. PCB MÓDULO DE COMUNICACIÓN CON DISPOSITIVOS.



XVI. GLOSARIO

AC: Acrónimo para “*alternating current*”, en español corriente alterna.

ARP: Del inglés *Address Resolution Protocol*, es el protocolo encargado encontrar una dirección hardware correspondiente a cada dirección IP dentro de una red.

ASK: Del inglés *Amplitude-Shift Keying*, hace referencia a una forma de modulación en la que los datos se representan mediante variaciones de amplitud de una onda portadora.

Baudio: Unidad de medición de datos utilizado en la transmisión de los mismos. La tasa de baudios indica la velocidad con la que los dispositivos (emisor y receptor) transmiten eventos de señalización; es decir la velocidad con la que pueden realizar un cambio en el dato transmitido.

Bits: Dígitos del sistema de numeración binario.

Bluetooth: El nombre hace referencia a una red inalámbrica que permite la comunicación entre distintos dispositivos mediante a un enlace por radiofrecuencia.

bps: Bits por segundo

Bticino: Compañía internacional de origen italiano que trabaja en el área de soluciones en domótica.

Byte: Unidad de información utilizada como múltiplo del bit, equivale a 8 bits.

CMOS: Acrónimo para “*Complementary Metal-Oxide Semiconductor*”. Familia lógica empleada en la construcción de circuitos impresos.

CPU: Del inglés *Control Processing Unit*, conocido también como procesador, es componente central de un computador encargado del procesamiento de datos y la interpretación de instrucciones.

DC: Acrónimo para “*direct current*”, en español corriente directa.

Debian: Sistema operativo libre basado en el núcleo Linux y las herramientas GNU utilizado para múltiples arquitecturas.

Demodulación: proceso de recuperar la señal original que fue modulada.

DHCP: De las siglas en inglés de *Dynamic Host Configuration Protocol*, es el protocolo encargado del direccionamiento dinámico IP.

Domótica: Conjunto de tecnologías capaces de automatizar dispositivos presentes en una vivienda.

Ethernet: Estándar internacional utilizado para la conexión de computadores en redes locales.

EUSART: *Enhanced Universal Synchronous Asynchronous Receiver Transmitter*

Frecuencia: cantidad de ciclos que se realizan por unidad de tiempo

FTP: Del inglés *File Transfer Protocol*, es un protocolo de redes que regula la transferencia de archivos en sistemas utilizando el protocolo TCP.

GNU: Acrónimo recursivo de GNU is Not Unix. Es un sistema operativo libre totalmente compatible con UNIX.

GPIO: Del inglés *General Purpose Input/Output*, se refiere a las salidas y entradas de un dispositivo que pueden ser utilizadas para propósitos generales.

GPU: De las siglas en inglés *Graphics Processing Unit*, es un coprocesador que se dedica al procesamiento de gráficos, permitiendo al CPU dedicarse a otras tareas.

Http: Del inglés *Hypertext Transfer Protocol*, es un protocolo desarrollado para la *World Wide Web* que define la sintaxis utilizada por los dispositivos conectados a la misma.

Inmótica: Conjunto de tecnologías capaces de automatizar dispositivos presentes en cualquier ambiente cerrado (oficinas, centros comerciales, etc.).

Intranet: Red local de computadores que utiliza los protocolos del Internet para compartir información.

iOS: Sistema operativo para dispositivos móviles Apple.

IP: Del inglés *Internet Protocol*, define un protocolo de comunicación basado en la transmisión de paquetes.

JSON: De las siglas de *JavaScript Object Notation*, representan un formato ligero para el intercambio de datos.

KNX: Estándar internacional de comunicación para sistemas domóticos e inmóticos.

LonWorks: Del inglés *Local Operation Network*, protocolo utilizado para la comunicación de sistemas domóticos e inmóticos.

Módem: Dispositivo que convierte señales digitales en analógicas y viceversa.

Modulación: Proceso de convertir una señal digital en una señal analógica para la transmisión de datos.

OOK: Acrónimo para “*On Off Key*”. Es otro nombre que recibe la modulación ASK.

ORM: *Object-Relational Mapping*, hace referencia la técnica realizada por ciertas aplicaciones para convertir datos entre lenguajes de programación orientados a objetos y el uso de bases de datos relacionales.

Período: Tiempo que toma en realizarse un ciclo completo.

Protocolo: Conjunto formal de convenciones que rigen el formato y control de las entradas y salidas entre dos dispositivos de comunicación.

Proxy: Dispositivo conectado a una red que tiene como función realizar acciones en representación de otro dispositivo conectado a la misma red.

Python: Lenguaje de programación interpretador orientado a objetos.

PWM: De las siglas en inglés *Pulse-Width Modulation*, es un tipo de modulación que modifica el ciclo de trabajo de señales periódicas.

RAM: Del inglés *Random Acces Memory*, es la memoria utilizada por las computadoras como memoria de trabajo. En ella se cargan las instrucciones a ejecutar por el CPU.

Raspbian: Sistema operativo basado en Debian diseñado específicamente para la computadora en placa *Raspberry Pi*.

SBC: De las siglas en inglés *Single-Board Computer*, hace referencia a computadores diseñados en placas reducidas y a bajos costos.

SMTP: *Simple Mail Transfer Protocol*, es un protocolo de red basado en texto y utilizado para el envío y recepción de correos electrónicos.

SoC: Del inglés *System On a Chip*, describe la fabricación de chips que integran todos (o la mayoría) de los módulos utilizados por un computador.

SPI: Del inglés *Serial Peripheral Interface*, se refiere a un protocolo de comunicación serial síncrono utilizado frecuentemente entre la comunicación de circuitos integrados.

SQL: De las siglas en inglés *Structured Query Languge*, es un lenguaje declarativo para el manejo de bases de datos relacionales.

Shell: Homónimo virtual de una terminal o consola de un computador. A través de ella se introducen y se muestran datos del sistema operativo.

TCP: Del inglés *Transmission Control Protocol*, es uno de los principales protocolos del internet utilizado para crear conexiones entre los dispositivos de la red.

TTL: Acrónimo para "*Transistor Transistor Logic*", en español "Lógica Transistor Transistor".

USART: Acrónimo para "*Universal Synchronous Asynchronous Receiver Transmitter*", en español "Transmisor Receptor Síncrono Asíncrono Universal". Es un controlador de puertos y dispositivos en serie.

URL: *Uniform Resource Locator*, es una cadena de texto utilizada para nombrar recursos en el internet.

USB: Estándar Universal de comunicación Serial que define el cableado y tipo de conectadores utilizados en computadores con entradas a este bus.

Wi-Fi: Mecanismo de conexión inalámbrica utilizado entre dispositivos y el internet.

X10: Protocolo de comunicación utilizado para sistemas domóticos e inmóticos.