



**UNIVERSIDAD TÉCNICA DEL NORTE**

**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y REDES DE  
COMUNICACIÓN**

**SISTEMA DOMÓTICO INCLUYENDO PLATAFORMAS DE  
HARDWARE Y SOFTWARE LIBRE PARA LA RESIDENCIA DE UNA  
PERSONA CON PARAPLEJIA**

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN  
ELECTRÓNICA Y REDES DE COMUNICACIÓN

**AUTOR: LUIS DIEGO ACOSTA HERRERÍA**

**DIRECTOR: ING. JAIME MICHILENA**

**Ibarra, Diciembre 2016.**



## UNIVERSIDAD TÉCNICA DEL NORTE

### BIBLIOTECA UNIVERSITARIA

## AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

### 1. IDENTIFICACIÓN DE LA OBRA

La UNIVERSIDAD TÉCNICA DEL NORTE dentro del proyecto Repositorio Digital Institucional, determinó la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información.

<b>DATOS DE CONTACTO</b>	
<b>CÉDULA DE IDENTIDAD</b>	1003333091-5
<b>APELLIDOS Y NOMBRES</b>	ACOSTA HERRERÍA LUIS DIEGO
<b>DIRECCIÓN</b>	SAN ANTONIO DE IBARRA- LOS NOGALES
<b>EMAIL</b>	diegoacosta.150@gmail.com
<b>TELÉFONO FIJO</b>	062550249
<b>TELÉFONO MÓVIL</b>	0995957579

<b>DATOS DE LA OBRA</b>	
<b>TITULO</b>	“SISTEMA DOMÓTICO INCLUYENDO PLATAFORMAS DE HARDWARE Y SOFTWARE LIBRE PARA LA RESIDENCIA DE UNA PERSONA CON PARAPLEJIA”
<b>AUTOR</b>	ACOSTA HERRERÍA LUIS DIEGO
<b>FECHA</b>	14 de febrero del 2017
<b>PROGRAMA</b>	PREGRADO
<b>TÍTULO POR EL QUE SE ASPIRA</b>	INGENIERÍA EN ELECTRÓNICA Y REDES DE COMUNICACIÓN
<b>DIRECTOR</b>	ING. JAIME MICHILENA

## **2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD**

Yo, Acosta Herrería Luis Diego, con cedula de identidad Nro. 100333091-5, en calidad de autor y titular de los derechos patrimoniales de la obra o trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en forma digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad de material y como apoyo a la educación, investigación y extensión, en concordancia con la ley de Educación Superior Artículo 144.

### 3. CONSTANCIAS

Yo, ACOSTA HERRERÍA LUIS DIEGO declaro bajo juramento que el trabajo aquí escrito es de mi autoría; y que este no ha sido previamente presentado para ningún grado o calificación profesional y que he consultado las referencias bibliográficas que se presentan en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondiente a este trabajo, a la Universidad Técnica del Norte, según lo establecido por las leyes de propiedad intelectual, reglamentos y normatividad vigente de la Universidad Técnica del Norte.

En la ciudad de Ibarra, Febrero del 2017

A handwritten signature in blue ink, appearing to read 'Luis Diego Acosta Herreria', with a large, stylized flourish above it.

Firma: .....

Nombre: Luis Diego Acosta Herreria

Cedula: 100333091-5



**UNIVERSIDAD TÉCNICA DEL NORTE**  
**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**

**CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A FAVOR DE LA**  
**UNIVERSIDAD TÉCNICA DEL NORTE**

Yo, Luis Diego Acosta Herrería, con cedula de identidad Nro. 100333091-5, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador artículos 4, 5 y 6, en calidad de autor del trabajo de grado con el tema: SISTEMA DOMÓTICO INCLUYENDO PLATAFORMAS DE HARDWARE Y SOFTWARE LIBRE PARA LA RESIDENCIA DE UNA PERSONA CON PARAPLEJIA. Que ha sido desarrollado con propósito de obtener el título de Ingeniero en Electrónica Redes de Comunicación de la Universidad Técnica del Norte, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia suscribo en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Técnica del Norte.

A handwritten signature in blue ink, appearing to read 'Luis Diego Acosta Herrería', is written over a white background.

Luis Diego Acosta Herrería

100333091-5

Ibarra, 14 de Febrero 2017



## UNIVERSIDAD TÉCNICA DEL NORTE

### FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

#### CERTIFICACIÓN

INGENIERO JAIME MICHILENA, DIRECTOR DEL PRESENTE TRABAJO DE  
TITULACIÓN CERTIFICA:

Que, el presente trabajo de Titulación “SISTEMA DOMÓTICO INCLUYENDO PLATAFORMAS DE HARDWARE Y SOFTWARE LIBRE PARA LA RESIDENCIA DE UNA PERSONA CON PARAPLEJIA” Ha sido desarrollado por el señor Luis Diego Acosta Herrería bajo mi supervisión.

Es todo en cuanto puedo certificar en honor de la verdad.

A handwritten signature in blue ink, appearing to read 'Jaime Michilena', is written over a horizontal line.

Ing. Jaime Michilena

100219843-8

DIRECTOR



**UNIVERSIDAD TÉCNICA DEL NORTE**  
**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**

**DECLARACIÓN**

Yo, LUIS DIEGO ACOSTA HERRERÍA declaro bajo juramento que el trabajo aquí escrito es de mi autoría; y que este no ha sido previamente presentado para ningún grado o calificación profesional y que he consultado las referencias bibliográficas que se presentan en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondiente a este trabajo, a la Universidad Técnica del Norte, según lo establecido por las leyes de propiedad intelectual, reglamentos y normatividad vigente de la Universidad Técnica del Norte.

En la ciudad de Ibarra, Febrero del 2017

EL AUTOR

A handwritten signature in blue ink, appearing to read 'Luis Diego Acosta Herrera', is written over a faint, circular stamp.

Luis Diego Acosta Herrera

C.I: 100333091-5

## **AGRADECIMIENTOS**

Agradecer a Dios con humildad por la oportunidad que muchos no tienen pero que a mí me ha brindado, a mis padres por todo el apoyo, la comprensión y por lo que me han enseñado, porque sin ellos jamás hubiera escrito estas líneas, a mi hermana por su ayuda en los momentos difíciles y a mi sobrino que sin saber siempre me ayudo con su amor y ternura.

Luis D. Acosta H.

## DEDICATORIA

Dedico este trabajo de titulación a mi familia, porque este es el verdadero reflejo de todo el esfuerzo que pusieron en educarme.

A mis maestros, por el tiempo que dedicaron en compartir sus conocimientos con nosotros los estudiantes y por toda la paciencia que nos tuvieron.

Y a todas las personas, familiares, amig@s y compañer@s que de una u otra forma me ayudaron a cumplir esta meta. Muchas gracias por compartir su tiempo conmigo.

"No vendas tu tiempo, no busques un trabajo... Emprende."

Luis D. Acosta H.

## ÍNDICE

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD	
TÉCNICA DEL NORTE.....	I
CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE GRADO A FAVOR DE LA	
UNIVERSIDAD TÉCNICA DEL NORTE.....	IV
CERTIFICACIÓN.....	V
DECLARACIÓN.....	VI
AGRADECIMIENTOS.....	VII
DEDICATORIA.....	VIII
RESUMEN.....	XX
ABSTRACT.....	XXI
PRESENTACIÓN.....	XXII
1.CAPITULO I.....	1
1.1.Problema.....	1
1.2.Objetivos.....	3
1.2.1.Objetivos General.....	3
1.2.2.Objetivos Específicos.....	3
1.3.Alcance.....	4
1.4.Justificación.....	5
2.CAPÍTULO II.....	8
INTRODUCCIÓN.....	8

2.1.ANÁLISIS DE HERRAMIENTAS LIBRES Y LOS ESTÁNDARES IEEE802.15.4 Y 802.11B/G.....	8
2.1.1.Herramientas libres.....	8
2.1.2.Placas de desarrollo ARM.....	9
2.1.3.Raspberry Pi.....	13
2.1.3.1.Procesadores ARM.....	15
2.1.4.Raspbian.....	16
2.1.4.1.Herramientas para el desarrollo de la aplicación web.....	17
2.1.4.2.Servidor web APACHE.....	18
2.1.4.3.Framework JQuery Mobile.....	20
2.1.4.4. Estructura de una página JQuery Mobile.....	22
2.1.5.Arduino.....	24
2.1.5.1.El software Arduino.....	25
2.1.5.2.El hardware Arduino.....	29
2.1.6.Estándar 802.15.4.....	32
2.1.6.1.Capa Física (PHY).....	33
2.1.6.2.Capa Enlace.....	36
2.1.6.3.Subcapa Control de acceso al medio MAC.....	37
2.1.6.4.Direccionamiento.....	38
2.1.6.5.Tipos de tramas.....	39
2.1.6.6.Supertramas.....	41
2.1.6.7.Tipos de dispositivos.....	43
2.1.7.Módulos de radiofrecuencia XBEE.....	45

	XI
2.1.8.Módulos de radio frecuencia NRF24L01.....	49
2.1.9.Estándar IEEE 802.11b/g.....	52
2.1.9.1.Capa Física.....	53
2.1.9.2.Capa enlace.....	54
2.1.9.3.Tipos de tramas.....	57
3.CAPÍTULO III.....	62
INTRODUCCIÓN.....	62
3.1.SITUACIÓN ACTUAL Y PARÁMETROS DEL DISEÑO.....	62
3.1.1.Características y situación de la persona con paraplejia.....	63
3.1.2.Características de la vivienda y áreas frecuentes.....	64
3.1.3.Parámetros del diseño.....	65
3.1.3.1.La interfaz de usuario.....	68
4.CAPÍTULO IV.....	70
INTRODUCCIÓN.....	70
4.1.DISEÑO DEL SISTEMA DOMÓTICO Y ANÁLISIS DE COBERTURA INALÁMBRICA.....	70
4.1.1.Planteamiento del diseño.....	71
4.1.1.1.Comunicación básica entre secciones de la red.....	73
4.1.2.Topología real de la red de dispositivos.....	75
4.1.2.1.Comunicación y relación entre elementos.....	78
4.1.3.Análisis de frecuencias utilizadas.....	81
4.1.4.El nodo central Raspberry pi.....	83
4.1.4.1.El Sistema Operativo.....	83

4.1.4.2.Instalación de herramientas y configuración.....	85
4.1.5.Librería RadioHead.....	86
4.1.6.Diseño y construcción de dispositivos.....	87
4.1.6.1.Nodo Principal.....	89
4.1.6.2.Los Nodos Intermedios.....	94
4.1.6.3.Nodo lámpara.....	98
4.1.6.4.Nodo manta.....	108
4.1.6.5.Nodo Puerta.....	111
4.1.6.6.Nodo Ventana.....	118
4.1.6.7.Nodo Cortina.....	121
4.1.7.La interfaz web.....	125
4.1.7.1.Partes y funcionamiento.....	125
4.1.7.2.El login o pantalla de ingreso.....	126
4.1.7.3.Página áreas.....	129
4.1.7.4.Página sala.....	131
4.1.7.5.Página cortina.....	135
4.1.7.6.Páginas ventana y puerta.....	137
4.1.7.7.Página manta.....	140
5.CAPÍTULO V.....	144
INTRODUCCIÓN.....	144
5.1.Implementación y pruebas de funcionamiento.....	144
5.1.1.Pruebas de comunicación entre los módulos xbee.....	145
5.1.2.Pruebas de nodo foco.....	148

5.1.3.Pruebas nodos cortina y ventana.....	150
5.2.Consideraciones e instalación en dispositivos conectados a Xbee A.....	154
5.2.1.Pruebas nodo puerta.....	160
5.2.2.Pruebas nodo manta.....	161
5.3.Consideraciones e instalación de dispositivos conectados a Xbee B.....	163
6.CAPÍTULO VI.....	165
INTRODUCCIÓN.....	165
6.1.Análisis costo beneficio.....	165
6.1.1.Entrevista al usuario.....	166
6.1.2.Costo total del sistema instalado.....	172
7.CAPÍTULO VII.....	174
7.1.Conclusiones y recomendaciones.....	174
7.2.Conclusiones.....	174
7.3.Recomendaciones.....	175
Referencias bibliográficas.....	177
Glosario.....	179
ANEXOS.....	181

## Índice del Figuras

Figura 1. Partes más relevantes de la placa Raspberry pi.....	15
Figura 2. Entorno de herramientas de trabajo web.....	20
Figura 3. Partes de una página web JQuery Mobile .....	22
Figura 4. Ejemplo de página web JQuery Mobile .....	23
Figura 5. Descripción del IDE Arduino.....	25
Figura 6. Descripción programa básico en arduino.....	28
Figura 7. Partes más relevantes de arduino UNO.....	30
Figura 8. Capas definidas por IEEE 802.15.4.....	32
Figura 9. Interoperabilidad IEEE 802.15.4 con IEEE 802.11b/g.....	33
Figura 10. Sincronización de dispositivos y header PHY.....	35
Figura 11. Trama vista desde Subcapa MAC.....	36
Figura 12. Tipos de tramas vistas desde Subcapa MAC.....	41
Figura 13. Tipos de tramas vistas desde Subcapa MAC.....	42
Figura 14. Xbee S1.....	46
Figura 15. Módulo NRF24L01.....	49
Figura 16. Diagrama nodo escondido.....	56
Figura 17. Trama IEEE 802.11.x.....	59
Figura 18. Características del sistema.....	71
Figura 19. Comunicación básica entre secciones de la red.....	73
Figura 20. Topología lógica de la red.....	75
Figura 21. Topología física de la red.....	76

Figura 22. Red implementada.....	77
Figura 23. Comunicación y relación entre elementos.....	78
Figura 24. Frecuencias utilizadas.....	82
Figura 25. Descarga de Raspbian.....	84
Figura 26. Partes de dispositivos electrónicos.....	87
Figura 27. Partes Nodo Principal.....	89
Figura 28. Fuente de alimentación nodo principal.....	90
Figura 29. Fuente de alimentación nodo principal.....	92
Figura 30. Diagrama de conexión eléctrico nodo principal.....	93
Figura 31. Partes de Nodos Intermedios.....	94
Figura 32. Diagrama de flujo nodos intermedios.....	95
Figura 33. Shield NRF24L01.....	96
Figura 34. Fuente nodos intermedios.....	97
Figura 35. Partes de nodo lámpara.....	98
Figura 36. Diagrama de flujo nodo lámpara.....	99
Figura 37. Fuente reactiva.....	100
Figura 38. Partes de una fuente reactiva.....	101
Figura 39. Partes de una fuente reactiva.....	104
Figura 40. Circuito opto-acoplado.....	105
Figura 41. Resistencia de entrada en opto-acoplador.....	106
Figura 42. Resistencia de disparo en triac.....	107
Figura 43. Resistencia de disparo en triac.....	107
Figura 44. Partes nodo lámpara.....	108

Figura 45. Diagrama de flujo nodo manta.....	109
Figura 46. Nodo puerta.....	112
Figura 47. Diagrama de flujo nodo puerta.....	113
Figura 48. Driver para motor.....	115
Figura 49. Cargador HP.....	117
Figura 50. Módulo LM2596.....	118
Figura 51. Diagrama eléctrico nodo puerta.....	118
Figura 52. Partes nodo ventana.....	119
Figura 53. Diagrama de flujo nodo ventana.....	120
Figura 54. Diagrama de flujo nodo ventana.....	121
Figura 55. Nodo cortina.....	122
Figura 56. Diagrama de flujo nodo cortina.....	123
Figura 57. Diagrama eléctrico nodo cortina.....	124
Figura 58. Diagrama interfaz web.....	126
Figura 59. Pagina login o pantalla de ingreso.....	127
Figura 60. Pagina áreas.....	129
Figura 61. Pagina sala.....	131
Figura 62. Pagina cortina.....	135
Figura 63. Paginas ventana y puerta.....	137
Figura 64. Pagina manta.....	140
Figura 65. Topología implementada.....	145
Figura 66. Ubicación de la placa Raspberry pi.....	146
Figura 66. Respuesta módulo xbeeA.....	147

Figura 67. Respuesta módulo xbeeB.....	148
Figura 68. Diagrama de conexión lámpara.....	149
Figura 69. Ubicación lámpara.....	149
Figura 70. Pruebas de conectividad lámpara.....	150
Figura 71. Ubicación nodos cortina y ventana.....	151
Figura 72. Partes nodo cortina.....	151
Figura 73. Partes nodos ventana.....	152
Figura 74. Prueba de conectividad ventana.....	152
Figura 75. Prueba de conectividad cortina.....	153
Figura 76. Ubicación nodos conectados a Xbee A instalados en el domicilio.....	154
Figura 77. Implementación nodo principal.....	155
Figura 78. Método de sujeción nodo lámpara.....	156
Figura 79. Implementación nodo lámpara.....	156
Figura 80. Punto eléctrico para nodos ventana y cortina.....	157
Figura 81. Implementación nodo ventana.....	157
Figura 82. Partes nodo ventana.....	158
Figura 83. Interruptores de fin de camino en nodo ventana.....	158
Figura 84. Soportes y polea en nodo cortina.....	159
Figura 85. Partes de nodo cortina implementado.....	159
Figura 86. Ubicación nodo puerta.....	160
Figura 87. Respuestas nodo puerta.....	161
Figura 88. Ubicación nodo Manta.....	161
Figura 89. Prueba de conectividad manta.....	162

Figura 90. Partes nodo puerta.....163

Figura 91. Interruptores de fin de camino nodo puerta.....163

Figura 92. Ubicación de todos los nodos en el domicilio.....164

**Índice de tablas**

Tabla 1. Comparación entre placas ARM.....	11
Tabla 2. Características mas relevantes Arduino UNO.....	31
Tabla 3 .- Módulos basados en IEEE802.15.4.....	44

## RESUMEN

El sistema implementado permite el control de puertas, ventanas, cortinas y una manta eléctrica en el domicilio de una persona con paraplejia. Para su desarrollo se estudiaron herramientas libres que permitan su construcción, determinando usar conexiones inalámbricas en las bandas ISM y adecuar su coexistencia con otras tecnologías similares.

Con el fin de obtener un sistema multiplataforma se orienta la aplicación a la web, que permite acceder a la interfaz desde cualquier dispositivo a través de un navegador, tanto desde la WLAN como desde Internet.

Se diseña y construye el hardware y software necesario adecuándolo a las instalaciones del domicilio, siempre tomando en cuenta que los dispositivos intervenidos puedan funcionar dual-mente, tanto con el sistema como manualmente.

Finalmente para un correcto funcionamiento se procede a la instalación y corrección de errores, garantizando así un sistema funcional.

El producto final obtenido es un sistema que reduce el tiempo de incomodidad en la interacción con los dispositivos intervenidos, mejorando el desenvolvimiento de la persona beneficiada.

## **ABSTRACT**

The implemented system allows the control of doors, windows, curtains and an electrical blanket in the domicile of a person with paraplegia. For his development there were studied free tools that allow his construction, determining to use wireless connections in the bands ISM and to adapt his coexistence with other similar technologies.

In order to obtain a system multiplatform orientates the application to the web, which it allows to accede to the interface from any device across a mariner, both from the WLAN and from Internet.

The hardware and necessary software is designed and constructs adapting it to the facilities of the domicile, always bearing in mind that the controlled devices could work dually, so much with the system as manually

Finally for a correct functioning one proceeds to the installation and correction of mistakes, guaranteeing this way a functional system.

The final obtained product is a system that reduces the time is inconvenience in the interaction with the controlled devices, improving the development of the benefited person.

## **PRESENTACIÓN.**

El presente trabajo abarca un estudio sobre el diseño y implementación de un sistema domótico en la residencia de una persona con paraplejia con el objetivo de mejorar su calidad de vida, utilizando la tecnología para mejorar su desenvolvimiento en el hogar reduciendo su dependencia de otras personas. El sistema propuesto que permite esto; es un sistema basado en software y hardware libre, específicamente utilizando las herramientas: Raspberry pi y Arduino las cuales son placas electrónicas libres, la primera es un computador de tamaño reducido con capacidad suficiente para ejecutar un sistema operativo GNU/LINUX y los servicios básicos que pueden instalarse en este, la segunda placa Arduino es un PCB que está conformada por un microcontrolador que permite recolectar información del entorno por medio de sensores permitiendo manejar actuadores; todo esto en tiempo real.

El trabajo en conjunto de estos dispositivos con la ayuda de más elementos de comunicación inalámbrica que trabajan en la bandas ISM, permiten tener una plataforma robusta para la construcción de un sistema domótico que se adapta a las necesidades de la persona con discapacidad, que será controlado desde un dispositivo móvil (smartphone) mediante una aplicación orientada a la web, facilitando al usuario el acceso desde una red local o desde Internet.

# 1. CAPITULO I

## Antecedentes

### 1.1. Problema

La discapacidad es una realidad humana que afecta a un gran porcentaje de personas limitando su desenvolvimiento en diferentes entornos. En el Ecuador según la “Misión Solidaria Manuela Espejo” se ha logrado determinar que existe un 2.43% de personas con algún tipo de discapacidad, esta información puede ser observada en la página web de la vicepresidencia de la república (Espejo 2013). Todas las personas que sufren algún tipo de discapacidad tiene necesidades especiales, tales necesidades deben ser solventadas de alguna forma para mejorar así su calidad de vida y reducir su nivel de dependencia.

Una persona con paraplejia presenta limitaciones físicas que hacen de su desenvolvimiento diario sea incómodo y con dificultad. Inclusive en su hogar realizar actividades comunes como, abrir una ventana o correr una persiana puede ser difícil para la persona desde una silla de ruedas; peor aún si la persona se encuentra recostada la situación se vuelve más dificultosa por todo lo que conllevaría. El sentirse dependiente de otras personas para realizar tareas básicas en el hogar, hace que la persona parapléjica tenga una baja autoestima y se sienta insegura. En la actualidad existen soluciones que mediante sistemas domóticos permiten un mejor desenvolvimiento, pero aun costo muy elevado. En su gran mayoría las personas que sufren paraplejia y sus familias, no tienen los recursos para adquirir un sistema como el indicado y como consecuencia se tiene la dependencia para poder desenvolverse dentro del hogar.

El desarrollo del software y hardware libre en la actualidad, hace que se tengan

herramientas de desarrollo avanzadas, con las cuales se pueden solucionar problemas como este. Placas del tamaño de una tarjeta de crédito, con la potencia de un ordenador y con sistema operativo Linux como la Raspberry Pi, son accesibles a un precio muy económico. Además, en combinación con proyectos como Arduino y tecnologías inalámbricas hacen que se puedan desarrollar soluciones específicas a problemas puntuales. Si a todo esto se suman los beneficios de la Internet, las soluciones orientadas a la web y los dispositivos móviles; se puede concebir un sistema de domótica funcional,

de calidad y económico especialmente dirigido a personas parapléjicas. Con lo cual se mejoraría su calidad de vida y reduciría considerablemente su nivel de dependencia aumentando su seguridad en el hogar.

La sociedad tiene el deber de dar soluciones a este tipo de problemas, y como se ha explicado en nuestro país no son pocas las personas que sufren algún tipo de discapacidad. Por muchos años este sector se ha encontrado en el olvido, pero hoy se cuenta con las herramientas, la tecnología y el conocimiento para poder desarrollar soluciones y mejorar su calidad de vida. El proyecto se enfoca en una persona parapléjica y en el desarrollo de un sistema domótico accesible, funcional, adaptable y económico que le permitirá desenvolverse de una mejor manera en su hogar.

## **1.2. Objetivos**

### **1.2.1. Objetivos General.**

- Mejorar la calidad de vida de una persona con paraplejia mediante la implementación de un sistema domótico incluyendo plataformas de hardware y software libre en su residencia.

### **1.2.2. Objetivos Específicos.**

- Analizar los estándares IEEE 802.15.4, 802.11b/g y las herramientas libres Raspberry Pi y Arduino para su uso conjunto en las bandas de frecuencia ISM.
- Identificar las áreas de la vivienda más críticas, así como las actividades en que la persona con paraplejia tiene problemas para desenvolverse, con el fin de identificar los dispositivos a controlar.
- Diseñar el hardware complementario para el sistema y redes de datos y de control para los diferentes dispositivos electrónicos a utilizarse, en la residencia de la persona con paraplejia.
- Diseñar la interfaz gráfica del sistema orientada a la web, compatible con dispositivos móviles que permita el acceso, control y monitoreo de los dispositivos; tanto localmente como a través de Internet.
- Implementar el sistema domótico, realizar pruebas para garantizar el correcto funcionamiento y corregir posibles fallos.
- Elaborar manuales de: usuario para la persona que utilizarán el sistema y administrador con el fin de dejar una documentación técnica para otro posible

administrador.

- Realizar un análisis costo beneficio con el fin de mostrar las facilidades que presta el sistema a la persona con paraplejía en base a su costo.

### **1.3. Alcance**

Para desarrollar el sistema se pretende hacer un estudio de las áreas más críticas del hogar de la persona con paraplejía, esto para determinar cuáles son sus verdaderas necesidades y en qué actividades tiene problemas de desenvolvimiento, tomando en cuenta las áreas en la cuales las limitaciones motrices de la persona no le permitan interactuar con las cosas.

Una vez identificados los puntos críticos se procederá con el desarrollo de la solución, para lo cual se utilizarán mayoritariamente herramientas libres; con el fin de no incrementar costos en licencias y de hacer el sistema lo más económico posible.

Para mejorar el desenvolvimiento de la persona se distribuirán actuadores en las áreas críticas identificadas, y se implementará el hardware complementario para que puedan funcionar adecuadamente, para el control de los actuadores se hará uso del estándar IEEE 802.15.4. Se realizará el estudio de cobertura e instalación de una red de datos basa en el estándar IEEE 802.11b/g, la cual servirá para la interconexión de dispositivos móviles (Smartphone), que a través de un nodo central basado en hardware y software libre podrá controlar los actuadores ubicados en la residencia de la persona con paraplejía. Para complementar el sistema y darle al usuario la capacidad de acceder remotamente se interconectará el nodo central con la Internet.

El sistema tendrá la capacidad de controlar dispositivos como; ventanas, persianas, luces, puertas que se identifiquen en las áreas críticas y una manta térmica. Todo esto controlado mediante una página web local especialmente diseñada para dispositivos móviles. Para que el usuario pueda acceder a la interfaz web deberá conectarse a un AP (Acces Point) local mediante Wi-Fi, a través de un dispositivo que soporte esta tecnología y tenga un navegador web o a través de Internet.

Se utilizará una solución orientada a la web ya que esta no dependerá de la marca del dispositivo con el que se acceda a la interfaz web ni el sistema operativo que tenga, teniendo así alternativas como smartphone's, tablet's, laptop's u ordenadores de escritorio. Con esto se tiene la ventaja de hacer la solución multiplataforma.

Para que el sistema pueda ser accesible desde Internet se utilizarán DNS's dinámicos, los cuales permiten la publicación de servicios en internet sin la adquisición de una ip pública esto con el fin de no aumentar el costo de la solución; se implementará un sistema de seguridad para que solo usuarios autorizados tengan acceso remoto a la interfaz del sistema.

Además, los elementos controlados por el sistema podrán utilizarse, tanto con la interfaz web como manualmente, esto con la finalidad de que no sea un sistema intrusivo, en el cual los demás miembros de la familia no puedan interactuar con las cosas de la vivienda.

#### **1.4. Justificación**

Uno de los sectores más vulnerables de la sociedad es el grupo de personas con discapacidad, dicho sector ha permanecido descuidado y olvidado por mucho tiempo. Sin embargo, en nuestro país iniciativas como la “Misión Solidaria Manuela Espejo” ha cobrado

fuerzas ayudando a estas personas en lo indispensable. Proyectos como este deben ser apoyados por la sociedad, ya que ahora se tienen en el panorama nacional la existencia de este problema y es su deber solventar sus necesidades, para que puedan tener una mejor calidad de vida.

Las limitaciones físicas que tienen las personas con paraplejia, son una realidad diaria en el hogar, causando; inseguridad, dependencia y baja autoestima en su propia vivienda. Para una persona con limitaciones motrices puede ser frustrante realizar tareas básicas en el hogar y depender de otra persona para estas tareas. Esta no es una solución.

Hoy se cuentan con diferentes herramientas para poder solucionar problemas como este. El uso de dispositivos tecnológicos en la vida diaria es una realidad, permitiéndonos realizar tareas que antes solo eran posibles desde un ordenador convencional. Porque no utilizar estas ventajas para solucionar problemas como el mencionado anteriormente y mejor aún con el gran desarrollo que han tenido el software y el hardware libre en los últimos años. El uso de placas de desarrollo como las Raspberry Pi y Arduino, permiten tener una plataforma de hardware y software completa, con muchísimas posibilidades de aplicaciones que pueden adaptarse a diferentes necesidades en entornos diversos. Además, las elecciones de herramientas libres permiten obtener soluciones, funcionales, escalables, adaptables y sobre todo económicas; ya que no es necesario adquirir licencias para su uso o modificación, como es el caso de las herramientas propietarias.

El uso de Raspberry Pi en este proyecto tiene algunas ventajas como; la funcionalidad de un ordenador con sistema operativo Linux en un tamaño muy reducido, la comunidad que se encuentra tras este dispositivo y su constante desarrollo, los diferentes accesorios de

hardware que se crean constantemente, su bajo consumo energético y la posibilidad de instalar diferentes servicios orientados al Internet. Para el control de hardware se tienen herramientas que llevan aún más tiempo de desarrollo, como la plataforma Arduino que presenta ventajas similares a la Raspberry Pi al ser hardware libre. Arduino permite la interconexión de múltiples sistemas, por medio de adaptadores (Shields) que manejan diferentes tecnologías de comunicación haciendo que estos puedan trabajar conjuntamente. Además, con el uso de tecnologías inalámbricas como Wi-Fi y Zigbee, que trabajan en bandas reservadas internacionalmente ISM y la inclusión de dispositivos móviles como smartphone's, tablet's o laptop's se puede desarrollar una solución domótica completa orientada a la persona con paraplejia.

Por último, uno de los aspectos más importantes de la misión de la “UNIVERSIDAD TÉCNICA DEL NORTE” es la vinculación con la sociedad, por lo cual esta institución tiene la obligación de dar soluciones a problemas como el mencionado.

## **2. CAPÍTULO II**

### **INTRODUCCIÓN**

En el presente capítulo se estudian las herramientas que se utilizan en el diseño y la construcción de un sistema domótico, que permite el mejor desenvolvimiento de una persona con paraplejia en su vivienda. Dicho sistema se compone tanto de hardware como de software, que permite el control de actuadores en la residencia por medio de una interfaz orientada a la web. Funciona gracias a la unión de diferentes subsistemas que cumplen con funciones reducidas, las cuales son controladas desde un dispositivo central. Por lo cual es necesario el estudio de las partes que forman al sistema por separado, para una mejor comprensión del mismo.

### **2.1. ANÁLISIS DE HERRAMIENTAS LIBRES Y LOS ESTÁNDARES IEEE802.15.4 Y 802.11B/G.**

#### **2.1.1. Herramientas libres.**

El software es un “conjunto de programas, instrucciones, y reglas informáticas para ejecutar ciertas tareas en una computadora” (RAE 2013).

Básicamente todas estas instrucciones se llevan a cabo por medio de circuitos electrónicos o hardware. Según la Organización (GNU 2013) cuando el código fuente de un software contempla libertades como: libre estudio, modificación, distribución y redistribución de los

cambios; y esta liberado con una licencia GPL<sup>1</sup> se lo considera software libre. Por otra parte cuando se habla de hardware libre, es notable hablar de ello con una comparación directa con el software libre en lo que se refiere a: estudio, modificación, distribución y redistribución del hardware. Pero compartir diseños de un dispositivo es mas complicado que compartir software. Al final, cabe destacar que no hay una definición exacta para “hardware libre”, solo esta presente la ideología de desarrollar algo y compartir el conocimiento con la sociedad para contribuir con ella.

### **2.1.2. Placas de desarrollo ARM.**

Actualmente las placas de desarrollo libres están en auge, con el avance de tecnologías como ARM es posible tener sistemas embebidos corriendo un sistema operativo en áreas no mayores a diez centímetros cuadrados. Estas herramientas permiten montar sistemas permanentemente conectados sin generar un gran consumo energético, sin ruidos y con un grado de procesamiento respetable.

La herramienta que se busca debe poder correr un servidor web que brinde su servicio a no mas de cinco usuarios y que este disponible las 24 horas del día; además, se debe poder conectar mediante una comunicación serial con otros dispositivos. Por funcionalidad la placa debe correr un sistema robusto que brinde estabilidad a la aplicación y permita una fácil configuración inicial para enfocar los esfuerzos en el desarrollo de la aplicación final, no es necesario un entorno gráfico para la configuración ya que se pretende usar configuración por consola. Finalmente como último requisito su consumo debe ser mínimo y su costo no debe representar una gran inversión.

---

1 GPL.- Licencia Pública General.

El objetivo de la placa en el sistema debe reunir estas características, para que pueda soportar el software que se va a instalar sin sub-utilizar recursos y tampoco saturar el procesamiento de la misma.

Con estos criterios se han tomado tres de las placas más populares para el desarrollo de aplicaciones de demanda reducida y los criterios a tomar en cuenta son los siguientes.

- *Capacidad de procesamiento.*- Se desea una placa que pueda manejar la carga que genere un servidor web, una sesión SSH y una comunicación serial simultáneamente, ya que este será un panorama común en el desarrollo.
- *Cantidad de memoria RAM.*- Este ítem es importante, por que la velocidad de ejecución depende directamente de la cantidad de memoria que los programas necesitan.
- *Interoperabilidad con periféricos.*- Al tratarse de una placa que pretende ser una parte vital de un sistema, debe brindar el mayor número de posibilidades en cuanto a conexión con periféricos. Esta placa va a permitir evaluar las diferentes alternativas que puedan ser útiles para desarrollar la aplicación.
- *Tipo de sistema operativo.*- De esta característica depende el tiempo de aprendizaje que el desarrollador debe invertir para comprender el funcionamiento y configuración del sistema operativo. El S.O debe permitir que el desarrollo se enfoque en la aplicación final y no debe ser un obstáculo, si no una ayuda para el fácil desarrollo de la aplicación final.
- *Soporte.*- El soporte por parte de los fabricantes y la comunidad de desarrollo es de vital importancia, esta característica está ligada con la calidad de la información que

se puede obtener de la herramienta, es conveniente que exista una comunidad activa y extensa en el desarrollo, esto permite tener una base sólida y en constante evolución que servirá como fuente de consulta y corrección de errores.

- *Precio.*- Finalmente el costo de esta herramienta no debe ser elevado ya que el sistema a desarrollarse tiene una causa social y debe por lo tanto ser accesible para el usuario.

Las placas elegidas son:

**Tabla 1.** Comparación entre placas ARM.

<b>Características</b>	<b>ODROID X</b>	<b>CUBIEBOARD</b>	<b>RASPBERRY PI</b>
Procesador	Exynos4412 4 núcleos 1,46 Ghz	Allwinner A10 1 núcleo 1Ghz	BCM2835 1 núcleo 700Mhz
Memoria	1 Gb DDR2	1 Gb DDR3	512 Mb
Interoperabilidad con periféricos	Ethernet 10/100 + USB3.0+USB2.0+ hdmi+UART+ GPIO	Ethernet 10/100 + USB 2.0+ sata +I2C+UART+SPI	Ethernet 10/100 + 2.0 USB 2.0 + HDMI + SPI +I2C+GPIO+UART
Sistemas operativos soportados	Android, Ubuntu	Android y Ubuntu	Debian, Fedora, Arch Linux, Slacware, Suse, RiscOS
Soporte	Moderado, comunidad reducida. Desarrollo enfocado en Android.	Pobre, comunidad reducida enfocada en Andorid y portar Linux.	Excelente, gran comunidad enfocada en el desarrollo de la placa y S.O a la medida.
Precio	130\$	49\$	35\$

**Fuente:**<http://www.gadgetoadicto.com/2012/10/cuadro-comparativo-placas-arm/>

De la tabla comparativa se pueden sacar algunas conclusiones, como se observa la tabla 1 contempla tres alternativas que se podrían utilizar para la aplicación requerida, pero tomando en cuenta los requerimientos mínimos que se necesitan para que la aplicación funcione de manera correcta, sin sobredimensionar los recursos de las mismas se llega ha los siguientes puntos:

- La placa Odroid X presenta la gran ventaja de un procesador potente con una memoria generosa, sus conexiones hacia el exterior son mas que suficientes para la aplicación. Las desventajas de esta placa están en el sistema operativo Android para el que fue construida y su elevado costo. Además se considera que se desperdician recursos al solo correr un servidor web para cinco usuarios y una comunicación serial. No se usa esta placa por lo mencionado.
- Continuando con la placa Cuibieboard, se puede observar que su procesador se reduce en capacidad con respecto a la primera alternativa pero la memoria RAM se conserva de igual tamaño mejorando su velocidad al ser DDR3, otra característica interesante es que ya se pueden instalar sistemas operativos basados en Linux, esto facilitará el desarrollo de la aplicación, finalmente su precio es atractivo. Sin embargo a pesar de presentar estas ventajas, se tiene en cuenta que los sistemas operativos para esta placa no están optimizados y se hace uso de adaptaciones de otras placas, dado como resultado inestabilidad. Por esta razón no se hace uso de esta placa en el proyecto.
- Al final se tiene la placa Raspberry Pi que cuenta con un procesador de núcleo simple, y una memoria modesta que representa la mitad de las otras alternativas, sin embargo tiene lo necesario para correr la aplicación requerida. La gran ventaja de esta placa es su desarrollo por parte de la comunidad, enfocado a la optimización del sistema operativo que utiliza, haciéndola una de las mas robustas para aplicaciones de alta disponibilidad. Su capacidad de proceso y memoria no se comparan con las placas revisadas anteriormente; pero sus características se encuentran adecuadas para este

tipo de aplicación. Además, presentan varias alternativas en cuanto a tipos de comunicaciones con periféricos externos, brindando mayor versatilidad en cuanto a posibilidades de uso. Finalmente su costo reducido y su soporte reflejado en su gran comunidad hacen que se decida usar esta placa en el proyecto.

### **2.1.3. Raspberry Pi**

Raspberry pi es una plataforma considerada un icono en el desarrollo del hardware y software libre; creado con el objetivo de incentivar la creatividad de los estudiantes en áreas como la electrónica, programación e informática. Raspberry pi es un SBC<sup>2</sup> de reducido tamaño con características propias de un computador, tiene la capacidad de ejecutar un sistema operativo GNU/Linux, entre sus características mas importantes destacan un procesador de 700Mhz con arquitectura ARM<sup>3</sup>, una memoria RAM de 512MB, comunicación Ethernet y USB; además de puertos de propósito general(GPIO) que permiten la interconexión con diferentes placas electrónica de expansión (Upton 2013).

Este SBC al considerarse un computador tiene la capacidad de interconexión con diferentes dispositivos y periféricos, haciendo posible mantener múltiples comunicaciones por medio de tecnologías como por ejemplo: I2C, RS232, USB y Ethernet.

La elección y utilización de la placa Raspberry pi se la ha echo por diversos motivos y ventajas entre las que se destacan.

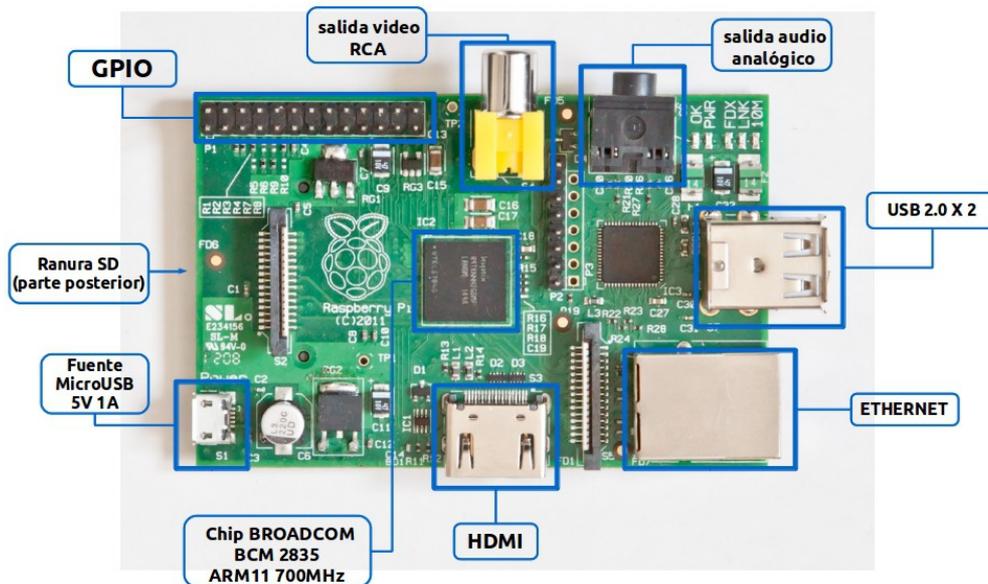
---

2 SBC.- Sigle Board Computer.

3 ARM.- Advanced RISC Machine (RISC.- Reduced Instruction Set Computer)

- Su bajo precio hace que la solución brindada no tenga un costo muy elevado.
- El consumo mensual eléctrico de esta placa esta en los 3,5[w], por lo cual es adecuada para un sistema de funcionamiento continuo, sin que el consumo eléctrico de la residencia se vea afectado mayormente.
- La comunidad de desarrollo que esta detrás de la Raspberry pi es impresionante y su constante desarrollo hace que se tenga siempre información actualizada, por ende el desarrollo de aplicaciones es ágil y rápido.
- Las diferentes alternativas que se tiene en sistemas operativos GNU/Linux. Raspbian, RiscOS, Raspbmc, Openelec, Arch y Pidora son algunos de los sistemas operativos que soporta esta placa.
- Su hardware, aunque no se puede comparar con un computador personal, es suficiente y capás de ejecutar programas para el control de periféricos y brindar servicios orientados a la web.

Se considera que para las necesidades del proyecto tanto en hardware como en software, Raspberry Pi es un elección idónea, pues se dispone de un gran abanico de herramientas de desarrollo de software; propias de un sistema operativo. Además, el consumo de recursos por parte de la aplicación propuesta no demanda una cantidad exagerada de procesamiento haciendo un uso normal y adecuado del hardware de la placa.



**Figura 1.** Partes más relevantes de la placa Raspberry pi.

**Fuente:** Fotografía y edición autor de la obra.

### **2.1.3.1. Procesadores ARM**

(Andrews 2004) hace un estudio sobre los procesadores ARM y el éxito de implementación en dispositivos móviles, tanto así que la mayoría de los dispositivos móviles poseen procesadores ARM en sus placas electrónicas. Estos procesadores ARM llamados así por sus siglas en inglés Advanced RISC Machine, son una variedad de chips que inicialmente fueron desarrollados por la empresa Acorn Computers en el año 1983, para luego continuar su desarrollo con la empresa que le dio su nombre, Advanced RISC Machine Ltd. Sus principales características son: manejo de instrucciones simples, tamaño pequeño, excelente respuesta en tiempo real y bajo consumo energético; ARM ha ganado fama de ser líder en el diseño pequeño y de bajo consumo.

Raspberry pi tiene como corazón un chip ARM fabricado por la empresa BROADCOM, que tiene un larga trayectoria en la fabricación de microcontroladores y chips. El chip

BCM2835 posee prácticamente casi todo lo necesario en su interior para funcionar como un computador, exactamente tiene: CPU<sup>4</sup>, GPU<sup>5</sup>, DSP<sup>6</sup>, SDRAM<sup>7</sup> y un puerto USB que por medio de un HUB externo integrado en la placa se transforma en dos puertos USB.

Es impresionante cómo la miniauturización de los dispositivos electrónicos han permitido llegar a tener prácticamente una computadora en la palma de la mano, este es el caso de la Raspberry pi que nos permite desarrollar aplicaciones a la medida sin la necesidad de recurrir a un gran presupuesto para llegar a obtener soluciones de calidad.

#### **2.1.4. Raspbian**

El sistema operativo que se utiliza en la Raspberry Pi es Raspbian, el cual está basado en Debian y por ende hereda sus cualidades robustas y estables. Según la Raspbian ORG (Raspbian, 2013), el S.O fue creado específicamente por la comunidad para el hardware de la Raspberry y cuenta con alrededor de 35 mil paquetes pre compilados listos para ser usados, además cuenta con un repositorio actualizado y un gestor de paquetes que se encarga de la instalación y configuración de los mismos.

Raspbian es un sistema operativo completo, donde se pueden instalar aplicaciones propias de un servidor, en el caso específico del sistema domótico se instala un servidor web donde se hospeda la interfaz, por medio de la cual se controlan los dispositivos remotos. Se puede

---

4 CPU.- Unidad Central de Proceso.

5 GPU.- Unidad de procesamiento gráfico.

6 DSP.- Procesador Digital de Señales.

7 SDRAM.- Memoria Sincrónica Dinámica de Acceso Aleatorio.

decir que Raspberry Pi junto con este S.O<sup>8</sup> son una plataforma compacta y económica en la cual se pueden desarrollar infinidad de proyectos que puedan sacar ventaja de un sistema GNU/Linux, una placa compacta, un consumo de energía reducido y varios tipos de puertos de comunicación con el exterior.

#### ***2.1.4.1. Herramientas para el desarrollo de la aplicación web.***

Para el desarrollar paginas web se debe disponer de diferentes herramientas que trabajan juntas para crear contenido web, a esto se le conoce como un marco de trabajo que a su vez permite desarrollar interfaces web con un estilo predefinido (Fernandez, 2013).

Una aplicación web es un conjunto de paginas web que se asocian por medio de enlaces. El contenido web es accesible por el usuario a través de un navegador y las páginas están alojadas en un servidor web. El usuario final visualiza la aplicación como si se tratase de una sola unidad pero en realidad cada página web tiene una conexión con otra por medio de un enlace; así en el servidor web existen varios archivos con código que se referencian unos con otros. Los contenidos mostrados al usuario a través de su navegador pueden ser dinámicos o estáticos, para poder interactuar con la aplicación web se opta por contenidos dinámicos, que se muestran al usuario haciendo uso de herramientas como PHP, Javascript, CCS y AJAX de las que se habla brevemente más adelante.

---

8 S.O.- Sistema operativo.

### **2.1.4.2. Servidor web APACHE**

Según la Apache ORG, el proyecto APACHE es un software de código abierto, de libre disposición y mantenido por una comunidad de desarrolladores dispersada por todo el mundo. El objetivo principal de este servidor web es ser seguro, eficiente y extensible; proporcionando servicios sobre el Protocolo de Transferencia de Hipertexto(HTTP<sup>9</sup>) y respetando sus estándares (Apache, 2013).

Apache a forjado una gran reputación en el mundo de los servicios web gracias a su gran robustez y estabilidad. Es un software que recibe actualizaciones constantemente y su desarrollo continúa.

Su forma extensible permite trabajar con módulos que pueden ser agregados en cualquier momento aumentando sus características y usabilidad. Los módulos permiten al desarrollador trabajar de una manera mas cómoda y rápida; permitiendo enfocar el servicio web según sus necesidades, es decir Apache es adaptable.

Este servidor web es parte fundamental en el desarrollo del sistema domótico, ya que este alberga y gestiona las peticiones hacia la interfaz web que permite la interacción con el entorno. A continuación se describen algunas características de los lenguajes de programación web utilizados.

- HTML.- Es el lenguaje de programación mas básico y fundamental para diseñar paginas web. Se basa en etiquetas las cuales dan formato a los contenidos de una página web. Los navegadores web interpretan el lenguaje HTML para ordenar los

---

9 HTTP.- Protocolo para la transferencia de hipertexto, permite la transferencia de información desde un servidor a un cliente para visualizar contenido en un navegador web.

contenidos para el usuario.

- CSS.- Define los estilos de el contenido de una página web por medio de hojas de estilos. Permite crear sombras, degradados, transiciones haciendo la página web visualmente más atractiva. Las hojas de estilos son archivos que se encuentran alojados en el servidor web y estos se asocian con las páginas donde se desee aplicar los estilos.
- PHP.- es un lenguaje de programación orientado a la web, se utiliza principalmente para construir scripts<sup>10</sup> que se ejecutan en el servidor, el código PHP puede ser incrustado directamente en el código HTML. El interprete PHP que se instala en el servidor genera código HTML que se visualiza en el navegador del usuario; es decir cuando existe código PHP dentro de una página web el interprete produce HTML a partir de las funciones definidas en PHP, siendo para el usuario transparente. Esta técnica permite obtener contenidos dinámicos en la página web ya que las funciones pueden obtener información de variables almacenadas en el sistema o dispositivos exteriores conectados a el mismo.
- Javascript .- otro lenguaje de programación utilizado para el desarrollo de contenido web dinámico. La principal diferencia con PHP es que este se ejecuta en el lado del cliente y se utiliza principalmente para la creación de eventos. Por ejemplo cuando el usuario interactúa con un control deslizante en su navegador, inmediatamente se ejecuta un evento que puede estar enlazado con una función específica o un script de PHP. De esta forma se puede cargar nueva información en la página web o ejecutar una función en el servidor.

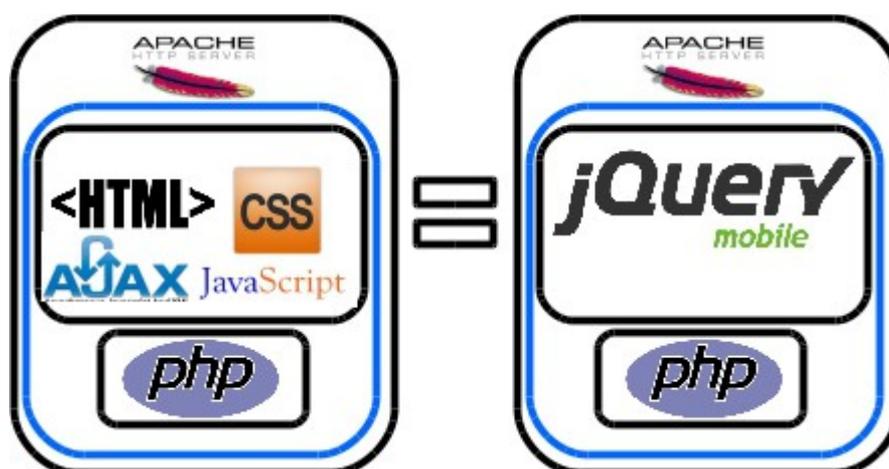
---

<sup>10</sup> Script.- conjunto de instrucciones en un archivo de texto para llevar a cabo una función específica.

- AJAX.- es la tecnología que permite cargar contenido en una página web de manera asíncrona, es decir cuando se necesita o se hace la petición. De esta manera no es necesario recargar o actualizar la página para mostrar nuevo contenido.

La aplicación web en el sistema domótico utiliza todas estas herramientas para generar interfaces amigables he intuitivas. El conjunto de estas herramientas permite la ejecución de funciones en el servidor con PHP y en el dispositivo del cliente mediante Javascript; para la transferencia de datos entre el servidor y el cliente se utiliza AJAX permitiendo obtener datos de una manera asíncrona.

#### 2.1.4.3. Framework Jquery Mobile



**Figura 2.** Entorno de herramientas de trabajo web.

**Fuente:** Diagrama diseñado por el autor de la obra con logos de páginas oficiales.

En el diagrama de la figura 2 se puede observar el entorno de trabajo de manera contenida que se utiliza para el desarrollo de la aplicación web, lo más destacable de este diagrama es la equivalencia entre los lenguajes HTML, Javascript, CSS y AJAX con Jquery Mobile.

Jquery Mobile es un framework o también conocido como un marco de trabajo, contiene las herramientas necesarias para el rápido desarrollo de aplicaciones web, de uso gratuito y constante desarrollo. Su principal característica es que está orientado para la creación de sitios web para móviles y posee una mejora progresiva, es decir tiene compatibilidad con dispositivos de gamas alta, media y baja; por ende es compatible con múltiples plataformas (Gallego, 2013).

Tiene una amplia variedad de elementos pre diseñados listos para ser usados, pudiendo personalizarlos en una sola línea de código con atributos llamados *roles*. Los *roles* son distintos atributos que se pueden aplicar a un elemento permitiendo que este cambie en su posición, color, tamaño, animación y comportamiento dentro de la página web.

El desarrollador puede manipular los eventos encadenándolos para llevar a cabo funciones específicas; por ejemplo, un control deslizante puede activar un evento por medio de Javascript, este a su vez puede crear una petición al servidor por medio de AJAX para recuperar información y visualizarla en un formulario, de esta manera la transmisión de información entre cliente y servidor, se da de una manera dinámica tan solo con que el usuario deslice un control.

El tiempo empleado para aprender a utilizar el framework es muy corto ya que su sintaxis es intuitiva y fácil de comprender.

Jquery Mobile permite que el desarrollador se concentre en el diseño del sitio web mientras el framework se encarga de la adaptación de los contenidos en el dispositivo del cliente. Como consecuencia se tiene que los contenidos siempre se visualizan de una manera correcta sin importar el tamaño de la pantalla en el dispositivo final.

#### 2.1.4.4. Estructura de una página JQuery Mobile.

Para una mejor comprensión de la estructura misma de una página web observe la figura 3.

```

<html>
<head>
  <!--enlace de archivos jquery y css-->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1"> ①
  <link rel="stylesheet" type="text/css" href="jquery.mobile-1.3.2.css"> ②
  <script type="text/javascript" src="jquery-1.10.2.min.js"></script>
  <script type="text/javascript" src="jquery.mobile-1.3.2.js"></script>
</head>
<body> ④
  <!--Cuerpo-->
  <div data-role="page" data-theme="c">
    <!--cabecera-->
    <div data-role="header"> ⑤
      <h2>Ejemplo</h2>
    </div>

    <!--contenido-->
    <div data-role="content" data-theme="c"> ⑥
      <div>
        <label data-inline="True">Contenido del sitio</label>
      </div>
      <select name="flag2" id="flag2" data-role="slider">
    </div>

    <!--pie de pagina-->
    <div data-role="footer"> ⑦
      <p>Pie de página</p>
    </div>
  </body>
</html>

```

**Figura 3.** Partes de una página web JQuery Mobile .

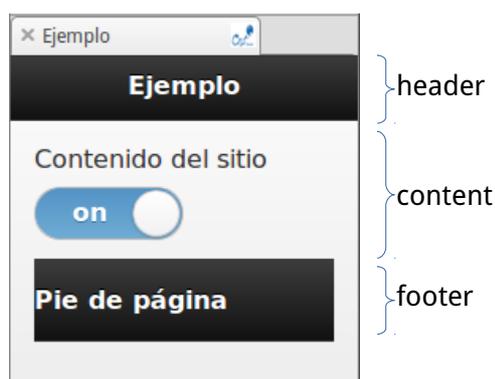
**Fuente:** Código y diagrama realizados por el autor de la obra.

Las partes fundamentales de una página hecha con JQuery Mobile son las siguientes:

- **Head.-** Se identifica con la etiqueta `<head>`, es la parte donde se enlazan los archivos necesarios para trabajar con JQuery y las hojas de estilos CSS, en la figura 3 están identificados con el número dos. La etiqueta “viewport” y su atributo `"width=device-width, initial-scale=1"` se encarga de acoplar la interfaz a la pantalla del dispositivo del usuario para que se vea correctamente, en la figura 3 señalado con el número uno.
- **Body.-** Señalado en la figura 3 con el número cuatro, dentro de la etiqueta `<body>` se ingresa todo el contenido de la pagina web. Aquí se especifican los diferentes

elementos predefinidos por JQuery Mobile a través de roles, básicamente existen tres elementos por los que se compone un sitio web; *header*, *content* y *footer*.

- *Header*.- especificado con la etiqueta `<div data-role = "header">` y señalado en la figura 3 con el número cinco, genera una barra en la parte superior de la página web en donde se pueden incrustar herramientas como por ejemplo botones.
- *Content*.- se lo identifica por medio de la etiqueta `<div data-role = "content">` y en la figura 3 corresponde al número seis, es la parte del sitio web donde se pone el contenido de la página, aquí se pueden incrustar los diferentes elementos definidos por el framework JQuery Mobile, como cajas de texto, imágenes, controles, etc.
- *Footer*.- Identificado por la etiqueta `<div data-role = "footer">` en la figura 3 se puede ver que corresponde al número siete. Esta etiqueta genera un barra en la parte inferior de la página web similar al la que genera la etiqueta *header*, en esta barra también se pueden incrustar herramientas que añadan funcionalidad al sitio web.



**Figura 4.** Ejemplo de página web JQuery Mobile .

**Fuente:** Propia.

En la figura 4 se observa resultado del código fuente de la figura 3, reflejado a través de un navegador web.

En la creación de aplicaciones web, el desarrollador tiene a su disposición infinidad de herramientas para llegar a sus objetivos, es el caso del sistema domótico se opta por el uso del framework descrito por su facilidad de uso, su gran biblioteca de elementos y su fácil interoperabilidad con otros lenguajes de programación.

### **2.1.5. Arduino**

Arduino es una placa de desarrollo electrónico dirigida a todo tipo de personas que estén interesadas en el desarrollo de aplicaciones electrónicas compuestas por software y hardware (Banzi, 2010).

Arduino el documental (2010), describe como empezó esta placa de desarrollo en el instituto Italiano Ivera en el año 2005, dirigida a los estudiantes del instituto como una alternativa económica en el desarrollo de aplicaciones electrónicas.

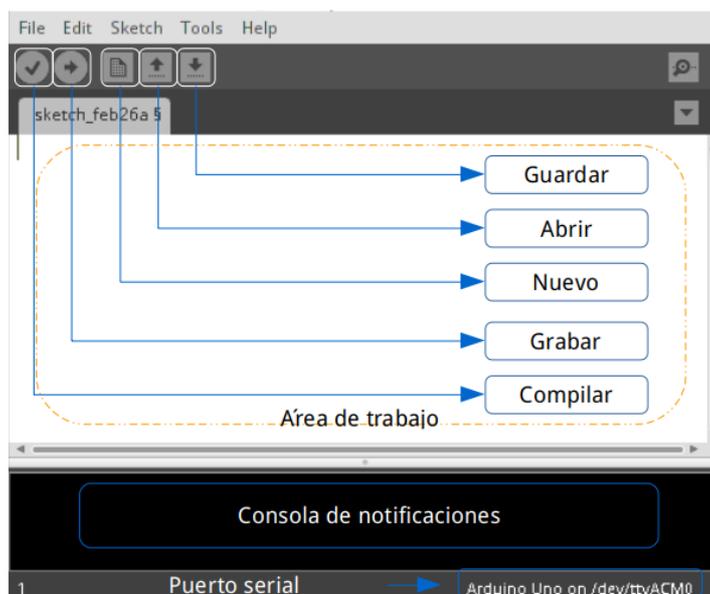
Hernando Barragán un estudiante colombiano en el mismo año, desarrolló una placa electrónica multipropósito basada en un microcontrolador que se puede programar por medio de un programa llamado Wiring creado por el mismo. Wiring es producto del estudio del lenguaje de programación Processing para hardware, que realizaba Barragán como proyecto de tesis.

Tomando como base Wiring y con la llegada de David Cuartielles que en ese entonces también trabajaba en el instituto como docente, se decide hacer de Arduino un proyecto de código abierto para que mas personas se involucren y puedan colaborar en el desarrollo del mismo. Al poco tiempo Banzi y Cuartielles crean la primera placa arduino y su entorno de desarrollo (IDE) llamado por el mismo nombre.

Como consecuencia de todo este proceso, varias personas con diferentes habilidades vieron en Arduino un proyecto con mucho potencial, con una filosofía nueva la cual trata de incluir a cualquier persona y hacerla parte del proyecto mismo, permitiéndoles desarrollar sus propios proyectos sin la necesidad de tener conocimientos previos en electrónica o programación. Por eso Arduino es ideal para el rápido desarrollo de aplicaciones basadas en microcontroladores, gracias a su gran comunidad, ejemplos de programación y toda la documentación que esta disponible en la web.

### 2.1.5.1. El software Arduino

Arduino IDE es un entorno de desarrollo integrado de código abierto disponible para los sistemas operativos mas utilizados: Windows, Linux y Mac. Esta basado en el lenguaje de programación Processing.



**Figura 5.** Descripción del IDE Arduino.

**Fuente:** IDE Arduino.

Este IDE cuenta con multitud de diferentes opciones para el fácil desarrollo de código; pero el objetivo de este trabajo no es describir todas y cada una de sus características ya que sería muy extenso y exhaustivo para el lector. Por eso solo se describirán las partes más importantes del entorno de desarrollo.

Como se puede observar en la figura 5 el software es muy intuitivo y es fácil de empezar a usar. Se señalan algunas de las partes básicas de Arduino IDE que se describen a continuación:

- *Guardar*.- El botón sirve para guardar los últimos cambios echos en el código, creando un archivo con extensión .ino dentro de la carpeta llamada arduino, la cual se puede ubicar en cualquier directorio. Para modificar la ubicación de la carpeta se debe acceder al menú *Archivo* y luego a *Preferencias*.
- *Abrir*.- Sirve para acceder rápidamente a todos los proyectos guardados previamente y además también permite abrir los ejemplos con los que viene el software. Esto permite al desarrollador tener guías en la escritura de código y la opción de probar fragmentos de código arduino en sus propios proyectos.
- *Nuevo*.- Abre una nueva ventana con un espacio de trabajo en blanco para la creación de un nuevo proyecto.
- *Grabar*.- Para utilizar esta opción previamente se debe seleccionar una placa Arduino en el menú *Herramientas* y luego *Tarjeta*. Además se debe elegir el puerto serial en el cual esta conectada la tarjeta Arduino en el menú *Herramientas* y posteriormente *Puerto Serial*. El botón Grabar permite compilar el código y verificar si existe algún tipo de error en el mismo, en caso de una compilación con errores se puede obtener

una descripción en la consola del IDE ubicada en la parte inferior; si no se tiene errores el software procede a grabar en la tarjeta Arduino el programa compilado.

- **Compilar.**- Este botón permite comprobar si el código escrito es correcto y si hay errores los describe en la consola del IDE. A diferencia del botón Grabar este solo compila el programa y no lo graba en la tarjeta Arduino.

Una vez descrito rápidamente como funciona el software de programación ahora se pasa a explicar las dos partes fundamentales de un programa arduino.

Básicamente todos los programas arduino tienen dos partes fundamentales, la primera conocida como *Setup* en la cual se debe inicializar cualquier tipo de comunicación, indicar que puertos se utilizaran como entradas o salidas y configurar cualquier módulo conectado a la placa Arduino. Por ejemplo en esta sección se podría inicializar una comunicación serial a una velocidad de 9600 baudios y configurar los pines 2, 3 y 4 como entradas digitales.

La segunda parte fundamental en un programa arduino se llama *Loop* y es la sección de código que se repetirá indefinidamente luego de haber ejecutado el *Setup* del programa. Por ejemplo en esta parte se podría tener el código correspondiente para que se envíen los datos de un sensor de temperatura cada tres segundos por medio de una comunicación serial.

Un programa arduino permite la utilización de recursos comunes de programación como métodos o funciones para hacer el código mas legible y ordenado, además también se puede reutilizar el código escrito anteriormente por medio de librerías; cabe destacar que se pueden añadir infinidad de librerías descargadas de Internet desarrolladas por terceros, con tan solo copiarlas dentro de la carpeta Arduino indicada en el IDE (*Archivo* y luego *Preferencias*) en el subdirectorio Library. Esta gran facilidad permite por ejemplo utilizar un nuevo módulo de

comunicación inalámbrica solo con una librería, que en la mayoría de los casos la desarrolla la comunidad arduino o el propio fabricante del módulo. Así el desarrollador utiliza lo que ya esta creado y lo anexa a sus propios proyectos, solo estudiando la librería y como utilizarla, sin la necesidad de desarrollar todo desde cero.

En la figura 6 se muestran las partes del código de un programa arduino.

```
//-----//
//Programa para hacer parpadear un led conectado
//en el pin 13 de la placa ARDUINO
int led 13;// Asocia el pin 13 con una variable
//de tipo entero y con nombre "led"
//-----//
void setup(){
pinMode(led, OUTPUT);//Declara el pin 13 como salida
}
//-----//
void loop(){
digitalWrite(led, HIGH);//Pone en el pin 13 un 1L.
delay(200);//Espera 200 milisegundos
digitalWrite(led, LOW);//Pone en el pin 13 un 0L.
delay(200);//Espera 200 milisegundos
}
//-----//
```

Declaración de variables globales e Inclusion de librerías

Setup del programa, configuración de puertos, comunicaciones, módulos.

Loop del programa, es la Parte que se repite indefinidamente luego de ejecutar el Setup.

**Figura 6.** Descripción programa básico en arduino.

**Fuente:** Código basado en el programa Blink de los ejemplos del IDE Arduino.

Algunas consideraciones generales con respecto a la figura 6 son:

- Como todo lenguaje de programación arduino permite la inclusión de comentarios en el código, en la figura en cuestión se pueden observar de color gris y precedidos por un doble // .
- La declaración de variables puede hacerse en cualquier parte del código pero el desarrollador debe saber en que fragmentos del código puede utilizar estas variables;

por ejemplo si se declara una variable dentro del fragmento de programa *void setup()* la variable solo se podrá utilizar en este fragmento y no en otros como *void loop()*. Al declarar las variables al inicio del programa, estas pueden ser utilizadas desde cualquier parte del programa ya que son variables globales.

- La utilización de funciones o métodos en Arduino IDE también esta permitida, así se pueden declarar cuantas funciones se desee en el programa; otra opción similar es la utilización de métodos incluidos en las librerías, para esto la librería debe ser incluida al inicio del programa antes de la declaración de las variables globales, por ejemplo se podría incluir fácilmente la librería *LiquidCrystal* con la línea *#include <LiquidCrystal.h>* que permite la utilización de un display de cristal liquido y controlar los pines de Arduino en los que este va ha ser conectado.
- Por último es importarte estar pendiente de el tamaño de los programas ya que la capacidad del microcontrolador en el que se basa Arduino es limitada.

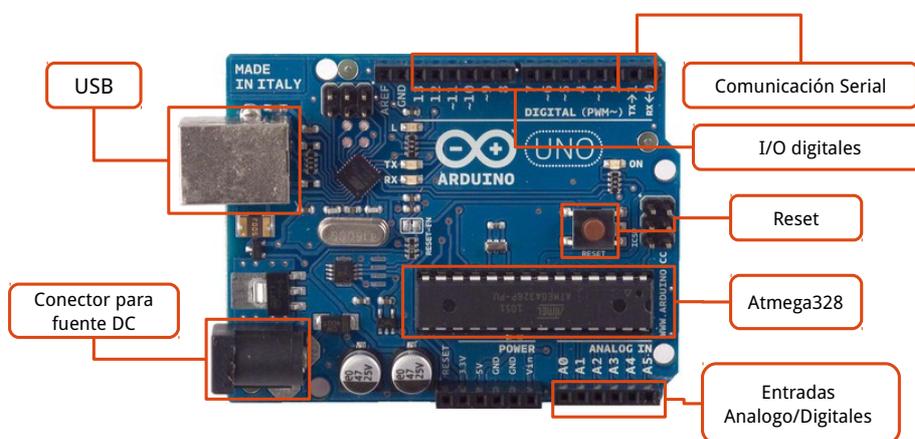
Si el lector desea profundizar en el lenguaje de programación arduino se recomienda visitar la página oficial de arduino, incluida en la bibliografía de este documento.

### **2.1.5.2. El hardware Arduino**

Al hablar de hardware Arduino implícitamente se esta hablando de las placas Arduino, existen una infinidad de placas Arduino que son compatibles con Arduino IDE, inclusive se pueden fabricar de forma artesanal en casa; esto es debido a que Arduino es hardware libre y cualquier persona o empresa lo puede fabricar sin la necesidad de un permiso. En este documento se describen las placas utilizadas en el desarrollo del sistema domótico, si el

lector desea informarse mas sobre las placas arduino disponibles y sus características puede visitar su página oficial.

Arduino UNO es una de las placas mas populares de esta plataforma, siendo una de las mas conocidas y utilizadas, cuenta con un microcontrolador atmega328<sup>11</sup> que viene previamente cargado con un bootloader<sup>12</sup>, el cual permite la fácil transferencia de programas directamente hacia el microcontrolador sin la necesidad de un grabador. La placa electrónica puede ser conectada a un ordenador por medio de USB y cuenta con un sistema de alimentación con autoselección entre tres fuentes; puede ser alimentada por USB, mediante un conector 2.1 mm de conexión DC con el positivo en su centro que soporta voltajes entre 7 a 12v y por último puede alimentarse directamente en dos pines de la placa señalados como *Vin* y *GND*, soportando voltajes entre 6.8 hasta los 12V. En la figura 7 y la Tabla 1 se pueden observar sus características.



**Figura 7.** Partes más relevantes de arduino UNO.

**Fuente:** Fotografía y edición autor de la obra.

<sup>11</sup> Atmega328.- microcontrolador del fabricante ATMEGA en el que se basa Arduino UNO.

<sup>12</sup> Bootloader.- es un pequeño programa previamente cargado en el microcontrolador y que se ejecuta inmediatamente luego de energizarlo, permite cargar programas sin un grabador de microcontroladores.

**Tabla 2.** Características mas relevantes Arduino UNO.

Microcontrolador	ATMEGA328
Fuente de alimentación	7 a 12 V
Pins I/O digitales	14 (6 utilizables como salidas PWM <sup>13</sup> )
Pins Analogos de entrada	6
Corriente de salida por pin	40 mA
Corriente de salida para pin 3.3V	50 mA
Memoria Flash	32 KB (0.5 KB usados por el bootloader)
SRAM <sup>14</sup>	2 KB
EEPROM <sup>15</sup>	1 KB
Velocidad de reloj	16 MHz

**Fuente:**<http://www.arduino.cl/int/caracteristicas.html>

Las entradas y salidas digitales pueden suministrar una máxima corriente de 40mA cada una y también pueden soportar la misma corriente de entrada, internamente cuenta con una resistencia PULL-UP<sup>16</sup> por cada I/O<sup>17</sup> con un valor comprendido entre 20Kohm a 50Kohm que se pueden activar mediante programación.

Existen pines que cumplen mas de una función y este es el caso de las 6 salidas PWM que pueden ser utilizadas con este fin o como una I/O normal. De manera similar los pines 2 y 3 se pueden utilizar para activar las interrupciones externas del microcontrolador. RX y TX que correspondientes a los pines 0 y 1, se pueden usar para realizar comunicaciones seriales con otros dispositivos y manejan voltajes TTL. Por último se tiene la capacidad de utilizar la

13 PWM.- Modulador por ancho de pulso.

14 SRAM.- Static Random Access Memory, Memoria Estática de Acceso Aleatorio.

15 EEPROM.- Electrically Erasable Programmable Read-Only Memory (memoria programable y borrable eléctricamente).

16 PULL-UP.- Configuración de una resistencia para suministrar un estado lógico definido.

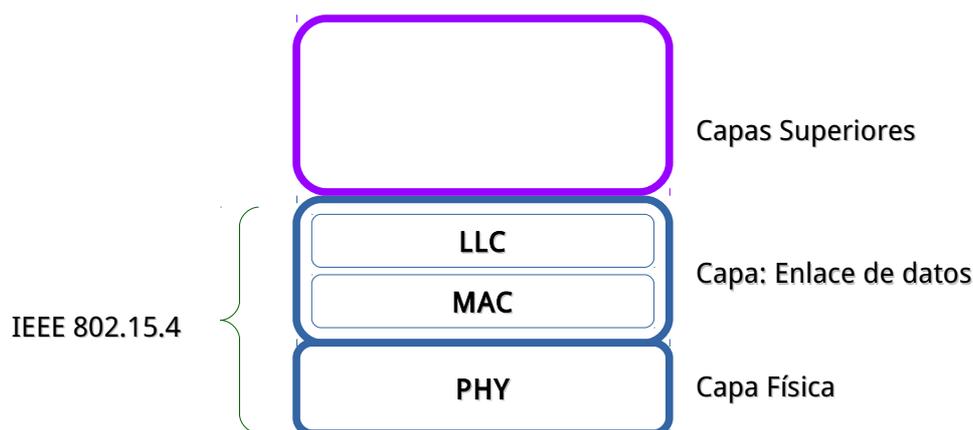
17 I/O.- Entrada o salida digital.

comunicación SPI por medio de los pines 10, 11, 12 y 13 que permite la transferencia de datos entre dispositivos y módulos que soporten este tipo de comunicación.

### 2.1.6. Estándar 802.15.4.

El estándar IEEE 802.15.4 se encarga básicamente de establecer una comunicación confiable mediante un enlace de radiofrecuencia (Caprile 2009). Define los niveles básicos de una WPAN(Wireless Personal Area Network) funcional, enfocándose principalmente en el consumo reducido de energía y un bajo procesamiento en sus dispositivos, dando como resultado una simpleza en la implementación; pero no por eso deja de tener características robustas presentes en otros estándares. Detección de errores, retransmisiones y acuses de recibo permiten obtener confiabilidad en la transmisión de información.

Este estándar define en concreto las dos primeras capas del modelo OSI para 802.15.4, en la figura 8 se puede evidenciar esto de una manera muy básica pero efectiva.



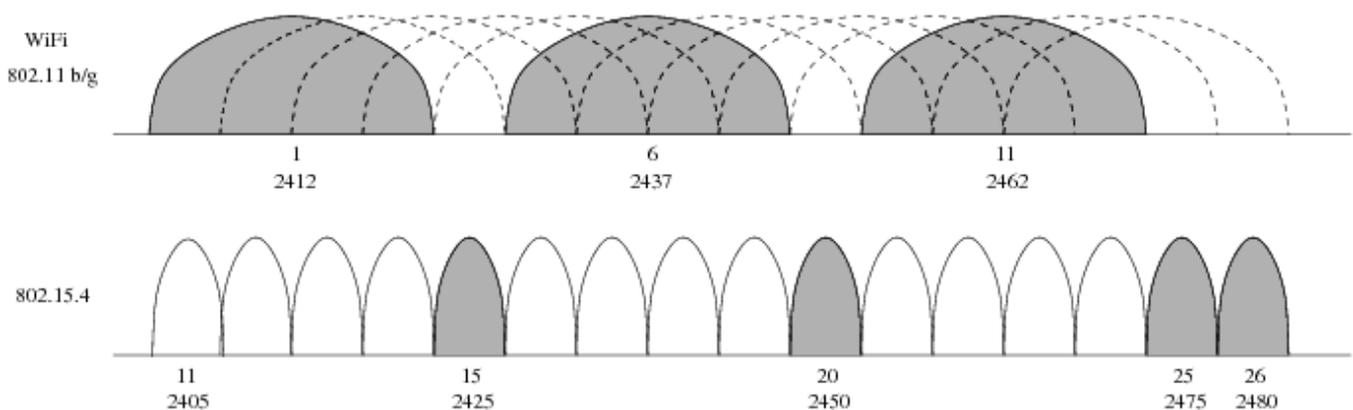
**Figura 8.** Capas definidas por IEEE 802.15.4.

**Fuente:** Edición autor de la obra.

### 2.1.6.1. Capa Física (PHY).

Esta capa define la conexión física del hardware, las bandas de frecuencia en las que funciona y algunas funciones especiales que se describen mas adelante. Respecto a las bandas de frecuencia de trabajo se definen tres bandas, en 868 Mhz, 915 Mhz y 2,4 Ghz. Las dos primeras bandas de trabajo tienen su respectiva regulación en muchos países, pero la banda de 2,4 Ghz esta dentro de las bandas ISM y su uso es libre en el Ecuador, motivo por el cual este documento se centra en el estudio de esta frecuencia por su ausencia de conflictos con el marco regulatorio.

En 2.4 Ghz se definen 16 canales de comunicación, con una separación por canal de 5 Mhz y numerados del 11 al 16, quedando en la frecuencia inicial 2405 Mhz y la final en 2480 Mhz. Pensando en la interoperabilidad de este tipo de redes con las definidas por 802.11 b/g (WIFI) que funcionan en la misma frecuencia, se utiliza una técnica para evitar utilizar frecuencias de canales utilizados por 802.11 b/g como se ve en la figura 9.



**Figura 9.** Interoperabilidad IEEE 802.15.4 con IEEE 802.11b/g.

**Fuente:** Libro Equisbí (Figura 2.2.1).

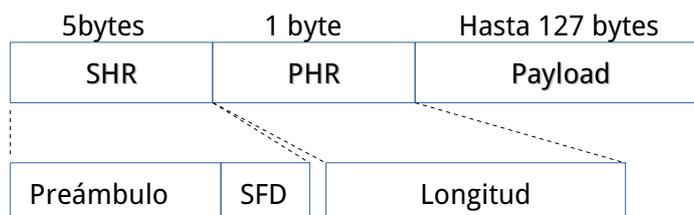
Inicialmente se puede evidenciar en la figura 9 (parte superior) algunos de los canales con sus frecuencias en donde funciona IEEE 802.11b/g y en la parte inferior se observa que los canales 15, 20, 25 y 26 de IEEE 802.15.4 que se encuentran en los espacios de espectro que no se utilizan por WIFI. De manera similar los canales 16, 21 y 22 de IEEE 802.15.4 evitan las frecuencias de IEEE 802.11b/g ubicandose entre los canales 1, 7 y 13. De esta manera se evitan interferencias entre redes en ambientes que tengan un número alto de redes WIFI, eligiendo adecuadamente los canales que no se interfieran.

(Caprile, 2009) indica que, para evitar la elección de canales que posiblemente estén ocupados, el estándar IEEE 802.15.4 establece funciones para evaluar el canal y son:

- *Energy Detection (ED)*.- Busca señales compatibles con el estándar, si se encuentra se verifica si supera un umbral predeterminado y si lo hace el canal esta ocupado.
- *Clear Channel Assessment (CCA)*.- Esta función puede escanear el canal de tres maneras para determinar si hay o no un canal limpio.
  - Nivel de energía suficiente.- El canal se encuentra ocupado si se detectan señales de energía que superen un umbral.
  - Presencia de portadora.- El canal esta ocupado si se detectan señales compatibles con 802.15.4.
  - Presencia de portadora con nivel de energía suficiente.- Es la combinación de los dos anteriores casos, y se considera el canal ocupado solo si cumple las condiciones anteriormente mencionadas.

Por la naturaleza de la comunicación inalámbrica se debe proveer un mecanismo de

sincronización entre dispositivos, por lo que en el encabezado de las tramas se envían los campos que se indican en la figura 10.



**Figura 10.** Sincronización de dispositivos y header PHY.

**Fuente:** Libro Equisbí (Figura 2.2.3).

- SHR<sup>18</sup>.- permite la sincronización de los relojes mediante un preámbulo de 32 ceros lógicos seguidos de la secuencia 10100111(SFD<sup>19</sup>), que indican que el siguiente campo contiene la longitud de la trama.
- PHR<sup>20</sup>.- Especifica la longitud de la trama con el numero de bytes que tiene la carga útil.
- Payload.- Contiene la información proporcionada por la subcapa MAC.

Según el estándar el tamaño máximo que puede tener la trama es de 133 bytes.

18 SHR.- Synchronization Header. Cabecera de sincronización.

19 SFD.- Start of Frame Delimiter. Delimitador de inicio de trama.

20 PHR.- PHY header. Encabezado de capa física.

### 2.1.6.2. Capa Enlace

En el nivel de enlace de datos IEEE 802.15.4 define solo la subcapa MAC(Control de acceso al medio), y se prevee la comunicación con capas superiores por medio de la subcapa LLC(Control de enlace lógico) que es parte de la capa enlace; un orden jerárquico se observa en la figura 7, entre las subcapas LLC y MAC la primera es superior .

LLC se implementa a base de software, es decir no depende del hardware en capas inferiores, mientras que la implementación de la subcapa MAC depende exclusivamente del hardware por el que esta compuesta la capa física. El estándar que define a LLC es IEEE 802.2 y al ser un estándar completo escapa al contenido de este documento; por lo cual se estudia unicamente lo definido por IEEE 802.15.4 con respecto a la capa enlace.

Su función principal es permitir la entrega de la información al dispositivo destino pero esto no incluye el servicio de ruteo, es decir permite el envío y recepción de datos entre dispositivos vecinos. Además permite la detección de errores y retransmisión de tramas.

Básicamente las tramas vistas desde MAC contienen información que corresponde a control, direccionamiento y un footer. El *header* contiene información correspondiente a direccionamiento y control que permite identificar el tipo de trama, en el *footer* se inserta el FCS (Frame Check Sequense) para poder corregir errores.

2 bytes	1 byte	4 a 20 bytes	Variable entre 0, 5, 10, 14 bytes	Variable	2 bytes
Control de trama	Número de secuencia de datos	Información de direcciones	Encabezado auxiliar de seguridad	Payload	FCS

**Figura 11.** Trama vista desde Subcapa MAC.

**Fuente:** <http://standards.ieee.org/getieee802/download/802.15.4-2007.pdf>

La descripción de los campos es la siguiente:

- **Control de trama.-** Tiene algunas funciones como especificar el tipo de dirección utilizada, identificar el tipo de trama y puede habilitar un procesamiento de seguridad.
- **Número de secuencia de datos.-** es una secuencia de números que contiene el identificador de la trama.
- **Información de direcciones.-** contiene información referente al direccionamiento, dirección origen, dirección destino y también identificadores de las redes PAN.
- **Encabezado auxiliar de seguridad.-** En el campo *control de trama* existe un *flag* de seguridad el cual si se activa hará que se genere este campo; si no se activa el *flag* este campo no existe. Cuando se desea saber que llave de seguridad MAC es usada, se emplea este campo.
- **FCS.-** es el campo que se utiliza para la detección de errores mediante una secuencia de redundancia cíclica.

La trama vista desde la subcapa MAC puede llegar a medir hasta 127 bytes, en otras palabras el payload de la figura 10 contiene todos estos campos.

### **2.1.6.3. Subcapa Control de acceso al medio MAC.**

El mecanismo para acceder al medio es CSMA-CA<sup>21</sup> (Acceso múltiple mediante censado de portadora evitando colisiones) ya que en la naturaleza del medio varias estaciones podrían empezar a transmitir generando una colisión, debe haber un orden en la manera en que las

---

21 CSMA-CA.- Carrier Sense Multiple Access with Collision Avoidance

estaciones acceden al medio de transmisión, con el fin de minimizar las colisiones y aprovechar el medio de una mejor manera. El algoritmo CSMA-CA establece los siguientes pasos:

- La estación que desee transmitir datos espera un tiempo aleatorio calculado localmente, luego del cual censa el canal en busca de una portadora.
- Si hay portadora se calcula un nuevo tiempo de espera aleatorio, para volver al paso uno. (Se repite este paso hasta agotar el número de intentos para acceder al canal, si se agotan se informa de un error).
- Si no hay portadora se procede a enviar los datos.

#### ***2.1.6.4. Direccionamiento***

En 802.15.4 elementalmente existen dos tipos de direccionamiento, largo y corto. El direccionamiento largo consta de 64-bits con una secuencia única grabada en el dispositivo al ser fabricado. En el caso de direccionamiento corto el desarrollador puede asignar a cada dispositivo una dirección de 16-bits que no se debe repetir dentro de la PAN<sup>22</sup>. Al tratarse de una red es natural que los dispositivos puedan agruparse en redes mas pequeñas, para esto se utiliza un parámetro configurable denominado PAN ID. Los dispositivos configurados dentro de una misma PAN ID y que estén al alcance pueden intercambiar mensajes entre si pero no podrán hacerlo con dispositivos que estén en otra PAN ID. Como dato importante el estándar no define como se podría llegar a intercambiar trafico entre dispositivos de distintas PAN, siendo necesaria la implementación de protocolos mas complejos u otra solución desarrollada

---

22 PAN.- Network Area Personal.(Red de área personal)

por el diseñador de la red.

En algunas aplicaciones es necesario enviar el mismo mensaje a todos los dispositivos de una PAN, para esto el estándar especifica una dirección de *broadcast* la cual debe ponerse como dirección destino y es *0XFFFF*.

#### **2.1.6.5. Tipos de tramas.**

El objetivo de la transmisión de información mediante tramas no solo es transportar datos de la aplicación en concreto, para llevar acabo ciertas funciones especificas en un dispositivo destino 802.15.4 especifica varios tipos de tramas, el tipo de trama se especifica en un campo de la trama llamado *Frame type*. Existen cuatro tipos de tramas:

- **Datos.-** Son tramas que tienen el objetivo de llevar la carga útil a dispositivo destino, a este tipo de tramas también se las llama *infotramas*.
- **Comandos.-** Son tramas administrativas de la red y son:
  - *Association request.-* Se genera cuando un dispositivo desea asociarse a una red.
  - *Association response.-* Es la respuesta al mensaje anterior, respondiendo si se acepta o no la petición de asociación.
  - *Disassociation notification.-* La trama contiene un mensaje que puede generar el coordinador de la red o un dispositivo indicando que sale de la red.
  - *Data request.-* Se utiliza cuando un dispositivo a estado ausente de la red durante un periodo de tiempo y envía este mensaje al coordinador solicitando que entregue los datos pendientes. Es decir si otro dispositivo envía datos al dispositivo ausente,

estos se almacenan en el coordinador para luego enviárselos cuando regrese.

- *PAN ID conflict notification.*- cuando un dispositivo que pertenece a una PAN-X con su respectivo coordinador, detecta otro coordinador(PAN-Y) con el mismo PAN ID, notifica a su coordinador por medio de este mensaje.
- *Orphan notification.*- este mensaje lo envía algún dispositivo que ha perdido el sincronismo con su coordinador (no se puede comunicar con el), declarandose huérfano.
- *Coordinador realignment.*- es la respuesta al mensaje anterior, si el mensaje del dispositivo que se declaró huérfano es escuchado por su coordinador este le responde con este tipo de mensaje, para restablecer su sincronismo.
- *Beacon request.*- generada un dispositivo que solicita a los coordinadores en su área que envíen una trama beacon para poderlos identificar.
- *GTS request.*- Se emplea en redes beacon cuando un dispositivo pide una ranura de tiempo para su propio uso.

Como se puede evidenciar existen muchas tramas de control que permiten que una red se mantenga operativa y funcional, tratando de evitar posibles conflictos e inconvenientes.

- ***Confirmación (ACK).***- Las tramas de confirmación solo se envían cuando se hace una petición de confirmación en una trama, esto se hace seteando la bandera *ACK request* en el campo *frame control* de la trama que necesita confirmación.
- ***Trama beacon.***- Esta trama la utiliza en coordinador de una red con el objetivo de darse a conocer a los dispositivos que están a su alcance y escuchando. Así los

dispositivos podrán asociarse o no a el.

Control de trama	Número de secuencia de datos				FCS	ACK
Control de trama	Número de secuencia de datos	Información de direcciones	Encabezado auxiliar de seguridad	Carga útil del comando	FCS	COMANDOS
Control de trama	Número de secuencia de datos	Información de direcciones	Encabezado auxiliar de seguridad	Carga útil de datos	FCS	DATOS
Control de trama	Número de secuencia de datos	Información de direcciones	Encabezado auxiliar de seguridad	Carga útil del Beacon	FCS	BEACON

**Figura 12.** Tipos de tramas vistas desde Subcapa MAC.

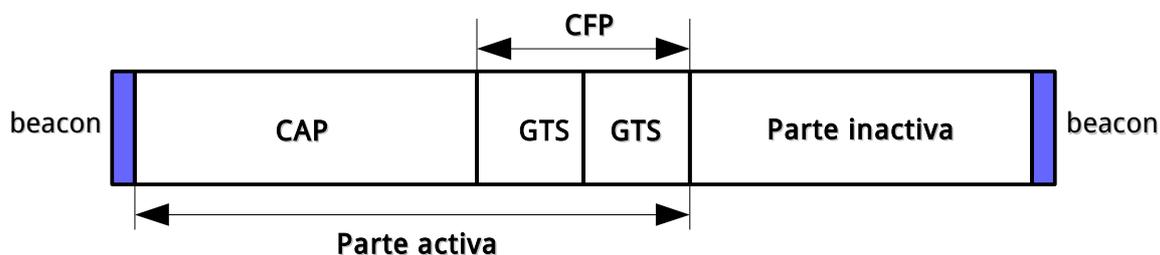
**Fuente:** <http://standards.ieee.org/getieee802/download/802.15.4-2007.pdf>

Debido a que las redes basadas en IEEE 802.15.4 están pensadas para soportar una cantidad de tráfico reducido y sus dispositivos no deben ser muy complejos, o de lo contrario consumiría mucha energía, el estándar contempla un tiempo de espera entre tramas para que el receptor pueda procesar la información de la última trama recibida antes de recibir otra. Este tiempo se calcula dependiendo del tamaño de la trama enviada y un umbral predefinido, sus tiempos de espera pueden ser cortos (192  $\mu$ s) o largos (640  $\mu$ s).

#### **2.1.6.6. Supertramas**

Existe otro método por el cual el coordinador de una red puede comunicarse con los dispositivos asociados, básicamente para el IEEE 802.15.4 existen dos tipos de redes sin beacons y con beacons. Las redes sin beacons utilizan CSMA-CA para poder acceder al canal de comunicaciones y las redes con beacons introducen una estructura llamada supertramas

que permite la transmisión de datos en ranuras de tiempo.



**Figura 13.** Tipos de tramas vistas desde Subcapa MAC.

**Fuente:** Libro Equisbí (Figura 2.4.5).

En la figura 13 se puede observar la estructura de una supertrama, el tiempo de duración de la supertrama es especificado por el coordinador y se envía mediante los beacons, las supertramas se encuentran confinadas entre los beacons. La estructura básica de la supertrama se divide en dos ranuras de tiempos, una parte activa donde existe comunicación y otra parte inactiva donde todos los dispositivos pueden entrar en estado de bajo consumo.

En la parte activa encontramos la subdivisión de CAP<sup>23</sup>(Periodo de Acceso por Contención) y CFP<sup>24</sup>(Periodo Libre de Contención). En el periodo CAP los dispositivos que deseen transmitir lo hacen utilizando CSMA-CA y en el periodo CFP determinados dispositivos asignados pueden transmitir sin utilizar CSMA-CA. Las siglas GTS<sup>25</sup> significan tiempos de ranura garantizados.

23 CAP.- Contention Access Period.

24 CFP.- Contention Free Period.

25 GTS.- Guaranteed Time Slots.

### ***2.1.6.7. Tipos de dispositivos***

Finalmente en el estudio del estándar IEEE 802.15.4 solo se definen dos tipos de dispositivos los FFD(Full Function Device) y los RFD(Reduce Function Device).

Los FFD son dispositivos que tienen todas las funciones que especifica el estándar y solo puede existir uno por PAN, su comunicación se da con los RFD.

Los RFD son dispositivos que por lo general están implementados en hardware de tamaño reducido con el objetivo de ser dispositivos finales, estos se ubican en los lugares donde se desean recolectar los datos. Por esta razón no suelen implementar todas las características del estándar y por lo general su hardware no es muy potente, su comunicación solo se puede dar con un FFD.

La consecuencia inmediata de esto es que solo se puede tener dos tipos de topologías, *punto a punto* y *estrella*, ya que el estándar no da servicio de ruteo; en todo caso si se desea esta característica y otras más el desarrollador debe recurrir a una solución propia ó a un protocolo de alto nivel como ZigBee<sup>26</sup>, el cual esta basado en IEEE 802.15.4 e implementa muchas mas funciones que las soportadas por el estándar.

De acuerdo ha este estándar existen en el mercado diferentes módulos que se basan en el estándar explicado, a continuación una comparativa en la tabla 3 de los que se han considerado posibles candidatos para aplicar en el proyecto.

De izquierda a derecha a derecha primero se tiene el módulo 579-MRF24J40MA-I/RM fabricado por la empresa Microchip, tiene un alcance en interiores de 20m los cuales se

---

26 ZigBee.- Es un protocolo de comunicación de alto nivel basado en IEEE 802.15.4, soporta ruteo y redes en malla.

consideran apropiados para el sistema, su velocidad de 250Kbps esta en lo esperado para soportar el bajo trafico de la red de dispositivos, los inconvenientes con este módulo empiezan con la conexión física hacia arduino ya que no cuenta con un shield y a pesar de comunicarse con el bus SPI del microcontrolador se tiene que adecuar la librería de arduino para efectuar la comunicación. Esto sumado ha que la plataforma arduino no soporta nativamente este hardware se tendrían que estudiar los registros de configuración del módulo.

Continuando se tienen los módulos xbee de la empresa digi, los cuales tiene un alcance interior de 40m con velocidades de 256Kbps, este módulo se comunica mediante comunicación serial y no necesita de configuración previa ya que tiene un firmware pre-configurado de fábrica, la comunicación con el microcontrolador es UART. Además, existen shields para conexión directa con placas arduino.

Tabla 3 .- Módulos basados en IEEE802.15.4

<b>Características</b>	<b>579-MRF24J40MA- I/RM</b>	<b>XBEE s1</b>	<b>579-MRF24J40MC- I/RM</b>
Distancias	20m interiores 50m exteriores	40m interior 100m exterior	30m interior 70m exteriores
Velocidad	250kbps	256kbps	128Kbps
Consumo	23 mA	50mA	120mA
Comunicación con microcontrolador	SPI	UART	SPI
Shield Arduino	no	si	no
Alimentación	3,6V	3,3	3,6

En la parte derecha se tiene a el módulo 579-MRF24J40MC-I/RM muy parecido al primer módulo expuesto con un alcance interior de 30m, la velocidad de este módulo se reduce hasta los 128Kbps y su consumo sube hasta los 120mA. En la parte de comunicación se puede ver que utiliza el bus SPI.

De acuerdo con las necesidades del proyecto, por facilidad de uso, y por menor complicación de conexión y configuración por emdío de arduino se eligen los módulos XBEE.

### **2.1.7. Módulos de radiofrecuencia XBEE.**

Los módulos XBEE son dispositivos desarrollados por la empresa Digi<sup>27</sup> y están basados en el estándar IEEE 802.15.4 y funcionan en 2,4GHz. En su fabricación se graba el conjunto de funciones descritas por el estándar además de una dirección única de 64 bits. Estos módulos fueron diseñados principalmente para el envío y recepción de datos en redes de bajo trafico, específicamente para redes de sensores. La utilización de estos módulos presentan varias ventajas en el desarrollo de proyectos que involucren una red de sensores con bajo tráfico, la amplia experiencia de la empresa que esta detrás de estos módulos permite tener la garantía de un producto maduro, robusto y estable.

La comunicación con dispositivos micro-controlados la hace mediante una simple comunicación serial, esto permite que cualquier dispositivo que posea un puerto serial pueda inmediatamente conectarse a una red IEEE 802.15.4 por medio de estos módulos.

Su hardware esta diseñado para alcanzar grandes distancias de cobertura por medio de

---

27 Digi.- Es una empresa de comunicaciones especializada en sensores.

receptores mas sensibles, como consecuencia se tiene que el transmisor no necesita mayor potencia para enviar datos a mas distancia, si no que el receptor podrá adquirir los datos gracias a su aguda sensibilidad, motivo por el cual solo se alimentan con 3.3V y su consumo de corriente ronda los 50mA. La velocidad máxima de estos módulos es de 250Kbps. Existen variedad de módulos XBEE pero el elegido para el desarrollo del sistema domótico se puede apreciar en la figura 14.



**Figura 14.** Xbee S1.

**Fuente:** [www.digi.com](http://www.digi.com)

Como se observa el módulo no solo esta enfocado a la transmisión y recepción de datos, si no que también posee un gran cantidad de pines configurables como entradas o salidas digitales, entradas analógicas, ADC y PWM. Su potencia de salida es 1mW(0dBm) con lo cual se pueden cubrir distancias de 100m en espacios abiertos y 30m en espacios con infraestructuras.

El modo de configuración se lo hace por medio de comandos AT que son enviados a través de su puerto serie con cualquier software que maneje el puerto serial o algún sistema

microcontrolado que tenga previamente grabados los comandos AT. Para entrar en el modo de configuración se deben enviar por su puerto serie los caracteres +++ y a continuación el comando responderá con un *OK* el cual indica que está listo para recibir el comando AT, luego de haber configurado los cambios se escapa del modo de configuración con el comando ATCN y el módulo vuelve a su estado normal de trabajo. El set de instrucciones está completo en el Anexo 1 de este documento.

Ya que XBEE funciona con 3,3V, diferentes empresas de hardware han diseñado adaptadores para acoplar los voltajes TTL del puerto serie a los 3,3V permitidos por XBEE.

La empresa Digi distribuye de forma gratuita XCURT, que es un programa condensado de funciones referentes a los módulos, permitiendo configurarlos en muy pocos pasos y hacer una gran variedad de pruebas. Este software facilita la configuración de los módulos pero no es indispensable.

La serie 1 de XBEE soporta todas las funciones descritas por el estándar IEEE 802.15.4 por consecuencia se los puede conectar de la siguiente manera:

- *Punto a Punto.*- Es la forma más básica de conexión entre módulos y consiste en poner las direcciones de destino en cada módulo correspondiente al módulo al que se desea transmitir, en otras palabras y como ejemplo si se tiene un XBEE(A) con una dirección(1111) y un módulo XBEE(B) con una dirección (2222), para que haya una comunicación punto a punto en el XBEE(A) la dirección destino debe ser (2222) y en el XBEE(B) (1111), de esta manera se comunicarán solo entre los dos módulos.
- *Punto a Multipunto.*- Es la consecuencia inmediata de aplicar una topología *punto a punto* entre varios XBEE y todos con la misma dirección destino, es decir se tiene un

módulo ubicado centralmente y todos los dispositivos alrededor tendrán como dirección de destino el módulo central. Cada vez que el nodo central quiera comunicarse con un vecino, el central deberá cambiar su dirección destino mediante comandos AT apuntando al vecino deseado, como el vecino tiene ya previamente configurado su destino hacia el nodo central, pueden intercambiar datos sin mayor dificultad. Este proceso conlleva a que los vecinos no pueden iniciar la comunicación ya que no saben si el central les va a responder por que su dirección destino puede ser cualquiera en cualquier instante, dando como resultado que el nodo central siempre inicie la conversación.

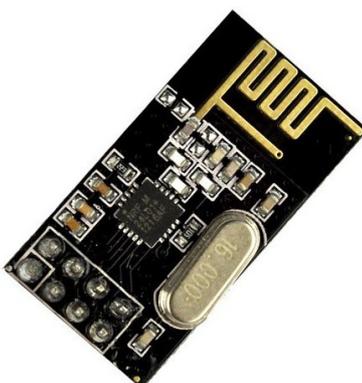
- *Estrella.*- En la topología estrella se tiene un módulo en una ubicación central el cual se comunica con los módulos que se encuentren a su alrededor. Para esto existen dos maneras de conexión y el diseñador de la aplicación debe saber cual elegir. La primera opción de este tipo es la detallada en el anterior inciso, su ventaja es que es muy fácil de configurar y sus desventajas son, que al acceder cada vez que se quiere cambiar el destino al modo de configuración, se pierde un pequeño tiempo que no todas las aplicaciones finales se pueden permitir, otra desventaja es que la comunicación se debe hacer una a una y no todas a la vez, no pueden todos los vecinos enviar información al nodo central al mismo tiempo. La segunda opción para llegar a tener una topología estrella es que se configure un modo de comunicación llamado API, en este modo si se pueden tener comunicaciones simultáneas hacia el nodo central pero también depende de la aplicación, ya que el nodo central no podrá soportar la comunicación de varias decenas de módulos a la vez por su limitación en su buffer. En la configuración API el arquitecto de la red debe definir un formato de

trama para poder identificar a los orígenes de los datos y diferenciar a los vecinos, esto conlleva mas complejidad y tiempo en el diseño, pero la gran ventaja es que si se dimensiona correctamente se obtienen comunicaciones con el mínimo retardo.

He ahí la importancia de saber elegir correctamente que tipo de topología elegir para cada aplicación.

### 2.1.8. Módulos de radio frecuencia NRF24L01.

Como se ha podido evidenciar los módulos XBEE son una solución para la implementación de redes de sensores con bajo tráfico, con el desarrollo de la IEEE y la empresa Digi se pueden obtener productos de gran calidad a disposición. El punto débil de estos dispositivos, es tal vez su elevado costo y la consecuencia que esto tiene en redes con varios dispositivos, se tiene una solución excelente pero a la vez muy costosa; además hay que tomar en cuenta que no todas la aplicaciones se pueden permitir esto.



**Figura 15.** Módulo NRF24L01.

**Fuente:** <http://hallard.me/nrf24l01-real-life-range-test/>

En la figura 15 se puede observar un modulo fabricado por la empresa NORDIC<sup>28</sup>, exactamente es el módulo de comunicación inalámbrico NRF24L01, la especialidad de la empresa NORDIC son los chips de comunicaciones de bajo consumo energético, según su pagina web oficial diseñados para permanecer largos periodos de tiempo funcionando conectados a una batería.

Este módulo de radio frecuencia cuesta el 6.6% de lo que cuesta un módulo XBEE serie 1 y posee características similares, aunque no llegan a igualar a las de XBEE ya que no esta basado en el IEEE 802.15.4. A pesar de esto este *transceiver* es capaz de transmitir información en la banda de 2,4GHz con alcances de hasta 100m en lugares despejados y 20m en lugares con infraestructura, algo muy parecido al los módulos XBEE.

Su velocidad de transmisión es configurable por software y puede ser 250kbps, 1Mbps ó 2Mbps con un consumo de 11,3mA a 13,5mA, esto depende de la ganancia que también se puede configurar ente 0, -6, -12, -18dBm.

Usan una modulación GFSK<sup>29</sup> y su canal de frecuencia puede ser elegido por el usuario entre 126 canales.

Sus pines de entrada son tolerantes a 5V(TTL), esto quiere decir que aunque el chip necesita una alimentación de voltajes entre 1,9 a 3,6 voltios para funcionar, el chip posee un regulador en sus entradas que le permiten comunicarse directamente con dispositivos que tengan tecnología TTL.

Para la implementación de estos módulos en una aplicación hace falta un microcontrolador

---

28 NORDIC.- Es una empresa de semiconductores, especializada en construcción de chips de bajo consumo.

29 GFSK.- Gaussian Frequency Shift Keying.

que posea una interfaz periférica en serie o también conocida como SPI. El chip NRF24L01 se configura mediante registros de configuración los cuales son accesibles mediante la interfaz periférica serial. Según la hoja de especificaciones proporcionada por su fabricante

“...el protocolo de banda base embebido se basa en la comunicación de paquetes y soporta varios modos de operación manual y un protocolo avanzado autónomo, FIFO's internos garantizan un flujo de datos sin problemas entre la parte de radio y el microcontrolador...”  
(NORDIC 2014)

Estos módulos poseen una interfaz común para la transmisión y recepción de datos que esta integrada en el PCB. Los dispositivos poseen características similares al estándar estudiado anteriormente y posee un protocolo embebido que por medio de su hardware se encarga del manejo de tramas en la capa enlace llamado *Enhanced ShokBurt* que se encarga entre otras funciones de:

- Ensamble automático de paquetes(preámbulo, dirección y CRC).
- Detección y validación de automática de paquete recibido.
- Longitud de paquete dinámica entre 1 a 32 bytes
- ACK automático.
- Retransmisión automática.

Como se observa estos dispositivos han sido diseñados para funcionar conjuntamente con un microcontrolador, una de sus ventajas principales es su protocolo embebido que se encarga de las tramas enviadas y recibidas estableciendo tamaños predefinidos, el diseñador puede

dedicarse concretamente en la aplicación final, aunque la comunicación SPI es un obstáculo superable le desarrollador puede utilizar librerías creadas previamente y utilizarlas en sus proyectos en vez de desarrollar todo desde cero.

La mezcla de elementos y recursos para la utilización de estos módulos son:

- Arduino.- se encarga de la parte microcontrolada.
- NRF24L01.- para la parte de comunicación inalámbrica.
- Librería RadioHead.- que es una librería que soporta varios módulos inalámbricos con características similares a los NRF24L01 y permite la comunicación entre varios módulos de este tipo de una manera ordenada.

Al final, se han incluido estos módulos como un complemento a los módulos XBEE utilizados en el sistema con el principal objetivo de reducir costos sin perder la calidad del producto final.

### **2.1.9. Estándar IEEE 802.11b/g.**

El estudio de este estándar es importante por la utilización de una red inalámbrica basada en IEEE 802.11, que permite la comunicación entre host's en la red de la residencia donde se instaló el sistema domótico. El uso de la banda ISM 2,4GHz utilizada para la red de sensores y la red de datos entre host's, hace que se considere la posibilidad de interferencia entre las dos redes; pero gracias a las consideraciones en el estándar IEEE 802.15.4 esto se puede evitar de una manera sencilla utilizando canales que no se interfieran, además las características de modulación en IEEE 802.11 que se ven mas adelante, permiten que los

datos se puedan transmitir de manera que no sufran alteraciones.

El estándar 802.11.x esta orientado a las redes inalámbricas y presenta muchas subdivisiones con diferentes características, este documento se centra en el estudio de las características básicas que comparten las divisiones que se desprenden del estándar y al final se enfoca en las dos divisiones utilizadas, IEEE 802.11b y IEEE802.11g (Gidoni, Eran, & Emanuel 2014).

IEEE 802.11.x define la capa física, en la capa enlace existe una subdivisión en dos subcapas MAC(Control de acceso al medio) que la define el estándar y LLC(Control de enlace lógico) definido por IEEE 802.2 que permite la comunicación con capas superiores.

### **2.1.9.1. Capa Física**

Es la encargada de la modulación y codificación de los datos, para la utilización de las frecuencias utiliza una técnica llamada espectro ensanchado, la cual consiste en utilizar un ancho de banda mas amplio que el mínimo necesario para la transmisión de los datos, el beneficio que esto aporta es que mediante esta técnica la red de datos puede coexistir con otras redes en las misma frecuencia y los datos no sufrirán alteraciones por interferencias. A continuación las dos técnicas utilizadas.

- FHSS<sup>30</sup>.- También conocida como espectro ensanchado por salto de frecuencia, esta técnica consiste en la división del ancho de banda total (2,400GHz a 2,4835GHz) en 79 canales, cada uno con un ancho de banda de 1MHz, la

---

30 FHSS.- Frecuency Hoping Spread Spectrum.

transmisión de los datos se hace a través de diferentes frecuencias, iniciando en una frecuencia y manteniéndose en ella por un tiempo menor a 400ms, para luego saltar a otra frecuencia y continuar con la transmisión de datos; esto se repetirá hasta terminar la transmisión de los datos. Para la sincronización el transmisor envía señales al receptor con información de la secuencia de saltos a seguir y sus tiempos. En su práctica esta técnica no es muy utilizada pero aporta una gran inmunidad al ruido y seguridad.

- DSSS<sup>31</sup>.- Se la conoce como espectro ensanchado por secuencia directa, consiste en la generación de un patrón redundante de bits que reemplazarán a cada bit de datos, el estándar recomienda una secuencia de 11 bits por cada bit de datos. En la práctica, si el número de bits de la secuencia es mayor entonces, mayor será la inmunidad al ruido pero utilizará más ancho de banda y se reducirá la velocidad de transmisión. Antes de la transmisión el transmisor envía al receptor la información referente a la secuencia utilizada y así este podrá obtener los datos originales. Esta técnica utiliza un ancho de banda mucho mayor al necesario para transmitir los datos directamente, pero aporta seguridad en la transmisión e inmunidad al ruido.

Estas técnicas sacrifican el ancho de banda y la velocidad por la inmunidad al ruido en el medio, ya que este puede estar también ocupado por otras tecnologías.

### **2.1.9.2. Capa enlace.**

En la capa de enlace de datos el estándar define la subcapa de control de acceso al medio

---

31 DSSS.- Direct Sequence Spread Spectrum.

MAC, para lo cual utiliza CSMA/CA. Las estaciones que deseen transmitir deberán escuchar el canal de comunicación antes de poder transmitir, si el canal esta libre la estación puede enviar los mensajes inmediatamente, si por lo contrario el canal se encuentra ocupado debe escuchar el canal hasta que la transmisión finalice, luego tiene que esperar un tiempo aleatorio antes de volver a censar el canal.

Para una transmisión de información segura se requieren acuses de recibo (ACK) los cuales son enviados desde el receptor hacia le transmisor confirmando la llegada de los datos de una manera correcta, los paquetes identificados como ACK's tienen prioridad frente a cualquier otro tipo de tráfico.

El estándar permite dos tipos de funcionamiento para la conformación de redes y son:

- ***Redes Ad-hoc.***

En el modo Ad-hoc los terminales pueden asociarse entre sí, sin la necesidad de un nodo central que coordine las comunicaciones, este modo de funcionamiento permite gracias a algoritmos de enrutamiento, que los paquetes sean enviados a través de los terminales que conforman la red, permitiendo tener comunicación entre cualquier equipo que conforme la red.

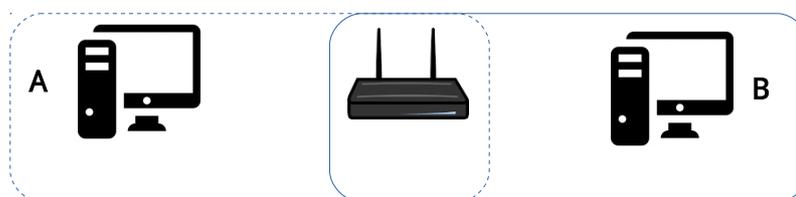
- ***Redes de infraestructura.***

En este modo se requiere de nodos centrales llamados Acces Points (AP) que coordinan las comunicaciones, las estaciones tienen que asociarse a un dispositivo central por el cual

circularán todos los paquetes.

En ambos casos se manejan parámetros de seguridad que generalmente se basan en una secuencia de caracteres(contraseña) que deben tener las estaciones que deseen asociarse a la red.

En la figura 16 se observa una topología multipunto, los host A y B están conectados de manera inalámbrica a un AP, el cual puede ver a las dos estaciones; pero las estaciones no pueden verse entre sí. A esta situación se la conoce como el *nodo escondido*.



**Figura 16.** Diagrama nodo escondido.

**Fuente:** Edición autor de la obra.

Por la naturaleza del medio las estaciones deben comunicarse de manera ordenada con el AP, a pesar de utilizar CSMA/CA para acceder al canal, este método no cubre el echo de que un nodo este alejado de los demás y por ende no pueda escuchar la portadora que generan los demás nodos al transmitir datos. Si este nodo escondido desea transmitir escuchará el canal para saber si hay portadora o no, al no poder detectar la portadora de los otros nodos, empieza a transmitir su información y causaría interferencia con la transmisión actual. Para solucionar eso el estándar define un segundo método a parte de CSMA/CA llamado, VCS<sup>32</sup> ó censado de portadora virtual, el cual permite a las estaciones reservar un espacio de tiempo de

<sup>32</sup> VCS.- Virtual Carrier Sense.

comunicación con el AP con tramas RTS<sup>33</sup>/CTS<sup>34</sup>. Refiriéndose a la figura 16 la estación A envía un RTS al AP, el paquete RTS contiene un campo que indica cuanto tiempo desea reservar el canal, este paquete no es escuchado por la estación B. El AP responde con un paquete CTS que contiene el tiempo asignado a la reserva del canal, esta vez la estación B si escucha la información y de esta manera sabe que no podrá transmitir por el tiempo indicado en el CTS, de esta manera se supera este problema.

### 2.1.9.3. Tipos de tramas

Existen diferentes tipos de tramas y se las puede clasificar de acuerdo a sus funciones:

- **Tramas de conexión.**- Tienen la función de mantener la comunicación entre los elementos de la red, dentro de este tipo de tramas se tienen:
  - *Trama de asociación.*- contienen una petición que se envía al coordinador de la red (AP), la trama contiene información sobre su velocidad y el SSID<sup>35</sup> al que desea asociarse. Si el AP decide aceptar la solicitud este reserva un espacio de memoria para el intercambio de paquetes y le asigna un número de asociación.
  - *Trama de respuesta de asociación.*- es la respuesta del AP ante una petición de asociación, contiene información relacionada con la velocidad de transmisión soportada y al número de asociación(Association ID).
  - *Trama de reasociación.*- esta trama es utilizada cuando un nodo se mueve dentro

---

33 RTS.- Request To Send.

34 CTS.- Clear to send.

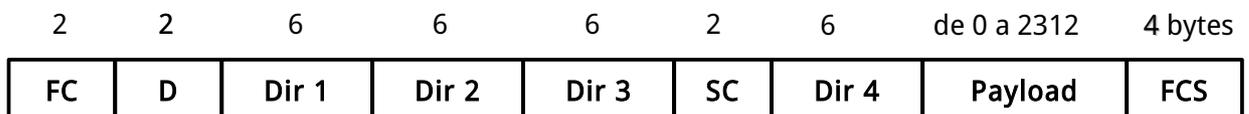
35 SSID.- Service Set Identifie

de una red y encuentra otro AP con mayor intensidad de señal, generalmente se trata de dos AP's que están trabajando juntos en una misma red. El nodo envía esta trama y los AP's comparten la información referente al nodo, luego si hay datos pendientes por entregarse al nodo se envían y finalmente se envía un paquete que contiene el nuevo número de asociación y las velocidades soportadas por el nuevo AP.

- *Trama de desasociación.*- Esta trama es utilizada por algún nodo que desee salir de la red y permite al AP liberar la memoria reservada, así como su número de asociación.
- *Trama de autenticación.*- Para la inclusión de un nuevo nodo a la red, este debe identificarse sin importar si la red tiene o no un sistema de seguridad. Si la red está abierta(sin seguridad), el nodo envía una trama de autenticación y el AP responderá con una respuesta afirmativa o negativa. Si por el contrario la red maneja algún tipo de seguridad, el nodo debe enviar los credenciales respectivos para que pueda ser aceptado en la red. El AP compara los credenciales y si son correctos, envía una trama aceptando la unión del nuevo nodo.
- *Trama de desautenticación.*- Es enviada para terminar una asociación entre nodos.
- *Trama beacon.*- Es un tipo de trama que el coordinador de la red está enviando periódicamente con el objetivo de informar de su presencia a los demás elementos de la red, también se utiliza para informar a los nodos sobre el tipo de seguridad manejada y las velocidades aceptadas, los nodos por su parte siempre están escaneando los canales de comunicación en busca de nuevos SSID(incluidos en

los beacons de otras redes).

- *Tramas de puebas.*- Se utilizan cuando un nodo desea saber que otras estaciones están al alcance, los nodos que reciben esta trama responde con un paquete que contiene información referente a la capacidad y velocidad soportada.
- **Tramas de control.**- Son tramas que permiten el intercambio de paquetes en la red.
  - *Tramas ACK.*- Contiene la confirmación de la llegada de una trama a su destino.
  - *Tramas RTS y CTS.*- Se utilizan para evitar problemas con los nodos ocultos que pueden existir en una red.
- **Tramas de datos.**- este tipo de tramas son las que transportan la carga útil de la aplicación final.



**Figura 17.** Trama IEEE 802.11.x

**Fuente:** Libro Transmisión de datos y redes de comunicaciones. Autor: Behrouz A. Forouzan (Mc Graw Hill 4ª Edición).

En la figura 17 se observa la estructura de la trama IEEE 802.11.x (Forouzan, 2007) a continuación se explican sus campos.

- FC.- se conoce como campo de control de trama consta de 2bytes, identifica el tipo de trama y contienen sub campos que indican, la versión del protocolo, tipo de dirección, tipo de trama, administración de seguridad y energía.
- D.- indica la duración o el tiempo de espera que las demás estaciones deben esperar

para censar el canal.

- Dir 1.- es la primera dirección contiene la dirección MAC<sup>36</sup> equipo receptor.
- Dir 2.- es la segunda dirección contiene la dirección MAC del equipo transmisor.
- Dir 3.- es la tercera dirección que contiene en ocasiones la dirección MAC de la puerta de enlace asociada al AP.
- SC.- conocido como control de secuencia, contiene dos subcampos número de secuencia de cada trama y número de fragmentos de cada trama que ha sido fragmentada.
- Dir 4.- Esta dirección es solo utilizada en el modo de funcionamiento Ad-hoc.
- Payload.- contiene la carga útil de la aplicación final.
- FCS.- contiene una secuencia de verificación de la trama, por medio de la cual se hace un control de errores a nivel de capa 2.

Hasta aquí se han estudiado todas las características referentes al las redes que norma el estándar IEEE 802.11, ahora se enumeran las las características de las subdivisiones b y g de este mismo estándar.

- IEEE 802.11b.- Utiliza una como técnica de modulación DSSS, trabaja en la frecuencia de 2,4GHz con una división de 11 canales de comunicación y se pueden alcanzar velocidades de hasta 11Mbps. Es altamente comercial y tiene un bajo costo, además de presentar un buen alcance (50m para interiores y 100m para

---

<sup>36</sup> Dirección MAC.- se refiere a la dirección única del dispositivo asignada es su fabricación, no es igual a control de acceso al medio(MAC).

exteriores) y permite ofrecer calidad de servicio.

- IEEE 802.11g.- Se emplea FHSS y DSSS para modular la señal, trabaja en 2,4GHz y se pueden alcanzar velocidades de hasta 54Mbps. También es una tecnología muy comercial y permite la interoperabilidad con IEEE 802.11b, el inconveniente de esta característica es que si hay la presencia de estos nodos, las velocidades de transferencia se reducen, aún así es una característica muy utilizada.

### **3. CAPÍTULO III**

#### **INTRODUCCIÓN**

En el presente capítulo se hace un recuento de la situación de la infraestructura física y de servicios del área de vivienda en donde se implementará el proyecto, para luego detallar las variables y parámetros de diseño a contemplar para el desarrollo del mismo, tomando en cuenta: planos, infraestructura, diagramas eléctrico, de servicios y sobre todo detalles de las zonas de recorrido permanente de la persona beneficiaria del sistema planteado.

#### **3.1. SITUACIÓN ACTUAL Y PARÁMETROS DEL DISEÑO.**

Es necesario establecer un panorama general referente a la situación de la persona beneficiada con el sistema a través de un estudio, para determinar cual es la mejor forma de ayudar a que pueda tener un mejor desenvolvimiento y por ende una mejor calidad de vida.

El principal objetivo del sistema es facilitar la interacción con las cosas comunes dentro del hogar, sin llegar a tener un sistema intrusivo; es decir los dispositivos a controlar no deben alterar mayormente su funcionamiento o causar incomodidades para otras personas.

De esta manera se llega a determinar que la mejor opción es enfocarse en las áreas más utilizadas por la persona con paraplejia, e instalar dispositivos que puedan funcionar conjuntamente con los controles y mecanismos anteriormente instalados. Es decir en lo posible el sistema tiene un grado de transparencia para los demás miembros del hogar.

### **3.1.1. Características y situación de la persona con paraplejia.**

El Sr. Jaime Cadena tiene 46 años de edad y sufre de paraplejia por un accidente a los 19 años, motivo por el cual no tiene sensibilidad ni movimiento en sus extremidades inferiores. Su hermano el Sr. Luis Cadena quien tiene una familia propia, comparte su vivienda con su hermano y de esta manera él y su familia le ayudan en lo que pueden, permitiendo que pueda llevar una vida cómoda, en lo posible.

La vivienda construida con hormigón y acero cuenta con los servicios básicos, agua, energía eléctrica, teléfono e Internet. Las instalaciones de la vivienda constan de sala, cuartos, comedor, cocina, garaje, dos baños normales y uno adecuado especialmente para satisfacer la necesidades de la persona con paraplejia; además se cuenta con accesos y rampas para que la movilidad no sea un problema.

La opinión verbal del Sr. Jaime Cadena ha sido que se siente cómodo en el que considera su hogar y que las dificultades o necesidades que se le presentan las trata de solucionar él mismo, cuando sus limitaciones físicas no lo permiten acude a algún miembro de la familia.

El Sr. Jaime Cadena es una persona joven que tiene una profesión y un trabajo estable, su deseo de superación ha hecho que pueda lograr gran independencia y su autoestima es muy sana y contagiante.

Por último cabe destacar que la mayoría del tiempo pasa fuera de la casa por su horario de trabajo, pero los fines de semana pasa con su familia.

### 3.1.2. Características de la vivienda y áreas frecuentes.

La vivienda esta compuesta por hormigón armado con una estructura de acero, el área total del inmueble es de 260m<sup>2</sup> los cuales se dividen en dos locales comerciales, un garaje, un patio pequeño, sala, cocina, tres baños y tres dormitorios. El plano con detalles se puede observar en el Anexo 2.

La vivienda es muy amplia, posee espacios diversos de uso común por todos los miembros de la familia; pero el sistema se centra en los espacios mas utilizados por la persona a quien se dedica el sistema. Así se llega a determinar que las áreas mas frecuentemente utilizadas por la persona con paraplejia, son las mostradas en el plano del Anexo 3, y son estas áreas a donde se enfocan los esfuerzos para eliminar las posibles incomodidades y dificultades que se le pudieran presentar.

A continuación se describen brevemente cada una de ellas:

- **Garaje y patio.-** Son dos áreas de acceso, si se desea entrar o salir de la vivienda siempre se debe pasar por estas áreas, el garaje es utilizado por dos carros, uno perteneciente a el Sr. Luis Cadena y el otro de el Sr. Jaime Cadena, se tiene instalado un portero eléctrico para el control del portón y por tratarse de un área de paso temporal, es decir no es un área de estancia se tiene instalados sensores de movimiento que controlan las luces. En el patio se tiene el acceso a la vivienda en si, por medio de una amplia puerta de madera, las luces de igual forma que el garaje cuenta con sensores de movimiento.
- **Sala.-** La sala es un área de 48m<sup>2</sup> (4m de ancho por 11,7m de largo), cuenta con tres luminarias, dos ventanas con sus respectivas cortinas y una puerta. Realmente esta

área esta compartida entre sala y comedor; es un área común y cuenta con los respectivos muebles.

- **Cocina.-** Esta es una área abierta hacia la sala y hacia el dormitorio número dos, no posee puertas, cuenta con una luminaria y una ventana con su respectiva cortina; el área es obviamente transitada por todos los miembros de la familia.
- **Dormitorio número dos.-** Es el dormitorio de la persona con paraplejia, no cuenta con puertas para su acceso, ni ventanas y posee dos luminarias; esta área esta conectada con la cocina a través de una rampa y con el baño número dos a través de una puerta, este baño esta especialmente diseñado para satisfacer las necesidades de esta persona.
- **Baño número dos.-** Es un baño que se diseño tomando en cuenta las necesidades de la persona con paraplejia, tiene su respectiva puerta, una luminaria y una pequeña ventana que se conecta con el patio.

Estas áreas son las tomadas en cuenta para el estudio, diseño e instalación del sistema.

### **3.1.3. Parámetros del diseño.**

Las limitaciones y parámetros del diseño están sujetos principalmente a dos factores, como en cualquier proyecto el aspecto económico no debe ser exagerado y se debe justificar mediante la usabilidad cada dispositivo. En otras palabras cada dispositivo diseñado he implementado debe aportar al sistema sin ser demasiado costoso. De esta manera se consideran el control y la inclusión de lámparas, cortinas, un sistema térmico para la cama del beneficiado, ventanas y puertas.

A continuación se detallan las funciones y limitaciones de cada control para cada dispositivo.

- ***lámpara.***- Se considera un punto importante el control de lámparas ya que aporta de manera sustancial al proyecto, permitiendo la inclusión de controles estilo *on/off* en la interfaz del sistema que trabajan conjuntamente con los controles tradicionales ya instalados. Se permite el control de las luminarias tanto desde la interfaz como desde los interruptores tradicionales, los cambios se ven reflejados en la interfaz y pueden ser consultados en cualquier momento. El control de este dispositivo esta basado en una solución combinada de software y hardware electrónico sin partes mecánicas.
- ***Cortinas.***- Las cortinas permiten el control de luminosidad en las habitaciones en horas del día, el nivel de esta debe ser ajustado por el usuario a través de la interfaz del sistema ó manualmente. Evidentemente esta solución requiere de software, hardware y una parte mecánica que permita el control de la cortina manualmente. Los cambios de estado de la cortina se pueden observar en la interfaz mientras estos estén siendo controlados desde la interfaz y no manualmente. Si hay un cambio manual la interfaz no reflejará esto, pero si el usuario manipula la interfaz la cortina regresará a un estado inicial desde el que podrá mover la cortina por medio de la interfaz.
- ***Sistema térmico para cama.***- Esta parte del sistema se incluye por petición especial de la persona con paraplejia, consiste en una manta termoeléctrica ubicada en la cama de la persona, el objetivo de esta manta es adecuar la

temperatura de la cama antes de que el usuario ingrese a esta. El motivo es un problema de circulación de sangre hacia las extremidades inferiores, debido a la paraplejia el usuario a explicado que la temperatura corporal en su cuerpo no es uniforme y tiene problemas para dormir si no se calienta la cama previamente. De esta forma la inclusión de la manta en el sistema se vuelve muy importante. La solución esta conformada por software y hardware electrónico que permite la activación temporizada de la misma y la verificación de su estado a través de la interfaz.

- ***Ventanas y puertas.-*** La inclusión de ventanas y puertas requiere de un desarrollo y construcción de hardware electrónico y partes mecánicas que puedan ser adaptadas a los accesos ya instalados anteriormente y permitan su control. Debido a la importancia de los accesos hacia la vivienda es importante la inclusión de controles para estos en el sistema; pero no se debe descuidar la funcionalidad y más importante aún la seguridad que las puertas y ventanas deben tener. La seguridad es una parte fundamental en los accesos y por esta razón los dispositivos no deben bloquearse en ninguna situación, es decir estos dispositivos deben permanecer funcionales sin importar las variables de su entorno, por ejemplo las personas no se pueden quedar encerradas o fuera de las instalaciones por un fallo eléctrico o mecánico; para entender mejor este punto se toma como ejemplo el caso de el portón ubicado en la entrada del garaje de la vivienda, este cuenta con un portero eléctrico y una cerradura eléctrica, esta última ante un fallo eléctrico puede funcionar con una llave tradicional y el acceso no se vería bloqueado, el caso del portón es similar, por medio de un control manual se desacopla el

mecanismo del portón de las hojas que conforman el mismo y se puede acceder libremente sin dificultades. El enfoque que se le da ha la automatización de puertas y ventanas no incluye seguros ni cerraduras, una de las razones es el gasto adicional que se añadiría al proyecto al instalar de estos mecanismos. Además considerando la seguridad de la vivienda se llega a determinar que no es conveniente liberar seguros de manera remota y necesariamente debe estar presente en el acceso una persona con su respectiva llave. En consecuencia ha esto el sistema se encarga de las funciones de abrir las puertas y ventanas siempre y cuando los seguros físicos se encuentren previamente liberados. El usuario puede abrir y cerrar puertas y ventanas mediante la interfaz y también simplemente empujando los accesos como se lo hace normalmente. De manera similar a las cortinas los cambios en la interfaz se ven reflejados siempre y cuando se accionen los accesos desde la misma, si se acciona un acceso manualmente o si el seguro esta puesto este se verá reflejado en la interfaz, si el seguro se encuentra puesto la interfaz se bloqueará hasta que el seguro sea liberado manualmente.

### **3.1.3.1. La interfaz de usuario.**

Las características del sistema y su interfaz deben ser amigables he intuitivas, permitiendo el fácil manejo de todos los actuadores instalados en los dispositivos incluidos y para mayor comodidad se los separa en áreas, que no es más que agrupar todos los controles de un área en una sub-interfaz que puede ser accedida desde un menú general; consecuentemente se obtiene una interfaz ordenada y delimitada por áreas.

Se debe contar con un diseño adaptable que permita la correcta visualización de los controles en los diferentes dispositivos desde donde se podrá acceder. Ya que el sistema no está orientado a un solo dispositivo final, la interfaz del sistema debe adaptarse a cualquier tamaño de pantalla, esto se logra utilizando el framework JQuery Mobile.

Los parámetros de seguridad están basados en un formulario que el usuario deberá llenar correctamente para poder acceder a la interfaz, para esto se maneja una sesión la cual se mantendrá abierta solo desde el terminal desde el que haya iniciado la misma y se cerrará una vez que el usuario finalice la sesión o un tiempo de inactividad haya transcurrido.

## **4. CAPÍTULO IV**

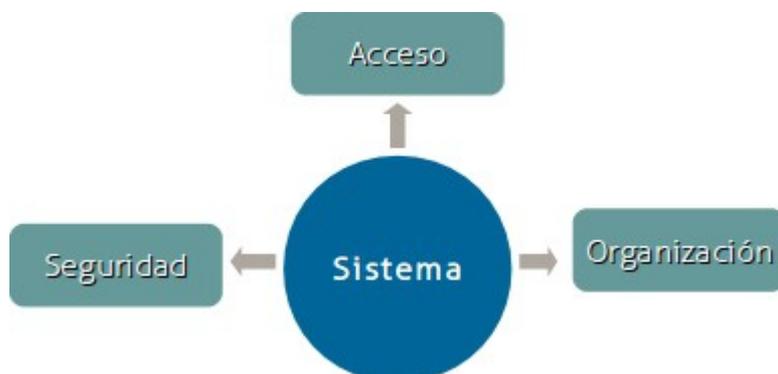
### **INTRODUCCIÓN**

El presente capítulo contiene todo lo referente al diseño y construcción de el sistema propuesto. Se hace un planteamiento referente al diseño del sistema domótico para luego desarrollarlo, debido a que se manejan dos tipos de redes se recurre a un análisis de la coexistencia de las redes de datos y dispositivos que se instalan en la vivienda, luego se estudia la relación de los elementos mediante una topología, para dar paso a la forma en que estos se comunican entre sí. Finalmente se pasa al desarrollo de la interfaz desde donde se controlarán los dispositivos.

#### **4.1. DISEÑO DEL SISTEMA DOMÓTICO Y ANÁLISIS DE COBERTURA INALÁMBRICA.**

En resumidas palabras el diseño del sistema se basa en la instalación de dos tipos de redes en la vivienda, la primera conteniendo a todos los dispositivos que se van a controlar y la segunda que permite la comunicación desde un dispositivo final hacia un nodo central basada en IEEE 802.11b/n; para que puedan relacionarse estas tecnologías, se utiliza un dispositivo que puede comunicarse con ambas, siendo la puerta de enlace entre ambos tipos de comunicaciones. Los dispositivos se configuran en frecuencias y canales que no se interfieran entre sí de tal manera que no haya conflictos en el tráfico de información.

#### 4.1.1. Planteamiento del diseño.



**Figura 18.** Características del sistema.

**Fuente:** Propia.

En la figura 18 se observan las características generales del sistema propuesto, cada una de ellas abarcan aspectos y funciones que son importantes y necesarias; la unión de estas características apuntan hacia el único objetivo de obtener un sistema funcional.

En accesibilidad se incluye la posibilidad de acceder al sistema desde cualquier dispositivo final, tanto de manera local como remota a través de Internet, esto elimina la dependencia de un solo dispositivo para su uso, obteniendo un sistema multiplataforma.

La organización de los elementos del sistema permiten una interacción y comunicación apropiada entre el usuario y el sistema; así también entre los diferentes elementos que lo conforman. La interfaz esta dividida en sub-menus que están relacionados con las áreas a controlar y permiten que el usuario pueda acceder si problemas a cada una de ellas. De manera similar los dispositivos instalados se organizan de tal forma que la información pueda fluir de manera adecuada y los resultados de las interacciones puedan reflejarse rápidamente en la interfaz.

En las medidas de seguridad se contemplan dos posibles casos con las siguientes consideraciones:

- El primer caso expone la situación cuando se desea acceder localmente a la interfaz, para esto el usuario debe primero obtener la clave WIFI de la red local, la cual puede ser proporcionada solo por un miembros de la familia o el administrador de la red. La red inalámbrica local esta protegida mediante WPA2 con cifrado AES para frustrar posibles ataques, además se tiene una clave de mayor a diez dígitos combinando números, símbolos y letras. Si el usuario ha accedido a la red con la clave correcta, debe apuntar a una dirección ip específica para que pueda acceder hacia la interfaz, donde se le pedirá la clave del sistema. Finalmente se protege al equipo que brinda WIFI cambiando las claves por defecto de fabrica, para que ningún usuario no autorizado pueda modificar la configuración del equipo.
- En el segundo caso se contempla el acceso a la interfaz desde Internet, para esto el dispositivo desde el que se quiere ingresar debe ser añadido previamente a la red VPN<sup>37</sup> de Hamachi<sup>38</sup>, esto requiere que el administrador acepte la unión del equipo por medio del panel de Hamachi. Gracias a las conexión PPTP<sup>39</sup> el trafico dentro de la VPN siempre viaja encriptado. Nuevamente aquí el usuario debe tener conocimiento de la ip privada y la respectiva contraseña de la interfaz.

Para terminar con la parte de seguridad hay que aclarar que, cuando el usuario accede a la interfaz se crea una sesión a nivel de aplicación que vincula al servidor con el dispositivo

---

37 VPN.- Red privada Virtual.

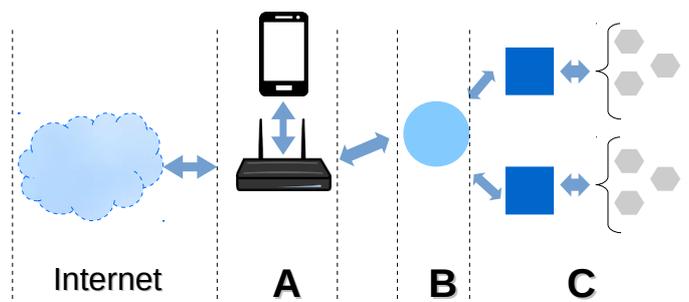
38 Hamachi.- Es una herramienta que brinda Logmein para crear VPNs, pudiendo se gratuitas o de pago.

39 PPTP.- Protocolo de tunelización punto a punto.

desde el que se accede. Como consecuencia se usa esta sesión cada vez que el usuario interactúa con la interfaz, comprobando que se haya iniciado sesión antes de cualquier acción. Si el usuario no ha iniciado sesión y aunque este tenga el link exacto de los dispositivos no podrá controlarlos y el sistema lo direccionará a la pantalla de ingreso para que introduzca su respectiva clave.

#### **4.1.1.1. Comunicación básica entre secciones de la red.**

La comunicación básica de las secciones del sistema se pueden ver en la figura 19, la relación que tienen estos se asocia mediante flechas en el gráfico indicando una comunicación en ambos sentidos, para una mejor explicación se ha dividido por secciones las cuales se explican a continuación.



**Figura 19.** Comunicación básica entre secciones de la red.  
Fuente: Propia.

En la parte izquierda encontramos la sección Internet, conexión contratada que provee la

empresa de telecomunicaciones CNT<sup>40</sup>, se utiliza la línea telefónica para acceder al servicio y como parte de la contratación también se ha instalado un equipo que brinda conexión para dispositivos que soporten WiFi y Ethernet cableado.

La sección **A** esta conformada por un equipo que esta conectado directamente a Internet, este equipo provee la instalación de una red IEEE 802.11b/n y puertos Ethernet. A el se conectaran todos los dispositivos de la red WLAN(dispositivos de usuario final).

Continuando con la descripción, en la sección **B** encontramos el nodo central de la red, el cual esta dotado con las tecnologías que permitan la comunicación entre Ethernet y IEEE 802.15.4, además este nodo tiene internamente un servidor web en donde se almacena la interfaz del sistema. Este es el nodo central del sistema, esta encargado de coordinar la interacción entre el usuario y los dispositivos instalados. El hardware de este dispositivo se basa en la placa Raspberry pi con un modulo XBEE conectado por medio de USB.

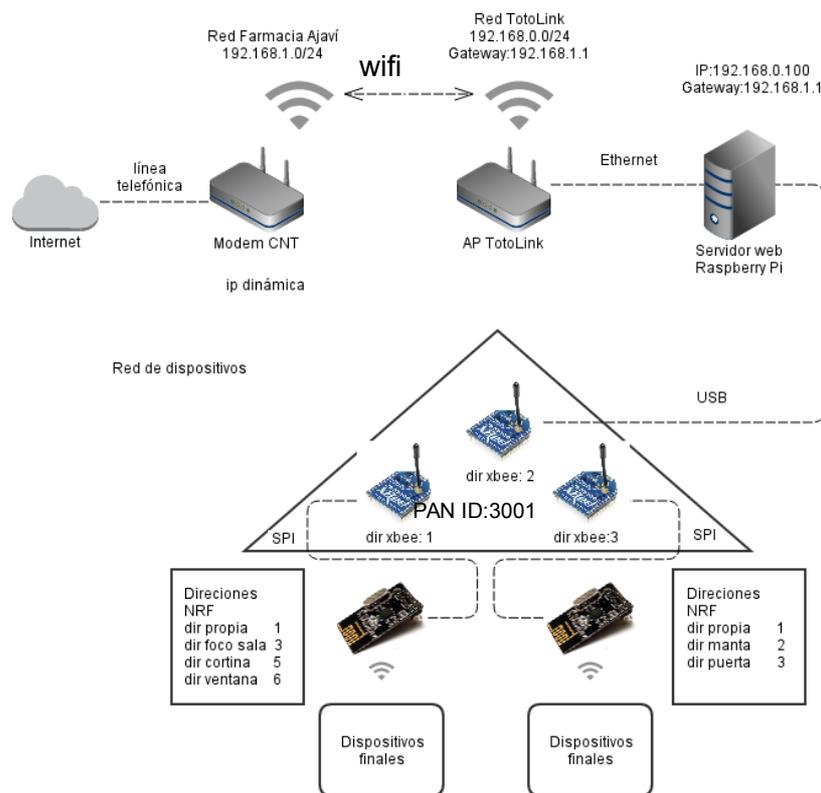
En la sección **C** se observan dos dispositivos de color azul, estos se encargan de la recepción y envío de información entre el nodo central(sección B) y los dispositivos instalados en las cosas a controlar. Su correcta ubicación permite aumentar el alcance de la red de dispositivos. Cada uno están formado por un sistema microcontrolado(Arduino), un modulo XBEE y un NRF24L01. Estos dispositivos permiten la comunicación entre la red formada por los módulos XBEE(IEEE 802.15.4) y los dispositivos conectados a los diferentes sensores y actuadores los que contienen un NRF24L01. Finalmente en color gris se pueden observar los dispositivos que se van ha controlar, todos ellos están constituidos básicamente por un sistema microcontrolado(Arduino) y un NRF24L01.

---

40 CNT.- Empresa nacional de telecomunicaciones, es uno de los proveedores de Internet mas grandes de Ecuador.

#### 4.1.2. Topología real de la red de dispositivos.

Para la construcción de los dispositivos primero se debe establecer una topología real a instalarse, y ya que toda topología de red consta de un diagrama de comunicación lógico y físico, se presenta a continuación la topología lógica de la red a implementarse.

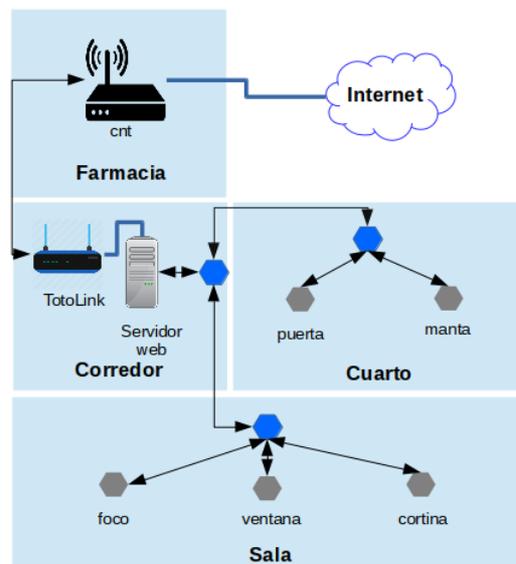


**Figura 20.** Topología lógica de la red.  
**Fuente:** Propia.

La figura 20 muestra la relación lógica que tienen los dispositivos y se detallan las direcciones de cada uno de ellos. Como se puede apreciar cada dispositivo se interconecta utilizando su propia tecnología de comunicación, por lo cual existen tres tipos de direccionamiento.

- *Direccionamiento IP.-* para los dos puntos de acceso inalámbricos y el servidor. En el caso de los puntos de acceso, estos mantienen una conexión vía WIFI en la cual el equipo TotoLink funciona como un equipo de enlace hacia el módem de el proveedor de Internet y también divide la red en dos segmentos distintos.
- *Direccionamiento Xbee.-* para los módulos xbee que se encuentran dentro del triangulo de la figura, como se observan estas direcciones son sumamente cortas e identifican a cada módulo dentro de su PAN ID.
- *Direccionamiento NRF.-* para los dispositivos finales que se van a controlar, se utiliza un método de direccionamiento muy similar al anterior pero basado en tecnología soportada por los módulos NRF24L01.

Una vez presentada la topología lógica de la red a implementarse ahora se procede a ilustrar la topología física de los dispositivos, mostrando como se distribuyen en las áreas del domicilio.

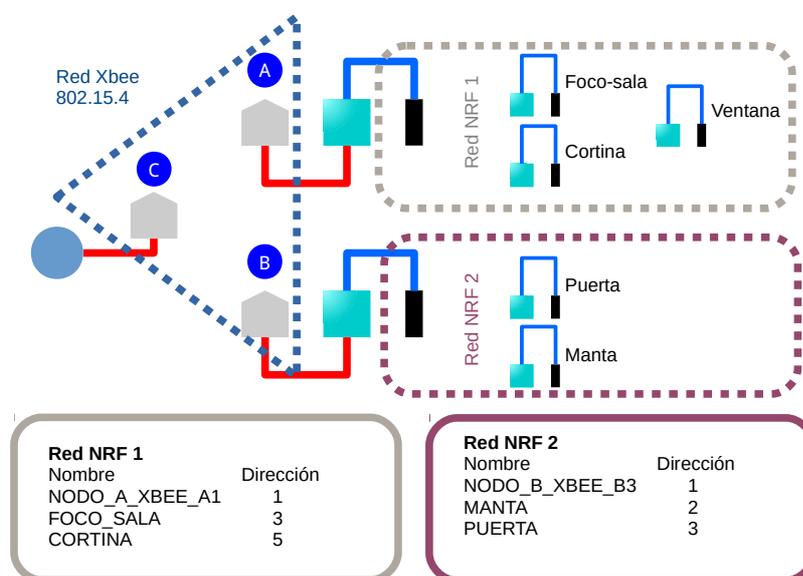


**Figura 21.** Topología física de la red.  
Fuente: Propia.

La figura 21 ilustra los elementos de la red distribuidos en una topología física en las áreas del domicilio, de color azul se muestran los nodos que conforman la red xbee, mientras que los dispositivos de color gris representan a los dispositivos finales. Si el lector desea ver exactamente donde se han ubicado los dispositivos puede dirigirse al Anexo 33, en donde se muestra como se distribuyen los dispositivos en relación al plano del domicilio.

De este modo se llega al diseño de la figura 22, en este diagrama se pueden observar los nombres reales de los dispositivos y como están asociados entre si. En la parte inferior se puede ver las tablas de direcciones correspondientes a la *Red NRF 1* y a la *Red NRF 2*.

Todos los diseños y consideraciones posteriores se refieren a este diagrama.

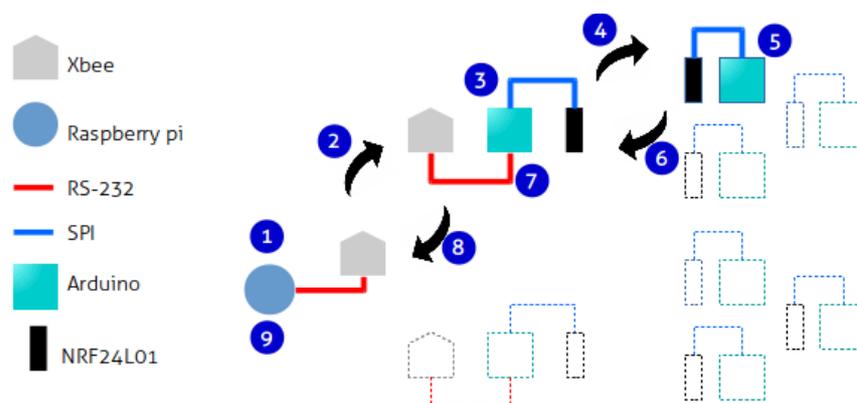


**Figura 22.** Red implementada.

**Fuente:** Propia

#### 4.1.2.1. Comunicación y relación entre elementos.

En este apartado se describe como se comunican los elementos de la red de dispositivos y como estos se relacionan.



**Figura 23.** Comunicación y relación entre elementos.

Fuente: Propia

La figura 23 describe los pasos para el cambio de estado de un dispositivo o una petición de estado del mismo, en la parte izquierda se pueden ver los elementos que conforman la red así como los tipos de comunicación en cada nodo.

El nodo principal, Raspberry pi y Xbee es el único nodo que puede hacer peticiones y cambios de estado en la red, todos los demás siempre están escuchando y pendientes por si tienen que hacer o reportar algún cambio. Los mensajes enviados y recibidos por el nodo principal se basan en simples caracteres; es decir letras, números o símbolos. Esta manera de trabajo hace que siempre se tenga un canal limpio libre de transmisiones, dando como resultado el mínimo retardo en las respuestas a las operaciones y que las comunicaciones siempre se realicen en un orden secuencial, una a una.

Los nodos en la parte derecha están conectados directamente a los actuadores, se han

diseñado para realizar cambios en los dispositivos a controlar y para detectar cambios manuales en los mismos, cuando sucede un cambio manual, este es almacenado en el mismo nodo y si el nodo principal necesita su estado actual simplemente lo envía; en caso de una petición de cambio de estado, se procede a cambiar el estado del dispositivo final y luego se almacena y envía su estado.

Como ejemplo se describen los pasos que se realizarían para el cambio o petición de estado en un nodo conforme a la figura 23:

- 1.- En el paso uno la Raspberry pi envía la petición de cambio o estado en un dispositivo final, dicha petición, para el caso algún carácter X viaja hacia el Xbee conectado directamente a ella por medio de una comunicación serial.
- 2.- Por medio de condiciones preestablecidas se direcciona el Xbee hacia el siguiente Xbee; que este a su vez esté conectado al nodo final por medio de NRF24L01. Se envía el dato.
- 3.- El dato es recibido por el Xbee destino y se envía al arduino por comunicación serial, por medio del programa grabado en el arduino el dato X entra a una serie de condiciones para determinar a que nodo final se debe enviar el mensaje, se envía el dato a través de comunicación SPI hacia el NRF24L01. Este nodo queda en estado de espera a una respuesta.
- 4.- Se envía el dato X al nodo final correspondiente.
- 5.-El NRF24L01 del dispositivo final recibe el dato X y lo envía al arduino por comunicación SPI, este revisa en el programa principal que acción debe realizar, en

caso de petición de estado simplemente responde con el estado actual con un dato Y (simple carácter), si el caso demanda de un cambio de estado se procede a realizar el cambio, se almacena localmente el estado y este se envía con un dato Y (nuevamente un simple carácter). La respuesta viaja por SPI hacia el NRF24L01 direccionado hacia el nodo intermedio del paso 3.

- 6.- Se envía el dato Y.
- 7.- El nodo que quedo en estado de espera recibe el dato Y como una respuesta por medio de su NRF24L01 lo comunica por SPI al arduino y este por comunicación serial a su Xbee.
- 8.- Se envía el dato Y hacia el nodo principal.
- 9.- El Xbee del nodo principal recibe el dato Y y lo envía por comunicación serial hacia la Raspberry Pi.

*Observaciones:*

-Es importante notar que el Xbee conectado directamente a la Raspberry Pi, es direccionado hacia un Xbee dependiendo del destino final cada vez que se genera una petición de cambio o estado. Se puede direccionar a cualquier nodo Xbee intermedio.

-Los Xbee intermedios siempre tiene como dirección destino al Xbee conectado a la Raspberry Pi, de esta manera estos se direccionan solo en una configuración inicial (antes de ser conectados al sistema) que nada tienen que ver con el programa grabado en el arduino del nodo intermedio, el arduino de este nodo ve al Xbee como una simple comunicación serial.

-El direccionamiento entre los nodos Xbee se puede observar en el Anexo 6.

-Los enlaces inalámbricos entre los Xbee son extremadamente eficientes, es decir los paquetes no necesitan de retransmisiones ya que muy difícilmente se pierden, esto último se cumple siempre y cuando se respeten los rangos de distancias. Mientras que los enlaces inalámbricos de los NRF24L01 si necesitan algún sistema de retransmisión de datos ya que funcionan con una tecnología diferente pero no representan un problema para el sistema ya que la cantidad de tráfico de datos es muy pequeña.

-Los tiempos de respuesta de la red son muy pequeños y están bajo el segundo de tiempo. Esto se debe gracias a que el canal siempre esta limpio y disponible para las comunicaciones. Ninguna comunicación se inicia sin orden del nodo central.

#### **4.1.3. Análisis de frecuencias utilizadas.**

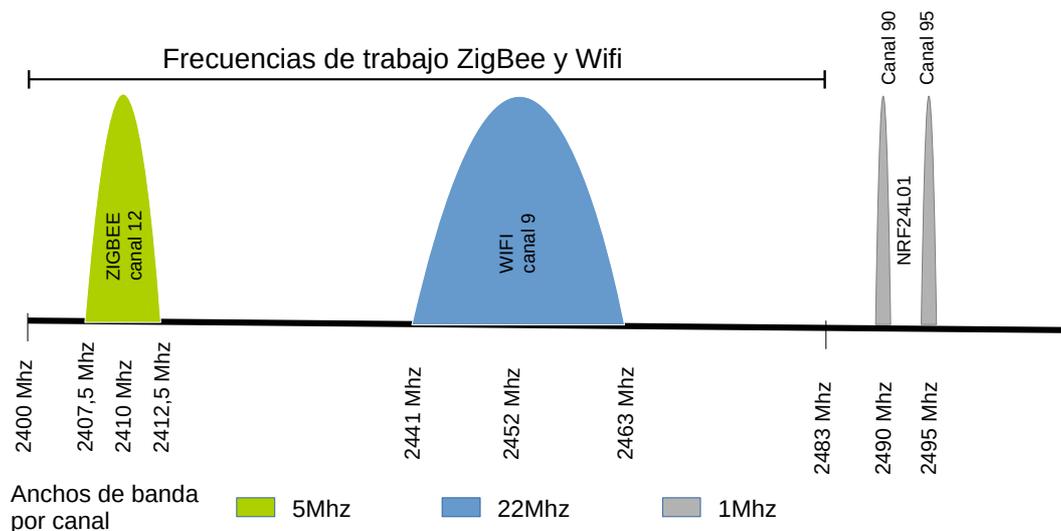
Como se había explicado anteriormente, se utilizan diferentes tecnologías para el correcto funcionamiento del sistema domótico, es necesario analizar los rangos de frecuencia utilizadas por cada una, para posteriormente elegir canales adecuados que no interfieran entre ellos.

- El estándar ZigBee trabaja en las frecuencias comprendidas entre los 2400 Mhz hasta los 2483,5 Mhz. Se tienen disponibles 11 canales de transmisión con una ancho de banda de 5 Mhz cada uno.
- Las redes basadas en IEEE 802.11b/n trabajan en el rango de 2402 Mhz hasta los 2483 Mhz. Aquí existen 13 canales de transmisión que se sobreponen con una ancho de banda de 22 Mhz por canal.

- Finalmente los módulos NRF24L01 trabajan desde los 2400 Mhz hasta los 2525Mhz. Estos módulos pueden trabajar en 126 canales de transmisión y su ancho de banda depende de la velocidad de transmisión configurada, teniendo 1 Mhz para 1 Mbps y 2 Mhz para 2 Mbps.

Con esta información se deduce que las tecnologías ZigBee, Wifi y los módulos NRF24L01 trabajan prácticamente en el mismo rango de frecuencias, con la diferencia que estos últimos pueden operar en frecuencias mas altas que las dos primeras tecnologías.

De esta manera se configuran las frecuencias de los dispositivos como se muestra en la figura 24.



**Figura 24.** Frecuencias utilizadas.

**Fuente:** Propia.

La elección de estas frecuencias se la hizo con el fin de que las tecnologías utilizadas no interfieran entre sí. Se aprovecha la ventaja de que los módulos NRF24L01 pueden trabajar

en frecuencias mayores para que en ningún caso interfieran con ZigBee y Wifi.

Finalmente se observa que ZigBee y Wifi deben coexistir en el mismo rango de frecuencias y para evitar interferencias se elijen canales adecuados.

#### **4.1.4. El nodo central Raspberry pi.**

La aplicación esta basada en su gran mayoría en esta placa ya que gracias a sus características y capacidades es posible construir un sistema centralizado, sobre su hardware y software que cumplen con las necesidades y expectativas del sistema domótico. Es conveniente comenzar el diseño de la aplicación por este nodo por la naturaleza centralizada de la aplicación, en otras palabras la Raspberry Pi es el elemento mas importante de todo el sistema; y como consecuencia todo se desprende de este nodo.

A continuación se detalla como se configuran las herramientas necesarias para la construcción del sistema, y se muestra el proceso de desarrollo del mismo.

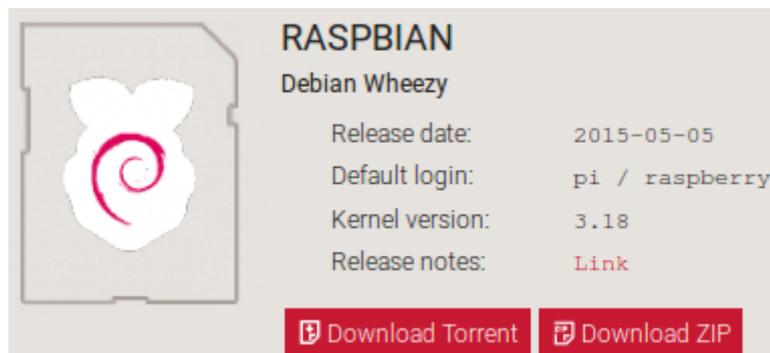
##### ***4.1.4.1. El Sistema Operativo***

Raspbian es el sistema operativo elegido para el desarrollo del proyecto, como se comento en el primer capítulo posee características heredadas de Debian por lo cual es muy robusto y estable, además este S.O es el mas ampliamente utilizado por los usuarios de Raspberry Pi teniendo una documentación detallada del mismo.

Para la instalación es necesario descargar una imagen del S.O desde su pagina oficial<sup>41</sup>.

---

41 Pagina oficial de Raspberry Pi.- [www.raspberrypi.org/download/](http://www.raspberrypi.org/download/)



**Figura 25.** Descarga de Raspbian.

**Fuente:** [www.raspberrypi.org/download/](http://www.raspberrypi.org/download/).

Por defecto el sistema operativo tiene configurados dos usuarios, uno con permisos de administrados y otro con permisos normales, por seguridad todos los credenciales de acceso por defecto deben ser cambiados para evitar posibles intrusiones.

La instalación de Raspbian, modificación de los credenciales y la asignación de una ip estática se pueden observar en el Anexo 4.

Resumiendo luego de la instalación y configuración se tiene:

- Configuradas las nuevas credenciales de acceso tanto para el usuario normal (pi) como para el súper usuario (root).
- El Sistema operativo Raspbian actualizado y funcionando.
- Y una IP estática dentro de la LAN.

#### 4.1.4.2. Instalación de herramientas y configuración.

- Con el sistema actualizado se instalan las herramientas que se utilizan en el desarrollo de la aplicación final con los siguientes comandos.

```
root@raspberrypi:~# apt-get install apache2 php5 proftpd minicom
```

Como se había mencionado en el capítulo correspondiente al análisis de las herramientas los elementos a instalarse son un servidor web Apache (apache2), el lenguaje de programación orientado a la web PHP (php5), un servidor FTP(proftpd) que posteriormente permitirá el fácil reemplazo de los archivos y finalmente una interfaz para manejar las comunicaciones seriales (minicom) con el hardware conectado a la placa.

- Para la publicación de una página web en Internet normalmente hace falta adquirir una IP pública y un nombre de dominio; pero en el caso de este proyecto se utilizará el servicio gratuito de Logmein, que consiste en la creación de una VPN a través de una de sus herramientas llamada Hamachi. Esto añade robustez a la aplicación final, ya que todo el tráfico se encontrará encriptado cuando el usuario haga uso de la VPN. El proceso de configuración se detalla en el Anexo 5.

Hasta aquí se tiene la configuración inicial del nodo central del sistema y se puede proceder al desarrollo de la aplicación web, en otras palabras el resto de partes del sistema se van añadiendo a este nodo.

#### 4.1.5. Librería RadioHead.

La utilización de librerías con la plataforma Arduino es un método que permite el rápido desarrollo de soluciones basadas en esta placa. RadioHead es una librería enfocada en módulos de comunicación inalámbricos utilizando el bus SPI de Arduino, la variedad de módulos que soporta esta librería es extensa, se puede leer mas sobre ella en su pagina oficial<sup>42</sup>(ver nota al pie de página). Precisamente esta librería es la que permite que Arduino trabaje con los módulos NRF24L01. Algunas de las ventajas que se aprovechan en este proyecto son:

- Capacidad de direccionamiento entre módulos, esto permite de una manera práctica crear redes compuestas por estos módulos, con un sistema de direcciones cortas, proporcionando un sistema de retransmisiones y acuses de recibo embebido en el mismo módulo.
- La librería se puede adaptar a cualquier placa compatible con el IDE Arduino, basta con que el microcontrolador abordo cuente con un bus SPI.
- Por último la optimización para trabajar con Arduino y derivados permite aprovechar la memoria del microcontrolador, de manera tal que los programas de aplicación final pueden ser muy complejos y relativamente grandes.

Exactamente las librerías utilizadas son:

<SPI.h> .- Se encarga de el bus SPI del microcontrolador.

<RH\_NRF24.h> .- Maneja la comunicación del módulo con Arduino por el bus SPI, parámetros como frecuencia de transmisión, velocidad de transmisión y la potencia de salida

---

42 RadioHead.- Pagina oficial <http://www.airspayce.com/mikem/arduino/RadioHead/>

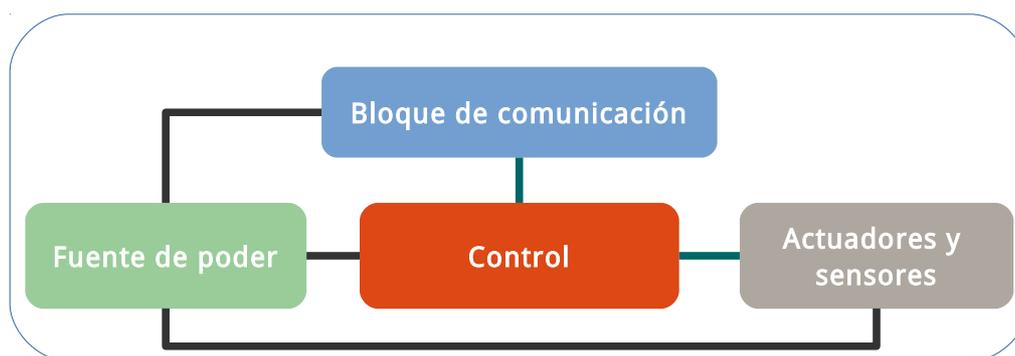
se pueden configurar con esta librería.

<*RHReliableDatagram.h*> .- Esta librería permite el envío y la recepción de los datos con confirmación, lo más importante de esta librería es la capacidad de direccionar los paquetes. Parámetros como número de reintentos y tiempo de espera por una respuesta pueden ser configurados gracias a esta librería.

El impacto del uso de las librerías RadioHead ha beneficiado profundamente al proyecto, ya que al utilizar estos módulos de comunicación se logra reducir significativamente los costos en el hardware utilizado, de otra manera se hubiere tenido que recurrir a tecnologías de comunicación mas costosas.

#### 4.1.6. Diseño y construcción de dispositivos.

Los dispositivos electrónicos se componen básicamente por las siguientes partes; pero existen alguna excepción en donde no se ha incluido algún bloque ya que no es necesario.



**Figura 26.** Partes de dispositivos electrónicos.

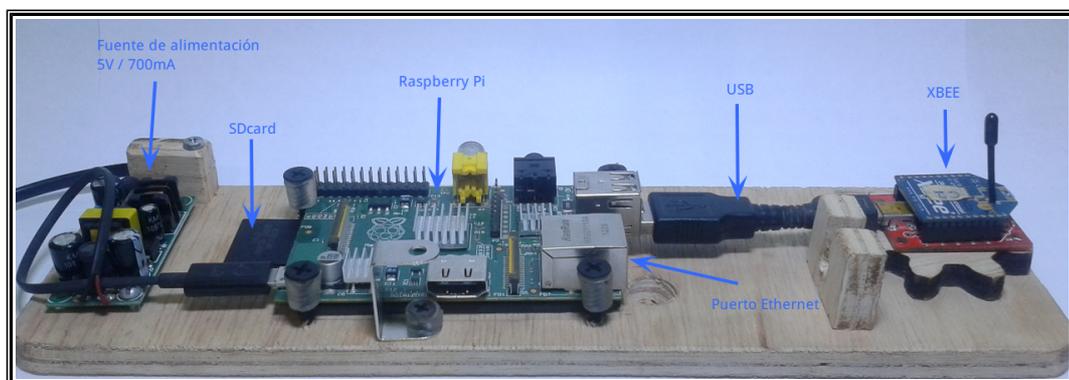
**Fuente:** Propia

En la figura 26 se muestran las partes esenciales que conforman a un nodo en la red de

dispositivos, a continuación una breve descripción de las mismas:

- **Control.-** Es la parte mas importante del nodo ya que aquí es donde se guarda el programa principal con las instrucciones necesarias para que todo funcione de una manera organizada y adecuada. Esta parte es básicamente un Arduino a excepción del nodo principal todos los nodos poseen un Arduino que es el encargado de procesar la información, administrar las comunicaciones e interactuar con el exterior por medio de actuadores o recolectar información por medio de sensores.
- **Bloque de comunicación.-** Este bloque esta formado por los módulos de comunicación conectados directamente al bloque de control mediante comunicación serial o SPI, evidentemente estos módulos permiten la interconexión del nodo con otros nodos.
- **Actuadores y sensores.-** Los actuadores permiten la interacción de los nodos con el exterior pudiendo por ejemplo encender o apagar una lámpara, mientras que los sensores permiten la recolección de estados de los dispositivos controlados. Algunos nodos carecen de este bloque ya que su función no es la de interactuar con el exterior o recolectar estados, si no la de intercomunicar a los nodos mismos.
- **Fuente de poder.-** Es la parte que alimenta al nodo de energía, proporcionando los niveles adecuados de voltaje y corriente para que este pueda funcionar de una manera correcta. Esta es una parte muy importante y fundamental ya que el sistema necesita estar conectado permanentemente a la red eléctrica.

#### 4.1.6.1. Nodo Principal.



**Figura 27.** Partes Nodo Principal.

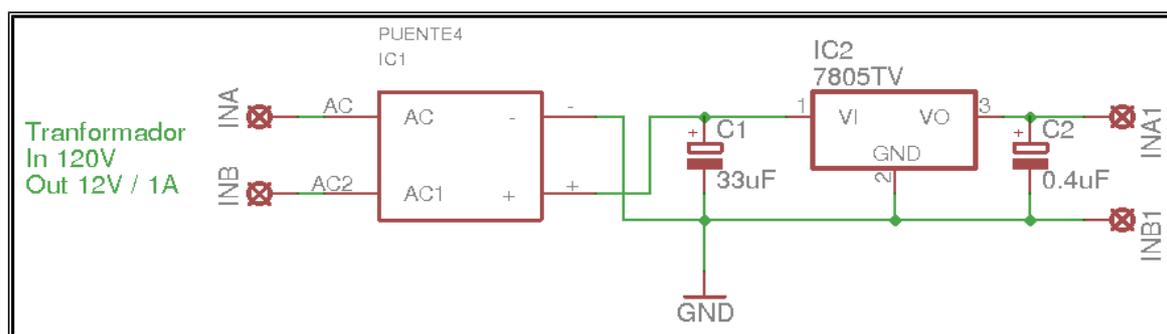
**Fuente:** Propia

La figura 27 muestra los elementos que conforman el nodo principal de la red, el corazón del sistema, la Raspberry Pi contiene la aplicación web por medio de la cual es usuario controla los dispositivos, este nodo esta encargado del envío y recolección de estados de los dispositivos controlados. Cuenta con dos tipos de comunicaciones hacia el exterior, una comunicación Ethernet que une a la placa a una red LAN por medio de un dispositivo central (Access Point) y este a su vez permite la interconexión con Internet; la dirección IP local de placa es la *192.168.0.100*. La segunda comunicación es la proporcionada por el módulo Xbee C (con dirección xbee = 2) que permite el envío y recepción de paquetes hacia los dispositivos controlados, en la figura referida se puede observar una conexión por medio de USB a este dispositivo, sin embargo la placa ve esta conexión como un puerto serial virtual, por lo que la comunicación es serial con una velocidad de 9600 baudios, sin control de flujo, 8bits de datos, sin paridad y con 1bit de parada.

En la parte izquierda se observa un tarjeta SD que contiene el sistema operativo y los archivos propios del sistema domótico.

- *Diseño de fuente de alimentación.*

Para la alimentación de la placa es necesaria una fuente de alimentación que entregue 5V y 700mA, estas especificaciones están señaladas por el fabricante. La fuente ideal para el nodo debe ser pequeña, eficiente y que cuente con algún tipo de seguridad.



**Figura 28.** Fuente de alimentación nodo principal.

**Fuente:** Propia

En la figura 28 se puede observar el diagrama electrónico de una fuente diseñada para el nodo central. Esta fuente está basada en un transformador que entrega 12V y 1A a partir de los 120V de la red eléctrica, luego se rectifica mediante un puente de diodos de 1A, para la regulación de voltaje se recurre al regulador LM7805 que entrega a su salida 5V con una corriente máxima de 1,5A. Este regulador trabaja con los condensadores  $C1 = 33\mu\text{F}$  y  $C2 = 0,1\mu\text{F}$  que recomienda el fabricante.

Las ventajas de esta fuente de alimentación son:

- Es fácil de implementar y construir.
- Brinda el voltaje y la corriente adecuada para la Raspberry pi.

Las desventajas de esta fuente son:

- Al utilizar un transformador de 12V y 1A el tamaño del circuito de la fuente sobrepasa al tamaño de la placa Raspberry pi, característica que dificulta su instalación en espacios reducidos.
- Este tipo de fuente es poco eficiente y transforma la energía excedente en calor, es decir el regulador debe disipar la diferencia entre los 12V que ingresan y los 5V que entrega en forma de calor, lo que traducido aun funcionamiento pleno solo el regulador disiparía 4,9W ( $7V \cdot 0,7A$ ), más la potencia entregada a la placa que es 3,5W sumarían 8,4W. Aquí se observa que el regulador consume mas que la propia Raspberry pi.
- Carece de algún sistema de seguridad, aunque se pueden añadir más dispositivos para dar mas robustez a este diseño, esto afecta directamente al costo total de la fuente y por ende al proyecto en si.

Como se observa la fuente propuesta es funcional y se puede construir, pero no es la óptima para el nodo por las desventajas expuestas anteriormente. Por lo cual se analizan otro tipo de fuentes que se adapten mejor a las necesidades del proyecto y se opta por utilizar una fuente conmutada. Pero el diseño de este tipo de fuentes es mucho mas complejo que la fuente expuesta anteriormente.

La ventajas de una fuente conmutada:

- Son muy eficientes, una fuente que entrega 5V y 1A solo disipa solo un 10% en sus componentes, en el caso de Raspberry pi se entrega 3,5W a la placa y en los componentes de la fuente solo se disipan 0,35W.

- Su tamaño es reducido ya que no necesitan un gran transformador para funcionar. Por su característica de trabajar a muy elevadas frecuencias hacen uso de transformadores muy pequeños.
- Por la gran acogida que ha tenido esta tecnología, estas fuentes se las puede encontrar comercialmente a bajos costos.
- Cuentan con sistemas de protección ante corto circuitos y subidas de tensión.

Desventajas de una fuente conmutada:

- La desventaja de una fuente conmutada es su diseño más elaborado, estas fuentes incorporan un gran número de componentes electrónicos que en su gran mayoría son de montaje superficial. En caso de daño es muy difícil de reparar.

Como se observa una fuente conmutada es la más conveniente para este nodo, la construcción de una fuente de este tipo es algo que escapa a este texto, ya que solo el diseño de una fuente así conllevaría una cantidad de tiempo considerable, por lo cual se opta por adquirir una fuente ya construida.



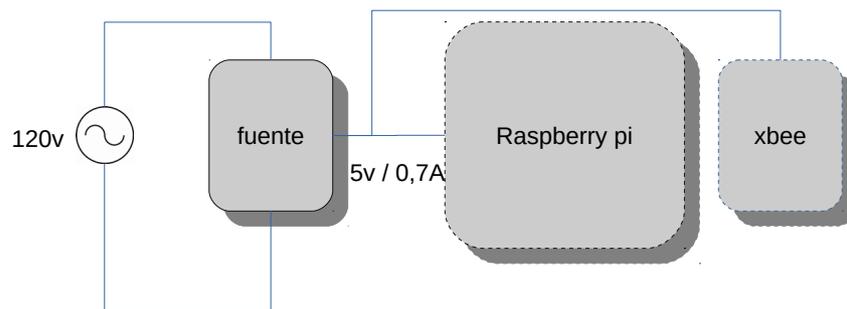
**Figura 29.** Fuente de alimentación nodo principal.

**Fuente:** Propia

La figura 29 muestra la fuente que se instala junto a la placa Raspberry pi, esta fuente puede funcionar con voltajes de entrada de 110V a 230V, a su salida entrega 5V con una

corriente máxima de 1A y posee protección contra cortocircuitos. La potencia entregada hacia la placa a pleno funcionamiento es de 3,5W.

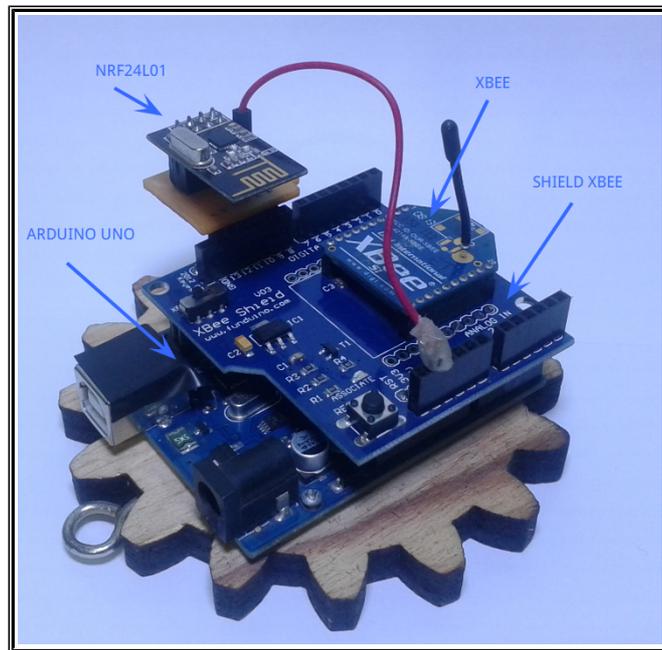
De esta manera el diseño final que da de la siguiente manera:



**Figura 30.** Diagrama de conexión eléctrico nodo principal.

**Fuente:** Propia

#### 4.1.6.2. Los Nodos Intermedios



**Figura 31.** Partes de Nodos Intermedios.

**Fuente:** Propia

En la figura 31 se pueden observar las partes que conforman a los nodos intermedios, la placa Arduino Uno es la base de este nodo la cual se encarga de administrar las comunicaciones y procesar los datos recibidos mediante un programa principal, se utiliza comunicación serial para la interconexión con el módulo Xbee, mientras que para conectarse con el NRF24L01 se utiliza el bus SPI.

Las funciones principales de este nodo son:

- Interconectar al nodo principal con los nodos finales (conectados a los dispositivos a controlar), mediante sus dos módulos de comunicación.
- Procesar la información recibida para que esta pueda llegar a su destino.

Los dos nodos intermedios utilizados en el proyecto son exactamente idénticos en

hardware, la diferencia entre ambos nodos radica en el programa cargado en cada nodo, los cuales pueden ser analizados en el Anexo 7, y su respectivo diagrama electrónico en el Anexo 8.

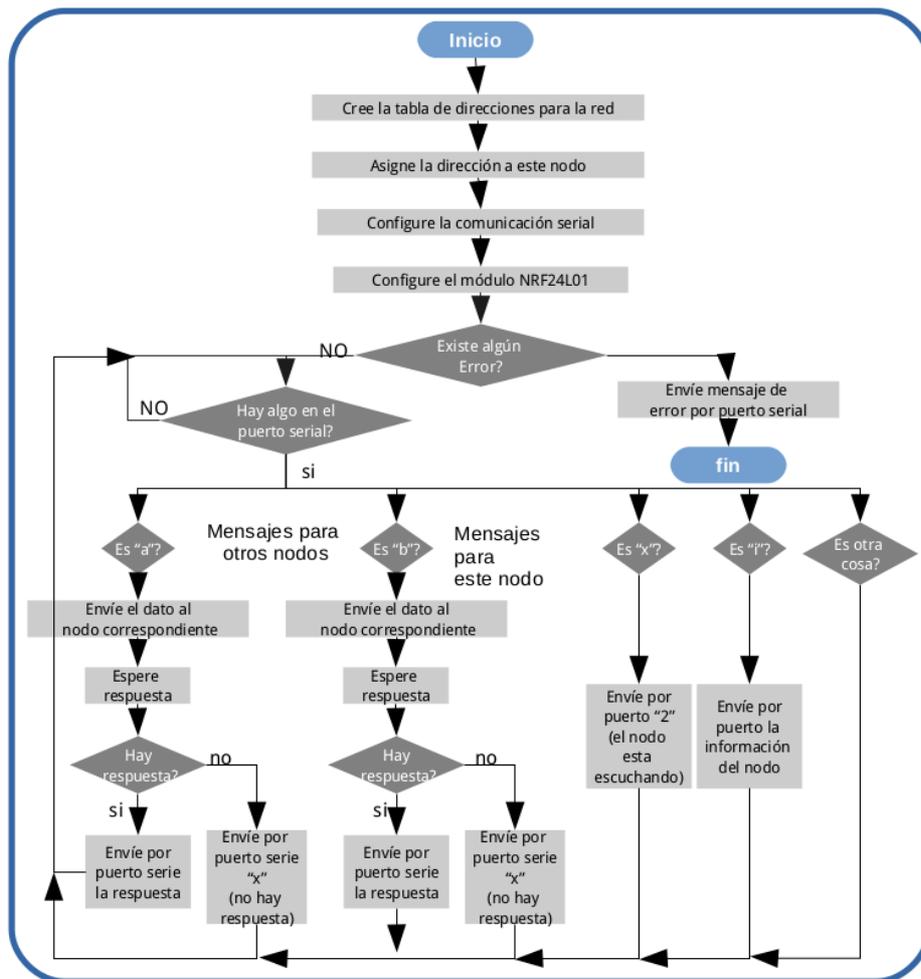
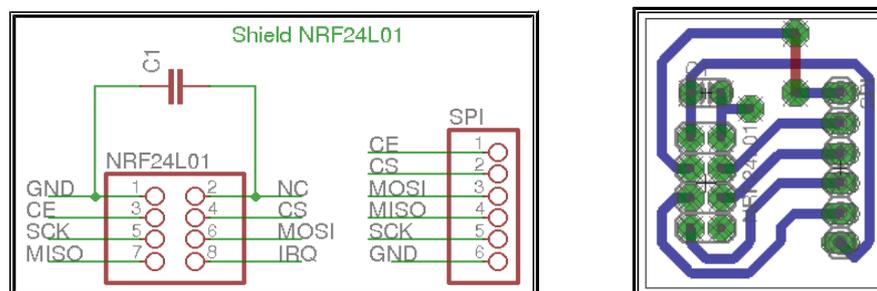


Figura 32. Diagrama de flujo nodos intermedios.

Fuente: Propia

En la figura 32 se observa el diagrama de flujo correspondiente a los programas que se alojan en los nodos intermedios.

- *Diseño de shield NRF24L01*



**Figura 33.** Shield NRF24L01.

**Fuente:** Propia

La conexión de los módulos NRF24L01 requieren del diseño y construcción de una placa, que permita conectar el módulo al puerto SPI de la placa arduino uno. En la figura 33 se observa el diseño de esta placa y los pines correspondientes, cabe destacar que los pines del bus SPI de la placa arduino uno son los siguientes:

GND	PIN 13
SCK	PIN 12
MISO	PIN 11
MOSI	PIN 10
CS	PIN 9
CE	PIN 8

- *Diseño de fuente de alimentación*

La fuente de alimentación de este dispositivo necesita suministrar un voltaje de 5V y una corriente de 120mA, su tamaño debe ser adecuado para que el nodo pueda ser movido en el domicilio en caso de requerir mejorar la cobertura. El planteamiento de la fuente de alimentación es el mismo que en la fuente para el nodo principal, es decir el diseño es el mismo; pero con la diferencia que para este dispositivo se requiere menos corriente, este

parámetro en el diseño modifica el tamaño del transformador pudiendo reducir el tamaño de la fuente. El transformador podría tan solo ser de 12V / 150mA, pero lamentablemente este valor no es comercial y el valor mas cercano de transformador comercial es 12V / 400mA. Ahora bien la comparación de los tamaños entre el transformador que se pretendió usar en el nodo principal de 1A y el que se pretende usar aquí se reduce a la mitad; pero a un así el tamaño de la fuente superaría al del nodo, además de que se siguen presentando las mismas desventajas de este tipo de fuentes, por lo tanto se opta por fuentes conmutadas para los nodos intermedios.

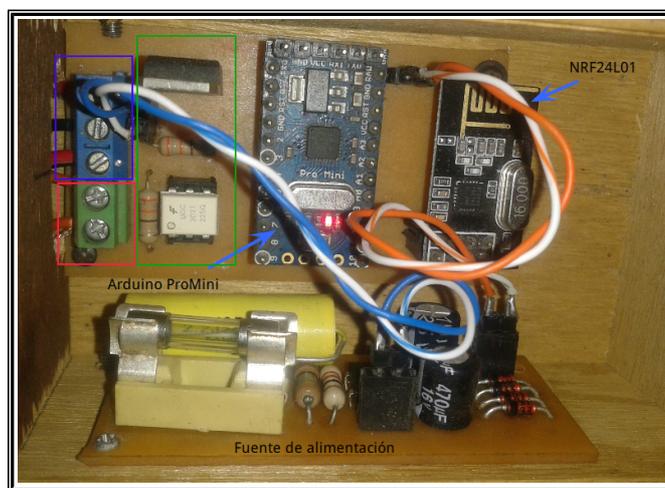


**Figura 34.** Fuente nodos intermedios.

**Fuente:** Propia

La fuente en cuestión mostrada en la figura 34, brinda 5V y una corriente máxima de 800mA, contando con una protección ante cortocircuito. Los nodos intermedios a pleno funcionamiento consumen una corriente máxima de 120mA, por lo que la fuente entrega a la placa una potencia de 0,6W.

### 4.1.6.3. Nodo lámpara.

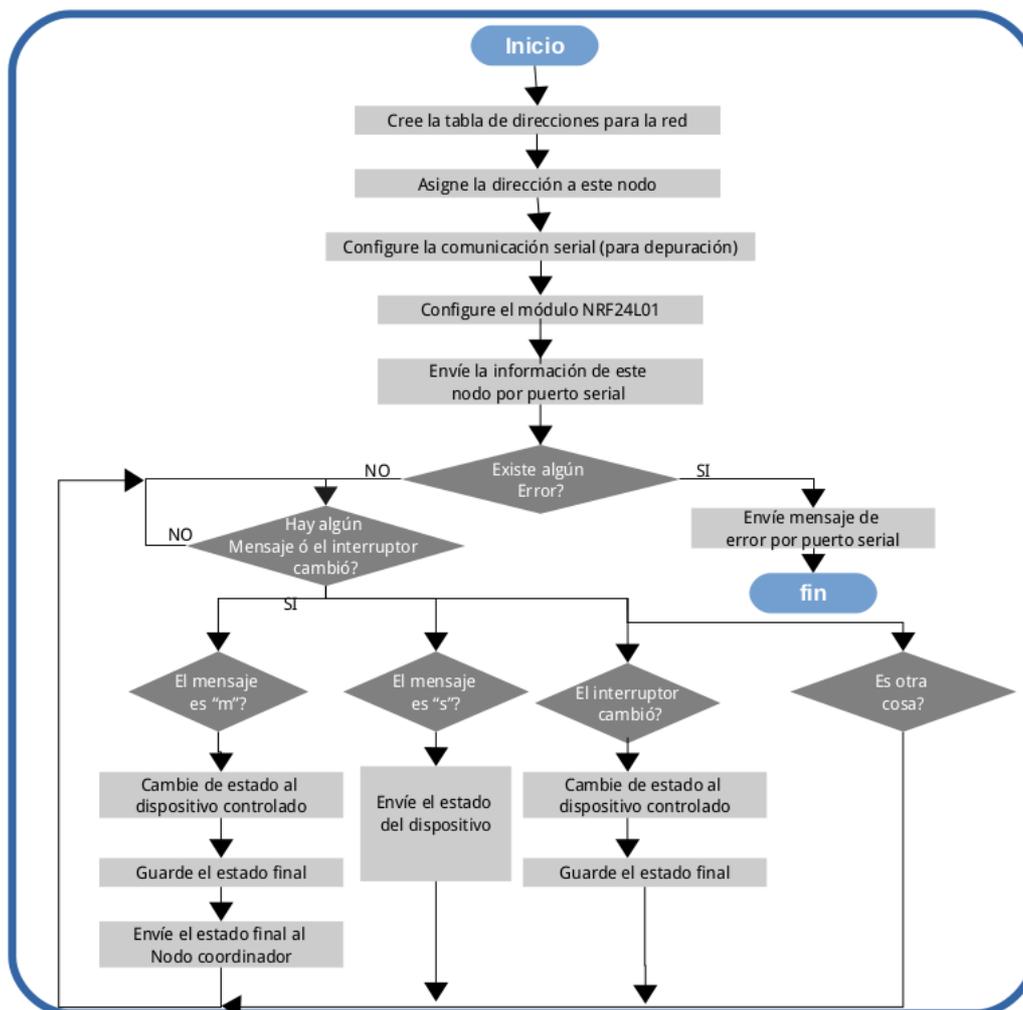


**Figura 35.** Partes de nodo lámpara.

**Fuente:** Propia

La figura 35 muestra el dispositivo que controla la lámpara del sistema, de izquierda a derecha se encuentra a un módulo NRF24L01 encargado de las comunicaciones, una placa arduino pro-mini que contiene el programa principal (observe el Anexo 9), este se encarga de administrar las variables del entorno, tanto del interruptor como las que llegan inalámbricamente. Enmarcado en verde se observa la etapa de potencia formada por un triac y un opto-acoplador, estos elementos permiten aislar la etapa de control (5V) de la etapa de potencia (120V). Finalmente a la izquierda enmarcado en rojo esta el conector donde va el interruptor convencional y enmarcado en violeta están los contactos donde se conecta la red eléctrica (120V contactos superiores), mientras que la lámpara a controlarse esta conectada en el contacto central e inferior.

El diagrama electrónico y su funcionamiento puede ser revisado en el Anexo 10. Mientras que el diagrama de flujo correspondiente se muestra a continuación.



**Figura 36.** Diagrama de flujo nodo lámpara.

**Fuente:** Propia

Para interactuar con este nodo se tienen dos formas, mediante el interruptor convencional y por medio del sistema. Se puede decir que el programa contenido en la placa arduino siempre esta pendiente de los cambios del interruptor y los caracteres que llegan a través del NRF24L01, los estados de la lámpara pueden ser encendido y apagado y el estado de esta puede ser consultado en cualquier momento desde el sistema. Cuando se realiza un cambio manual desde el interruptor, el estado final de la lámpara se almacena en el nodo para poder ser consultado desde el sistema posteriormente.

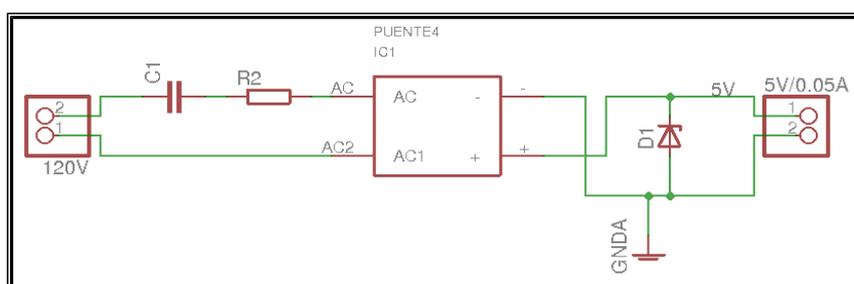
- *Diseño electrónico*

Como se observa en la figura 35 este nodo se basa en un arduino micro-pro acompañado de un módulo NRF24L01, una fuente de alimentación y una etapa de potencia, partes que se diseñan y describen a continuación.

- *Fuente de alimentación*

El nodo lámpara necesita 5V y 50mA para funcionar correctamente, en los casos anteriores la corriente demandada por los nodos estaba sobre los 100mA por lo que no se había considerado una fuente de alimentación reactiva<sup>43</sup> utilizando condensadores.

El Anexo 34 de este documento contiene la información de donde se obtuvieron las ecuaciones para llegar a plantear el siguiente diseño, que cumple con las necesidades del nodo en cuestión.



**Figura 37.** Fuente reactiva.

**Fuente:** Propia

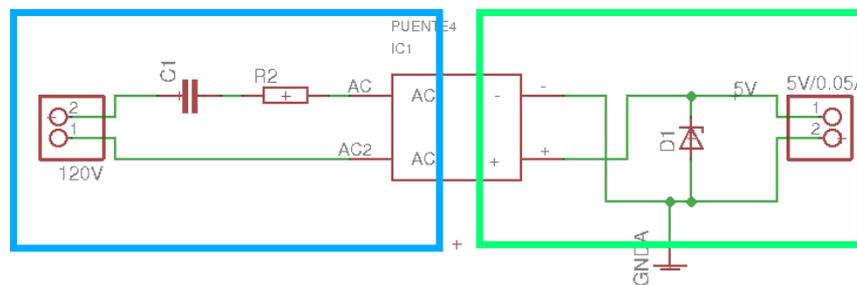
Este tipo de fuentes aprovecha la característica reactiva de los condensadores que se opone al paso de la corriente alterna, atenuando así la tensión sin la necesidad de un transformador, la ventaja de utilizar esta fuente es que es fácil de diseñar y construir con elementos

43 Fuente de alimentación reactiva.- es un tipo de fuente de alimentación sin transformador que utiliza la oposición al paso de la corriente alterna que presentan los condensadores.

electrónicos comunes y resulta menos costosa que las fuentes con transformador y las conmutadas.

El primer inconveniente de esta fuente es que por su naturaleza se deben aislar en un gabinete, ya que están conectadas directamente a la red eléctrica y su manipulación puede causar descargas eléctricas. El segundo inconveniente es que siempre necesitan que la carga alimentada este conectada, si la carga se desconecta la corriente tiende a fluir por el elemento regulador que en este caso es un diodo zener. Estas dos desventajas no son un inconveniente para este caso ya que se pretende instalar el nodo junto con la fuente en el techo, cerca a la lámpara en un gabinete.

El circuito en cuestión esta dividido en dos partes como se muestra en la figura 38



**Figura 38.** Partes de una fuente reactiva.

**Fuente:** Propia

En la parte izquierda enmarcado en color azul, se encuentra el condensador que reducirá la tensión seguido de una resistencia que limita la corriente que ingresa al circuito, mientras que en la parte derecha, enmarcado en verde se encuentra un diodo zener que se encargará de establecer la tensión en 5V.

A continuación lo cálculos y consideraciones correspondientes:

- La intensidad de entrada siempre es mayor o igual a la corriente de salida, la ecuación que describe esto es:

$$1. \quad I_{in} = \frac{V_{HFRMS}}{RT} \geq I_{out} \quad \text{donde:} \quad \mathbf{I_{in}}: \text{ intensidad de entradas}$$

$\mathbf{I_{out}}$ : intensidad de salida

$\mathbf{V_{HRMS}}$ : es el voltaje RMS de una media onda de una onda seno de CA.

$$2. \quad RT = X_C + R$$

$\mathbf{RT}$ : es la suma de la reactancia con la resistencia limitadora que tiene un valor muy pequeño por lo que se puede decir que  $RT \cong X_C$  donde  $X_C$  es la reactancia capacitiva.

$$3. \quad X_C = \frac{1}{2\pi fC} \quad \text{donde:} \quad \mathbf{f}$$
: es la frecuencia de la red eléctrica 60Hz.

$\mathbf{C}$ : es el valor del condensador en faradios.

El término  $V_{HFRMS}$  es igual a:

$$4. \quad V_{HFRMS} = \frac{\sqrt{2}VRMS - V_Z}{2} \quad \text{donde:} \quad \mathbf{VRMS}$$
: es el voltaje de la red eléctrica

$\mathbf{V_Z}$ : es el voltaje que cae en el diodo zener

Con estas ecuaciones definidas procedemos a calcular los valores correspondientes con los siguientes datos.

$V_Z = 5.1V$  : este valor se lo obtiene por que se utiliza un diodo zener 1N4733A de 5,1V y con una potencia máxima de disipación de 1W.

$VRMS = 120V$  de la red eléctrica.

$f = 60\text{Hz}$  que es la frecuencia de la red eléctrica.

$I_{out} = 50\text{mA}$  este dato es la intensidad que se desea en la salida para la carga.

De la ecuación número 4 se obtiene:

$$V_{HFRMS} = \frac{\sqrt{2}(120) - 5,1}{2} = 82,3\text{ V}$$

Con este dato se utiliza la ecuación 1, reemplazando datos:

$$I_{out} \cong \frac{V_{HFRMS}}{RT} \quad \text{y por que } RT \cong X_C \quad \text{se llega a:}$$

$$I_{out} \cong \frac{V_{HFRMS}}{X_C} \quad \text{entonces } 0,05\text{ A} \cong \frac{80,3\text{ V}}{X_C} \quad \text{y por consiguiente}$$

$$X_C \cong \frac{80,3\text{ V}}{0,05\text{ A}} = 1606\ \Omega$$

Con este valor ya se puede calcular el valor del condensador que se debe utilizar para obtener los 50mA deseados, entonces usando la ecuación 3 :

$$X_C = \frac{1}{2\pi fC} \quad \text{reemplazando y despejando}$$

$$C = \frac{1}{2\pi(60)(1606)} = 1,65\ \mu\text{F} \quad \text{pero este valor no existe comercialmente así que se aproxima}$$

hacia el valor comercial superior próximo que es 2,2 $\mu\text{F}$ . Con este nuevo valor se calcula la corriente de salida que suministrará la fuente.

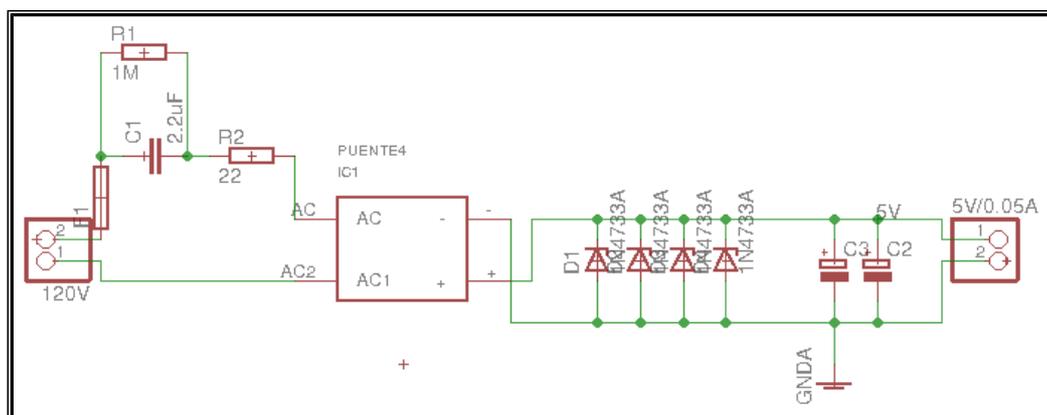
$$I_{out} \cong \frac{V_{HFRMS}}{X_C} \quad \text{y } X_C = \frac{1}{2\pi fC} = \frac{1}{2\pi(60)(2,2 * 10^{-6})} = 1205,7\ \Omega$$

$$I_{out} \cong \frac{80,3\text{ V}}{1205,7\ \Omega} = 0,0665\text{ A} = 66,5\text{ mA}$$

Como se observa este valor es suficiente para alimentar la carga del nodo foco.

Finalmente una consideración importante a tener en cuenta es que si la carga consume 50mA los 16.5mA restantes tienen que circular por el diodo zener, teniendo así que el diodo debe disipar  $0,084W$  ( $5,1V \cdot 0,016mA$ ) valor que no supera el 1W de potencia para el que está diseñado. Otro panorama importante a considerar es que cuando la carga se desconecta toda la corriente, los 66,5mA deben circular por el diodo, este panorama es poco probable pero en caso de darse el diodo debe poder soportar esta corriente. Así que se calcula la potencia que se disiparía sin carga, siendo  $0,33W$  ( $5,1V \cdot 0,0665A$ ) que no supera los parámetros de diseño del diodo.

Al final luego de todos los cálculos correspondientes se llega al siguiente circuito.



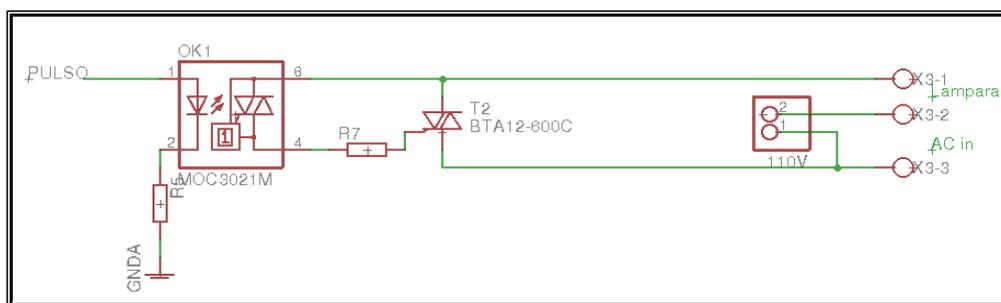
**Figura 39.** Partes de una fuente reactiva.

**Fuente:** Propia

Como se observa en la figura 39 se han aumentado algunos elementos para mejorar la robustez de esta fuente, en la parte izquierda se puede ver un fusible que protege al circuito en caso de una sobre corriente o cortocircuito, continuando se ha ubicado una resistencia de 1M ohmio  $\frac{1}{4}$  de watio en paralelo con el condensador de 2,2μF con el objetivo de descargar el

condensador cuando la fuente se desconecte, seguido esta una resistencia de 22 ohmios  $\frac{1}{4}$  de watio que limita la corriente de entrada, los valores de potencia de las resistencias se eligieron a partir de que no circula una corriente mayor a 66,5mA, en la parte derecha se observa que se han puesto cuatro diodos zeners, esto se lo hace con el fin de aumentar la vida útil de los diodos al tener que soportar aun menos corriente cada uno y en la parte derecha se han aumentado dos condensadores de 470uF para filtrar posibles impurezas.

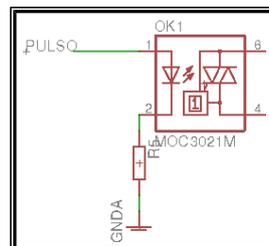
- *Circuito opto-acoplado*



**Figura 40.** Circuito opto-acoplado.

**Fuente:** Propia

El nodo lámpara tiene que conmutar una carga de 120V, para lo cual se utiliza un circuito que aísla la parte lógica del circuito de la parte de potencia. Para lograr esto se hace uso de un opto-acoplador MOC3021 y un triac BTA12-600C. El funcionamiento del circuito es sencillo pero eficaz, en la parte izquierda de la figura 40 se observa que el pin 1 del opto-acoplador es el que recibe el pulso de control que viene desde el arduino pro-mini, este pulso es de 5v y su corriente debe siempre ser menor a 40mA, para limitar la corriente se calcula la resistencia que limitaría la corriente.



**Figura 41.** Resistencia de entrada en opto-acoplador.

**Fuente:** Propia

De la hoja de datos del opto-acoplador MOC3021 se sacan los siguientes datos:

El voltaje máximo del led es igual a 3V para aumentar la vida útil del led se reduce este valor a 2V. La intensidad máxima del led es igual a 15mA, por el mismo motivo anterior se reduce este valor a 10mA. Con estos valores se calcula el voltaje que cae en la resistencia:

$V_t = V_{led} + V_r$  entonces  $V_r = 5V - 2V = 3V$  con este dato se calcula la resistencia con:

$$R = \frac{V_r}{I} \text{ donde } I = 10\text{mA}$$

$$R = \frac{3V}{0,01} = 300\Omega \text{ pero este valor no existe comercialmente, entonces se lo aproxima al}$$

valor superior que es 330 ohm. Con el cual la corriente que la placa arduino pro-mini debe

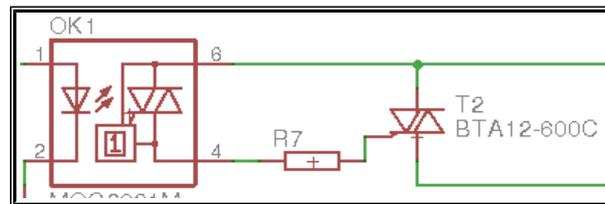
entregar  $I = \frac{3V}{330\Omega} = 9\text{mA}$ . Según la hoja de datos del fabricante los valores de corriente

para este dispositivo deben estar entre los 8mA a los 15mA por lo que este valor es correcto y

garantiza el correcto funcionamiento de la entrada del opto-acoplador. La potencia de esta

resistencia es  $(3V * 9\text{mA})$  igual a 0,027W por lo que se pone una de  $\frac{1}{4}$  de watio.

Para la conmutación de la carga se utiliza un triac BTA12-600C que según su hoja de datos permite alimentar cargas de hasta 12A a 600V. Entonces se tiene que, el voltaje en la compuerta debe ser 3V. Se calcula la resistencia de la compuerta del triac( $R_g$ ).



**Figura 42.** Resistencia de disparo en triac.

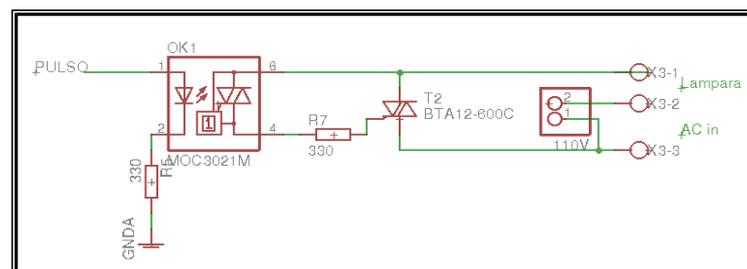
**Fuente:** Propia

$R_g = \frac{V_g}{I_f}$  Donde  $I_f$  es la corriente máxima de conducción del opto-acoplador y es

10mA (este valor se lo obtiene de la hoja de datos del opto-acoplador), mientras que  $V_g$  es el voltaje necesario en el Gate del triac.

$R_g = \frac{3V}{0,01A} = 300\Omega$ , Y una vez mas se aproxima este valor a 330 ohm y a  $\frac{1}{4}$  de watio.

Con todos estos datos el circuito opto-acoplado queda de la siguiente manera.



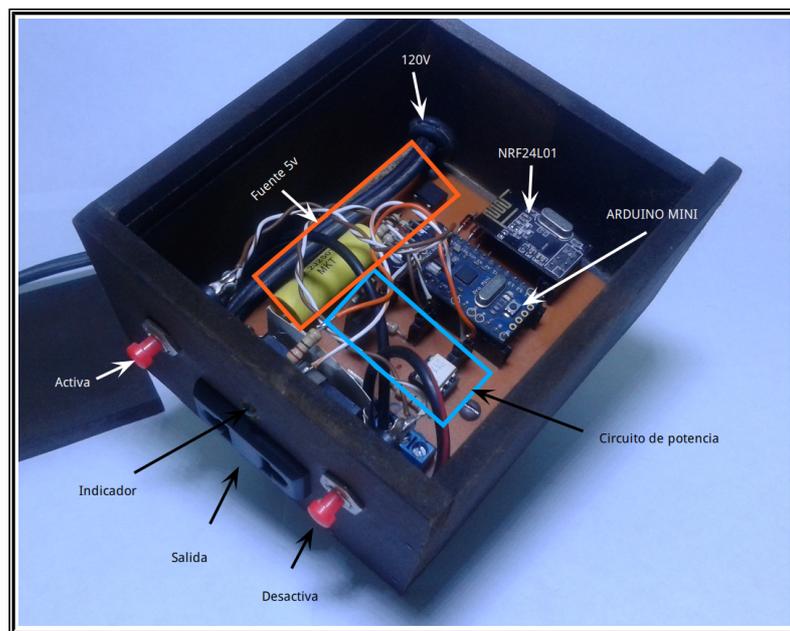
**Figura 43.** Resistencia de disparo en triac.

**Fuente:** Propia

La figura 43 muestra en su parte izquierda tres puntos de conexión para la entrada de 120V y la lámpara.

#### 4.1.6.4. *Nodo manta.*

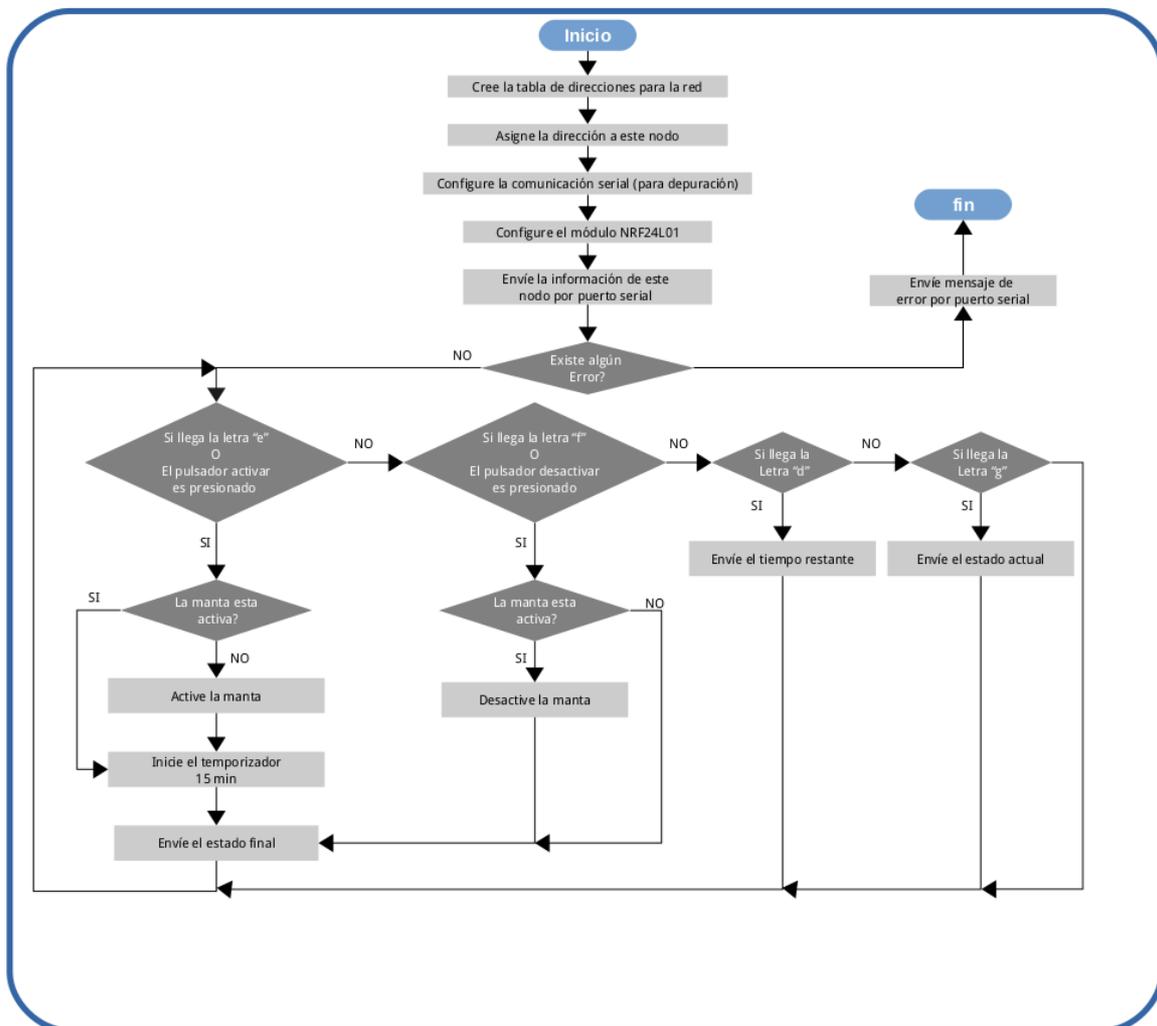
El nodo manta es un dispositivo diseñado para conectar y desconectar una carga eléctrica (para este caso una manta termoeléctrica), para realizar esto se ha implementado un nodo muy parecido en hardware al nodo lámpara, pero se ha tenido que implementar también un temporizador por software, que permite la desconexión de la carga después de 15 minutos, tiempo en el cual la manta termoeléctrica alcanza a poner la cama del usuario a una temperatura adecuada. El Anexo 11 y el Anexo 12 contienen el programa y el diagrama electrónico correspondiente.



**Figura 44.** Partes nodo lámpara.

**Fuente:** Propia

Como se puede observar en la figura 44 se aprecian las partes de este dispositivo, a diferencia del nodo lámpara este tiene incluido en el PCB su fuente de alimentación, el diagrama de flujo correspondiente a este nodo se observa a continuación.



**Figura 45.** Diagrama de flujo nodo manta.

**Fuente:** Propia

El flujograma de la figura 45 ilustra el funcionamiento de este dispositivo, se tienen dos pulsadores los cuales permiten una interacción manual para la conexión y desconexión de la manta térmica. Además paralelamente se puede interactuar mediante letras recibidas por el NRF24L01 desde el sistema teniendo estas condiciones:

- Si llega el carácter “e”.- La manta se activará y un temporizador de 15 minutos empezará una cuenta regresiva para su desconexión. Si la mata ya estaba activa el temporizador simplemente se reiniciará. Al final se envía el estado final.
- Si llega el carácter “f”.- La manta de se desconectará y se envía el estado final.
- Si llega el carácter “d”.- El nodo reporta el tiempo restante para que la manta se desconecte, si la manta esta desconectada se envía cero.
- Si llega el carácter “g”.- El nodo reporta el estado actual de la manta. Indica si esta conectada o desconectada.

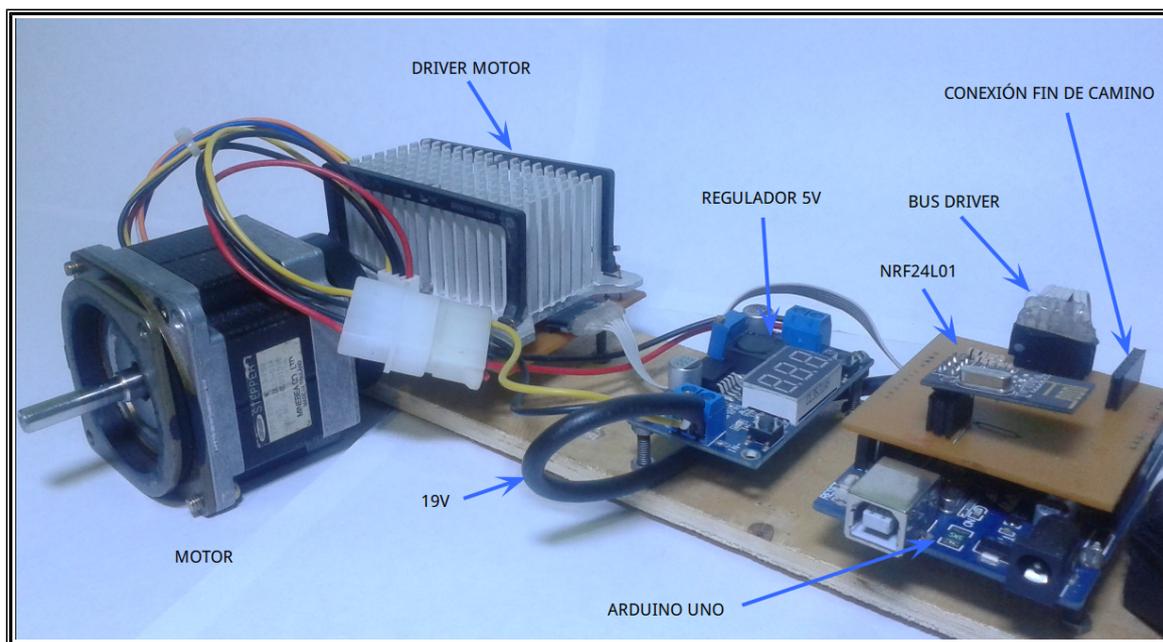
- ***Diseño electrónico***

Finalmente este nodo al ser muy parecido al nodo foco, requiere los mismos 5V y 50mA para su correcto funcionamiento, así que se ha implementado la misma fuente de alimentación diseñada para el nodo anterior, con la única diferencia que en este caso se la ha integrado en el mismo PCB. Como se menciona antes el circuito se lo puede observar en el Anexo 12.

#### **4.1.6.5. Nodo Puerta.**

El propósito de este nodo es controlar una puerta abatible interior, el control de la puerta no incluye el seguro de la misma, el dispositivo instalado cuenta con tres interruptores de fin de camino utilizados como sensores de presión y un motor de pasos. De esta manera la puerta puede estar abierta, cerrada, entre-abierta o asegurada.

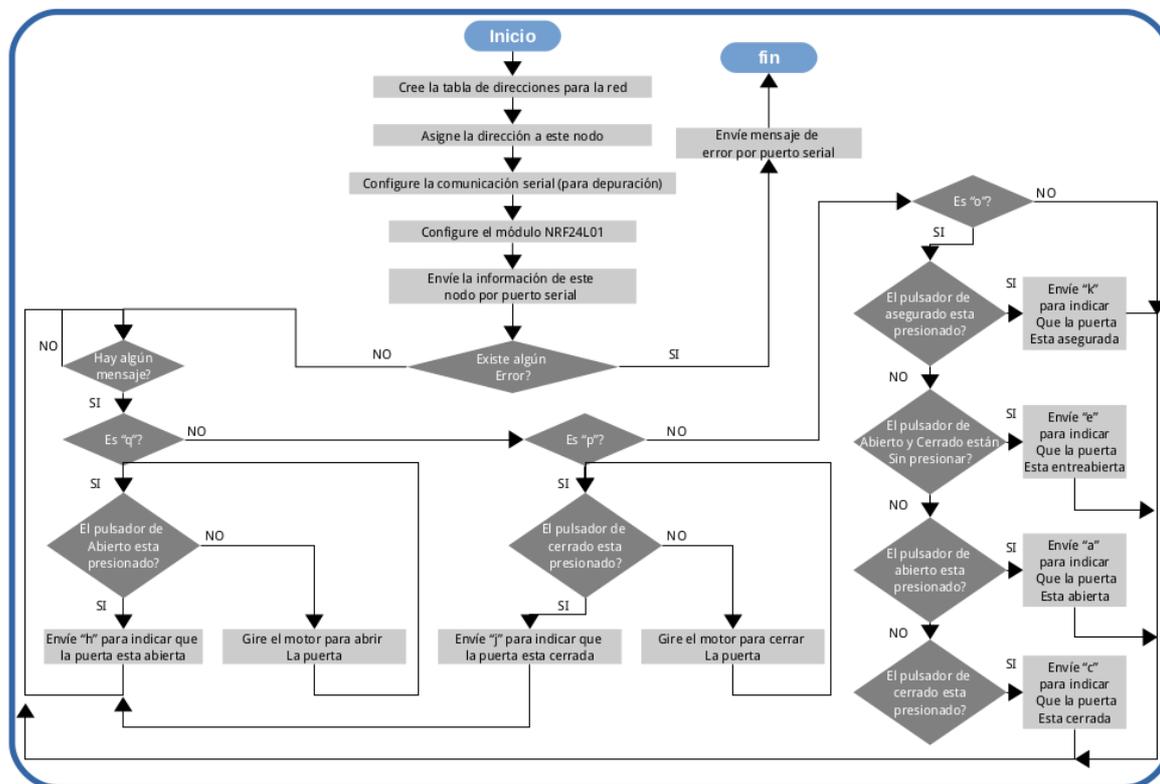
- Puerta abierta.- en este estado la puerta esta totalmente abierta, para esto es necesario que el interruptor de fin de camino correspondiente este presionado. Este interruptor se ubica convenientemente entre la puerta y la pared que debe tocar la puerta para estar abierta.
- Puerta cerrada.- este estado indica una puerta cerrada pero no asegurada, osea sin el seguro puesto, para esto se utiliza un interruptor de fin de camino ubicado en el marco de la puerta, de manera tal que se presione justo antes de que la puerta se cierre totalmente y se asegure.
- Puerta entre-abierta.- Si los interruptores de cerrado y abierto están sin presionar esto quiere decir que la puerta esta abierta parcialmente, pero no se conoce su posición exacta.
- Puerta asegurada.- Este estado indica una puerta totalmente cerrada, es decir la puerta esta con el seguro puesto, esto imposibilita la interacción desde la interfaz siendo necesaria la intervención de una persona para quitar el seguro de la puerta.



**Figura 46.** Nodo puerta.

**Fuente:** Propia

En la figura 46 se observan las partes electrónicas de este nodo, de izquierda a derecha se tiene un motor de pasos que brinda la fuerza necesaria para mover la puerta, continuando tenemos un arreglo de cuatro mosfets que permiten gobernar las bobinas del motor con un voltaje de 19V y picos de corrientes hasta de 1,5A acompañados de un disipador de calor, a continuación se observa un módulo regulador de voltaje que a partir de 19V entrega 5V para la electrónica lógica del nodo. Finalmente en este nodo se tiene un arduino uno encargado de controlar y censar los dispositivos conectados, este arduino contiene el programa correspondiente y sobre el se observa una placa electrónica que permite la comunicación con el módulo NRF24I01, el control del motor a través de un bus de datos sencillo y el censado de los tres interruptores de fin de camino que determinan el estado de la puerta.



**Figura 47.** Diagrama de flujo nodo puerta.

**Fuente:** Propia

El diagrama de flujo que se observa en la figura 47 explica el funcionamiento de este nodo, esto corresponde con las condiciones expuestas anteriormente. Los caracteres utilizados para la interacción con el nodo son:

- q.- para abrir la puerta.
- p.- para cerrar la puerta
- o.- para conocer el estado de la puerta.

Las respuestas a estos caracteres son:

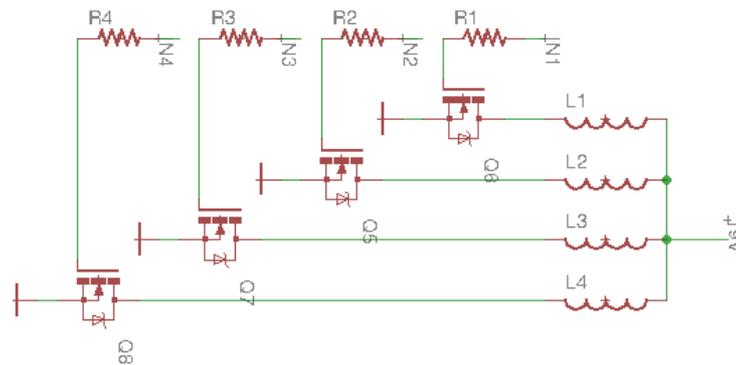
- h.- para indicar que la puerta se abrió.

- j.- para indicar que la puerta se cerró.
- a.- indica que la puerta esta abierta.
- c.- indica que la puerta esta cerrada.
- e.- indica que la puerta esta entre-abierta.
- k.- indica que la puerta esta cerrada y asegurada.

El diagrama electrónico de este nodo y su respectivo programa se pueden observar en los Anexos 15 y 16 respectivamente.

- ***Diseño electrónico***

Este nodo requiere el uso de un driver para gobernar las bobinas del motor NEMA17 de pasos, dicho motor para empezar el movimiento requiere de una cantidad considerable de corriente, de hasta de 3A. Si bien existen driver comerciales que soportan esta corriente no se los ha utilizado por que se considera que estarían trabajando al limite cuando el motor arranca. Para solventar esto se ha utilizado cuatro mostfets IRFZ44N, que pueden conmutar cargas de hasta 49A y voltajes de 55V, con lo cual se garantiza un correcto funcionamiento sin saturación de corriente en el driver.



**Figura 48.** Driver para motor.

**Fuente:** Propia

La figura 48 muestra el driver diseñado, de derecha a izquierda se observan las cuatro bobinas del motor NEMA17, este motor necesita de 12V a 24V, hasta 3A para arrancar y valores entre 2A hasta 2,5A para un trabajo a plena capacidad. Continuando se observan los cuatro mosfets que conmutan la alimentación en cada una de las bobinas, las puertas de los mosfets están gobernadas mediante cuatro salidas digitales de la placa arduino, hay que recordar que cada salida digital no puede entregar mas 40mA por lo que se hace necesario poner una resistencia en cada entrada para que limite la corriente. A pesar de que el mosfet es un componente que se gobierna por voltaje, se pone estas resistencias para mayor seguridad.

En la hoja de datos se observa que la corriente necesaria para que el mosfet empiece a conducir es de 100nA, por lo que se tendrían que poner resistencias de 50Mohm en cada entrada. Este valor se lo obtiene mediante:

$$R = \frac{V}{I_{in}} = \frac{5V}{100nA} = 50 M\Omega$$

- *Fuente de alimentación*

La fuente de alimentación para este nodo debe manejar dos voltajes, 5V para alimentar el circuito de control que en este caso es un arduino uno y otro bloque que brinde un voltaje entre 12V a 24V con una salida de corriente que este entre 3A a 4A. Como se observa el planteamiento de esta fuente demanda un diseño mas elaborado que las fuentes anteriores. Para poder llegar ha un diseño primero se toman algunas consideraciones.

- La fuente de alimentación junto con el circuito se deben poder empotrar en la puerta.
- Esta fuente debe ser pequeña y no evitar el fácil desplazamiento de la misma.
- La fuente debe ser eficiente, con el fin de no causar un impacto en el consumo eléctrico de la vivienda.
- Debe contar con algún tipo de protección contra sobrecarga o cortocircuitos.

Los requerimientos de esta fuente plantean un reto en el diseño de este nodo, se considera construir una fuente basada en transformador con un rectificador adecuado y dos reguladores de voltaje que brinden los niveles adecuados; pero este diseño no es viable por que no cumpliría con todos los requerimientos y presentaría inconvenientes por estas razones:

- La fuente resultaría tener dimensiones demasiado grandes para poder ser empotrada en la puerta, el principal problema sería el tamaño del transformador ya que al pretender entregar 3,5A seria muy grande.
- La alternativa al inciso anterior seria separar el transformador del circuito ó la fuente del circuito, para su posterior unión mediante cables. Esto requeriría de cables de

sección considerable que soporten los 3A máximos del sistema y esto conllevaría afectar de manera más intrusiva la estética de la vivienda, al tener que buscar el espacio para la fuente y además para tender los cables de conexión.

- El regulador de voltaje que debe entregar los 3A debe ser un regulador especial y debe contar con un disipador de tamaño considerable, añadiendo mas espacio necesitado.

Por estas tres importantes razones no se plantea un diseño de una fuente para este nodo, cabe aclarar que la fuente se puede construir pero no es viable por lo expuesto.

Se contempla la utilización de una fuente de computadora ATX pero por su tamaño tampoco es una opción aceptable, finalmente se decanta por la utilización de una fuente de poder de un pc portátil HP.

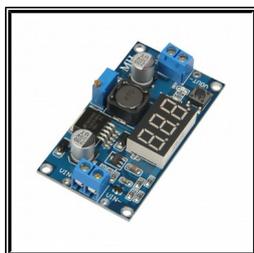


**Figura 49.** Cargador HP.

**Fuente:** Propia

La fuente de la figura 49 entrega 3.5A con un voltaje de 19V, cuenta con protección contra cortocircuitos, la potencia máxima que entrega esta fuente es de 66.5W. Esta fuente cumple con todos los requerimientos del nodo ya que los valores entregados están dentro de las necesidades energéticas del nodo.

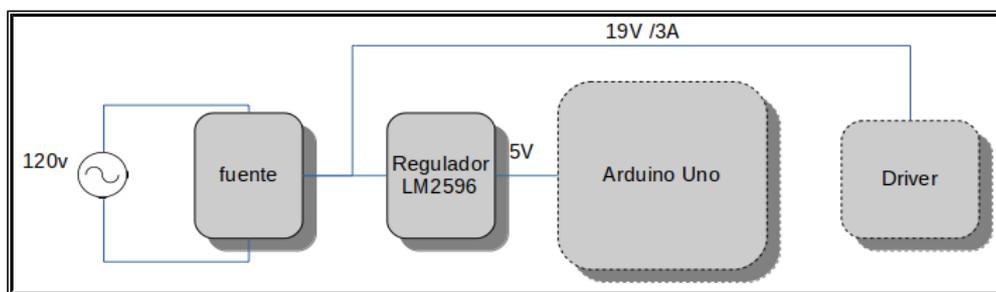
Para la regulación del voltaje a 5V se utiliza un módulo regulador no lineal con el integrado LM2596, que garantiza un voltaje y corriente estable para la placa arduino uno.



**Figura 50.** Módulo LM2596.

**Fuente:** Propia

El módulo en cuestión tiene una entrada que acepta voltajes entre 1.2V hasta 37V y permite entregar voltajes entre, 1V hasta 36V que pueden ser regulados mediante un potenciómetro. Este módulo puede entregar hasta 3A como máximo, pero este valor siempre depende de la corriente de entrada. Finalmente el diagrama de conexión ecléctico queda se muestra en la figura



**Figura 51.** Diagrama eléctrico nodo puerta.

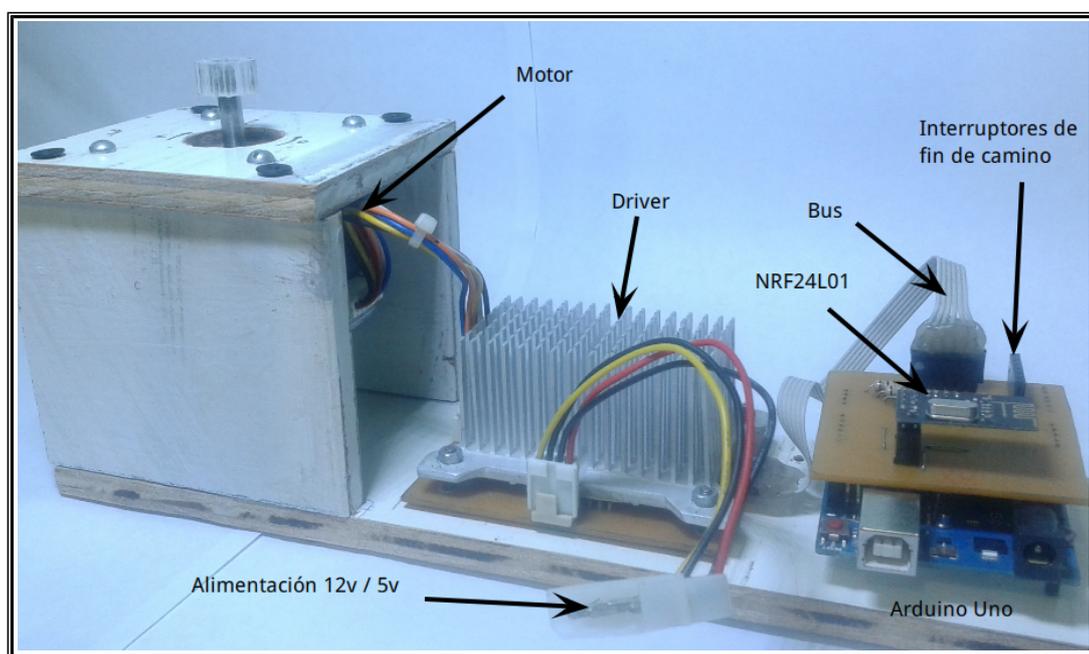
**Fuente:** Propia

#### **4.1.6.6. Nodo Ventana.**

El nodo ventana esta compuesto casi idénticamente al de la puerta, por su naturaleza similar de abrir o cerrar un acceso. Se ha duplicado el circuito utilizado en la puerta y se lo ha utilizado para interactuar con la ventana. Consecuentemente, la ventana puede tener los

mismos estados que la puerta, abierta, cerrada, entre-abierta y asegurada.

Otra ventaja utilizada es que gracias a que los dispositivos se ubican en distintas redes compuestas por los NRF24L01, se pueden utilizar los mismos caracteres que se utilizaron con la puerta para el control de la ventana. Esto gracias ha que los módulos trabajan en diferentes frecuencias.



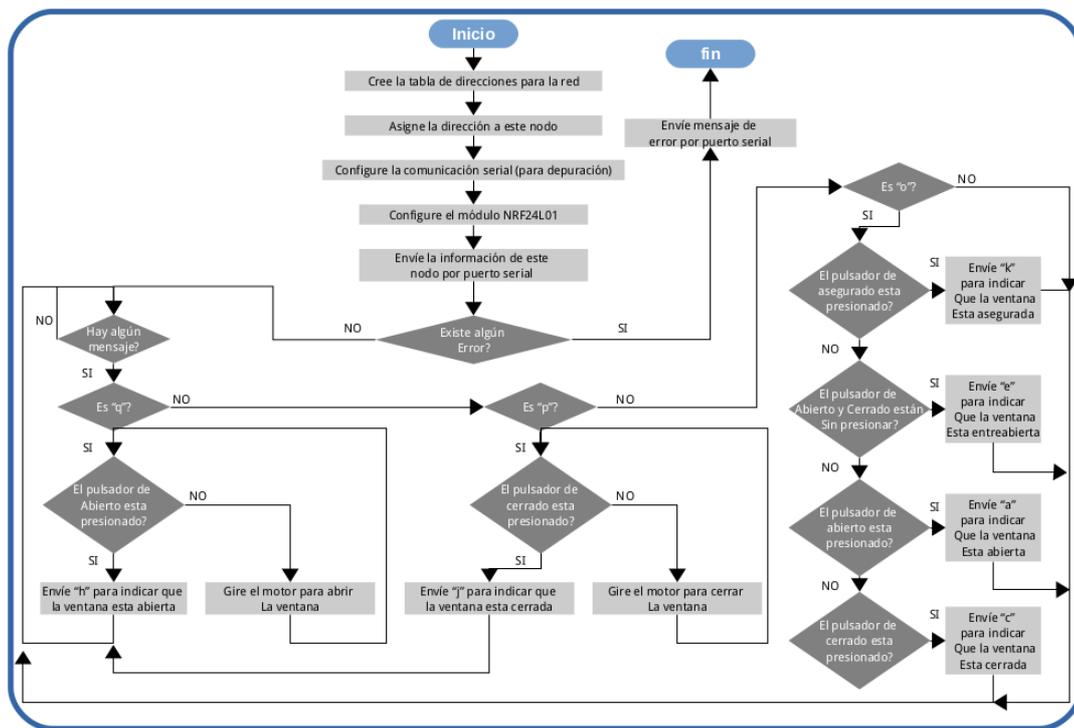
**Figura 52.** Partes nodo ventana.

**Fuente:** Propia

En este nodo la disposición de los interruptores de fin de camino se ubican convenientemente en el marco de la ventana, gracias a esto se puede censar el estado de la ventana en todo momento.

El diagrama de flujo correspondiente a este nodo se aprecia en la figura 53 muestra como funciona el programa cargado en este nodo, como se explico los caracteres para interactuar son los mismos que los de el nodo puerta y prácticamente no hay diferencias en los

programas, mas que las frecuencias de trabajo de los NRF24101. El diagrama electrónico de este nodo puede ser consultado en el Anexo 15, mientras que el programa cargado se lo puede encontrar en el Anexo 17.



**Figura 53.** Diagrama de flujo nodo ventana.

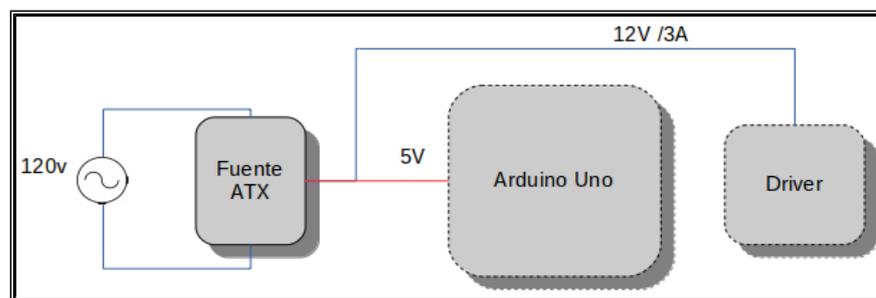
**Fuente:** Propia

- ***Diseño electrónico***

La pequeña diferencia electrónica entre el nodo puerta y ventana es su fuente de alimentación, como se observó en el nodo anterior existe una sola entrada de voltaje de 19V y es necesario un regulador de voltaje para obtener 5V para la etapa de control.

Al ser esta la única diferencia los requerimientos de voltajes y corrientes son los mismos, es decir se necesitan 5V para la placa arduino uno y 3A para el motor NEMA17.

Las consideraciones para este nodo son similares al anterior, pero aquí no es necesario que la fuente de alimentación y el circuito estén empotrados en la ventana, por lo que no se moverá y estarán fijos en la base de la ventana. Esta diferencia hace que se pueda utilizar una fuente de pc ATX y se aprovecha que ya brinda los voltajes necesarios para el funcionamiento del nodo. Por esta razón se utiliza una fuente ATX que entrega 12V con una corriente máxima de 18A y los 5V que necesita la etapa de control. Quedando el diagrama eléctrico de la siguiente manera.

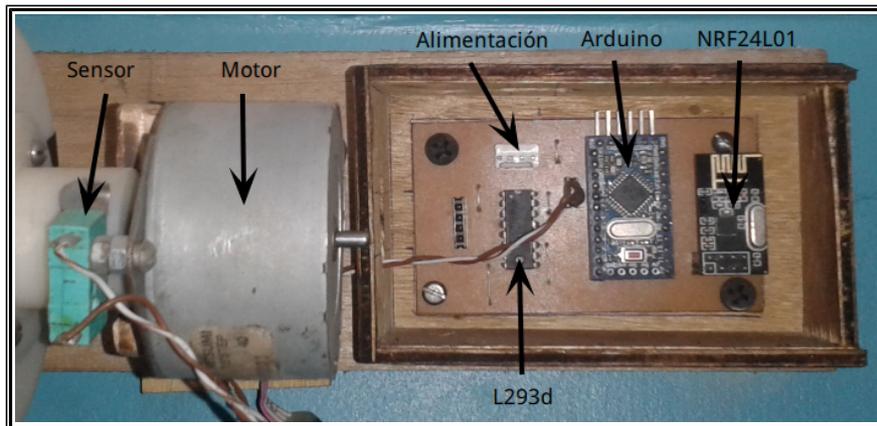


**Figura 54.** Diagrama de flujo nodo ventana.

**Fuente:** Propia

#### **4.1.6.7. Nodo Cortina.**

Este nodo permite controlar una cortina enrollable, para esto se utiliza un motor de pasos capaz de mover el eje en donde se enrolla la cortina. De manera similar que en otros nodos en la parte lógica y de comunicación se utilizan un arduino pro-mini en conjunto con un módulo NRF24L01. El diagrama electrónico y programa correspondiente a este nodo se encuentran en los Anexos 18 y 19 respectivamente.



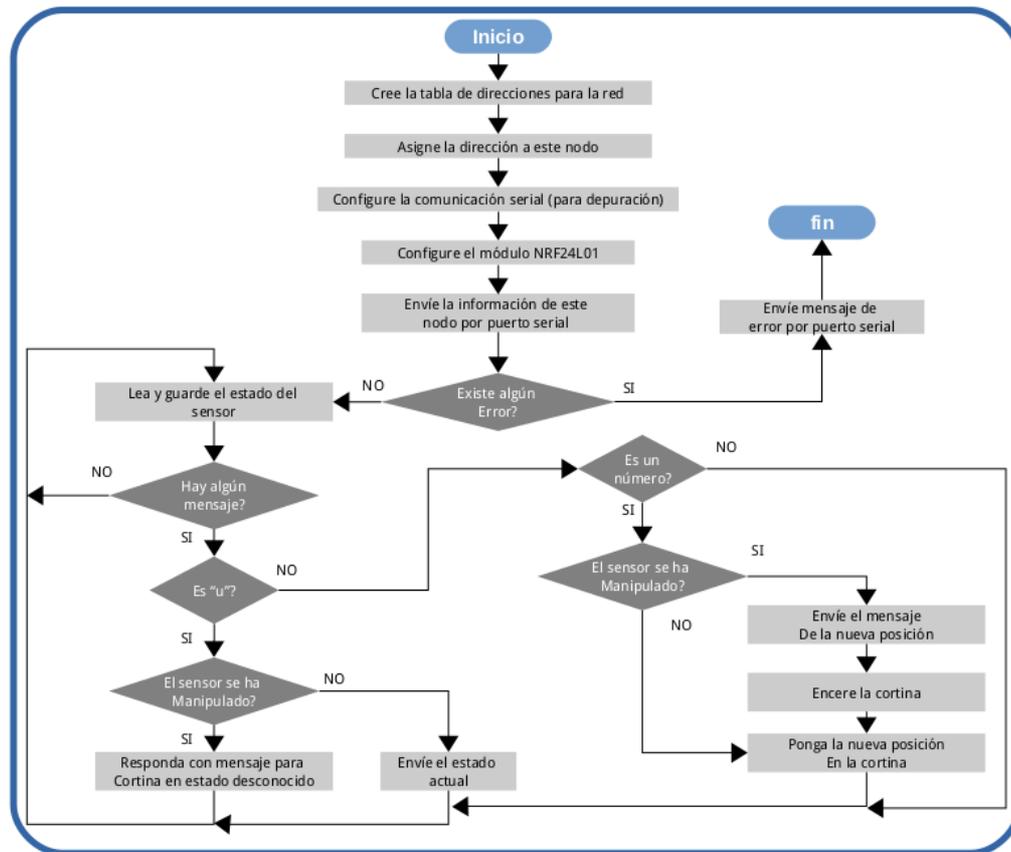
**Figura 55.** Nodo cortina.

**Fuente:** Propia

En la figura 55 se observan los elementos que conforman el nodo cortina, de izquierda a derecha se observa un sensor que permite contar la revoluciones que da el motor y también sirve como un indicador de movimiento manual, cuando el usuario envía una orden desde la interfaz, el nodo actualiza la posición de la cortina basado en el número de revoluciones que da el motor, cuando usuario mueve la cortina manualmente el sensor cambia de estado y este cambio se almacena en el nodo.

El siguiente elemento es un motor de pasos acoplado directamente al eje que enrolla la cortina y a una polea desde la cual se pueden hacer cambios manuales. Este motor trabaja conjuntamente con dos integrados L293D que son capaces de soportar 600 mA en condiciones normales y picos de corrientes hasta de 1 A, cada uno.

En la parte derecha se observan un arduino pro- mini y un NRF24L01 que se encargan del control y comunicación del nodo.



**Figura 56.** Diagrama de flujo nodo cortina.

**Fuente:** Propia

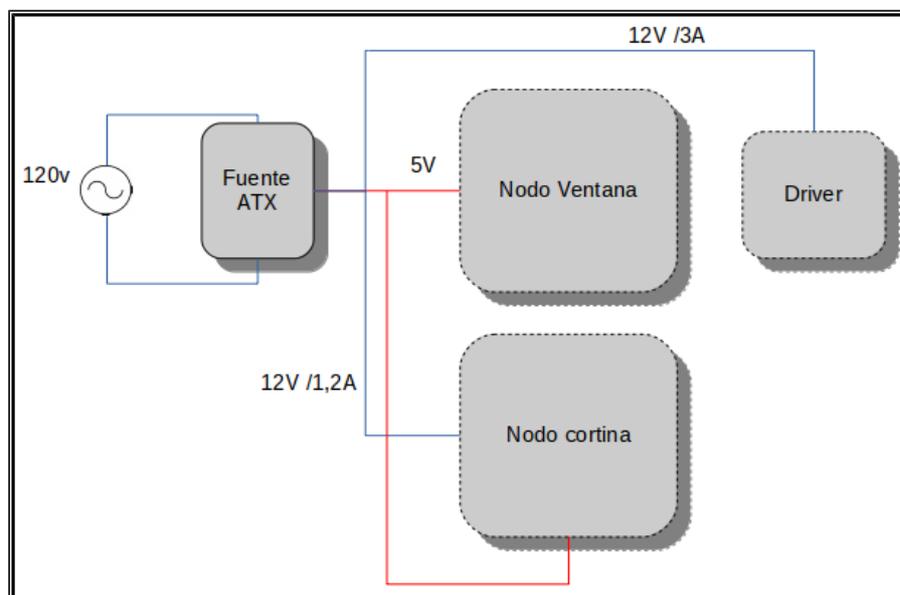
El funcionamiento de este nodo corresponde con el diagrama de flujo expuesto en la figura 56. Para interactuar con este nodo se espera recibir la letra “u” con la cual el nodo responderá con el estado actual de la cortina.

Y si en cambio llega un número entre X y Y este número representará la nueva posición de la cortina. En caso de que la cortina se mueva manualmente, el nodo cuenta con un sensor que permite detectar la interacción manual, entonces el usuario puede mover manualmente la cortina libremente y si el usuario luego necesita mover la cortina desde la interfaz, esta vuelve a su posición inicial y termina moviéndose a la posición indicada.

- **Diseño electrónico.**

La electrónica de este nodo consiste en una fuente de alimentación que suministra dos voltajes, el primero de cinco voltios para la parte lógica del circuito y el segundo de doce voltios para el motor utilizado. Los niveles de corriente suministrados hacia el motor deben ser los adecuados ya que el motor utilizado demanda 1,5A para arrancar y 900 mA para su trabajo normal, por esto se vio la necesidad de sobreponer dos integrados L293D doblando así su capacidad, soportando picos de corrientes de hasta 2,4A y brindando 1,2A en condiciones normales de trabajo lo cual es suficiente para el propósito del nodo.

Se ha visto conveniente hacer uso de la misma fuente ATX que se utiliza en el nodo ventana, ya que la instalación de la cortina se pretende hacerla en le mismo lugar de la ventana, es decir la cortina pertenece a la misma ventana donde se instala en nodo anterior. El diagrama eléctrico queda de la siguiente manera.



**Figura 57.** Diagrama eléctrico nodo cortina.

**Fuente:** Propia

#### **4.1.7. La interfaz web**

Como se había revisado en el primer capítulo de este documento, se utilizan herramientas orientadas a la web para crear la interfaz que permite interactuar con los diferentes nodos descritos anteriormente.

Esta interfaz esta basada en un marco de trabajo que consiste en:

- Un servidor web apache.
- El lenguaje de programación PHP.
- JQuery Mobile

Estas tres herramientas interactúan entre si para dar como resultado una interfaz amigable e intuitiva para el usuario con un diseño responsivo que se adapta a cualquier tamaño de pantalla.

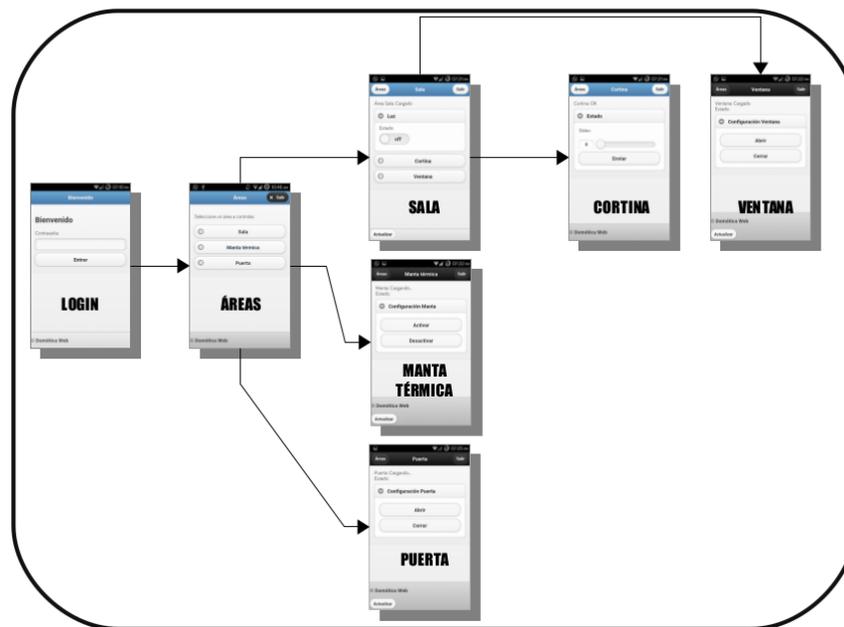
El funcionamiento de la interfaz web en realidad es simple y esta basado en archivos que contienen lenguaje html, javascript y php conjuntamente. Cada nueva interfaz o nueva pagina que se muestra al usuario es un archivo alojado en el servidor. Estos archivos están relacionados unos a otros de manera tal que cuando el usuario interactúa con la interfaz estos se van desplegando en la pantalla creando una interfaz interactiva. Claro esta que la relación que estos tiene entre si corresponde con los nodos a controlar.

##### **4.1.7.1. Partes y funcionamiento.**

La interfaz cuenta con siete pantallas o ventanas, la figura 58 muestra como se relacionan. Cada una de las ventanas cuenta con controles que permiten interactuar con los nodos,

permitiendo conocer su estado y cambiándolo de acuerdo a lo que el usuario desee.

Se a tratado de agrupar los dispositivos por áreas, con el objetivo de obtener una interfaz lo mas ordenada posible.



**Figura 58.** Diagrama interfaz web.

**Fuente:** Propia

A continuación se explica una a una las ventanas las funciones que cumple en la interfaz.

#### **4.1.7.2. El login o pantalla de ingreso.**

Es la primera pagina que se muestra al apuntar en el navegador a la dirección de la interfaz.



**Figura 59.** Pagina login o pantalla de ingreso.

**Fuente:** Propia

Consiste en un pequeño formulario en donde el usuario tiene que ingresar una clave, si la clave no es correcta la pagina devuelve un mensaje de error y se recarga. Los caracteres ingresados en la interfaz son ocultados visualmente y para su envío al servidor se encriptan. De esta manera se previene el ingreso a personas no autorizadas, además se debe recordar que la única forma que el usuario pueda acceder a esta interfaz es mediante la conexión por una VPN o por la conexión directa a la LAN donde se encuentra el servidor. Con estas dos características combinadas se obtiene un nivel de seguridad aceptable para esta aplicación.

El código fuente de esta página se lo puede encontrar en el Anexo 20 y básicamente consiste crear una sesión a nivel de aplicación con PHP, es decir si el usuario ingresa una clave válida se crea una sesión que vincula al dispositivo desde el que se esta ingresando y PHP en el servidor. Esta sesión es necesaria para que el usuario pueda continuar utilizando la

interfaz y siempre se verificará que el usuario haya iniciado sesión antes de cargar una nueva página. Con este método se evita que otro usuario pueda apuntar simplemente a la URL de la página en cuestión y pueda interactuar con la interfaz, de modo que si no inició sesión, el sistema lo regresa automáticamente a la página de login.

El código encargado de comprobar los credenciales y crear la sesión es:

```
<?php
session_start(); //crea la cookie... sesión
if(isset($_POST['login']))
{
$completa = $_POST['pass'];
$comparar = md5 ($completa); //codifica lo escrito en md5
$clave = '****'; //contraseña
$final = md5 ( $clave ); //la codifica md5
if ( $comparar == $final ) { //Condición para comprobar la contraseña
$_SESSION['listo'] = true;
header('Location: lugares.php'); //redirige a la pagina después de iniciar sesión
exit;
} else
{
?>
<script type="text/javascript">
<!--
alert('Password Incorrecto') //Mensaje en caso de contraseña incorrecta
/-->
</script>
<?php
}
}
?>
```

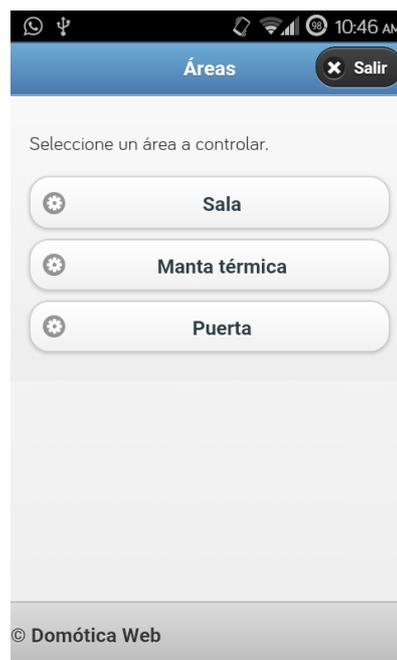
Si se observa detenidamente el código se puede o ver que se recupera el contenido de una variable llamada *pass*, que contiene la clave ingresada y se la comprueba con la almacenada en el servidor. Si esta clave es correcta se procede a crear la sesión y direccionar a una página llamada *lugares*, caso contrario se envía un mensaje por medio de javascript alertando que la clave no es válida. En la parte de código mismo de la página escrito con jquery mobile la parte mas destacable es:

```
</div>
<!--contenido-->
<div data-role="content" data-theme="c">
<h2>Bienvenido</h2>
<form action="" method="post" data-ajax="false">
<label for="contra">Contraseña:</label>
<input type="password" name="pass">
<input type="submit" value="Entrar" name="login">
</form>
</div>
```

Aquí se observa el contenido de la página en si, como se puede observar los nombres *pass* y *login* corresponden con una caja de texto y un botón que permiten enviar los datos al servidor para ser comprobados.

#### 4.1.7.3. Página áreas.

El objetivo de esta página es agrupar los elementos que se desean controlar en conjuntos que contienen a los mismos, se pretende agruparlos de acuerdo a donde están ubicados en la casa.



**Figura 60.** Pagina áreas.

**Fuente:** Propia

En la figura 60 se observa el menú que contiene, el área *Sala* y los elementos *Manta térmica* y *Puerta* a los que el usuario puede acceder. Algunas de las partes más importantes del código de esta página son:

```
<?php session_start(); require 'cookie.php';?>
```

Esta línea de código es la que comprueba si el usuario ha iniciado sesión. El contenido del archivo *cookie.php* es:

```
<?php
session_start();
if (!isset($_SESSION['listo']))
    || $_SESSION['listo'] !== true) {
    header('Location:loginjqr.php'); //Redirige al inicio de sesión en caso de que haya no iniciado
    sesión
    exit;
}
?>
```

De esta manera al incluir este archivo en todas las páginas que conforman el sistema, siempre se comprueba la sesión iniciada.

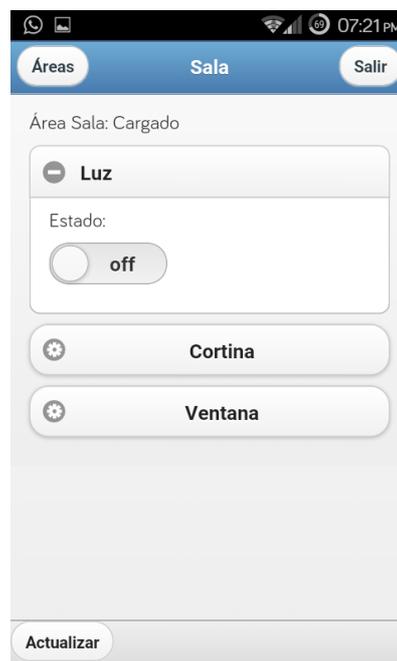
El contenido de la página consta de tres enlaces hacia las páginas correspondientes con el área o dispositivo a controlar y para que se vea más agradable visualmente se les ha aplicado el formato de botón.

```
<!--contenido-->
<div data-role="content" data-theme="c">
  <p>Seleccione un área a controlar.</p>
  <a href="sala.php" data-role="button" data-icon="gear" data-transition="Pop" data-ajax="false"
    data-theme="c">Sala</a>
  <a href="manta-termica.php" data-role="button" data-icon="gear" data-transition="Pop" data-
    ajax="false" data-theme="c">Manta térmica</a>
  <a href="puerta.php" data-role="button" data-icon="gear" data-transition="Pop" data-
    data-theme="c">Puerta</a>
</div>
```

Como se observa aquí se hace uso de los roles de jquery mobile que se habían mencionado en el capítulo uno. Gracias a esto los controles pueden ser personalizados de diferentes maneras. Otra parte importante es que en ninguna de las páginas diseñadas se hace uso de *ajax*, por eso es necesario agregar el rol *data-ajax="false"*, como se recordará *ajax* es una parte de jquery mobile que permite sincronizar contenido entre el dispositivo utilizado y el servidor; pero para la aplicación diseñada se utiliza javascript para este fin. Para un mejor análisis de esta página se proporciona el código fuente de la misma en el Anexo 21.

#### 4.1.7.4. Página sala.

En el área sala se encuentran agrupados tres nodos que constan de una lámpara, una cortina y una ventana. En la figura 61 ya se puede observar un control deslizante llamado *flip switch* que corresponde al control de la lámpara, y en la parte inferior un botón que permite actualizar el estado de la lámpara en caso de intervención manual.



**Figura 61.** Pagina sala.

**Fuente:** Propia

En la parte superior se observan dos botones que permiten regresar a la página áreas por medio de un enlace o salir del sistema cerrando la sesión iniciada por el usuario y redireccionando a la pagina de login. Estos botones están presentes en todas las páginas que permiten el control de algún dispositivo.

Para el control de la lámpara se hace uso de javascript trabajando conjuntamente con PHP. Inicialmente cuando la página es cargada, se ejecuta una subrutina en javascript llamada

*configurar()* que direcciona el módulo xbee de la Raspberry pi hacia el xbee que tiene asociado la lámpara, en este caso el xbee A. El código es el siguiente:

```
function configurar(){
//Direcciona el XBEE A
$.ajax({
  url: 'control/direccionA.php',
  type: 'post',
  beforeSend: function () {
    $("#Carga").html("Cargando...");
  },
  success: function (response) {
    $("#Carga").html(response);
    var estado = $("#Carga").html();
    if(estado[1]=='O'){
      if(estado[2]=='K'){
        $("#Carga").html("Cargado");
      }
    }
  },
  complete : function (){
    estado();
  }
});
}
```

Como se observa la función configurar apunta a un archivo llamado direccionA.php dentro del directorio control y este a su vez esta ubicado dentro del directorio principal de apache. Este archivo envía por el puerto serial los comandos necesarios para cambiar la dirección destino del xbee conectado a la Raspberry. Lo que se espera como respuesta es un “OK” para confirmar los cambios. Una vez direccionado, ahora se procede ha comprobar el estado de la lámpara con la función estado. El archivo direccionA.php puede observarse en el Anexo 22.

Cuando la función *configurar()* termina de ejecutarse inmediatamente se ejecuta la función *estado()* que contiene el siguiente código:

```
function estado(){
$.ajax({
  url: 'control/control-sala/estado-foco-sala.php',
  type: 'post',
  beforeSend: function () {
    $("#estadofoco0").html("");
  },
  success: function (response) {
    var x=response;
    if( x[0]==1){
      var fts = $('#flag0');
      fts.val('on');
      fts.slider('refresh');
    }
    if( x[0]==0){
      var fts = $('#flag0');
      fts.val('off');
      fts.slider('refresh');
    }
  }
});
}
```

```

        }
        if( x[0]==3){
            $("#estadofoco0").html("Problemas de comunicación, intente nuevamente.");
        }
    });
}

```

Como se observa esta función apunta a un archivo llamado estado-foco-sala.php ubicado dentro de la carpeta control-sala, el código de este archivo esta en el Anexo 23. Lo que se espera recibir es un uno (1) en caso de que la lámpara este encendida, un cero (0) en caso de que la lámpara este apagada o un tres (3) si existe algún problema de comunicación. Dependiendo de lo recibido el control *flip switch* de la pagina cambia de posición automáticamente reflejando el estado de la lámpara. En el caso de que exista algún problema en la comunicación se mostrará un mensaje en la misma interfaz.

Ahora se muestra la parte correspondiente al control deslizante que muestra la página.

```

<div data-role="content" data-theme="c">
  <!--aviso estado actual-->
  <div
    <label data-inline ="True">Área Sala:</label>
    <label id="Carga" data-inline ="true"></label>
  </div>

  <div data-role="collapsible" data-theme="c" data-content-theme="d" data-collapsed="false">
    <h4>Luz</h4>
    <!--switch-Luz-->
    <div
      <label data-inline ="True">Estado:</label>
      <label id="estadofoco0" data-inline ="true"></label>
    </div>

    <!--slider Luz-->
    <select name="flag0" id="flag0" data-role="slider">
      <option value="off">off</option>
      <option value="on">on</option>
    </select>
  </div>

```

Aquí se observa un label llamado *estadofoco0* que se encarga de mostrar el mensaje en caso de una falla de comunicación. Al final del código se muestra el flip switch utilizado con el nombre *flag0*.

Cuando se mueve este control deslizante se ejecuta la función *foco1()*

```

function foco1(){
  //Controla el foco 1
  $.ajax({
    url: 'control/control-sala/foco_sala.php',

```

```

type: 'post',
beforeSend: function () {
$("#estadofoco0").html("");
},
success: function (response) {
var x=response;
if( x[0]==1){
var fts = $('#flag0');
fts.val('on');
fts.slider('refresh');
}
if( x[0]==0){
var fts = $('#flag0');
fts.val('off');
fts.slider('refresh');
}
if( x[0]==3){
$("#estadofoco0").html("Problemas de comunicación, intente nuevamente.");
}
}
});
}

```

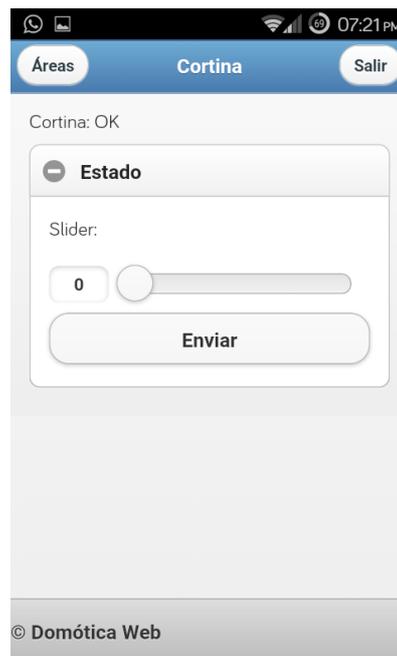
Esta función se ejecuta de manera similar a la función estado, apunta a un archivo llamado `foco_sala.php` que puede ser analizado en el Anexo 23. Se esperan las mismas respuestas que la función `estado()` para actualizar el control deslizante de acuerdo al estado de la lámpara.

Para poder actualizar el estado de la lámpara se utiliza un botón ubicado en la parte inferior de la página, de manera tal que cuando se presione ejecute la función `estado()` actualizando el control deslizante.

Finalmente sobre esta pagina, se debe explicar que la interacción con la misma desemboca en en envío y recepción de caracteres por medio del puerto serial. Las respuestas recibidas de los nodos son interpretadas mediante condiciones y gracias a esto es que se pueden actualizar visualmente los controles en la página. El código fuente de esta página se puede analizar en el Anexo 24.

#### 4.1.7.5. Página cortina.

La página cortina contiene un control deslizante que permite elegir el nivel en el cual se desea poner la cortina. Para que los cambios tengan efecto hay que enviar el valor con el botón enviar.



**Figura 62.** Pagina cortina.

**Fuente:** Propia

Como en las anteriores páginas aquí también se hace uso de javascript para ejecutar archivos PHP alojados en el servidor. Al cargar la página se procede a verificar la sesión y a direccionar el xbee hacia el xbee A como en la anterior pagina. Al completar la función inmediatamente se procede ha ejecutar la función *estado-cortina()*.

```
function Estado_cortina(){
    $.ajax({
        url: 'control/control-cortina/estado.php', //ejecuta este archivo
        type: 'post',
        beforeSend: function () {
        },
        success: function (response) {
            $("#Estado").html(response); //Pone la respuesta en el label #Estado
            var estado = $("#Estado").html();
        }
    });
}
```

```

    var fts = $('#slider-10');
    fts.val(parseInt(estado)); //Transforma lo recibido a un entero
    fts.slider('refresh'); //Refresca la posición de la barra deslizante
  }
});
}

```

Esta función ejecuta el archivo estado.php que se encuentra dentro del directorio control-cortina (mire los archivos contenidos en esta carpeta en el Anexo 25) , este archivo envía la letra “u” al nodo y devuelve la posición actual de la cortina. Para finalmente poner esta posición en la barra deslizante (*slider-10*).

El contenido de la pagina esta formado por:

```

<div
  <label data-inline = "True">Cortina:</label>
  <label id="Carga" data-inline = "true"></label>
</div>

<div data-role="collapsible" data-theme="c" data-content-theme="d" data-collapsed="false">
  <h4>Estado</h4>
  <form>
    <div>
      <label for="slider-10">Slider:</label>
      <label id="Estado" data-inline = "true"></label>
    </div>
    <input name="slider-10" id="slider-10" min="0" max="14" step="1" value="0" type="range">
    <button id="button" type="button">Enviar</button>
  </form>
</div>
</div>

```

En el código se puede observar el label *Estado* en donde se carga el estado de la cortina y también un *slider* con un valor mínimo de cero (0), un máximo de cien (14), un incremento de uno en uno y un valor inicial o por defecto de cero.

Para enviar el dato ingresado al nodo se utiliza la función *cortina()*:

```

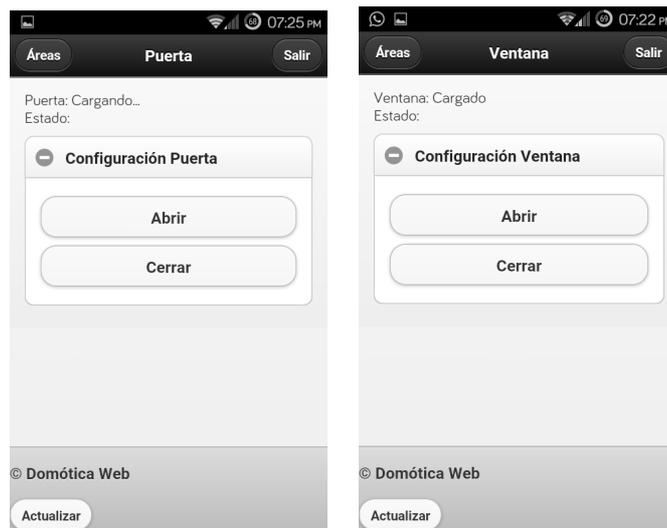
<script>
function cortina(){
var valor = $('#slider-10').val(); //recupera el valor ingresado en el slider
$.ajax({
  url: 'control/control-cortina/mover.php?valor='+valor, //enviá el valor al archivo mover.php
  type: 'post',
  beforeSend: function () {
    //$('#Carga').html("");
  },
  success: function (response) {
    $('#Estado').html(response); //recupera la respuesta y la pone en el label Estado
  }
});
}

```

Esta función toma el valor que el usuario indica en el *slider* y lo envía hacia el archivo *mover.php*, la respuesta de este archivo es ingresada en el label *Estado*. Un análisis más a fondo del código correspondiente a esta página se lo puede hacer en el Anexo 26.

#### 4.1.7.6. Páginas ventana y puerta.

La páginas ventana y puerta tienen un funcionamiento casi idéntico, la única diferencia son los archivos que ejecutan las funciones de javascript y los nombres que aparecen en la interfaz.



**Figura 63.** Páginas ventana y puerta.

**Fuente:** Propia

En el caso de la página ventana se utiliza el archivo *direccionA.php* para dirigirse al xbee A, mientras que para la puerta se usa el archivo *direccionB.php* para cambiar la dirección hacia el xbee B. Esta y los nombres mostrados en las páginas son las únicas diferencias que se pueden destacar en ambas páginas. Para explicar el funcionamiento de estas páginas se toma el archivo correspondiente a la página ventana.

Inmediatamente luego de hacer el direccionamiento del xbee se ejecuta la función *estado()* mostrada a continuación.

```
function estado(){
$.ajax({
  url: 'control/control-ventana/estado.php',
  type: 'post',
  beforeSend: function () {
  },
  success: function (response) {
    var x=response;
    if( x[0]=='k'){ //respuesta para ventana asegurada
      $('#button1').button('disable');
      $('#button2').button('disable');
      $('#Estado').html("La ventana esta ASEGURADA.");
    }

    if( x[0]=='c'){ //respuesta para ventana cerrada
      $('#button2').button('disable');
      $('#button1').button('enable');
      $('#Estado').html("La ventana esta CERRADA.");
    }

    if( x[0]=='a'){ //respuesta para ventana abierta
      $('#button2').button('enable');
      $('#button1').button('disable');
      $('#Estado').html("La ventana esta ABIERTA.");
    }

    if( x[0]=='e'){ // respuesta para ventana entre-abierta
      $('#Estado').html("Ventana entreabierta");
      $('#button1').button('enable');
      $('#button2').button('enable');
    }

    if( x[0]==3){ // respuesta para indicar problemas de comunicación.
      $('#Estado').html("Problemas de comunicación, intente nuevamente.");
    }
  },

```

Como se observa esta función apunta a un archivo llamado *estado.php* que se encuentra en el directorio control-ventana (vea el Anexo 27). Este archivo al ejecutarse envía la letra “o” y devuelve uno de los caracteres mostrados en el código. Finalmente el estado se imprime en el label *Estado*. El contenido de la página es el siguiente:

```
<!--contenido-->
<div data-role="content" data-theme="c">
  <div>
    <label data-inline = "True">Ventana:</label>
    <label id="Carga" data-inline = "true"></label>
  </div>
  <div>
    <label data-inline = "True">Estado:</label>
    <label id="Estado" data-inline = "true"></label>
  </div>

  <!--Menu colapsable opciones manta-->
  <div data-role="collapsible" data-theme="c" data-content-theme="d" data-collapsed="false">
  <h4>Configuración Ventana</h4>
    <button id="button1" type="button">Abrir </button>
    <button id="button2" type="button">Cerrar </button>
  </div>
</div>
```

Aquí se observan la etiqueta *Estado* donde aparece el mensaje correspondiente al estado del acceso en este caso la ventana y también dos botones que se habilitan o se deshabilitan dependiendo de estado de la ventana. Cuando los botones son presionados se ejecutan estas funciones de javascript:

```
<script type="text/javascript">
  $('#button1').click(function() {miAbrir()}); //si presiona el botón1 ejecute la funcion
  $('#button2').click(function() {miCerrar()}); //si presiona el botón1 ejecute la funcion
  $('#button3').click(function() {estado()}); //si presiona el botón1 ejecute la funcion
</script>
```

Las funciones relacionadas con este pedazo de código son:

```
function miAbrir()
{
  $.ajax({
    url: 'control/control-ventana/abrir.php',
    type: 'post',
    beforeSend: function () {

    },
    success: function (response) {
      var x=response;
      if( x[0]=='k'){
        $('#button1').button('disable');
        $('#button2').button('disable');
        $('#Estado').html("La ventana esta ASEGURADA.");
      }
      if( x[0]=='h'){
        $('#button1').button('disable');
        $('#button2').button('enable');
        $('#Estado').html("Abriendo ventana.");
      }
      if( x[0]==3){
        $('#Estado').html("Problemas de comunicación, intente nuevamente.");
      }
    }
  },
}
```

Como se observa esta función ejecuta un archivo llamado *abrir.php* (mire el Anexo 27), este archivo envía “q” y devuelve “k” para indicar que la ventana esta asegurada, “h” para indicar que la ventana esta abierta y “3” para notificar que existen problemas de comunicación. Dependiendo de estas respuestas los botones se habilitan o se deshabilitan

correspondiendo con el estado de la ventana. Este funcionamiento se refleja con la función *miCerrar()* que realiza las mismas acciones pero apuntando al archivo *cerrar.php*, este archivo envía y devuelve la letra “k” para mostrar una ventana asegurada, la letra “j” para

una ventana cerrada y “3” para indicar que existen problemas de comunicación. De manera similar los botones se actualizan para mostrar el estado actual de la ventana.

Finalmente el botón *Actualizar* ejecuta la misma función estado que se ejecuta al cargar la página *estado()*. El código fuente de las páginas puerta y ventana se muestran en el Anexo 28.

#### 4.1.7.7. Página manta.

La página manta es muy parecida visualmente a las páginas puerta y ventana, pero los archivos que ejecuta detrás de la interfaz son algo distintos ya que la naturaleza del nodo que controla esta página es diferente, como se indicó antes el nodo manta esta encargado de conectar y desconectar una manta por medio de un temporizador. Entonces esta página a pesar de ser visualmente muy parecida a las páginas que permiten el acceso a la vivienda, es mas parecida a la página que controla la lámpara, pero aquí se la ha implementado con botones.



**Figura 64.** Pagina manta.

**Fuente:** Propia

La figura 44 muestra la interfaz que controla en nodo manta, consta de una pequeña área delante la etiqueta *Estado* donde se mostrará el estado de la misma. Luego se han ubicado dos

botones para el control de la manta. De manera similar a las paginas anteriores en esta página se direcciona el xbee de la Raspberry con el archivo direccionB.php hacia el xbee B, para luego comprobar el estado con esta función:

```
function estado() {
    $.ajax({
        url: 'control/control-manta/estado.php',
        type: 'post',
        beforeSend: function () {
        },
        success: function (response) {
            var x=response;
            if( x[0]==1){
                $('#button1').button('disable');
                $('#button2').button('enable');
                miTiempo();
            }

            if( x[0]==0){
                $('#button1').button('enable');
                $('#button2').button('disable');
                $('#Estado').html("La manta esta desactivada.");
            }

            if( x[0]==3){
                $('#Estado').html("Problemas de comunicación, intente nuevamente.");
            }
        }
    },

```

Esta función ejecuta el archivo *estado.php* ubicado en el directorio control-manta(vea el Anexo 29) que envía la letra “g” para consultar el estado actual de la misma. Las posibles respuestas a esta petición pueden ser uno (1) para una manta activa, cero (0) para una manta inactiva y (3) para indicar problemas de comunicación. Dependiendo de estas respuestas los botones se habilitan o deshabilitan correspondientemente con el estado actual de la manta.

Para activar la manta se ejecuta la siguiente función:

```
function miActivar()
{
    $.ajax({
        url: 'control/control-manta/activar.php',
        type: 'post',
        beforeSend: function () {
            $('#Estado').html("");
        },
        success: function (response) {
            var x=response;
            if( x[0]==1){
                $('#button1').button('disable');
                $('#button2').button('enable');
                miTiempo();
            }
            if( x[0]==3){
                $('#Estado').html("Problemas de comunicación, intente nuevamente.");
            }
        }
    });
}
```

Esta función ejecuta el archivo `activar.php`, este archivo envía la letra “e” hacia el nodo manta y este puede responder con tres (3) para reportar problemas de comunicación, uno (1) para indicar que la manta esta activa, además cuando esta función devuelve uno como respuesta se ejecuta la función `miTiempo()` mostrada a continuación.

```
function miTiempo()
{
$.ajax({
url: 'control/control-manta/tiempo.php',
type: 'post',
beforeSend: function () {
$("#Estado").html("");
},
success: function (response) {
$("#Estado").html(response);
},
complete : function (){
}
});
}
```

Esta función ejecuta el archivo `tiempo.php` y este a su vez envía la letra “d” para preguntar el tiempo restante que la manta va ha estar activa, finalmente esta función pone la respuesta en el label `Estado`. Ahora bien cuando el nodo manta recibe la letra “d” este responde con el tiempo que la manta va ha estar activa en milisegundos, el archivo `tiempo.php` es el encargado de transformar este tiempo a un formato mas agradable a la vista minutos y segundos. La función para desactivar la manta es muy similar a la función `miActivar()`:

```
function miDesactivar()
{
$.ajax({
url: 'control/control-manta/desactivar.php',
type: 'post',
beforeSend: function () {
$("#Estado").html("");
},
success: function (response) {
var x=response;
if( x[0]==0){
$("#button1").button('enable');
$("#button2').button('disable');
$("#Estado").html("La manta esta desactivada.");
}
if( x[0]==3){
$("#Estado").html("Problemas de comunicación, intente nuevamente.");
}
},
complete : function (){
}
}
```

Esta función apunta al archivo *desactivar.php* y la respuesta esperada puede ser cero (0) para indicar que la manta se desactivo o tres (3) para notificar que existen un problema en la comunicación. En la parte inferior de la página se ubica un botón para actualizar el contenido de la página de acuerdo al estado actual de la manta, este al ser activado ejecuta la misma función `estado()` que se ejecuta al cargar inicialmente la página. El código fuente de esta página se expone en el Anexo 30.

## **5. CAPÍTULO V**

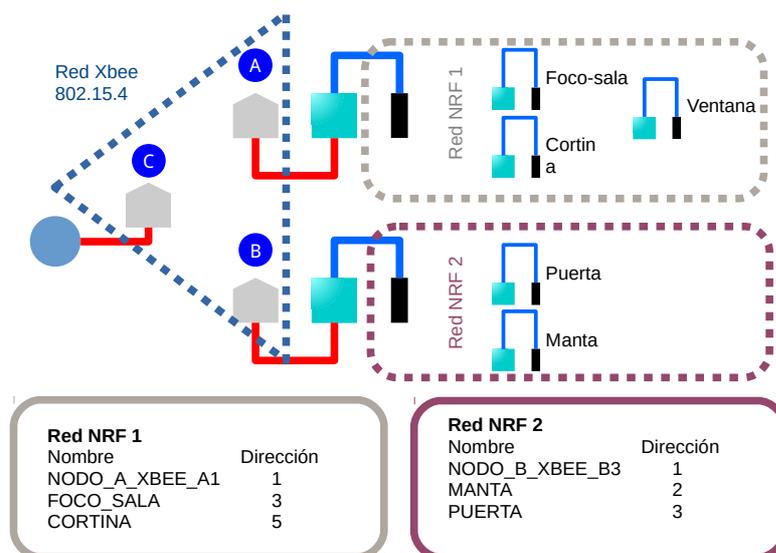
### **INTRODUCCIÓN**

El presente capítulo describe el proceso de instalación de los dispositivos construidos en la vivienda del usuario, uno de los objetivos del proyecto aunque no sea específico es conservar la estética de la casa y ser lo menos intrusivo. El sistema debe ser lo más transparente posible hacia el usuario y otros miembros del hogar.

#### **5.1. Implementación y pruebas de funcionamiento.**

La implementación del sistema domótico conlleva hacer las pruebas necesarias, tanto de comunicación como de funcionamiento de los nodos, recordando la topología propuesta en la figura 22 de este documento se probarán los enlaces de comunicación para garantizar tráfico entre dispositivos, en caso de presentarse problemas con los alcances por distancias se procede cambiar de lugar los nodos intermedios, para dichos cambios se toma en cuenta que se está trabajando con módulos que tienen un alcance en interiores de más o menos 40 metros de distancia.

La topología a implementarse es la mostrada en la figura 65, como se explicó anteriormente las peticiones y cambios de estado siempre se originan en la Raspberry, entonces para realizar las pruebas pertinentes de comunicación la información siempre inicia y termina en la Raspberry, es decir se considerará que la prueba es un éxito cuando el dato pueda llegar a su destino y la respuesta del nodo regrese a la Raspberry.



**Figura 65.** Topología implementada.

**Fuente:** Propia

Para realizar estas pruebas se hace uso de una terminal abierta en la placa Raspberry, para esto se usa el protocolo SSH y ya en la placa se procede a ejecutar el programa MINICOM, que permite administrar la comunicación serial entre el módulo xbee y la placa.

### 5.1.1. Pruebas de comunicación entre los módulos xbee.

Esta prueba consiste en el envío de un carácter desde el xbee C conectado a la placa hacia los módulos xbee A y xbee B, se realiza esta comunicación con el objetivo de comprobar las comunicaciones en el núcleo de la red de dispositivos, hay que tomar en cuenta que la red formada por los módulos xbee es la que interconecta a todos los dispositivos. Las pruebas de comunicación se las hace mediante una comunicación serial mediante el programa Minicom y el puerto serial donde está conectado el módulo xbee corresponde al /dev/ttyUSB0.

Para esta prueba se ubica la placa Raspberry pi como se muestra en la figura 66, mientras

que los nodos intermedios se los ubica al alcance de este nodo principal, es decir a distancias no mayores a 40m.



**Figura 66.** Ubicación de la placa Raspberry pi.

**Fuente:** Propia

Convenientemente en cada uno de los programas de los nodos intermedios se ha incrustado un segmento de código que al recibir la letra “i” responde con la información de los nodos.

Entonces se procede a la prueba de comunicación hacia el xbee A, primero se debe iniciar una sesión como root mediante SSH hacia la placa Raspberry pi, para esto es necesario ejecutar el archivo direccionA.php como se muestra.

```
root@raspberrypi:/home/pi# php /var/www/html/xbee4/control/direccionA.php
OK
root@raspberrypi:/home/pi#
```

Ahora se procede a ejecutar el programa minicom y enviar el carácter “i”.

```

pi@raspberrypi: -
pi@raspberrypi: - 77x37
Welcome to minicom 2.7

OPTIONS: I18n
Compiled on Jan 12 2014, 05:42:53.
Port /dev/ttyUSB0, 05:05:30

Press CTRL-A Z for help on special keys

*****
NODO: NODO_A_XBEE_A1
Nombre de programa: NODO_A_XBEE_A1
Canal utilizado: 90
Velocidad NRF24L01 250kbps
Potencia NRF24L01 0dBm
Elementos de red
Nodo: FOCO_SALA <----> NODO_A_XBEE_A1
-Interactue con:
-Letra 'm' <----> cambia estado
-Letra 's' <----> reporta el estado

Nodo: CORTINA <----> NODO_A_XBEE_A1
-Interactue con:
u -Devuelve la posicion de la cortina
-Cadena cXX donde XX es la poscion que se desea en la cortina

Nodo: VENTANA <----> NODO_A_XBEE_A1
Letras para interaccion Respuesta significado
q h Abre la ventana
p c Cierra la ventana
o Comprueba el estado
e La ventana esta entreabierta
c La ventana esta cerrada
a La ventana esta abierta
k La ventana esta asegurada
*****
CTRL-A Z for help | 9600 8N1 | NOR | Minicom 2.7 | VT102 | Offline | ttyUSB0

```

**Figura 66.** Respuesta módulo xbeeA.

**Fuente:** Propia

La respuesta vista en la figura 66 de este módulo, muestra información detallada del mismo y de los dispositivos conectados a él. Como se observa la comunicación se ha realizado con éxito.

Ahora se procede a direccionar el xbee C hacia el xbee B con la ejecución del archivo `direccionB.php`. Para esto es necesario cerrar el puerto serial y ejecutar los comandos.

```

root@raspberrypi:/home/pi# php /var/www/html/xbee4/control/direccionB.php
OK
root@raspberrypi:/home/pi#

```

Ahora se debe abrir el puerto serial mediante minicom y finalmente se realiza la prueba de comunicación enviando la letra “i” a través del puerto serial.

```

pi@raspberrypi: -
pi@raspberrypi: - 77x37
OPTIONS: I18n
Compiled on Jan 12 2014, 05:42:53.
Port /dev/ttyUSB0, 05:15:04

Press CTRL-A Z for help on special keys

NODO_B_XBEE_B3
Nombre de programa: NODO_B_XBEE_B3
Canal utilizado: 95
Velocidad NRF24L01 250kbps
Potencia NRF24L01 0b8m
Elementos de red
Nodo: Manta <----> NODO_B_XBEE_B3
Interactue con
-Letra      Funcion          Devuelve
e           activa            1 si se activo
f           desactiva         0 si se desactivo
d           tiempo            tiempo restante ms
g           comprueba         1 si esta activa o 0 si no esta activa

Nodo: Puerta <----> NODO_B_XBEE_B3
Interaccion  Respuesta  Significado
q           h          Abre la puerta
p           c          Cierra la puerta
o           o          Comprueba el estado
           e          La puerta esta entreabierta
           c          La puerta esta cerrada
           a          La puerta esta abierta
           k          La puerta esta asegurada

Nodo: Foco_aux falta configurar <----> NODO_B_XBEE_B3
-Interactue con
-Letra 'c' <----> cambia estado
-Letra 'w' <----> reprota el estado
*****
CTRL-A Z for help | 9600 8N1 | NOR | Minicom 2.7 | VT102 | Offline | ttyUSB0

```

**Figura 67.** Respuesta módulo xbeeB.

**Fuente:** Propia

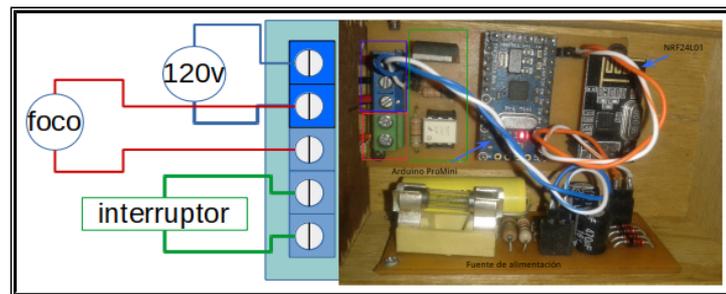
La comunicación con el xbee B también muestra información sobre el nodo y que dispositivos están conectados a él, como se observa la información llega sin ningún tipo de error al Raspberry por lo cual se considera una prueba favorable.

Como comentario sobre estos enlaces se puede decir que la velocidad de respuesta es impresionante y su alcance es considerable, esto facilita mucho el proceso de pruebas con los demás dispositivos ya que los nodos intermedios pueden ser movidos casi ha cualquier lugar de la vivienda.

### 5.1.2. Pruebas de nodo foco.

Para esta prueba se procede a instalar el nodo ya en el sitio en donde permanecerá, la

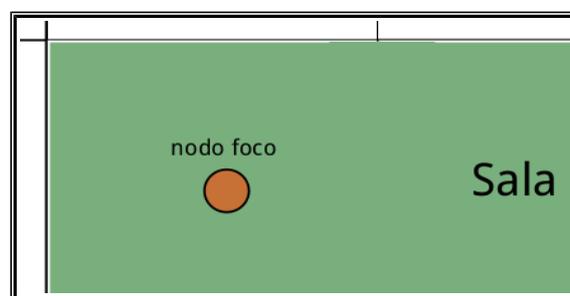
conexión del mismo se hace como se muestra en la figura 68, el cajetin de una lámpara cuenta con un voltaje de alimentación de 120V, dos conductores para el interruptor (en la figura de color verde) y los contactos donde conecta la lámpara misma representados en la figura de color rojo.



**Figura 68.** Diagrama de conexión lámpara.

**Fuente:** Propia

Ya en el sitio la conexión del nodo lámpara se ubica de la siguiente manera



**Figura 69.** Ubicación lámpara.

**Fuente:** Propia

La prueba de conectividad de este nodo se la realiza de manera similar a la prueba anterior, si se observa la figura 66 que corresponde a la información de los dispositivos conectados al xbee A esta la lámpara, entonces se ubica este nodo intermedio en la sala del domicilio. Se procede a direccionar el xbee C hacia el xbeeA de la misma manera que en la prueba anterior

y se enviar la letra “s” por medio de minicom para comprobar su estado.

```

pi@raspberrypi: -
pi@raspberrypi: ~ 77x37
Nodo: FOCO_SALA <----> NODO_A_XBEE_A1
-Interactue con:
-Letra 'm' <----> cambia estado
-Letra 's' <----> reporta el estado

Nodo: CORTINA <----> NODO_A_XBEE_A1
-Interactue con:
  u                -Devuelve la posicion de la cortina
-Cadena cXX donde XX es la poscion que se desea en la cortina

Nodo: VENTANA <----> NODO_A_XBEE_A1
Letras para interaccion Respuesta significado
q                h Abre la ventana
p                c Cierra la ventana
o                o Comprueba el estado
                 e La ventana esta entreabierta
                 c La ventana esta cerrada
                 a La ventana esta abierta
                 k La ventana esta asegurada
*****
0
0
0
0
0
1
0
1
0
1
0
3
3
1
3
0
CTRL-A Z for help | 9600 8N1 | NOR | Minicom 2.7 | VT102 | Offline | ttyUSB0

```

**Figura 70.** Pruebas de conectividad lámpara.

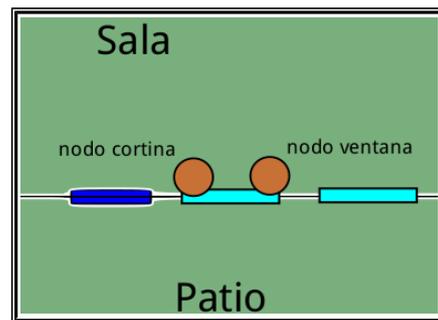
**Fuente:** Propia

En la figura 70 se observan las respuestas del nodo lámpara al enviar varias veces el carácter “s”, las respuestas que muestran los cambios realizados correctamente corresponden a los unos y los ceros, mientras que los números tres significa que existe algún problema de conectividad. Este problema se lo solventa con una ubicación más adecuada del nodo intermedio que contiene el xbee A.

### 5.1.3. Pruebas nodos cortina y ventana.

Para terminar con los dispositivos conectados al nodo xbee A, se procede a instalar los nodos cortina y ventana, como se espera estos nodos están ubicados en la misma ventana y

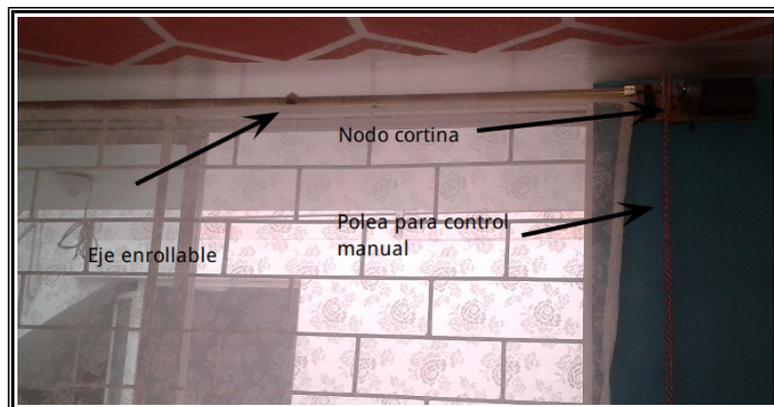
gracias ha esto comparten la misma fuente de alimentación, como se había explicado en el anterior capítulo estos nodos funcionan con dos voltajes, 12V y 5V. Las cargas se conectan en paralelo con la fuente de alimentación como se mostró en el diagrama eléctrico del nodo cortina. Ahora se muestra la ubicación exacta de los nodos en cuestión.



**Figura 71.** Ubicación nodos cortina y ventana.

**Fuente:** Propia

Entonces el nodo cortina luego de la instalación queda de la siguiente manera.



**Figura 72.** Partes nodo cortina.

**Fuente:** Propia

Y el nodo ventana una ves instalado queda así.



prueba se ha puesto la cortina en un estado cero, que equivale un estado inicial por lo que las respuestas deben ser este número

Tomando en cuenta que el nodo cortina y el nodo ventana están casi en el mismo lugar se espera que no haya ningún tipo de problema en la comunicación.



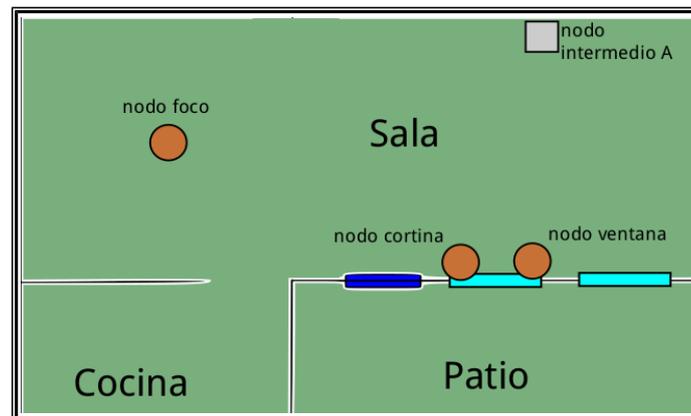
```
pi@raspberrypi: ~  
pi@raspberrypi: ~ 80x24  
Welcome to minicom 2.7  
OPTIONS: I18n  
Compiled on Jan 12 2014, 05:42:53.  
Port /dev/ttyUSB0, 07:41:44  
Press CTRL-A Z for help on special keys  
0  
0  
0  
0  
0  
0
```

**Figura 75.** Prueba de conectividad cortina.

**Fuente:** Propia

Y efectivamente en la figura 75 muestra las respuestas del nodo cortina. En seis envíos del carácter “u” se obtiene una respuesta de cero, cuyo significado es que la cortina esta en un estado inicial, valor esperado por su configuración inicial.

Con este dispositivo se terminan las pruebas de funcionamiento de los nodos conectados al Xbee A. Entonces se procede ha instalar definitivamente todos los nodos ya que no existe ningún problema de comunicación, de esta manera los dispositivos se ubican como se muestra en la figura 76.



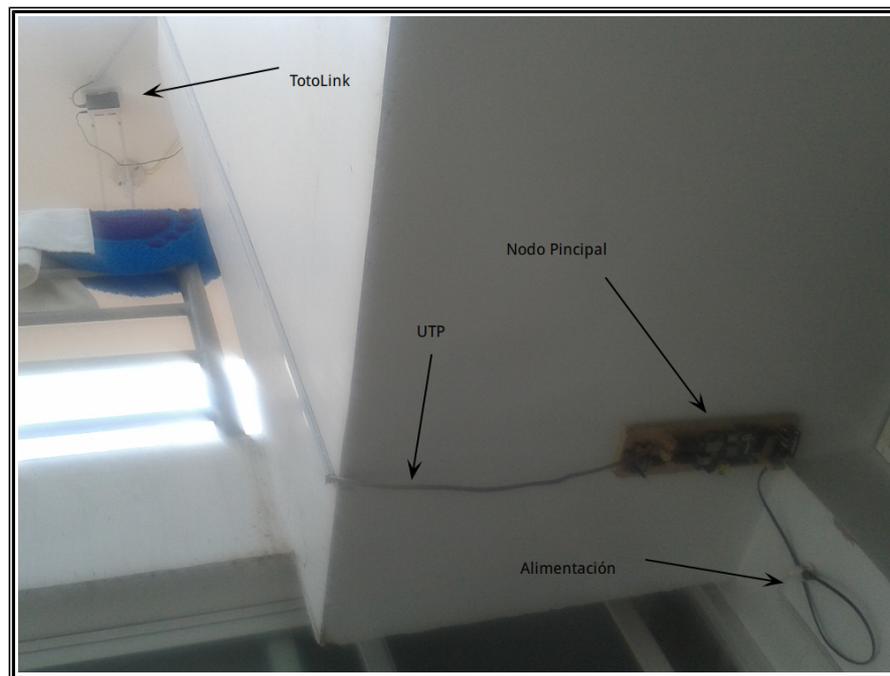
**Figura 76.** Ubicación nodos conectados a Xbee A instalados en el domicilio.

**Fuente:** Propia

## 5.2. Consideraciones e instalación en dispositivos conectados a Xbee A.

Como era de esperarse la instalación de estos dispositivos, requiere una conexión eléctrica que requiere de algunos pasos que se detallan a continuación.

En la instalación del nodo principal se tubo que sacar una toma de corriente extra desde el patio hacia la placa Raspberry pi, esto no demandó mayor esfuerzo ya que en el patio cuenta con una toma eléctrica, entonces se construyó una extensión eléctrica y se la instaló con canaleta para no alterar mayormente la estética del domicilio. Se fabricó un cable conexión ethernet hacia el equipo TotoLink y también se utilizó canaleta para mantener el orden y la estética. La figura 77 muestra la implementación del nodo principal.



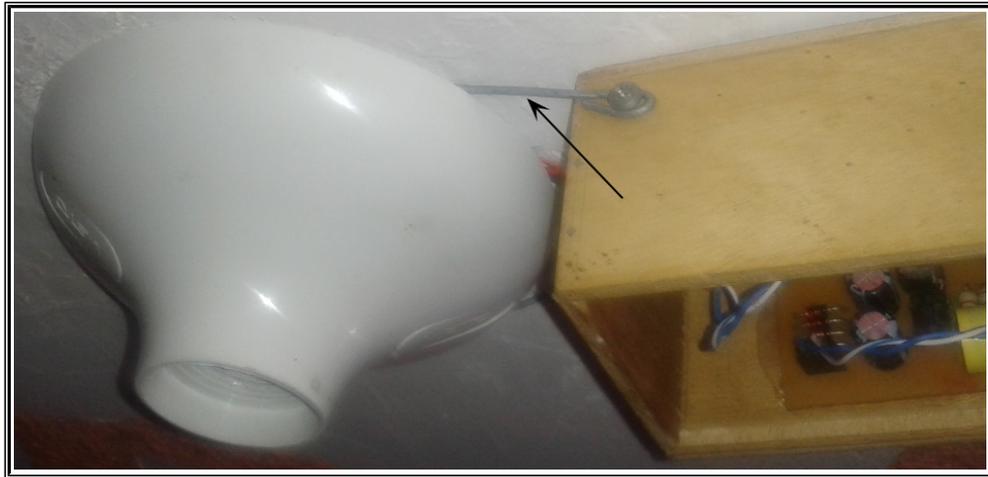
**Figura 77.** Implementación nodo principal.

**Fuente:** Propia

Como se observa fue necesario sujetar el nodo principal al techo, esto por las condiciones de conexión, algo que benefició al nodo en todos los sentidos, primero la ubicación del nodo en este lugar beneficia a la cobertura del módulo xbee conectado al mismo ya que esta en una zona casi central del domicilio, segundo las conexiones ethernet y eléctricas no se encontraban tan alejadas del lugar de la instalación, por lo que fue relativamente fácil su conexión

En la instalación del nodo lámpara, se requirió desconectar momentáneamente la energía eléctrica del domicilio ya que se requería desconectar la boquilla de la lampara he identificar los cables para su posterior conexión hacia el nodo.

Para sujetar el nodo lámpara se requirió construir un gabinete que cuente con algún método de sujeción a la boquilla de la lámpara misma.



**Figura 78.** Método de sujeción nodo lámpara.

**Fuente:** Propia

Finalmente el nodo lámpara quedó instalado de la siguiente manera.



**Figura 79.** Implementación nodo lámpara.

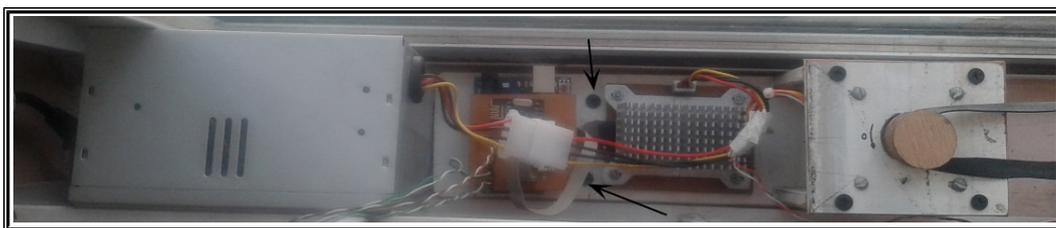
**Fuente:** Propia

Para la instalación del nodo ventana se tuvo que sacar un punto de conexión eléctrica de una luminaria que se encontraba cerca.



**Figura 80.** Punto eléctrico para nodos ventana y cortina.

**Fuente:** Propia

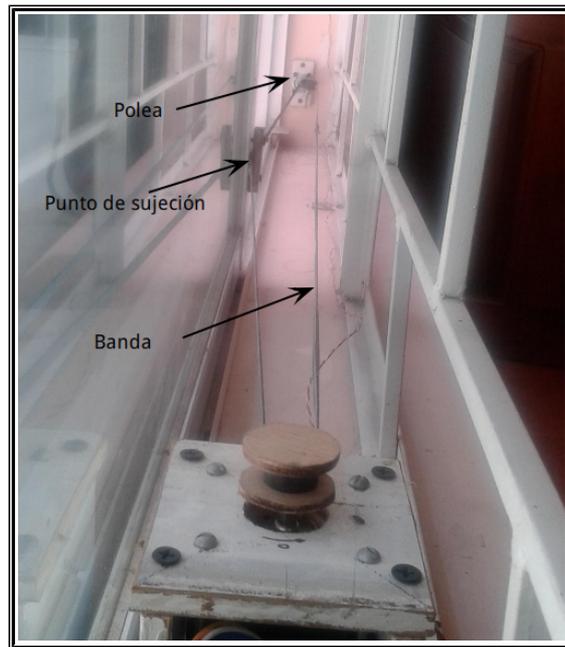


**Figura 81.** Implementación nodo ventana.

**Fuente:** Propia

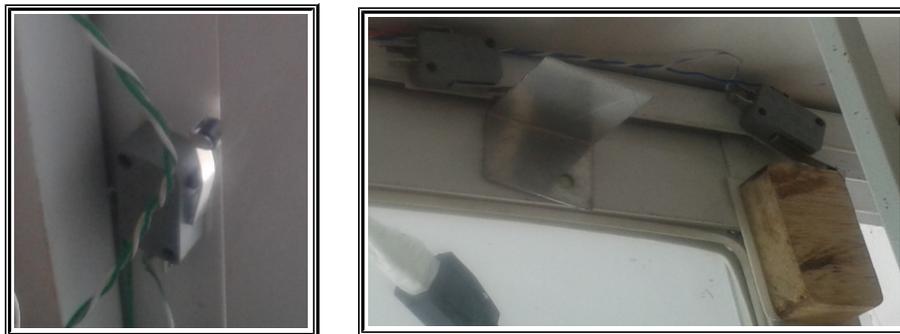
La figura 81 muestra la ubicación del nodo ventana, para sujetarlo se utilizaron dos tornillos anclados al concreto señalados con las flechas.

Para transformar el movimiento circular del motor en movimiento lineal, se recurrió a una banda anclada al la parte móvil de la ventana y a la instalación de una polea en un punto fijo en el marco de la ventana.



**Figura 82.** Partes nodo ventana.

**Fuente:** Propia

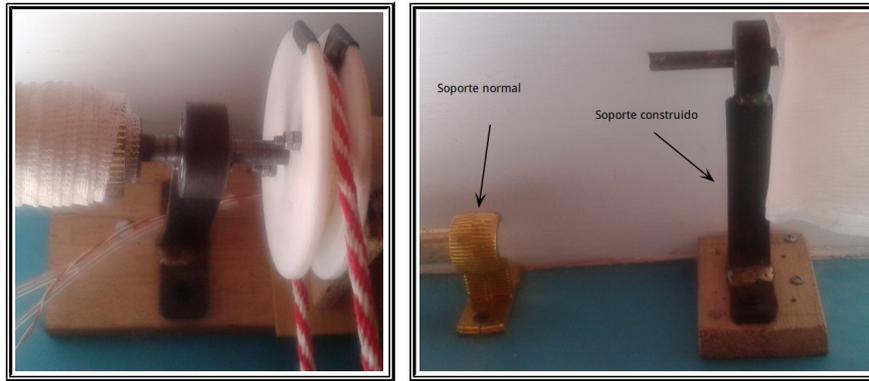


**Figura 83.** Interruptores de fin de camino en nodo ventana.

**Fuente:** Propia

Finalmente se instalaron interruptores de fin de camino en el marco de la ventana para poder detectar los estados de la misma.

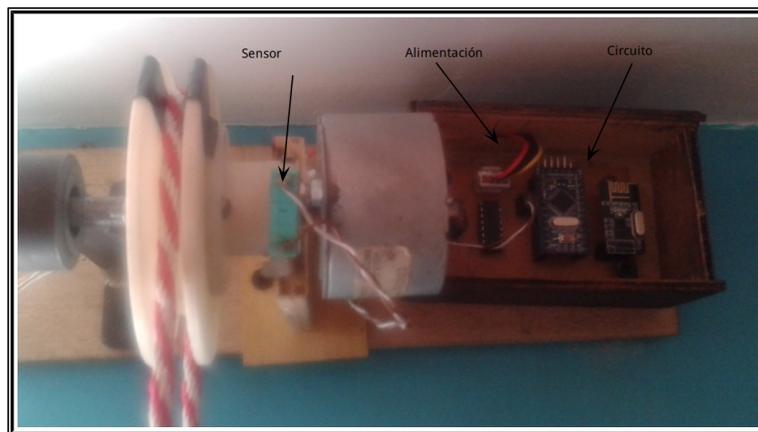
La implementación de la cortina requirió que se cambien los soportes y la construcción de los ejes que soporten la cortina y permitan que esta se enrolle.



**Figura 84.** Soportes y polea en nodo cortina.

**Fuente:** Propia

En la figura 84 se puede también observar una polea que se construyó para el control manual de la cortina.



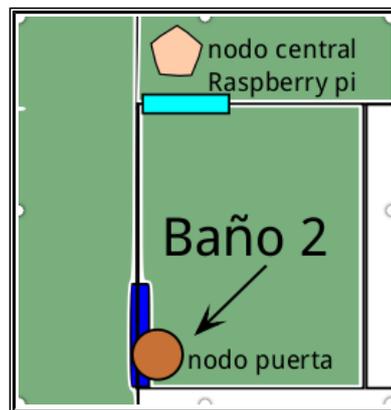
**Figura 85.** Partes de nodo cortina implementado.

**Fuente:** Propia

Con esto se finaliza la instalación los nodos conectados al Xbee A, ahora se continua con las pruebas de los nodos conectados al Xbee B.

### 5.2.1. Pruebas nodo puerta.

La prueba de conectividad de este nodo es similar a la de la ventana, primero se procede ha ubicar el nodo en el lugar donde esta la puerta.



**Figura 86.** Ubicación nodo puerta.

**Fuente:** Propia

Como se indico en el anterior capítulo los nodos ventana y puerta son casi idénticos, inclusive funcionan con las misma letras, el cambio que se debe hacer es direccionar el xbee C al xbee B, para esto se debe ejecutar el siguiente comando:

```
root@raspberrypi:/home/pi# php /var/www/html/xbee4/control/direccionB.php
OK
root@raspberrypi:/home/pi#
```

y luego mediante minicom enviar la letra “o” que comprueba el estado de la puerta. En la figura 87 se observa las respuestas del nodo puerta, las primeras letras “e” significan que la puerta estaba entre-abierta, cuando la respuesta cambia a la letra “k” significa que la puerta se cerró y se aseguro manualmente.

```

pi@raspberrypi: -
pi@raspberrypi: - 80x24

Welcome to minicom 2.7

OPTIONS: I18n
Compiled on Jan 12 2014, 05:42:53.
Port /dev/ttyUSB0, 08:27:42

Press CTRL-A Z for help on special keys

e
e
e
k
k
k

```

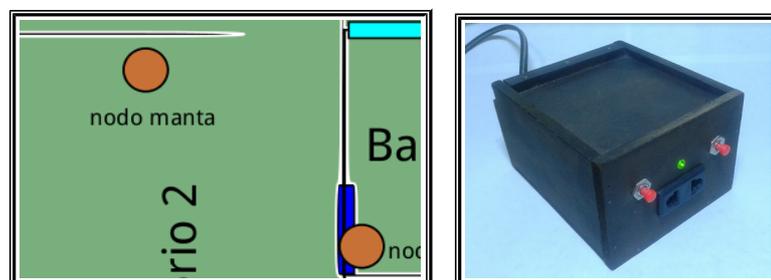
**Figura 87.** Respuestas nodo puerta.

**Fuente:** Propia

Esto demuestra que se se tiene una comunicación fiable y no es necesario mover el nodo intermedio el cual se ubico a unos 5m del baño.

### 5.2.2. Pruebas nodo manta.

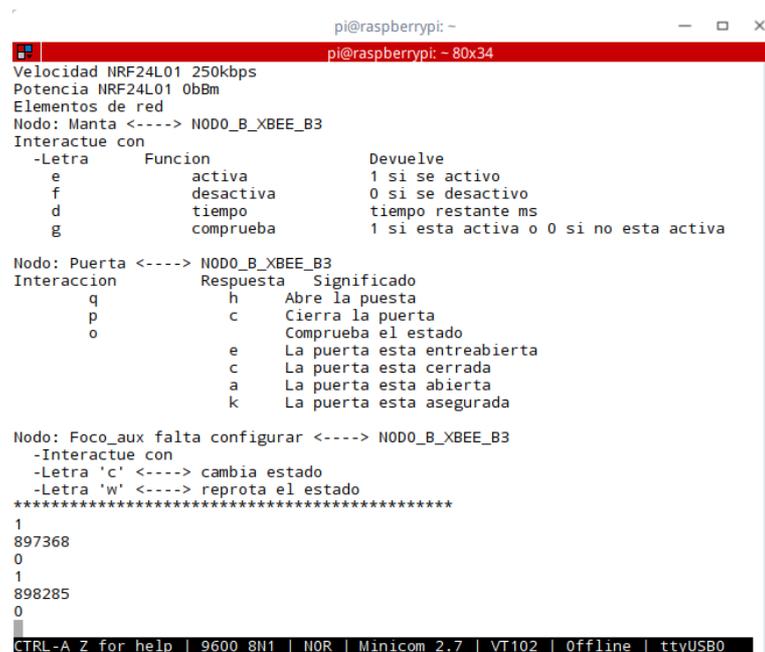
En nodo manta es un dispositivo pequeño, cuenta con un enchufe para conectarlo a la red eléctrica y se lo puede mover al antojo del usuario. Si se observa la figura 88 en la parte frontal existen dos botones que permiten el uso manual del nodo, mientras que un led funciona como indicador.



**Figura 88.** Ubicación nodo Manta.

**Fuente:** Propia

Como en el caso anterior para la prueba se debe direccionar el xbee C hacia el xbee B y luego se envían los caracteres “e” para activar la manta, “d” para recuperar el tiempo restante en milisegundos y finalmente “f” para apagar el dispositivo.



```

pi@raspberrypi: ~
pi@raspberrypi: ~ 80x34
Velocidad NRF24L01 250kbps
Potencia NRF24L01 0dBm
Elementos de red
Nodo: Manta <----> NOD0_B_XBEE_B3
Interactue con
-Letra   Funcion      Devuelve
e         activa          1 si se activo
f         desactiva       0 si se desactivo
d         tiempo          tiempo restante ms
g         comprueba      1 si esta activa o 0 si no esta activa

Nodo: Puerta <----> NOD0_B_XBEE_B3
Interaccion Respuesta Significado
q           h         Abre la puerta
p           c         Cierra la puerta
o           o         Comprueba el estado
           e         La puerta esta entreabierta
           c         La puerta esta cerrada
           a         La puerta esta abierta
           k         La puerta esta asegurada

Nodo: Foco_aux falta configurar <----> NOD0_B_XBEE_B3
-Interactue con
-Letra 'c' <----> cambia estado
-Letra 'w' <----> reprot el estado
*****
1
897368
0
1
898285
0
CTRL-A Z for help | 9600 8N1 | NOR | Minicom 2.7 | VT102 | Offline | ttyUSB0

```

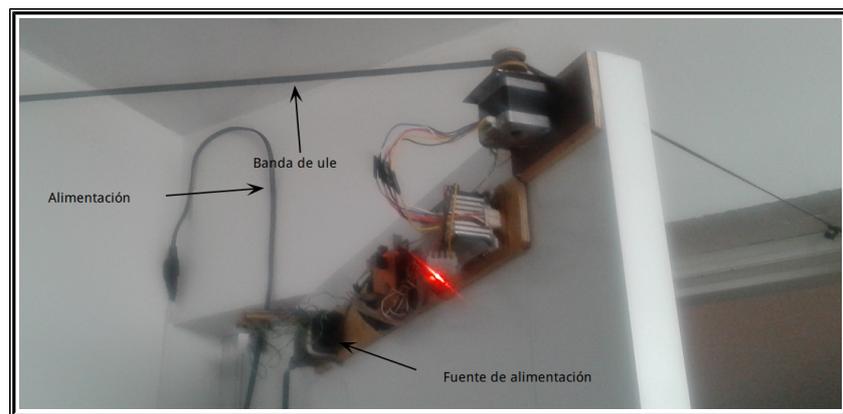
**Figura 89.** Prueba de conectividad manta.

**Fuente:** Propia

Como se observa se ha enviado dos veces la secuencia indicada obteniendo respuestas acertadas, uno (1) significa que la manta esta conectada, el número 897368 son los milisegundos que faltan para que la manta sea desconectada y finalmente cero (0) indica que la manta ha sido desconectada.

### 5.3. Consideraciones e instalación de dispositivos conectados a Xbee B.

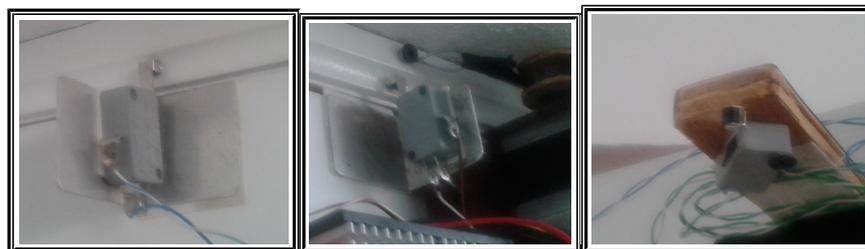
El nodo puerta fue empotrado en la puerta misma, como consecuencia los componentes se están moviendo contante mente, pero esta condición no impide tener una muy buena comunicación con el nodo, la parte mas complicada en la instalación de este nodo fue la ubicación y sujeción del motor de pasos.



**Figura 90.** Partes nodo puerta.

**Fuente:** Propia

En la figura 90 se observa como quedó finalmente el dispositivo empotrado en la puerta, el funcionamiento mecánico de este nodo aprovecha la elasticidad de una banda de hule, que sirve como guía para el motor de pasos.

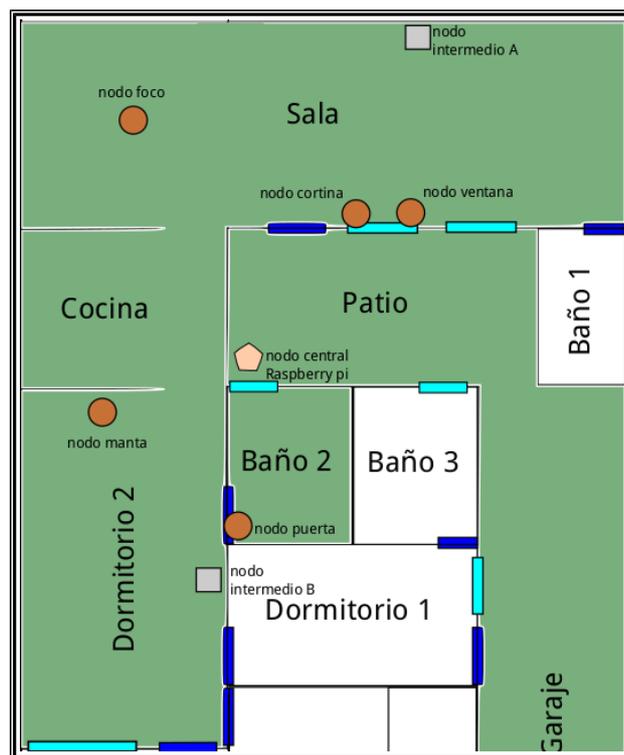


**Figura 91.** Interruptores de fin de camino nodo puerta.

**Fuente:** Propia

Dependiendo del giro del motor por medio de fricción la puerta se mueve abriendo o cerrando el acceso., el movimiento dura hasta que se presionan los interruptores de fin de camino. Finalmente gracias a que el eje del motor queda libre cuando se termina su movimiento el usuario puede abrir y cerrar la puerta normalmente.

Para la alimentación de este nodo se utilizó un toma de corriente ubicada en el baño y con el fin de no restar un conector a este, se opto por conectar una extensión eléctrica directamente a los cables internos. Finalmente se le recuerda al lector que el nodo manta puede conectarse en cualquier toma y por lo tanto no se lo ha incluido en esta sección. De esta manera los nodos quedan ubicados como se muestra.



**Figura 92.** Ubicación de todos los nodos en el domicilio.

**Fuente:** Propia

## 6. CAPÍTULO VI

### INTRODUCCIÓN

Debido que todo proyecto tiene una inversión monetaria y se espera recibir un beneficio a cambio, en este capítulo se realizara un estudio en base a la información recolectada a lo largo del proyecto, para medir la satisfacción del propio usuario, en donde se pretenderá demostrar que las actividades en las que el sistema ayuda a la persona, son realmente ventajosas en contraste con las actividades que tendría que hacer manualmente. De esta manera se demostraría como el sistema optimiza el tiempo del usuario y mejora su calidad de vida.

#### 6.1. Análisis costo beneficio.

Este proyecto de titulación tiene como objetivo principal mejorar la calidad de una persona con paraplejia, por lo que es un proyecto social si una tasa de retorno económica, es decir existe una inversión económica pero el beneficio no es económico. El beneficio que se obtiene se puede enfocar en las actividades independientes, que el usuario puede realizar después de la instalación del sistema domótico.

Los beneficios primordiales de la domótica es que ofrece un entorno más favorable y óptimo para una vida lo más autónoma posible. El hecho de que los sistemas de la vivienda se pueden programar ya sea para que realicen ciertas funciones con sólo tocar un botón o que las lleven a cabo en función de otras condiciones del entorno, produce un aumento del confort y un ahorro de tiempo.

La posibilidad de poder ingresar desde cualquier parte del mundo por medio de Internet y poder acceder a la interfaz, representan una gran ventaja ya que el usuario puede verificar y controlar dispositivos si así lo quiere desde lugares muy alejados geográficamente.

En el tema de costos de los equipos utilizados para la construcción de los nodos, siempre se ha procurado adquirirlos por medio de tiendas electrónicas desde el extranjero, esto reduce los costos de manera beneficiosa ya que los equipos que se adquieren en el país se los oferta en un precio de hasta 4 veces mayor.

Para percibir con claridad la satisfacción del usuario se ha realizado una pequeña entrevista, recogiendo la opinión del usuario directamente beneficiado.

### **6.1.1. Entrevista al usuario.**

La entrevista al usuario comienza en la comodidad de su hogar y se enfoca en sus opiniones sobre el sistema instalado y los dispositivos intervenidos.

-Comencemos:

-¿Cual es su opinión sobre el proyecto en conjunto?

*“Brinda mucha comodidad, imagínese que se pueden controlar las cosas sin tener que levantarse y también desde Internet ”*

-¿Qué expectativas tenía sobre el proyecto y cuales se cumplieron?

*“La cobija o la manta me parecía una excelente idea, había pensado antes si existía alguna forma de automatizarla. El caso de los otros dispositivos pienso que cada*

*uno tiene un aporte sustancial para la comodidad en el hogar, yo personalmente me encuentro muy satisfecho.”*

*-¿Qué opina sobre la interfaz?*

*“Se ve como una aplicación nativa de Android, no parece una página web. Me parece una muy buena idea que los dispositivos se hayan dividido por áreas, para evitar confusiones. Otra cosa que me parece excelente es que cuando se abren las áreas correspondientes a los dispositivos se pueda reflejar en que estado están, entonces también puedo utilizar los controles manuales”*

*-¿Qué opina sobre el dispositivo que controla la manta?*

*“Me parece excelente este dispositivo, por mi trabajo yo salgo al rededor de las dos de la mañana y cuando llego a mi casa tengo que conectar la manta manualmente, el proceso hasta que la manta caliente la cama y yo puedo dormir es de al rededor de 45 minutos, este tiempo yo lo podría utilizar para descansar. Gracias a este aparato ahora ya puedo activarlo desde el trabajo antes de salir y cuando llego yo puedo dormir directamente. Cuida mi salud y me permite tener un mejor desempeño al siguiente día de trabajo”*

*-¿Qué opina sobre el control de luminosidad por medio de las lamparas y cortinas?*

*“En mi caso pienso que me ayuda mucho cuando yo ya estoy acostado en mi*

*cama, así me evita el tener que levantarme o pedir a alguien más que me ayude, estos dispositivos me permiten apagar las luces sin levantarme de la cama, así mismo con las cortinas. Otra ventaja que le veo a esto es que ayudaría a evitar la delincuencia, en el caso de que yo me vaya de viaje yo puedo prender y apagar las luces y mover las cortinas, de esta manera sorprender y engañar a la delincuencia simulando que alguien esta dentro de la casa.”*

-¿Cual es su opinión sobre las puertas y las ventanas?

*“Es genial el beneficio que me brinda, ya que para mi es un poco complicado mover ventanas, dependiendo de donde me halle a veces pido ayuda. En le caso de las puertas pienso que lo que aporta es la posibilidad de ver y limitar las áreas por ejemplo para que no entren las mascotas o evitar que mis sobrinos pequeños salgan”*

-¿Qué puede decir sobre la seguridad?

*“Me parece acertado el que se tengan que configurar previamente los dispositivos que van a acceder al sistema desde Internet, por que hoy los hackers están prácticamente en todos lados”*

-¿Usted cree que el sistema le ahorra tiempo en sus actividades?

*“Justamente, el caso de la manta es un ejemplo muy claro de ello, imagínese la*

*gran ventaja de ahorrarme 45 minutos y aprovecharlos descansando en vez de estar esperando, hablando de los otros dispositivos me ahorran movimientos incómodos dentro de mi casa, y además ya no tengo que molestar a nadie para que me ayude”*

-¿Como cree que se podría mejorar el proyecto?

*“Ampliando la aplicación a otros dispositivos, por ejemplo si me quisiera prepara un café remotamente, para cuando yo llegue este listo, me refiero a automatizar la cafetera y así con otros dispositivos”*

*“...yo deseaba automatizar algunas partes de mi auto que también me hacen falta, pero el gran limitante siempre fue el costo súper elevado que tenía esto. Ahora veo que es posible construir estas cosas a un costo razonable”*

-¿Cree que el sistema es accesible?

*“Como en mi caso, creo que siempre va a depender del tamaño de la casa. Como clase media el precio es un poco alto para una inversión de golpe, lo mejor siempre sería un pequeño plan de pagos, pero esto no debe ser inconveniente para quien en realidad vea las ventajas de este sistema y como puede ayudar a las personas en mi condición”*

-¿Tiene algún comentario final?

*“Por su puesto, me imagino las cosas que se podrían hacer con esta tecnología, hoy en un principio estos dispositivos parecen algo rústicos, pero con las experiencias y las pruebas que se vayan haciendo, es como una retroalimentación que va a permitir que estos dispositivos se mejoren, me imagino que en unos años ya existirán maravillas con respecto a este tema. De mi parte una felicitación a usted y a la universidad por apoyar y sacar adelante este tipo de proyectos. Muchas gracias.”*

Como se puede evidenciar en la entrevista el usuario se muestra entusiasmado por el sistema y sus funcionalidades, de esta manera se demuestra que el nivel de satisfacción es alto y aunque el sistema se puede mejorar de muchas formas el usuario se encuentra muy satisfecho con los logros obtenidos.

La automatización y control de dispositivos en la vivienda permite al beneficiado conseguir una mejora en su comodidad ayudando a reducir el tiempo en actividades cotidianas.

Gracias a las nuevas tecnologías como; dispositivos inalámbricos y electrónicos, red de Internet, los beneficios obtenidos por el usuario pueden ser diversos al utilizar esta tecnología para un sistema domótico.

Con la implementación del sistema en puertas y ventanas, el usuario tiene la facilidad de controlar estos dispositivos sin necesidad de moverse, puede abrir o cerrar los accesos

cuando el lo crea conveniente desde la comodidad de su tablet, celular, computador e independientemente de donde se encuentre ubicado. Además de estos beneficios el usuario siempre puede conocer en que estado se encuentran las puertas y ventanas de su domicilio, pudiendo corregir los estados de las mismos si así lo desea.

En el caso de las puertas se orientó la implementación a interiores pudiendo abarcar áreas del hogar como cocina, dormitorio o baño. Mientras que en el caso de las ventanas por la mecánica de funcionamiento que se tiene en el domicilio, el diseño fue orientado hacia ventanas corredizas.

Mientras que los dispositivos descritos anteriormente controlan accesos hacia la vivienda con elementos mecánicos y electrónicos, el dispositivo que se instala en las lamparas es puramente electrónico y permite controlar la luminosidad en cada área de la vivienda, esto beneficia al usuario evitando movimientos que antes tenía que hacer para encender o apagar las luminarias, además que si el usuario no esta en el domicilio por medio de la interfaz puede simular presencia en la casa y así evitar que personas mal intencionadas irruman en la misma.

Las cortinas instaladas en la viviendas suelen ser dispositivos difíciles de controlar para una persona desde una silla de ruedas, ya que esta siempre están en un nivel muy alto. Una cortina adaptada para las necesidades del usuario es lo que el sistema domótico ofrece.

Gracias a la instalación de este dispositivo el usuario ya no requiere de la ayuda de otro miembro de la familia para realizar esta tarea, y simplemente puede controlar a voluntad el nivel de luminosidad que ingresa desde el exterior a la vivienda. Esta cortina cuenta con la capacidad de poder ubicarse en diferentes niveles lo cual brinda confort al usuario.

Con la instalación de este pequeño dispositivo el usuario ha podido resolver un problema cotidiano. Es usuario tiene un trabajo que le obliga ha llegar ha altas horas de la noche a su domicilio, su rutina la llegar a la vivienda era conectar una manta eléctrica que calentaba la superficie de su cama, el motivo de este habito es que el usuario tiene una mala circulación de sangre en sus piernas debido a su condición física. El tiempo de espera necesario para que la cama este a una temperatura agradable era 15 minutos, los cuales el debía esperar despierto para desconectar posteriormente la manta. Si el caso era que por el cansancio el usuario se quedaba dormido antes de desconectar la manta esta caliente demasiado la cama e interrumpe el sueño.

La ventaja de el dispositivo instalado es que la persona puede acceder desde su trabajo o desde cualquier lugar que el este a la interfaz y para previamente calentar la superficie de la cama, entonces cuando el usuario llega a su casa puede descansar de manera inmediata sin preocuparse de la desconexión.

Los materiales incluidos en los nodos descritos y sus precios unitarios pueden ser consultados en el Anexo 31.

### **6.1.2. Costo total del sistema instalado.**

Para finalizar con el análisis costo beneficio del sistema, se toma en cuenta el precio total del sistema instalado el cual asciende a los **687,99\$** dólares, frente a todas las ventajas descritas anteriormente, más el confort, la comodidad y el tiempo que el usuario se ahorra con la instalación del sistema en su casa.

Cabe destacar que el sistema domótico mas halla de beneficiar a la persona con paraplejia

también beneficia a su familia.

Como se observa, la cifra indicada a la que asciende el sistema, se eleva un poco mas que la de dos mensuales básicos en el Ecuador que se podrían amortizar para ayudar a más personas en la misma condición, por tal motivo se considera una buena inversión para una persona con este tipo de discapacidad. Inversión que no retorna ningún capital pero que si ayuda de manera importante a este tipo de personas.

Lo mas importante al finalizar la instalación del sistema es que se mejoró notablemente el autoestima del usuario beneficiado.

## **7. CAPÍTULO VII**

### **7.1. Conclusiones y recomendaciones**

### **7.2. Conclusiones**

- El sistema instalado mejora notablemente la calidad de vida del usuario, reduciendo el nivel de dependencia, el tiempo y la incomodidad en la realización de actividades cotidianas en su hogar.
- Las herramientas de hardware estudiadas conjuntamente con el trabajo en las bandas ISM, permitieron obtener una red básica de bajo tráfico para el control y recolección de información, que se puede expandir a otros campos similares a la domótica.
- Mediante algunas modificaciones en los dispositivos intervenidos, se logró un funcionamiento transparente para el usuario beneficiado y los demás miembros de la familia, demostrando que no son necesarios dispositivos especializados para la implementación del sistema.
- El proyecto se beneficia favorablemente de los módulos NRF24L01 que a pesar de no estar basados en un estándar y gracias a los esfuerzos de la comunidad de software libre, han llegado a ser parte fundamental del proyecto haciendo al sistema más accesible económicamente.
- Una manera gratuita y factible de publicar contenido a Internet es utilizar VPNs, de

esta manera se añade encriptación punto a punto y se restringe el acceso a solo un grupo de usuarios.

- Por medio de la activación de los dispositivos a través de Internet, el usuario ahorra hasta cuarenta y cinco minutos al activar la manta eléctrica de forma remota. Permitiendo que pueda utilizar este tiempo para descansar cuando llega a su hogar.

### **7.3. Recomendaciones**

- Para un mejor desarrollo de proyectos con la placa Raspberry pi y Raspbian, se recomienda mantener actualizado el sistema para corregir errores y si es necesario actualizar todo el sistema operativo por una versión mas nueva, de esta manera se obtiene un mejor rendimiento con el mismo hardware.
- Cuando se trabaja con arduino se recomienda investigar exhaustivamente sobre librerías que puedan agilizar el desarrollo de la aplicación final, esto ahorra mucho tiempo y permite enfocar los esfuerzos en la aplicación final.
- El diseño e implementación del sistema se puede mejorar notablemente con la inclusión de nuevos protocolos orientados a domótica como MQTT, que permitiría la reducción del hardware y la orientación ha Internet desde otro enfoque, abriendo la posibilidad de añadir otras tecnologías como BDD ó algún sistema de LOGS para el seguimiento de eventos y seguridad, permitiendo construir un sistema más robusto. Se

recomienda estudiar estos temas para tener una visión mas amplia y actual del llamado Internet de las cosas.

- Al trabajar con comunicaciones inalámbricas es necesario contar con el mayor número de herramientas posibles, ya que las bandas ISM son de libre uso en Ecuador, en muchos lugares se presenta saturación en el espectro electromagnético, siendo en algunos casos indispensable de la ayuda de alguno de estos instrumentos de medida y detección para evitar inconvenientes.

## Referencias bibliográficas

- Alcalde San Miguel, P. (2014). *Electrónica aplicada*. Paraninfo.
- Andrews, J. (2004). *Co-verification of Hardware and Software for ARM SoC Design*.
- Apache, O. (2013). Apache. Retrieved from <https://www.apache.org/>
- Arias, M. Á. (2015). *Aprende Programación Web con PHP y MySQL*. IT Campus Academy.
- Banzi, M. (2010). *Arduino The Documentary (2010) Spanish HD*. Retrieved from <https://vimeo.com/18390711>
- Caprile, S. R. (2009). *Equisbí* (2009th ed.).
- Dennis, A. K. (2016). *Raspberry Pi Computer Architecture Essentials*. Packt Publishing Ltd.
- D'Onofrio, L. (2013, Agosto). Fundamentos de JQuery. Retrieved from <http://librojquery.com/>
- Eben Upton, G. H. (2013). *Raspberry Pi*.
- Espejo, M. (2013). Misión Solidaria Manuela Espejo. Retrieved from <http://www.vicepresidencia.gob.ec/>
- Fernandez, E. (2013). Entorno de desarrollo para aplicaciones web dinámicas. Retrieved from <http://www.neoteo.com/entorno-de-desarrollo-para-aplicaciones-web-dinamicas>
- Forouzan, B. (2007). *Libro Transmisión de datos y redes de comunicaciones* (4ta ed.). INTERAMERICANA DE ESPAÑA.
- Gallego, J. (2013). JQueryMobile: Introducción al desarrollo web para móviles. Retrieved from <https://www.genbetadev.com/frameworks/jquerymobile-introduccion-al-desarrollo-web-para-moviles>
- Gauchat, J. D. (2011). *El gran libro de HTML5, CSS3 y Javascript*. Marcombo.
- Gidoni, T., Eran, S., & Emanuel, C. (2014). Digital predistortion using piecewise

- memory polynomial for 802.11 WiFi applications. Retrieved from <http://ieeexplore.ieee.org/document/7806082/authors>
- GNU. (2013). El sistema operativo GNU. Retrieved from <https://www.gnu.org/>
- Huidobro, J. M., & Millán, R. (2004). *Domótica: edificios inteligentes*. Creaciones Copyright.
- Jack J Purdum. (2012). *Beginning C for Arduino: Learn C Programming for the Arduino*.
- Mezzanotte, J. (2006). *Cómo Funcionan las máquinas simples*. Gareth Stevens.
- Michael Margolis. (2014). *Arduino Cookbook*. O'Reilly Media.
- Mott, R. L. (2006). *Diseño de elementos de máquinas*. Pearson Educación.
- Norbom, H. (2013). *Bootloader Source Code for Atmega328P Using Stk500 for Debian Linux: Including Makefile and Test Program*. CreateSpace Independent Publishing Platform.
- NORDIC, S. (2014). nRF24L01Pluss\_Preliminary\_Product\_Specification\_v1\_0. Retrieved from [https://www.sparkfun.com/datasheets/Components/SMD/nRF24L01Pluss\\_Preliminary\\_Product\\_Specification\\_v1\\_0.pdf](https://www.sparkfun.com/datasheets/Components/SMD/nRF24L01Pluss_Preliminary_Product_Specification_v1_0.pdf)
- RAE. (2013). *Diccionario de la real academia de la lengua*. España. Retrieved from <http://dle.rae.es/?w=diccionario>
- Raspbian, O. (2013). Raspbian. Retrieved from <https://www.raspbian.org/>
- Sergio R. Caprile. (2012). *Desarrollo con microcontroladores ARM Cortex-M3*.
- Simon Monk. (2013). *Raspberry Pi Cookbook*. O'Reilly Media.
- Stefan, J., Xavier Passaret, & Daniel Vázquez. (2004). *Domótica y hogar digital*. Paraninfo.
- Upton, E. (2013). Que es Raspberry Pi. Retrieved from <https://www.raspberrypi.org/>

## **Glosario.**

**GPL.-** Licencia Pública General.

**SBC.-** Single Board Computer.

**ARM.-** Advanced RISC Machine (RISC.- Reduced Instruction Set Computer)

**CPU.-** Unidad Central de Proceso.

**GPU.-** Unidad de procesamiento gráfico.

**DSP.-** Procesador Digital de Señales.

**SDRAM.-** Memoria Sincrónica Dinámica de Acceso Aleatorio.

**S.O.-** Sistema operativo.

**HTTP.-** Protocolo para la transferencia de hipertexto, permite la transferencia de información desde un servidor a un cliente para visualizar contenido en un navegador web.

**Script.-** conjunto de instrucciones en un archivo de texto para llevar a cabo una función específica.

**Atmega328.-** microcontrolador del fabricante ATMEGA en el que se basa Arduino UNO.

**Bootloader.-** es un pequeño programa previamente cargado en el microcontrolador y que se ejecuta inmediatamente luego de energizarlo, permite cargar programas sin un grabador de microcontroladores.

**PWM.-** Modulador por ancho de pulso.

**SRAM.-** Static Random Access Memory, Memoria Estática de Acceso Aleatorio.

**EEPROM.-** Electrically Erasable Programmable Read-Only Memory (memoria programable y borrrable eléctricamente).

**PULL-UP.-** Configuración de una resistencia para suministrar un estado lógico definido.

**I/O.-** Entrada o salida digital.

**SHR.-** Synchronization Header. Cabecera de sincronización.

**SFD.-** Start of Frame Delimiter. Delimitador de inicio de trama.

**PHR.-** PHY header. Encabezado de capa física.

**CSMA-CA.-** Carrier Sense Multiple Access with Collision Avoidance

**PAN.-** Network Area Personal.(Red de área personal)

**CAP.-** Contention Access Period.

**CFP.-** Contention Free Period.

**GTS.-** Guaranteed Time Slots.

**ZigBee.-** Es un protocolo de comunicación de alto nivel basado en IEEE 802.15.4, soporta ruteo y redes en malla.

**Digi.-** Es una empresa de comunicaciones especializada en sensores.

**NORDIC.-** Es una empresa de semiconductores, especializada en construcción de chips de bajo consumo.

**GFSK.-** Gaussian Frequency Shift Keying.

**FHSS.-** Frecuency Hoping Spread Spectrum.

**DSSS.-** Direct Secuence Spread Spectrum.

**VCS.-** Virtual Carrier Sense.

**RTS.-** Request To Send.

**CTS.-** Clear to send.

**SSID.-** Service Set Identifie

**Dirección MAC.-** se refiere a la dirección única del dispositivo asignada es su fabricación, no es igual acontrol de acceso al medio(MAC).

**CNT.-** Empresa nacional de telecomunicaciones, es uno de los proveedores de Internet mas grandes de Ecuador.

## **ANEXOS**

**Anexo 1.-** Documento con el set de comandos para xbee proporcionado por diggi.

**Anexo 2.-** Plano de la vivienda.

**Anexo 3.-** Áreas de la vivienda mas utilizadas.

**Anexo 4.-** Instalación y configuración de Raspbian.

**Anexo 5.-** Configuración de VPN con el servicio Hamachi de Logmein.

**Anexo 6.-** Direccionamiento de módulos Xbee.

**Anexo 7.-** Programas grabados en los nodos intermedios.

**Anexo 8.-** Diagrama electrónico de nodo intermedio.

**Anexo 9.-** Programa Arduino nodo lámpara.

**Anexo 10.-** Diagrama electrónico del nodo lámpara.

**Anexo 11.-** Programa nodo manta

**Anexo 12.-** Diagrama electrónico nodo manta.

**Anexo 13.-** Programa nodo cortina.

**Anexo 14.-** Circuito electrónico nodo cortina.

**Anexo 15.-** Diagrama electrónico nodo puerta.

**Anexo 16.-** Programa nodo puerta.

**Anexo 17.-** Programa nodo ventana.

**Anexo 18.-** Diagrama electrónico nodo cortina.

**Anexo 19.-** Programa nodo cortina.

**Anexo 20.-** Página web login.

**Anexo 21.-** Página áreas.

**Anexo 22.-** Archivos PHP para direccionamiento.

**Anexo 23.-** Archivos de la carpeta control-sala

**Anexo 24.-** Código fuente página sala.

**Anexo 25.-** Archivos de la carpeta control-cortina

**Anexo 26.-** Código fuente página cortina.

**Anexo 27.-** Archivos de las carpetas control-ventana y control-puerta

**Anexo 28.-** Código fuente páginas puerta y ventana.

**Anexo 29.-** Archivos de la carpeta control-manta.

**Anexo 30.-** Código fuente página manta.

**Anexo 31.-** Materiales y costos de cada nodo.

**Anexo 32.-** Manuales de usuario y administrador.

**Anexo 33.-** Distribución de dispositivos en el domicilio.

**Anexo 34.-** Diagrama eléctrico de las cargas aumentadas en el domicilio.

**Anexo 35.-** Fuentes de voltaje resistivas y capacitivas.

**Anexo 36.-** Hojas de datos de los elementos electrónicos utilizados.

## **Anexo 1**

**Documento con el set de comandos para xbee proporcionado por diggi.**

# 10. XBee Command Reference Tables

## Addressing

### Addressing Commands

AT Command	Name and Description	Node Type <sup>1</sup>	Parameter Range	Default
DH	<b>Destination Address High.</b> Set/Get the upper 32 bits of the 64-bit destination address. When combined with DL, it defines the 64-bit destination address for data transmission. Special definitions for DH and DL include 0x000000000000FFFF (broadcast) and 0x0000000000000000 (coordinator).	CRE	0 - 0xFFFFFFFF	0
DL	<b>Destination Address Low.</b> Set/Get the lower 32 bits of the 64-bit destination address. When combined with DH, it defines the 64-bit destination address for data transmissions. Special definitions for DH and DL include 0x000000000000FFFF (broadcast) and 0x0000000000000000 (coordinator).	CRE	0 - 0xFFFFFFFF	0xFFFF(Coordinator) 0 (Router/End Device)
MY	<b>16-bit Network Address.</b> Read the 16-bit network address of the module. A value of 0xFFFE means the module has not joined a ZigBee network	CRE	0 - 0xFFFE [read-only]	0xFFFE
MP	<b>16-bit Parent Network Address.</b> Read the 16-bit network address of the module's parent. A value of 0xFFFE means the module does not have a parent.	E	0 - 0xFFFE [read-only]	0xFFFE
NC	<b>Number of Remaining Children.</b> Read the number of end device children that can join the device. If NC returns 0, then the device cannot allow any more end device children to join.	CR	0 - MAX_CHILDREN (maximum varies)	read-only
SH	<b>Serial Number High.</b> Read the high 32 bits of the module's unique 64-bit address.	CRE	0 - 0xFFFFFFFF [read-only]	factory-set
SL	<b>Serial Number Low.</b> Read the low 32 bits of the module's unique 64-bit address.	CRE	0 - 0xFFFFFFFF [read-only]	factory-set
NI	<b>Node Identifier.</b> Stores a string identifier. The register only accepts printable ASCII data. In AT Command Mode, a string can not start with a space. A carriage return ends the command. Command will automatically end when maximum bytes for the string have been entered. This string is returned as part of the ND (Node Discover) command. This identifier is also used with the DN (Destination Node) command. In AT command mode, an ASCII comma (0x2C) cannot be used in the NI string	CRE	20-Byte printable ASCII string	ASCII space character (0x20)
SE	<b>Source Endpoint.</b> Set/read the ZigBee application layer source endpoint value. This value will be used as the source endpoint for all data transmissions. SE is only supported in AT firmware. The default value 0xE8 (Data endpoint) is the Digi data endpoint	CRE	0 - 0xFF	0xE8
DE	<b>Destination Endpoint.</b> Set/read Zigbee application layer destination ID value. This value will be used as the destination endpoint all data transmissions. DE is only supported in AT firmware. The default value (0xE8) is the Digi data endpoint.	CRE	0 - 0xFF	0xE8
CI	<b>Cluster Identifier.</b> Set/read Zigbee application layer cluster ID value. This value will be used as the cluster ID for all data transmissions. CI is only supported in AT firmware. The default value 0x11 (Transparent data cluster ID).	CRE	0 - 0xFFFF	0x11
NP	<b>Maximum RF Payload Bytes.</b> This value returns the maximum number of RF payload bytes that can be sent in a unicast transmission. If APS encryption is used (API transmit option bit enabled), the maximum payload size is reduced by 9 bytes. If source routing is used (AR < 0xFF), the maximum payload size is reduced further. <b>Note:</b> NP returns a hexadecimal value. (e.g. if NP returns 0x54, this is equivalent to 84 bytes)	CRE	0 - 0xFFFF	[read-only]
DD	<b>Device Type Identifier.</b> Stores a device type value. This value can be used to differentiate different XBee-based devices. Digi reserves the range 0 - 0xFFFFF. For example, Digi currently uses the following DD values to identify various ZigBee products: 0x30001 - ConnectPort X8 Gateway 0x30002 - ConnectPort X4 Gateway 0x30003 - ConnectPort X2 Gateway 0x30005 - RS-232 Adapter 0x30006 - RS-485 Adapter 0x30007 - XBee Sensor Adapter 0x30008 - Wall Router 0x3000A - Digital I/O Adapter 0x3000B - Analog I/O Adapter 0x3000C - XStick 0x3000F - Smart Plug 0x30011 - XBee Large Display 0x30012 - XBee Small Display	CRE	0 - 0xFFFFFFFF	0x30000

Node types that support the command: C=Coordinator, R=Router, E=End Device

## Networking

### Networking Commands

AT Command	Name and Description	Node Type <sup>1</sup>	Parameter Range	Default
CH	<b>Operating Channel.</b> Read the channel number used for transmitting and receiving between RF modules. Uses 802.15.4 channel numbers. A value of 0 means the device has not joined a PAN and is not operating on any channel.	CRE	<b>XBee</b> 0, 0x0B - 0x1A (Channels 11-26) <b>XBee-PRO (S2)</b> 0, 0x0B - 0x18 (Channels 11-24) <b>XBee-PRO (S2B)</b> 0, 0x0B - 0x19 (Channels 11-25)	[read-only]
ID	<b>Extended PAN ID.</b> Set/read the 64-bit extended PAN ID. If set to 0, the coordinator will select a random extended PAN ID, and the router / end device will join any extended PAN ID. Changes to ID should be written to non-volatile memory using the WR command to preserve the ID setting if a power cycle occurs.	CRE	0 - 0xFFFFFFFF	0
OP	<b>Operating Extended PAN ID.</b> Read the 64-bit extended PAN ID. The OP value reflects the operating extended PAN ID that the module is running on. If ID > 0, OP will equal ID.	CRE	0x01 - 0xFFFFFFFF	[read-only]
NH	<b>Maximum Unicast Hops.</b> Set / read the maximum hops limit. This limit sets the maximum broadcast hops value (BH) and determines the unicast timeout. The timeout is computed as (50 * NH) + 100 ms. The default unicast timeout of 1.6 seconds (NH=0x1E) is enough time for data and the acknowledgment to traverse about 8 hops.	CRE	0 - 0xFF	0x1E
BH	<b>Broadcast Hops.</b> Set/Read the maximum number of hops for each broadcast data transmission. Setting this to 0 will use the maximum number of hops.	CRE	0 - 0x1E	0
OI	<b>Operating 16-bit PAN ID.</b> Read the 16-bit PAN ID. The OI value reflects the actual 16-bit PAN ID the module is running on.	CRE	0 - 0xFFFF	[read-only]
NT	<b>Node Discovery Timeout.</b> Set/Read the node discovery timeout. When the network discovery (ND) command is issued, the NT value is included in the transmission to provide all remote devices with a response timeout. Remote devices wait a random time, less than NT, before sending their response.	CRE	0x20 - 0xFF [x 100 msec]	0x3C (60d)
NO	<b>Network Discovery options.</b> Set/Read the options value for the network discovery command. The options bitfield value can change the behavior of the ND (network discovery) command and/or change what optional values are returned in any received ND responses or API node identification frames. Options include: 0x01 = Append DD value (to ND responses or API node identification frames) 002 = Local device sends ND response frame when ND is issued.	CRE	0 - 0x03 [bitfield]	0
SC	<b>Scan Channels.</b> Set/Read the list of channels to scan. <b>Coordinator</b> - Bit field list of channels to choose from prior to starting network. <b>Router/End Device</b> - Bit field list of channels that will be scanned to find a Coordinator/Router to join. Changes to SC should be written using WR command to preserve the SC setting if a power cycle occurs. Bit (Channel): 0 (0x0B)    4 (0x0F)    8 (0x13)    12 (0x17) 1 (0x0C)    5 (0x10)    9 (0x14)    13 (0x18) 2 (0x0D)    6 (0x11)    10 (0x15)    14 (0x19) 3 (0x0E)    7 (0x12)    11 (0x16)    15 (0x1A)	CRE	<b>XBee</b> 1 - 0xFFFF [bitfield] <b>XBee-PRO (S2)</b> 1 - 0x3FFF [bitfield] (bits 14, 15 not allowed) <b>XBee-PRO (S2B)</b> 1-0x7FFF (bit 15 is not allowed)	1FFE
SD	<b>Scan Duration.</b> Set/Read the scan duration exponent. Changes to SD should be written using WR command. <b>Coordinator</b> - Duration of the Active and Energy Scans (on each channel) that are used to determine an acceptable channel and Pan ID for the Coordinator to startup on. <b>Router / End Device</b> - Duration of Active Scan (on each channel) used to locate an available Coordinator / Router to join during Association. Scan Time is measured as: (# Channels to Scan) * (2 ^ SD) * 15.36ms - The number of channels to scan is determined by the SC parameter. The XBee can scan up to 16 channels (SC = 0xFFFF). Sample Scan Duration times (13 channel scan): If SD = 0, time = 0.200 sec SD = 2, time = 0.799 sec SD = 4, time = 3.190 sec SD = 6, time = 12.780 sec <b>Note:</b> SD influences the time the MAC listens for beacons or runs an energy scan on a given channel. The SD time is not a good estimate of the router/end device joining time requirements. ZigBee joining adds additional overhead including beacon processing on each channel, sending a join request, etc. that extend the actual joining time.	CRE	0 - 7 [exponent]	3
ZS	<b>ZigBee Stack Profile.</b> Set / read the ZigBee stack profile value. This must be set the same on all devices that should join the same network.	CRE	0 - 2	0

Networking Commands

AT Command	Name and Description	Node Type <sup>1</sup>	Parameter Range	Default
NJ	<b>Node Join Time.</b> Set/Read the time that a Coordinator/Router allows nodes to join. This value can be changed at run time without requiring a Coordinator or Router to restart. The time starts once the Coordinator or Router has started. The timer is reset on power-cycle or when NJ changes.	CR	0 - 0xFF [x 1 sec]	0xFF (always allows joining)
JV	<b>Channel Verification.</b> Set/Read the channel verification parameter. If JV=1, a router will verify the coordinator is on its operating channel when joining or coming up from a power cycle. If a coordinator is not detected, the router will leave its current channel and attempt to join a new PAN. If JV=0, the router will continue operating on its current channel even if a coordinator is not detected.	R	0 - Channel verification disabled 1 - Channel verification enabled	0
NW	<b>Network Watchdog Timeout.</b> Set/read the network watchdog timeout value. If NW is set > 0, the router will monitor communication from the coordinator (or data collector) and leave the network if it cannot communicate with the coordinator for 3 NW periods. The timer is reset each time data is received from or sent to a coordinator, or if a many-to-one broadcast is received.	R	0 - 0x64FF [x 1 minute] (up to over 17 days)	0 (disabled)
JN	<b>Join Notification.</b> Set / read the join notification setting. If enabled, the module will transmit a broadcast node identification packet on power up and when joining. This action blinks the Associate LED rapidly on all devices that receive the transmission, and sends an API frame out the UART of API devices. This feature should be disabled for large networks to prevent excessive broadcasts.	RE	0 - 1	0
AR	<b>Aggregate Routing Notification.</b> Set/read time between consecutive aggregate route broadcast messages. If used, AR should be set on only one device to enable many-to-one routing to the device. Setting AR to 0 only sends one broadcast	CR	0 - 0xFF	0xFF

## Security

### Security Commands

AT Command	Name and Description	Node Type <sup>1</sup>	Parameter Range	Default
EE	<b>Encryption Enable.</b> Set/Read the encryption enable setting.	CRE	0 - Encryption disabled 1 - Encryption enabled	0
EO	<b>Encryption Options.</b> Configure options for encryption. Unused option bits should be set to 0. Options include: 0x01 - Send the security key unsecured over-the-air during joins 0x02 - Use trust center (coordinator only)	CRE	0 - 0xFF	
NK	<b>Network Encryption Key.</b> Set the 128-bit AES network encryption key. This command is write-only; NK cannot be read. If set to 0 (default), the module will select a random network key.	C	128-bit value	0
KY	<b>Link Key.</b> Set the 128-bit AES link key. This command is write only; KY cannot be read. Setting KY to 0 will cause the coordinator to transmit the network key in the clear to joining devices, and will cause joining devices to acquire the network key in the clear when joining.	CRE	128-bit value	0

## RF Interfacing

### RF Interfacing Commands

AT Command	Name and Description	Node Type <sup>1</sup>	Parameter Range	Default
PL	<b>Power Level.</b> Select/Read the power level at which the RF module transmits conducted power. For XBee-PRO (S2B) Power Level 4 is calibrated and the other power levels are approximate.	CRE	<b>XBee</b> (boost mode disabled) 0 = -8 dBm 1 = -4 dBm 2 = -2 dBm 3 = 0 dBm 4 = +2 dBm  <b>XBee-PRO (S2)</b> 4 = 17 dBm <b>XBee-PRO (S2)</b> <b>(International Variant)</b> 4 = 10dBm  <b>XBee-PRO (S2B)</b> (Boost mode enabled) 4 = 18dBm 3 = 16dBm 2 = 14dBm 1 = 12dBm 0 = 10dBm <b>XBee-PRO (S2B)</b> <b>(International Variant)</b> (Boost mode enabled) 4 = 10dBm 3 = 8dBm 2 = 6dBm 1 = 4dBm 0 = 2dBm	4
PM	<b>Power Mode.</b> Set/read the power mode of the device. Enabling boost mode will improve the receive sensitivity by 1dB and increase the transmit power by 2dB Note: Enabling boost mode on the XBee-PRO (S2) will not affect the output power. Boost mode imposes a slight increase in current draw. See section 1.2 for details.	CRE	0-1, 0= -Boost mode disabled, 1= Boost mode enabled.	1
DB	<b>Received Signal Strength.</b> This command reports the received signal strength of the last received RF data packet. The DB command only indicates the signal strength of the last hop. It does not provide an accurate quality measurement for a multihop link. DB can be set to 0 to clear it. The DB command value is measured in -dBm. For example if DB returns 0x50, then the RSSI of the last packet received was -80dBm. As of 2x6x firmware, the DB command value is also updated when an APS acknowledgment is received.	CRE	0 - 0xFF Observed range for XBee-PRO: 0x1A - 0x58 For XBee: 0x 1A - 0x5C	
PP	<b>Peak Power.</b> Read the dBm output when maximum power is selected (PL4).	CRE	0x0-0x12	[read only]

1. Node types that support the command: C = Coordinator, R = Router, E = End Device

## Serial Interfacing (I/O)

### Serial Interfacing Commands

AT Command	Name and Description	Node Type <sup>1</sup>	Parameter Range	Default
AP	API Enable. Enable API Mode. The AP command is only supported when using API firmware: 21xx (API coordinator), 23xx (API router), 29xx (API end device).	CRE	1 - 2 1 = API-enabled 2 = API-enabled (w/escaped control characters)	1
AO	<b>API Options.</b> Configure options for API. Current options select the type of receive API frame to send out the Uart for received RF data packets.	CRE	0 - Default receive API indicators enabled 1 - Explicit Rx data indicator API frame enabled (0x91) 3 - enable ZDO passthrough of ZDO requests to the UART which are not supported by the stack, as well as Simple_Desc_req, Active_EP_req, and Match_Desc_req.	0
BD	<b>Interface Data Rate.</b> Set/Read the serial interface data rate for communication between the module serial port and host. Any value above 0x07 will be interpreted as an actual baud rate. When a value above 0x07 is sent, the closest interface data rate represented by the number is stored in the BD register.	CRE	0 - 7 (standard baud rates) 0 = 1200 bps 1 = 2400 2 = 4800 3 = 9600 4 = 19200 5 = 38400 6 = 57600 7 = 115200 0x80 - 0xE1000 (non-standard rates up to 921kbps)	3
NB	<b>Serial Parity.</b> Set/Read the serial parity setting on the module.	CRE	0 = No parity 1 = Even parity 2 = Odd parity 3 = Mark parity	0
SB	<b>Stop Bits.</b> Set/read the number of stop bits for the UART. (Two stop bits are not supported if mark parity is enabled.)	CRE	0 = 1 stop bit 1 = 2 stop bits	0
RO	<b>Packetization Timeout.</b> Set/Read number of character times of inter-character silence required before packetization. Set (RO=0) to transmit characters as they arrive instead of buffering them into one RF packet The RO command is only supported when using AT firmware: 20xx (AT coordinator), 22xx (AT router), 28xx (AT end device).	CRE	0 - 0xFF [x character times]	3
D7	<b>DIO7 Configuration.</b> Select/Read options for the DIO7 line of the RF module.	CRE	0 = Disabled 1 = CTS Flow Control 3 = Digital input 4 = Digital output, low 5 = Digital output, high 6 = RS-485 transmit enable (low enable) 7 = RS-485 transmit enable (high enable)	1
D6	<b>DIO6 Configuration.</b> Configure options for the DIO6 line of the RF module.	CRE	0 = Disabled 1 = RTS flow control 3 = Digital input 4 = Digital output, low 5 = Digital output, high	0

1. Node types that support the command: C = Coordinator, R = Router, E = End Device

## I/O Commands

### I/O Commands

AT Command	Name and Description	Node Type <sup>1</sup>	Parameter Range	Default
IR	<b>IO Sample Rate.</b> Set/Read the IO sample rate to enable periodic sampling. For periodic sampling to be enabled, IR must be set to a non-zero value, and at least one module pin must have analog or digital IO functionality enabled (see D0-D8, P0-P2 commands). The sample rate is measured in milliseconds.	CRE	0, 0x32:0xFFFF (ms)	0
IC	<b>IO Digital Change Detection.</b> Set/Read the digital IO pins to monitor for changes in the IO state. IC works with the individual pin configuration commands (D0-D8, P0-P2). If a pin is enabled as a digital input/output, the IC command can be used to force an immediate IO sample transmission when the DIO state changes. IC is a bitmask that can be used to enable or disable edge detection on individual channels. Unused bits should be set to 0. Bit (IO pin): 0 (DIO0)4 (DIO4)8 (DIO8) 1 (DIO1) 5 (DIO5) 9 (DIO9) 2 (DIO2) 6 (DIO6) 10 (DIO10) 3 (DIO3) 7 (DIO7) 11 (DIO11)	CRE	: 0 - 0xFFFF	0
P0	<b>PWM0 Configuration.</b> Select/Read function for PWM0.	CRE	0 = Disabled 1 = RSSI PWM 3 - Digital input, monitored 4 - Digital output, default low 5 - Digital output, default high	1
P1	<b>DIO11 Configuration.</b> Configure options for the DIO11 line of the RF module.	CRE	0 - Unmonitored digital input 3- Digital input, monitored 4- Digital output, default low 5- Digital output, default high	0
P2	<b>DIO12 Configuration.</b> Configure options for the DIO12 line of the RF module.	CRE	0 - Unmonitored digital input 3- Digital input, monitored 4- Digital output, default low 5- Digital output, default high	0
P3	<b>DIO13 Configuration.</b> Set/Read function for DIO13. This command is not yet supported.	CRE	0, 3-5 0 – Disabled 3 – Digital input 4 – Digital output, low 5 – Digital output, high	
D0	<b>AD0/DIO0 Configuration.</b> Select/Read function for AD0/DIO0.	CRE	1 - Commissioning button enabled 2 - Analog input, single ended 3 - Digital input 4 - Digital output, low 5 - Digital output, high	1
D1	<b>AD1/DIO1 Configuration.</b> Select/Read function for AD1/DIO1.	CRE	0, 2-5 0 – Disabled 2 - Analog input, single ended 3 – Digital input 4 – Digital output, low 5 – Digital output, high	0
D2	<b>AD2/DIO2 Configuration.</b> Select/Read function for AD2/DIO2.	CRE	0, 2-5 0 – Disabled 2 - Analog input, single ended 3 – Digital input 4 – Digital output, low 5 – Digital output, high	0

## I/O Commands

AT Command	Name and Description	Node Type <sup>1</sup>	Parameter Range	Default
D3	<b>AD3/DIO3 Configuration.</b> Select/Read function for AD3/DIO3.	CRE	0, 2-5 0 – Disabled 2 - Analog input, single ended 3 – Digital input 4 – Digital output, low 5 – Digital output, high	0
D4	<b>DIO4 Configuration.</b> Select/Read function for DIO4.	CRE	0, 3-5 0 – Disabled 3 – Digital input 4 – Digital output, low 5 – Digital output, high	0
D5	<b>DIO5 Configuration.</b> Configure options for the DIO5 line of the RF module.	CRE	0 = Disabled 1 = Associated indication LED 3 = Digital input 4 = Digital output, default low 5 = Digital output, default high	1
D8	<b>DIO8 Configuration.</b> Set/Read function for DIO8. This command is not yet supported.	CRE	0, 3-5 0 – Disabled 3 – Digital input 4 – Digital output, low 5 – Digital output, high	
LT	<b>Assoc LED Blink Time.</b> Set/Read the Associate LED blink time. If the Associate LED functionality is enabled (D5 command), this value determines the on and off blink times for the LED when the module has joined a network. If LT=0, the default blink rate will be used (500ms coordinator, 250ms router/end device). For all other LT values, LT is measured in 10ms.	CRE	0, 0x0A - 0xFF (100 - 2550 ms)	0
PR	<b>Pull-up Resistor.</b> Set/read the bit field that configures the internal pull-up resistor status for the I/O lines. "1" specifies the pull-up resistor is enabled. "0" specifies no pullup.(30k pull-up resistors) Bits: 0 - DIO4 (Pin 11) 1 - AD3 / DIO3 (Pin 17) 2 - AD2 / DIO2 (Pin 18) 3 - AD1 / DIO1 (Pin 19) 4 - AD0 / DIO0 (Pin 20) 5 - RTS / DIO6 (Pin 16) 6 - DTR / Sleep Request / DIO8 (Pin 9) 7 - DIN / Config (Pin 3) 8 - Associate / DIO5 (Pin 15) 9 - On/Sleep / DIO9 (Pin 13) 10 - DIO12 (Pin 4) 11 - PWM0 / RSSI / DIO10 (Pin 6) 12 - PWM1 / DIO11 (Pin 7) 13 - CTS / DIO7 (Pin 12)	CRE	0 - 0x3FFF	0 - 0x1FFF
RP	<b>RSSI PWM Timer.</b> Time the RSSI signal will be output on the PWM after the last RF data reception or APS acknowledgment.. When RP = 0xFF, output will always be on.	CRE	0 - 0xFF [x 100 ms]	0x28 (40d)
%V	<b>Supply Voltage.</b> Reads the voltage on the Vcc pin. Scale by 1200/1024 to convert to mV units. For example, a %V reading of 0x900 (2304 decimal) represents 2700mV or 2.70V.	CRE	-0x-0xFFFF [read only]	-
V+	<b>Voltage Supply Monitoring.</b> The voltage supply threshold is set with the V+ command. If the measured supply voltage falls below or equal to this threshold, the supply voltage will be included in the IO sample set. V+ is set to 0 by default (do not include the supply voltage). Scale mV units by 1024/1200 to convert to internal units. For example, for a 2700mV threshold enter 0x900. Given the operating Vcc ranges for different platforms, and scaling by 1024/1200, the useful parameter ranges are: XBee 2100-3600 mV, 0,0x0700-0x0c00 PRO 3000-3400 mV, 0,0x0a00-0x0b55 S2B 2700-3600 mV, 0,0x0900-0x0c00	CRE	0-0xFFFF	0
TP	Reads the module temperature in Degrees Celsius. Accuracy +/- 7 degrees. 1° C = 0x0001 and -1° C = 0xFFFF. Command is only available in PRO S2B.	CRE	0x0-0xFFFF	-

## Diagnostics

### Diagnostics Commands

AT Command	Name and Description	Node Type <sup>1</sup>	Parameter Range	Default
VR	<p><b>Firmware Version.</b> Read firmware version of the module. The firmware version returns 4 hexadecimal values (2 bytes) "ABCD". Digits ABC are the main release number and D is the revision number from the main release. "B" is a variant designator.</p> <p>XBee and XBee-PRO ZB modules return: 0x2xxx versions.</p> <p>XBee and XBee-PRO ZNet modules return: 0x1xxx versions. ZNet firmware is not compatible with ZB firmware.</p>	CRE	0 - 0xFFFF [read-only]	Factory-set
HV	<p><b>Hardware Version.</b> Read the hardware version of the module. This command can be used to distinguish among different hardware platforms. The upper byte returns a value that is unique to each module type. The lower byte indicates the hardware revision.</p> <p>XBee ZB and XBee ZNet modules return the following (hexadecimal) values: 0x19xx - XBee module 0x1Axx - XBee-PRO module</p>	CRE	0 - 0xFFFF [read-only]	Factory-set
AI	<p><b>Association Indication.</b> Read information regarding last node join request:</p> <ul style="list-style-type: none"> <li>0x00 - Successfully formed or joined a network. (Coordinators form a network, routers and end devices join a network.)</li> <li>0x21 - Scan found no PANs</li> <li>0x22 - Scan found no valid PANs based on current SC and ID settings</li> <li>0x23 - Valid Coordinator or Routers found, but they are not allowing joining (NJ expired)</li> <li>0x24 - No joinable beacons were found</li> <li>0x25 - Unexpected state, node should not be attempting to join at this time</li> <li>0x27 - Node Joining attempt failed (typically due to incompatible security settings)</li> <li>0x2A - Coordinator Start attempt failed<sup>1</sup></li> <li>0x2B - Checking for an existing coordinator</li> <li>0x2C - Attempt to leave the network failed</li> <li>0xAB - Attempted to join a device that did not respond.</li> <li>0xAC - Secure join error - network security key received unsecured</li> <li>0xAD - Secure join error - network security key not received</li> <li>0xAF - Secure join error - joining device does not have the right preconfigured link key</li> <li>0xFF - Scanning for a ZigBee network (routers and end devices)</li> </ul> <p><b>Note:</b> New non-zero AI values may be added in later firmware versions. Applications should read AI until it returns 0x00, indicating a successful startup (coordinator) or join (routers and end devices)</p>	CRE	0 - 0xFF [read-only]	--

1. Node types that support the command: C = Coordinator, R = Router, E = End Device

## AT Command Options

### AT Command Options Commands

AT Command	Name and Description	Node Type <sup>1</sup>	Parameter Range	Default
CT	<b>Command Mode Timeout.</b> Set/Read the period of inactivity (no valid commands received) after which the RF module automatically exits AT Command Mode and returns to Idle Mode.	CRE	2 - 0x028F [x 100 ms]	0x64 (100d)
CN	<b>Exit Command Mode.</b> Explicitly exit the module from AT Command Mode.	CRE	--	--
GT	<b>Guard Times.</b> Set required period of silence before and after the Command Sequence Characters of the AT Command Mode Sequence (GT + CC + GT). The period of silence is used to prevent inadvertent entrance into AT Command Mode.	CRE	1 - 0x0CE4 [x 1 ms] (max of 3.3 decimal sec)	0x3E8 (1000d)
CC	<b>Command Sequence Character.</b> Set/Read the ASCII character value to be used between Guard Times of the AT Command Mode Sequence (GT + CC + GT). The AT Command Mode Sequence enters the RF module into AT Command Mode. The CC command is only supported when using AT firmware: 20xx (AT coordinator), 22xx (AT router), 28xx (AT end device).	CRE	0 - 0xFF	0x2B (+' ASCII)

1. Node types that support the command: C = Coordinator, R = Router, E = End Device

## Sleep Commands

### Sleep Commands

AT Command	Name and Description	Node Type <sup>1</sup>	Parameter Range	Default
SM	<b>Sleep Mode</b> Sets the sleep mode on the RF module. An XBee loaded with router firmware can be configured as either a router (SM set to 0) or an end device (SM > 0). Changing a device from a router to an end device (or vice versa) forces the device to leave the network and attempt to join as the new device type when changes are applied.	RE	0-Sleep disabled (router) 1-Pin sleep enabled 4-Cyclic sleep enabled 5 - Cyclic sleep, pin wake	0 - Router 4 - End Device
SN	<b>Number of Sleep Periods.</b> Sets the number of sleep periods to not assert the On/Sleep pin on wakeup if no RF data is waiting for the end device. This command allows a host application to sleep for an extended time if no RF data is present	CRE	1 - 0xFFFF	1
SP	<b>Sleep Period.</b> This value determines how long the end device will sleep at a time, up to 28 seconds. (The sleep time can effectively be extended past 28 seconds using the SN command.) On the parent, this value determines how long the parent will buffer a message for the sleeping end device. It should be set at least equal to the longest SP time of any child end device.	CRE	0x20 - 0xAF0 x 10ms (Quarter second resolution)	0x20
ST	<b>Time Before Sleep</b> Sets the time before sleep timer on an end device. The timer is reset each time serial or RF data is received. Once the timer expires, an end device may enter low power operation. Applicable for cyclic sleep end devices only.	E	1 - 0xFFFFE (x 1ms)	0x1388 (5 seconds)
SO Command	<b>Sleep Options.</b> Configure options for sleep. Unused option bits should be set to 0. Sleep options include: 0x02 - Always wake for ST time 0x04 - Sleep entire SN * SP time Sleep options should not be used for most applications. See chapter 6 for more information.	E	0 - 0xFF	0
WH	<b>Wake Host.</b> Set/Read the wake host timer value. If the wake host timer is set to a non-zero value, this timer specifies a time (in millisecond units) that the device should allow after waking from sleep before sending data out the UART or transmitting an IO sample. If serial characters are received, the WH timer is stopped immediately.	E	0 - 0xFFFF (x 1ms)	
SI	<b>Sleep Immediately.</b> See Execution Commands table below..			
PO	<b>Polling Rate.</b> Sets the polling rate for the end device.	E	0 - 0x1770 (10msec)	0x00 (100 msec)

## Execution Commands

Where most AT commands set or query register values, execution commands cause an action to be executed on the module. Execution commands are executed immediately and do not require changes to be applied.

### Execution Commands

AT Command	Name and Description	Node Type <sup>1</sup>	Parameter Range	Default
AC	<b>Apply Changes.</b> Applies changes to all command registers causing queued command register values to be applied. For example, changing the serial interface rate with the BD command will not change the UART interface rate until changes are applied with the AC command. The CN command and 0x08 API command frame also apply changes.	CRE	-	
WR	<b>Write.</b> Write parameter values to non-volatile memory so that parameter modifications persist through subsequent resets. Note: Once WR is issued, no additional characters should be sent to the module until after the "OK\r" response is received. The WR command should be used sparingly. The EM250 supports a limited number of write cycles."	CRE	--	--
RE	<b>Restore Defaults.</b> Restore module parameters to factory defaults.	CRE	--	--
FR	<b>Software Reset.</b> Reset module. Responds immediately with an OK status, and then performs a software reset about 2 seconds later.	CRE	--	--
NR	<b>Network Reset.</b> Reset network layer parameters on one or more modules within a PAN. Responds immediately with an "OK" then causes a network restart. All network configuration and routing information is consequently lost. If NR = 0: Resets network layer parameters on the node issuing the command. If NR = 1: Sends broadcast transmission to reset network layer parameters on all nodes in the PAN.	CRE	0 - 1	--
SI	<b>Sleep Immediately.</b> Cause a cyclic sleep module to sleep immediately rather than wait for the ST timer to expire.	E	-	-
CB	<b>Commissioning Pushbutton.</b> This command can be used to simulate commissioning button presses in software. The parameter value should be set to the number of button presses to be simulated. For example, sending the ATCB1 command will execute the action associated with 1 commissioning button press.	CRE		

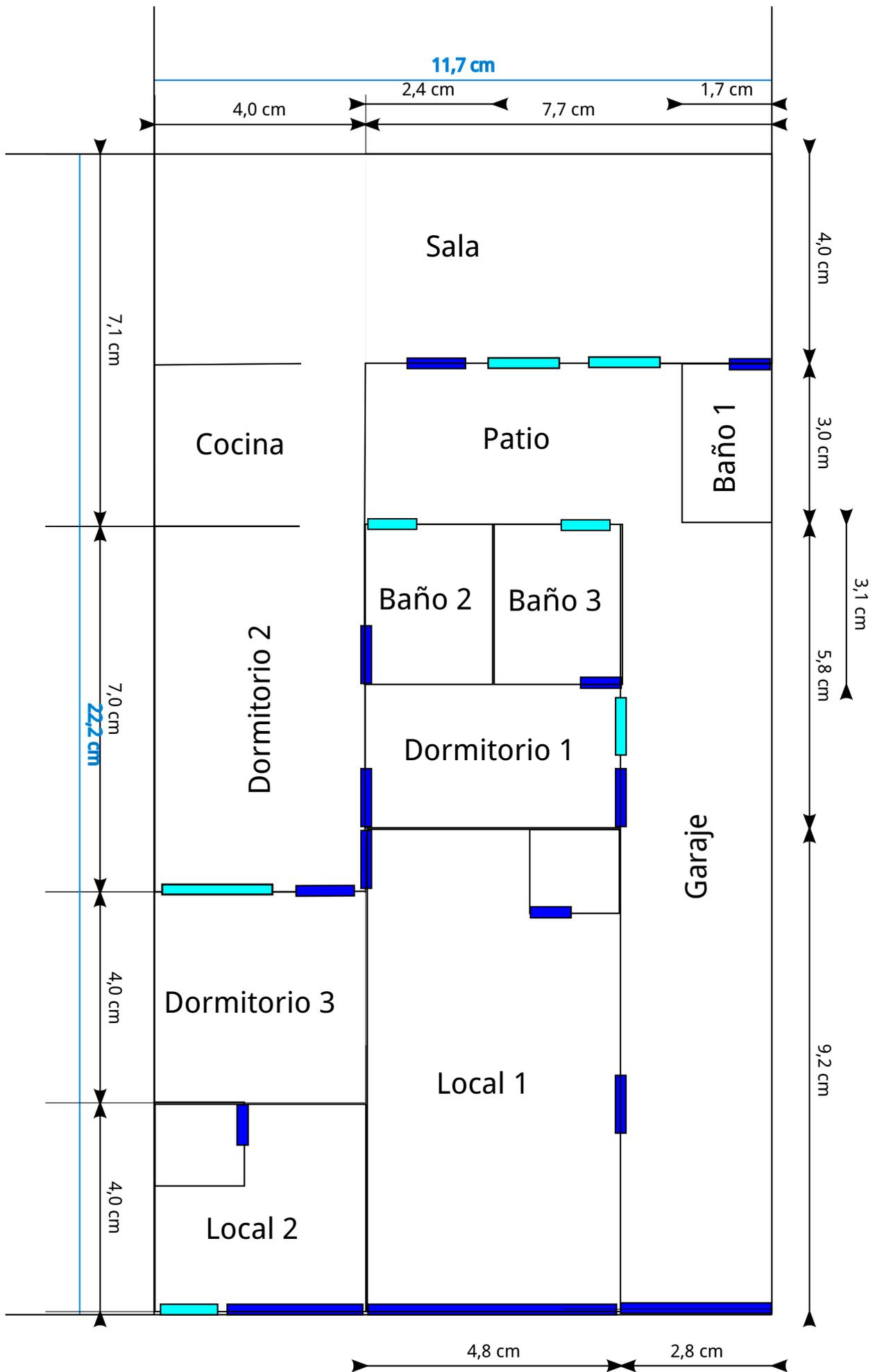
Execution Commands

AT Command	Name and Description	Node Type <sup>1</sup>	Parameter Range	Default
ND	<p><b>Node Discover.</b> Discovers and reports all RF modules found. The following information is reported for each module discovered.</p> <p>MY&lt;CR&gt;                      SH&lt;CR&gt;                      SL&lt;CR&gt;                      NI&lt;CR&gt; (Variable length)                      PARENT_NETWORK ADDRESS (2 Bytes)&lt;CR&gt;                      DEVICE_TYPE&lt;CR&gt; (1 Byte: 0=Coord, 1=Router, 2=End Device)                      STATUS&lt;CR&gt; (1 Byte: Reserved)                      PROFILE_ID&lt;CR&gt; (2 Bytes)                      MANUFACTURER_ID&lt;CR&gt; (2 Bytes)                      &lt;CR&gt;</p> <p>After (NT * 100) milliseconds, the command ends by returning a &lt;CR&gt;. ND also accepts a Node Identifier (NI) as a parameter (optional). In this case, only a module that matches the supplied identifier will respond.</p> <p>If ND is sent through the API, each response is returned as a separate AT_CMD_Response packet. The data consists of the above listed bytes without the carriage return delimiters. The NI string will end in a "0x00" null character. The radius of the ND command is set by the BH command.</p>	CRE	optional 20-Byte NI or MY value	--
DN	<p><b>Destination Node.</b> Resolves an NI (Node Identifier) string to a physical address (case-sensitive). The following events occur after the destination node is discovered:</p> <p>&lt;AT Firmware&gt;</p> <ol style="list-style-type: none"> <li>DL &amp; DH are set to the extended (64-bit) address of the module with the matching NI (Node Identifier) string.</li> <li>OK (or ERROR)r is returned.</li> <li>Command Mode is exited to allow immediate communication</li> </ol> <p>&lt;API Firmware&gt;</p> <ol style="list-style-type: none"> <li>The 16-bit network and 64-bit extended addresses are returned in an API Command Response frame.</li> </ol> <p>If there is no response from a module within (NT * 100) milliseconds or a parameter is not specified (left blank), the command is terminated and an "ERROR" message is returned. In the case of an ERROR, Command Mode is not exited. The radius of the DN command is set by the BH command.</p>	CRE	up to 20-Byte printable ASCII string	--
IS	<b>Force Sample</b> Forces a read of all enabled digital and analog input lines.	CRE	--	--
1S	<b>XBee Sensor Sample.</b> Forces a sample to be taken on an XBee Sensor device. This command can only be issued to an XBee sensor device using an API remote command.	RE	-	-

Node types that support the command: C = Coordinator, R = Router, E = End Device

## **Anexo 2**

**Plano de la vivienda.**

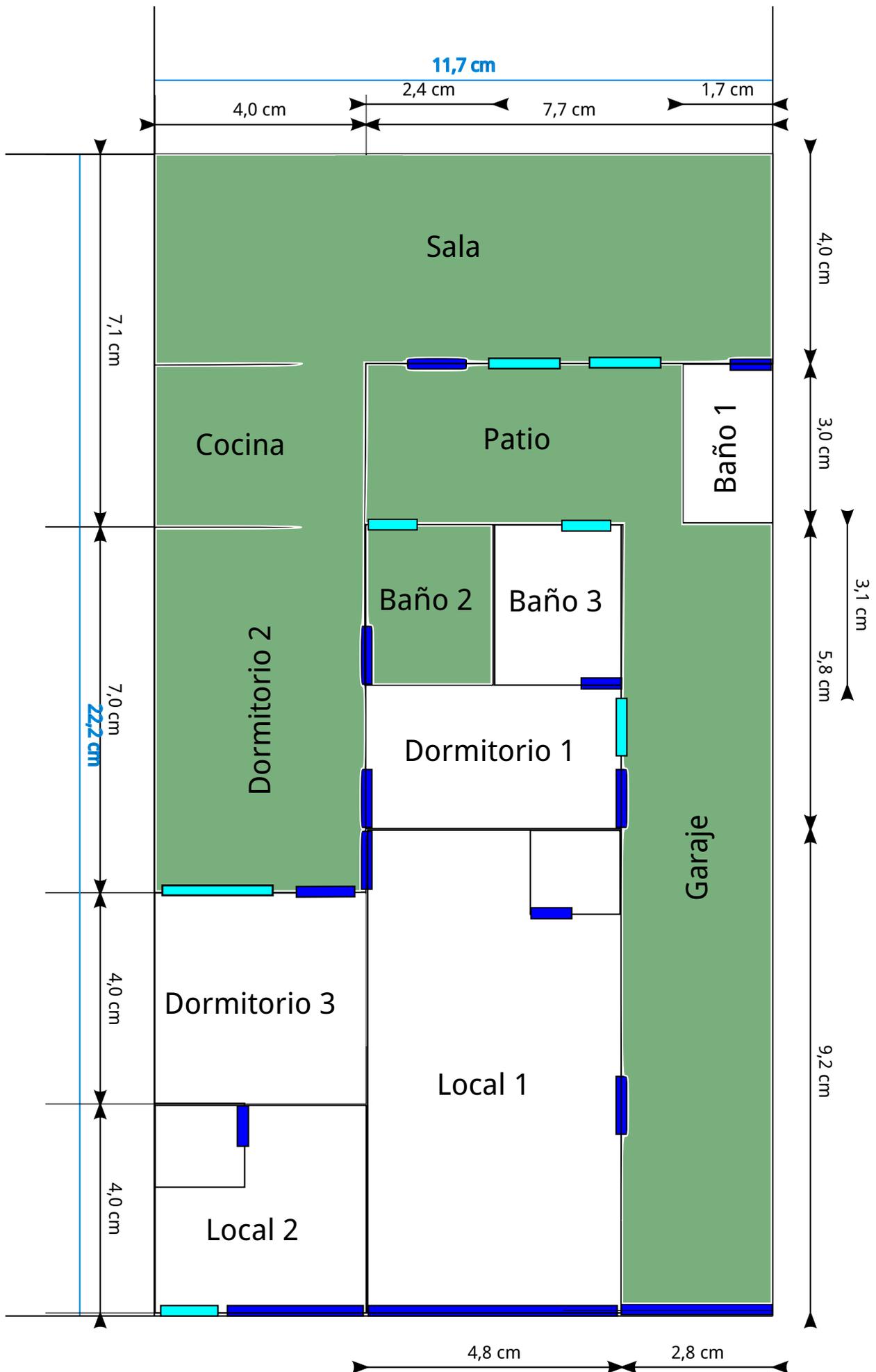


- Accesos o puertas
- Ventanas

Escala 1cm : 1m

## **Anexo 3**

**Áreas de la vivienda mas utilizadas.**



- Accesos o puertas
- Áreas más utilizadas
- Ventanas

Escala 1cm : 1m

## Anexo 4

### Instalación y configuración de Raspbian

Posterior a la descarga, la imagen del S.O debe ser grabada en una tarjeta SD de 4GB por medio de un programa que permite la transferencia de todos los archivos contenidos en la imagen a la SD, esto se lo puede realizar con el programa **dd** en sistemas GNU Linux de la siguiente manera.

- Desde una terminal y con la SD conectada se procede a identificar el dispositivo en el directorio **/dev** con el siguiente comando.

```
root@HP-Mini:/# ls /dev/sd*  
/dev/sda /dev/sda2 /dev/sda4 /dev/sda6 /dev/sdb /dev/sdb2  
/dev/sda1 /dev/sda3 /dev/sda5 /dev/sda7 /dev/sdb1
```

A terminal window showing the command 'ls /dev/sd\*' and its output. The output lists various storage devices: /dev/sda, /dev/sda1, /dev/sda2, /dev/sda3, /dev/sda4, /dev/sda5, /dev/sda6, /dev/sda7, /dev/sdb, and /dev/sdb1. Two arrows point from the text above to the first and fifth items in the list, /dev/sda and /dev/sdb.

El resultado de este comando resulta en el listado de los dispositivos de almacenamiento y sus particiones, en este caso el existen dos dispositivos de almacenamiento */dev/sda* que corresponde al disco duro primario y luego esta */dev/sdb* que corresponde a la tarjeta SD. Lo siguiente es ubicarse en el directorio donde se encuentra la imagen del sistema operativo (.img) para proceder a copiar el mismo a la SD con el siguiente comando.

```
root@HP-Mini:/# cd /home/xxx/Raspberry-img/  
root@HP-Mini:/home/xxx/Raspberry-img# ls *.img  
Raspbian.img  
root@HP-Mini:/home/xxx/Raspberry-img# dd if=Raspbian.img of=/dev/sdb
```

- Una vez copiada la imagen a la SD se prepara las conexiones de la placa para poder acceder ha ella después del primer arranque del sistema operativo. Por defecto se puede acceder a la placa de dos maneras distintas, la primera es conectando los periféricos necesarios para poder interactuar con la placa como son un monitor, teclado y mouse; mientras que la segunda forma consiste en conectar la placa a una

red Ethernet por medio de un cable y luego desde otro dispositivo se puede acceder ha ella mediante el protocolo SSH(Secure Shell) mediante una dirección IP. La opción elegida es la última por lo que se procede a insertar la SD con el S.O en la placa, luego conectar esta a la red por medio de un cable Ethernet a un dispositivo que brinde DHCP<sup>44</sup>(asignación de ip dinámica) y por último se conecta la fuente de alimentación de la placa que debe suministrar 5V y 700mA por el conector MicroUSB. Para acceder a la placa por primera vez se debe esperar al rededor de dos minutos hasta que el sistema Raspbian arranque en la misma, luego desde otro terminal que se encuentre en la misma red se procede ha escanear la red para identificar la ip de la Raspberry Pi, para esto se utiliza una herramienta llamada fping ejecutando el siguiente comando.

```
root@HP-Mini:/# fping
10:57:15 > Discovery profile: Default discovery profile
10:57:15 > Discovery class:  data-link (data-link layer)
10:57:15 > Discovery on:      192.168.1.0/28

10:57:15 > Discovery round starting.
10:57:15 > Discovery progress 25%
10:57:15 > Host is up:      192.168.1.13
           HW Address:     EC:55:F9:2C:99:2E (Hon Hai Precision)
           Hostname:       HP-Mini.local

10:57:15 > Host is up:      192.168.1.1
           HW Address:     00:26:B6:82:F1:2A (Askey Computer)

10:57:15 > Host is up:      192.168.1.10
           HW Address:     B8:27:EB:95:AA:60 (Raspberry Pi Foundation)
           Hostname:       raspberrypi
```

El resultado es un listado de los dispositivos conectados a la red con sus respectiva IP y otros datos mas que hacen referencia al mismo.

---

44 DHCP.- (Dynamic Host Configuration Protocol) permite asignar IP's bajo demanda de forma dinámica en una red.

- Con la IP de la placa Raspberry Pi se procede ha acceder ha ella por medio de SSH desde cualquier dispositivo que pueda utilizar este protocolo, para esto se ha utilizado una PC con el S.O GNU/Linux Elementary OS que tiene incorporada esta herramienta por defecto en su terminal. Para poder acceder a la placa el protocolo SSH necesita un nombre de usuario y su contraseña en la maquina remota además de su ip. Por defecto en Raspbian vienen pre-instalados dos usuarios *root* el cual tiene todos los privilegios de administración y *pi* que es un usuario normal con privilegios reducidos y clave *raspberrypi* pero es este el único que puede recibir peticiones SSH; de esta manera se accede a la placa con el siguiente comando:

```

root@HP-Mini:/# ssh pi@192.168.1.10
pi@192.168.1.10's password:
Linux raspberrypi 3.18.11+ #781 PREEMPT Tue Apr 21 18:02:18 BST 2015
Lo armv6l

```

```

The programs included with the Debian GNU/Linux system are free soft
ware;
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.
pi@raspberrypi:~$

```

siguiente es cambiar las contraseñas por seguridad ya que los credenciales utilizados hasta el momento son los que vienen por defecto y solo se deben utilizar para acceder inicialmente. Para esto es necesario acceder al modo súper usuario con el comando **sudo su** y poniendo la contraseña por defecto *raspberry*, luego se utilizan los siguientes comandos para modificar los credenciales de los usuarios.

Para cambiar las credenciales de cualquier usuario en linux se utiliza el siguiente

comando:

```
sudo passwd [nombre del usuario]
```

donde se debe reemplazar el nombre correspondiente del usuario, luego de esto se pedirá la nueva contraseña con su respectiva confirmación.

- Como súper usuario se procede ha actualizar los paquetes que se encuentran en Raspbian, esto con el fin de obtener los últimos parches de seguridad y paquetes estables. Los comandos a ejecutar son:

Para actualizar la lista de paquetes:

```
root@raspberrypi:~# apt-get update
```

Para actualizar los paquetes:

```
root@raspberrypi:~# apt-get upgrade
```

- Finalmente se asigna una ip estática a la placa con la finalidad de acceder rápidamente a ella sin tener que escanear la red para saber su dirección ip.

El archivo a editar es /etc/network/interfaces, quedando de la siguiente manera:

```
auto lo

iface lo inet loopback
#iface eth0 inet dhcp
Para #IP estatica casa
a)iface eth0 inet static
que losif address 192.168.1.10
wp netmask 255.255.255.240
if network 192.168.1.0
cambios broadcast 192.168.1.15
gateway 192.168.1.1
```

surtan efecto ese necesario reiniciar la placa.

## Anexo 5

### Configuración de VPN con el servicio Hamachi de Logmein.

- Primero hay que crear una cuenta en la pagina [www.vpn.net](http://www.vpn.net) con un correo electrónico y seguir las instrucciones.



Figura 5.1

- Luego de haber llenado el formulario, se envía un correo electrónico con un link de activación, posterior a este paso ya se puede ingresar a la pagina con los respectivos credenciales.
- El siguiente paso es crear la red a la cual se conectaran los dispositivos.

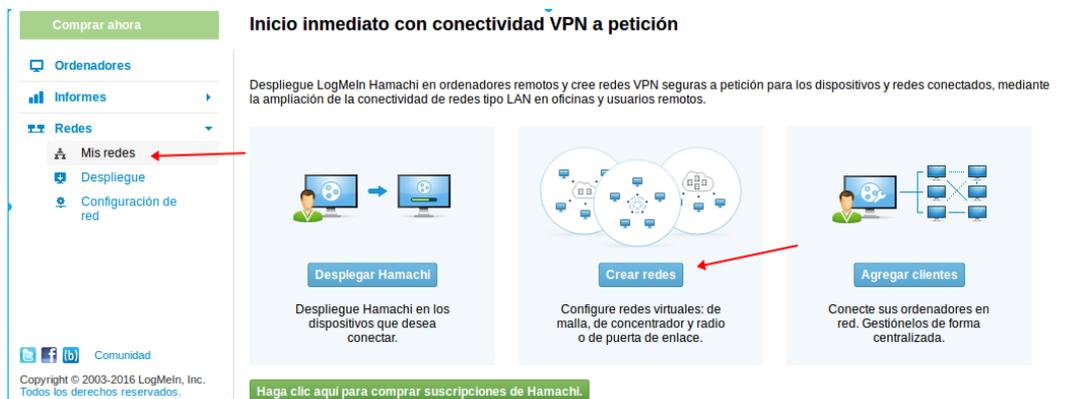


Figura 5.2

- En el apartado *Mis redes* se debe elegir la opción *Crear redes*.
- En este paso se debe llenar el formulario correspondiente a la información de la red y el tipo de red a crear. En este caso se eligió la red malla por que es la que mejor se adapta a las necesidades del proyecto.

Comprar ahora

### Añadir red (Paso 1)

**Tipo y nombre de red**

Nombre de red:

Descripción de red (opcional):

Tipo de red:

Malla  Concentrador y radio  Puerta de enlace

**Red de malla**  
En una red de malla, cada miembro está conectado a cada uno de los demás miembros fundamental llegar a cada uno de los miembros de la red.

Figura 5.3

- Lo siguiente es configurar los parámetros mismos de la red, indicando como se admitirán los nuevos equipos, una clave de seguridad y el tipo de suscripción, se elige una red de cinco equipos (red gratuita).

Comprar ahora

### Añadir red (Paso 2)

**Solicitudes de admisión**

Aceptar automáticamente

Debe ser aprobado

Los miembros solo pueden añadirse por Internet

**Network password**

Se necesita una contraseña para unirse a esta red

Contraseña de red

Confirmar contraseña

Los clientes que soliciten admisión a la red deben ingresar la contraseña. Si no configuró una contraseña, recomendamos configurar el comportamiento de Solicitud de admisión a **Debe ser aprobado** o **Los miembros solo pueden añadirse por Internet**.

**Suscripción**

Free (hasta 5 miembros) - Sin caducidad

Comprar Standard (hasta 32 miembros/red) - €44,00/año

Comprar Premium (hasta 256 miembros/red) - €179,00/año

Comprar Multi-network (hasta 256 miembros, cualquier número de redes) - €269,00/año

Figura 5.4

- Finalmente se termina el proceso de la creación de la VPN.



Figura 5.5

- En esta imagen se ve la información referente a la red recién creada. Lo mas relevante es el ID de la red y el nombre.

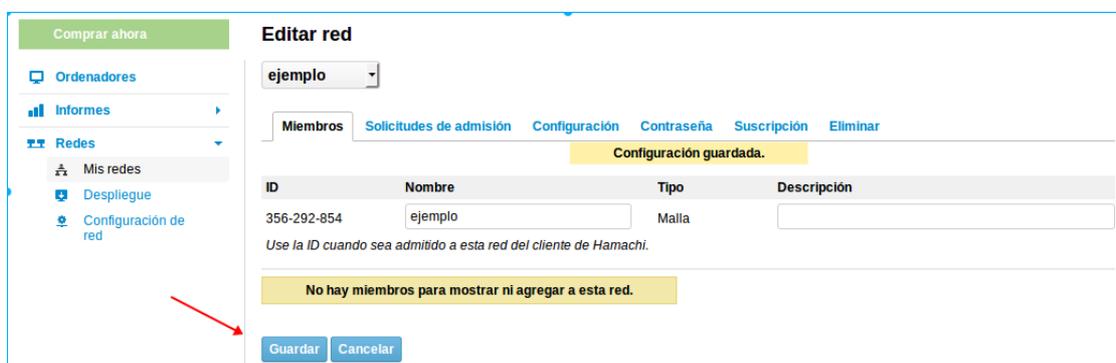


Figura 5.6

Se debe guardar la configuración.

- Ahora ya se pueden añadir los miembros de la red.

- En el apartado *Mis redes* se puede agregar los miembros de la red.



Figura 5.7

- El primer equipo en añadir es la Raspberry pi, para ello descargamos el paquete correspondiente en:

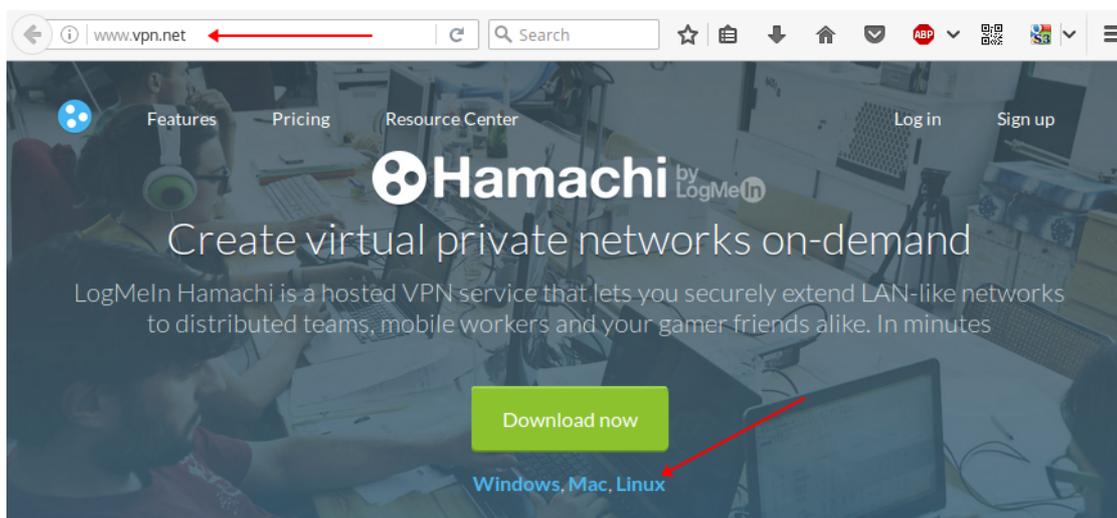


Figura 5.8

el las opciones que aparecen se debe descargar el siguiente paquete

ARM HF version (BETA):

[logmein-hamachi-2.1.0.174-1.armhf.rpm](#)

[logmein-hamachi\\_2.1.0.174-1\\_armhf.deb](#)

[logmein-hamachi-2.1.0.174-armhf.tgz](#)

- Para mayor comodidad se lo descarga directamente desde la Raspberry Pi, para esto se accede por medio de ssh como súper usuario y se descarga el paquete con el siguiente comando:

```
wget http://www.vpn.net/installers/logmein-  
hamachi_2.1.0.174-1_armhf.deb
```

Esto descarga el paquete, lo siguiente es instalar todas la dependencias para que Hamachi pueda funcionar con el siguiente comando.

```
sudo apt-get install lsb-core
```

Y ahora ya se puede instalar el paquete anteriormente descargado con:

```
sudo dpkg -i hamachi_2.1.0.174-1_armhf.deb
```

- Para que la Raspberry pi pueda conectarse a la red privada se debe configurar el paquete anteriormente instalado con estos comandos.

```
sudo hamachi login
```

```
sudo hamachi attach <email_asociado>
```

```
sudo hamachi set-nick "nombre del quipo"
```

- La conexión se reflejada en el panel web Hamachi de la siguiente forma.

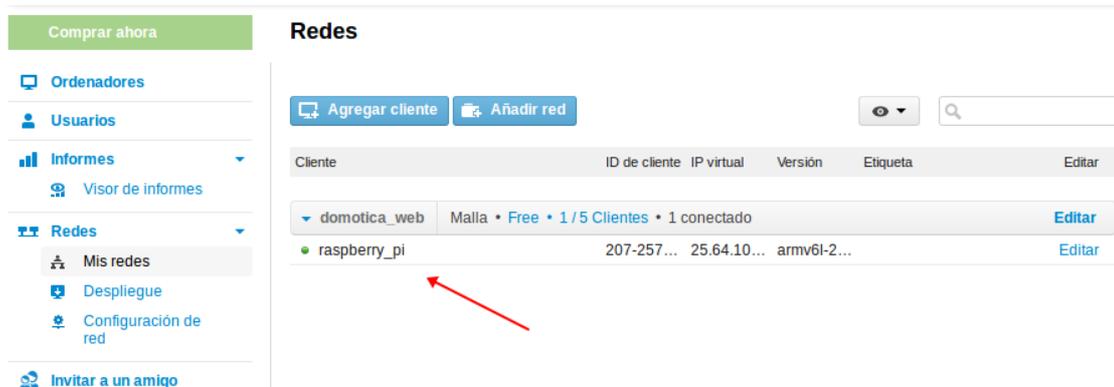


Figura 5.9

- Para que el servicio de Hamachi siempre se inicie con la Raspberry pi se debe añadir el programa a el archivo *etc/rc.local*.

```
sudo echo > /etc/init.d/logmein-hamachi start
/etc/rc.local
```

- Lo siguiente es añadir los demás equipos que el usuario desee que se puedan conectar a la aplicación desde Internet. Se presenta dos ejemplos ilustrativos de la unión de un smartphone Android y un pc con windows.
- Para conectar un Android se siguen los siguientes pasos.
  - En el apartado *Mis Redes* del panel de control Hamachi se elije la opción *Agregar cliente*.

**Redes**

Comprar ahora

- Ordenadores
- Usuarios
- Informes
  - Visor de informes
- Redes
  - Mis redes
  - Despliegue
  - Configuración de red

Agregar cliente Añadir red

Cliente	ID de cliente	IP virtual	Versión
domotica_web Malla • Free • 1 / 5 Clientes • 1 conectado			
raspberry_pi		207-257...	armv6l-2...

- Se elije la opción *Agregar cliente móvil*. Y luego click en continuar.

**Agregar cliente**

LogMeIn Hamachi

Instalar LogMeIn Hamachi en este ordenador [\(Más información\)](#)  
 Desplegar LogMeIn Hamachi en ordenadores remotos [\(Más información\)](#)  
 Añadir cliente móvil [\(Más información\)](#)

Continuar Cancelar

Figura 5.11

- Se le da un nombre al dispositivo y se debe unir a la red como se muestra.

**Agregar cliente**

Este cliente móvil se asociará a la siguiente cuenta: [account ID]

Nombre de cliente: Andorid 1

Redes:  domotica\_web [Malla] (Free)

Crear Cancelar

Figura 5.12

- En este paso se pueden utilizar algunas opciones para enviar los parámetros a configurar vía mail al usuario que se desee unir. Y también se puede configurar directamente en el teléfono. El protocolo utilizado para la conexión es PPTP.

Nombre	Redes	Configuración	<b>Móvil</b>	Eliminar
--------	-------	---------------	--------------	----------

**Enviar configuración por correo electrónico a: (Más información)**

Incluir contraseña en archivo de configuración

**Enviar a IOS** **Enviar a otro dispositivo**

**Configuración del dispositivo**

ID de usuario	207-XXXXXX
Contraseña	XXXXXXXXXX (generar nuevo)
Dirección del servidor	m.hamachi.cc
Último inicio de sesión	Aún no se ha iniciado sesión

**Mostrar ejemplo de IOS** **Mostrar ejemplo de Android**

Figura 5.13

- Los credenciales a configurar en el smartphone son los que están dentro del cuadrado rojo.
- En el smartphone del usuario se accede a configuraciones.

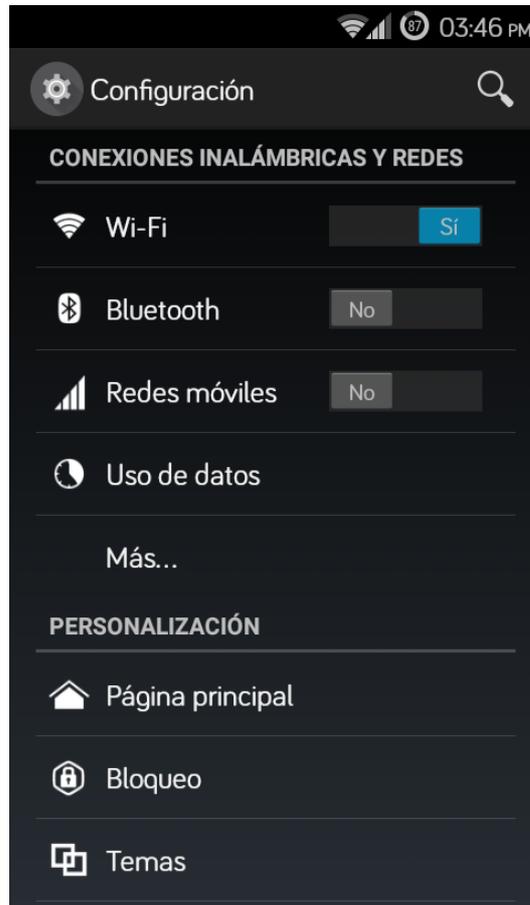


Figura 5.14

Luego se accede a la opción *Más...*

- Lo siguiente es acceder a el apartado *VPN*.



Figura 5.15

- La configuración debe quedar así.



Figura 5.16

- Se guardan los cambios y se procede a conectar a la VPN.

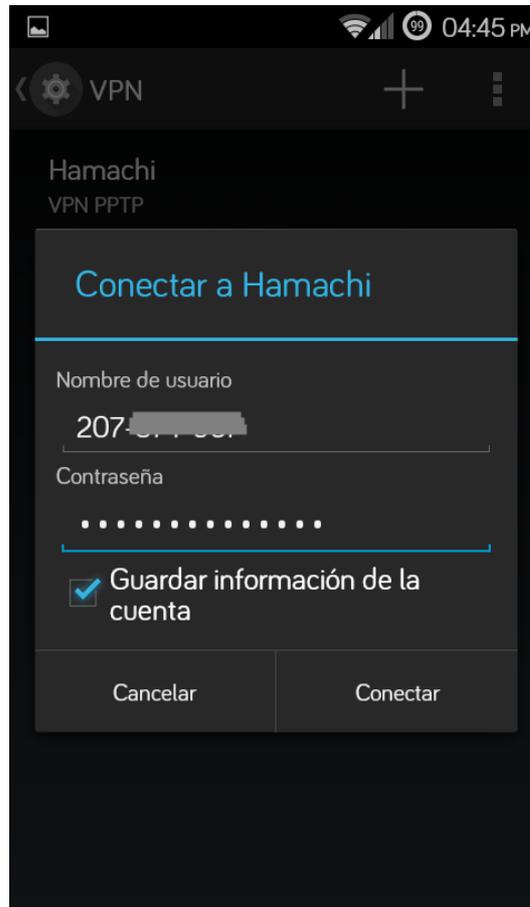


Figura 5.17

- Cuando se haya establecido la comunicación aparece una llave en la parte superior y si se accede a ella se observa.



Figura 5.18

De esta forma ya se puede acceder a la interfaz web desde cualquier conexión a Internet.

- Aquí se observa la interfaz alojada en la Raspberry pi.

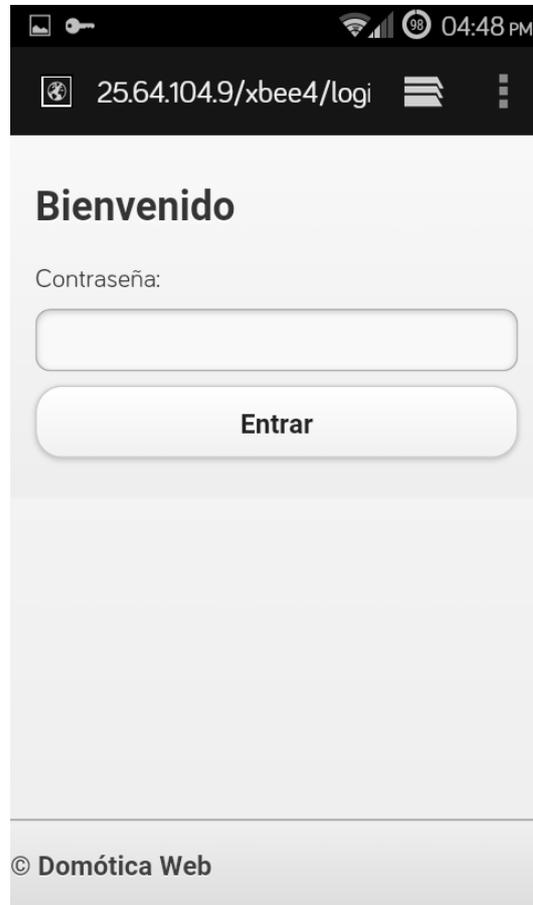


Figura 5.19

- Ahora se muestra la configuración en una pc con Windows.
  - Lo primero sera descargar el software des la pagina [www.vpn.net](http://www.vpn.net).



Figura 5.20

- Luego se procede ha instalar el programa. Una ves instalado se tiene:



Figura 5.21

- La interfaz es muy intuitiva. Para conectar se debe dar click en el botón de encendido y aparecerán las siguientes opciones.



Figura 5.22

Como se ve el servicio de Hamachi proporciona una ip y ya es posible unirse a la red privada.

- En este paso se procede a llenar los credenciales de la red a la que se desea unir.

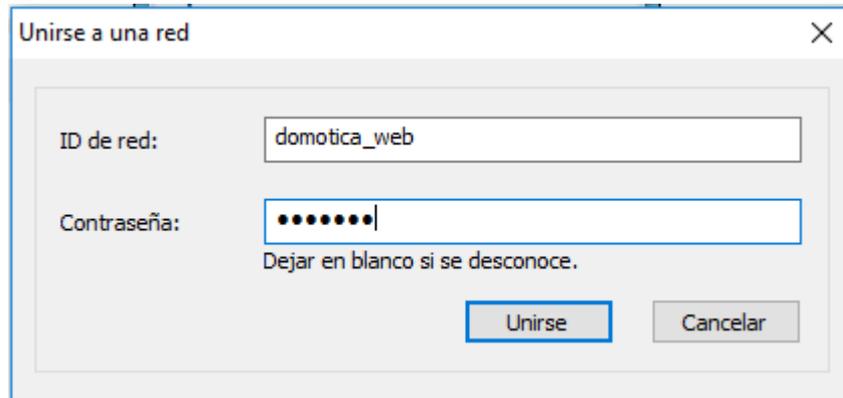


Figura 5.23

- Luego de llenar y enviar los credenciales aparecerá esta ventana.

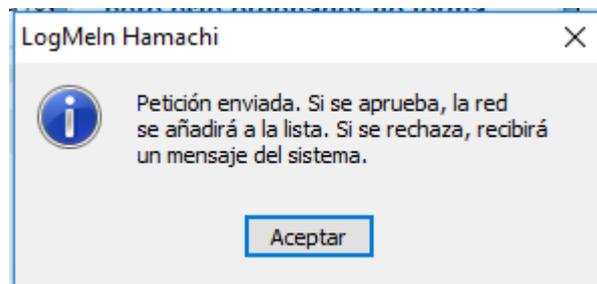


Figura 5.24

- Para que esta pc pueda ingresar a la red, se debe aceptar el equipo en el panel de Hamachi.

Comprar ahora

Ordenadores

Usuarios

Informes

Visor de informes

Redes

Mis redes

Despliegue

Configuración de red

Invitar a un amigo

### Editar red

domotica\_web

Miembros Solicitudes de admisión Configuración Contraseña Suscripción Eliminar

Aceptar	Rechazar	Nombre de cliente	ID de cliente
<input checked="" type="radio"/>	<input type="radio"/>	DESKTOP-FMTDS07	207-380-227

Guardar Cancelar

Figura 5.25

- Para comprobar que el equipo esta dentro de la red.

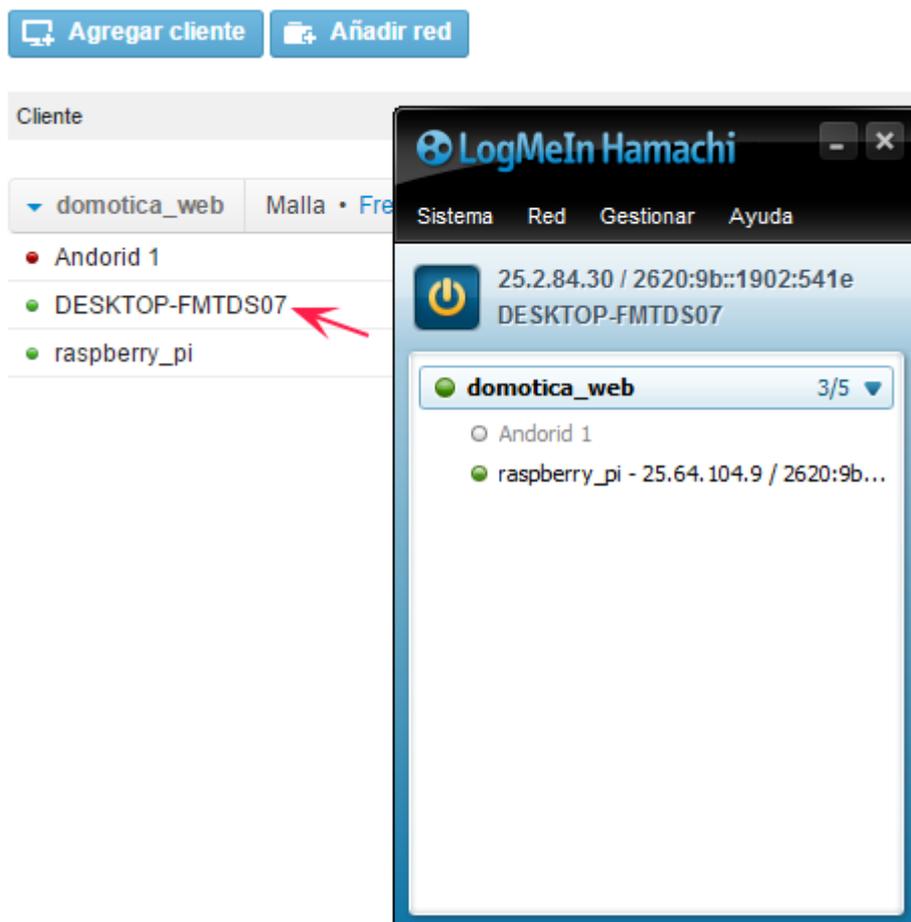


Figura 5.26

Con esto la maquina queda dentro de la red y ya es posible acceder desde cualquier lugar con una conexión a Internet.

## Anexo 6

### Configuración y direccionamiento XBEE.

Para el direccionamiento de los tres módulos Xbee se utiliza un método llamado transparente, consiste en direccionar dos módulos entre si es decir poner sus correspondientes direcciones como destino, al hacer esto los dos módulos “*emparejados*” se comportan como un cable conectado a los puertos serie de los dispositivos que se desea comunicar.

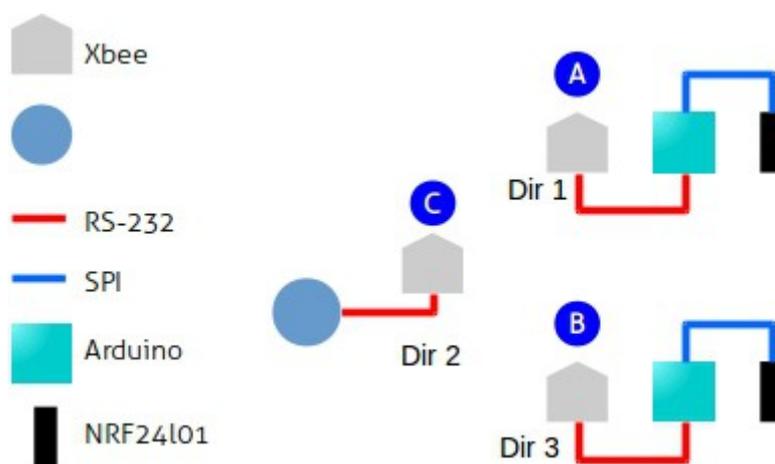


Figura 6.1

En este caso el nodo principal, con los nodos intermedios se deben “*emparejar*” a conveniencia dependiendo del destino al que se desee enviar el dato de información. En otras palabras cada vez que se desee enviar datos a un Xbee diferente la dirección destino del Xbee que está en el nodo principal debe cambiar. Mientras que los Xbee de los nodos intermedios siempre tendrán como dirección destino al Xbee del nodo principal.

La configuración inicial (antes de conectar a la red) de cada Xbee se debe hacer con algún programa que maneje el puerto serial, pero para una mayor comodidad la empresa

fabricante Diggi distribuye gratuitamente el software X-CUT que facilita la configuración.

- Con el Xbee conectado a la PC por medio de un XBEE Explorer, se inicia el programa.

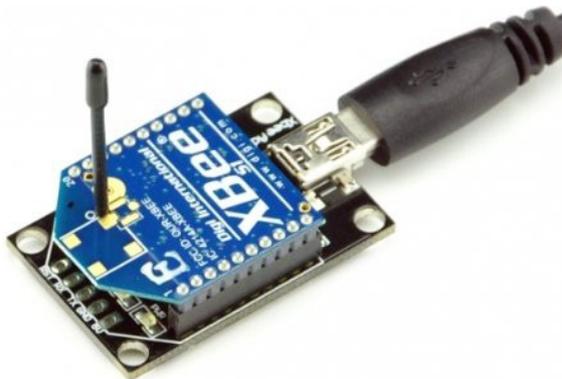


Figura 6.2

- Se obtiene la siguiente ventana.

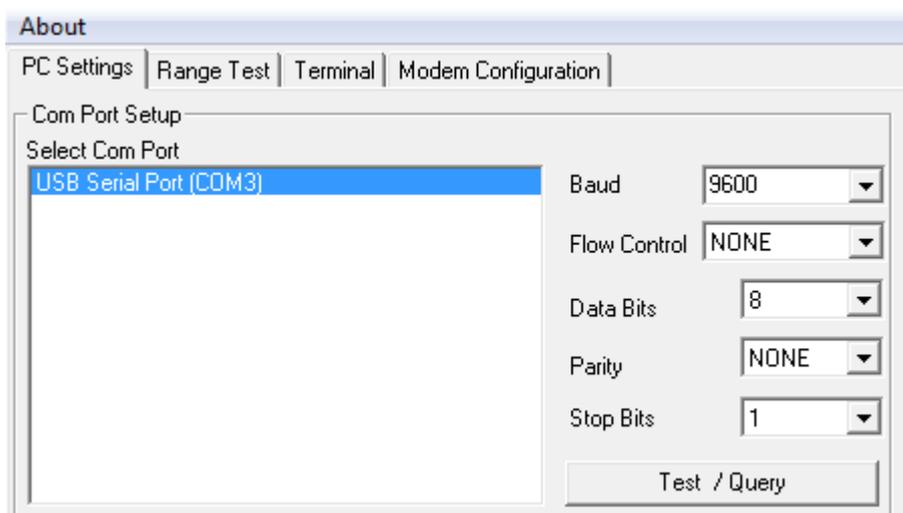


Figura 6.3

Los valores propios de la comunicación se dejan por defecto.

- Se verifica la lectura del Xbee con el botón "Test / Query".

```

Communication with modem..OK
Modem firmware version = 10CD
Modem type = XB24

Serial Number = 13A2004090F053

```

Figura 6.4

- Una vez comprobada la comunicación con el dispositivo se procede a la configuración, en la pestaña “*Modem Configuration*” se presiona el botón “*Read*” para obtener los valores por defecto del Xbee.
- Los valores para el **Xbee C** quedan de la siguiente manera:

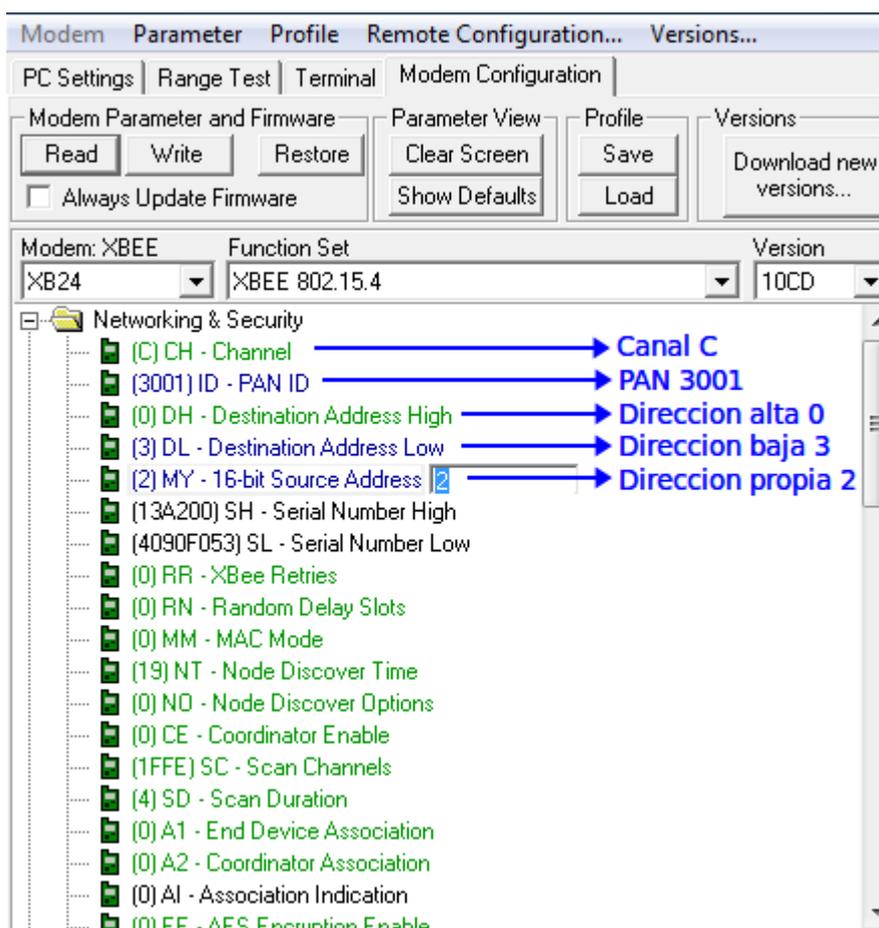


Figura 6.5

Los valores modificados corresponden a:

- Channel.- Canal de comunicación, en este caso se elige el canal C que corresponde al canal 12 (0x0C) para trabajar en el rango de frecuencias inferior 2,4075 Ghz  
central 2,4100 Ghz  
superior 2,4125 Ghz
- PAN ID.- Es el identificador de la red PAN en este caso la 3001.
- Destination Address Hihg.- Se utiliza el modo de direccionamiento corto por lo que la dirección alta del dispositivo debe ser cero.
- Destination Address Low.- Es la dirección destino que tendrá el Xbee en este caso 3, los paquetes se enviarán al Xbee que tenga esta dirección. Este parámetro y en este Xbee es el que cambiará cada vez que se quiera direccionar a un nuevo Xbee.
- Source Address.- Esta es la dirección propia del Xbee en este caso 2.

Los Xbee A y B tienen que correlacionarse con esta configuración de forma tal que coincidan los parámetros de la PANID y el canal de comunicación para que pueda existir una comunicación entre los dispositivos.

- Los valores para el **Xbee A** quedan de la siguiente manera:

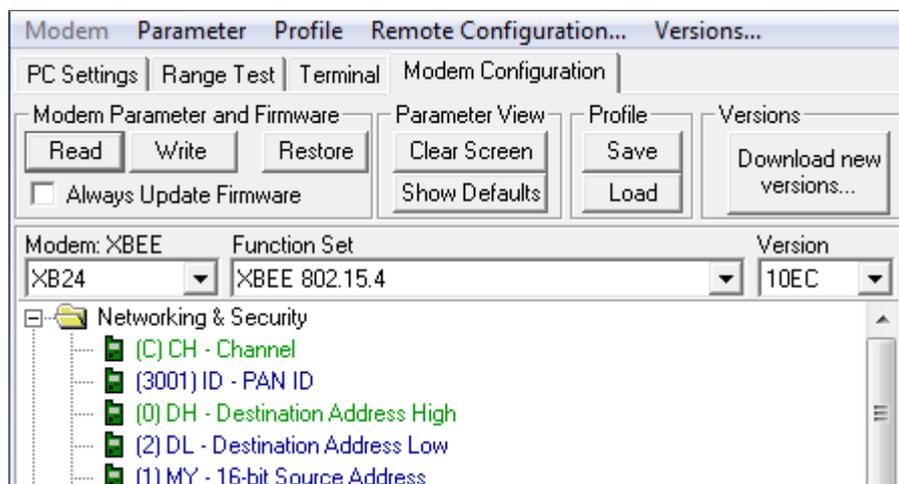


Figura 6.6

Como se observa la dirección de este Xbee es 1, y la dirección destino es 2 (el Xbee C)

- Los valores para el **Xbee B** quedan de la siguiente manera:

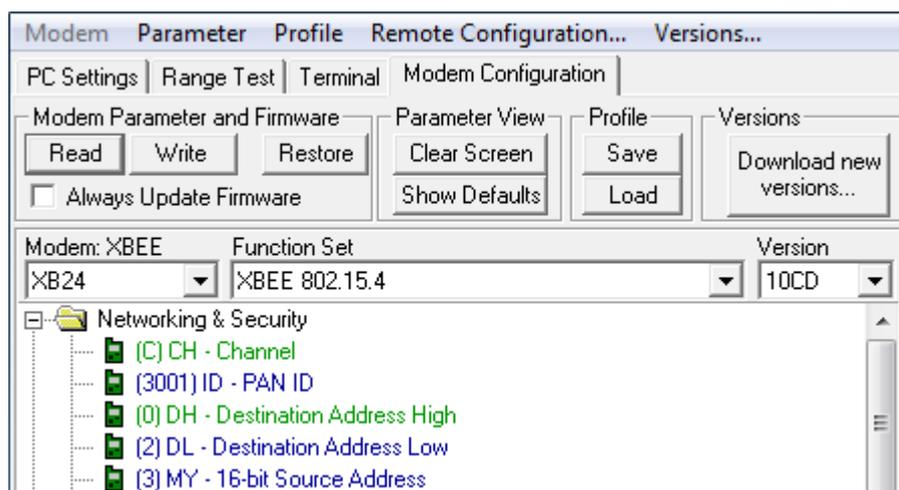


Figura 6.7

La dirección en este Xbee es 3, y la dirección destino es 2 (el Xbee C)

En resumen se tienen las siguientes direcciones.

- Nodo A con la dirección 1 y su dirección destino la 2.
- Nodo B con la dirección 3 y su dirección destino la 2.
- **Nodo C** con la dirección 2 y su dirección destino la 3.

Los nodos A y B siempre van a transmitir hacia C por lo que su configuración no cambia, pero la dirección destino del Nodo C cambia convenientemente con la siguiente secuencia de comandos AT:

Por el puerto serial:

- Se envía “+++”
- Se espera un “OK”
- Se envía la nueva dirección:
  - Para el Xbee A “atd11”
  - Para el Xbee B “atd13”
- Se espera un “OK”
- Se sale del modo de configuración con “atcn”
- Se espera un “OK”

De esta forma se logra cambiar el destino de los paquetes de información.

## Anexo 7

### Programas Arduino grabados en los nodos intermedios.

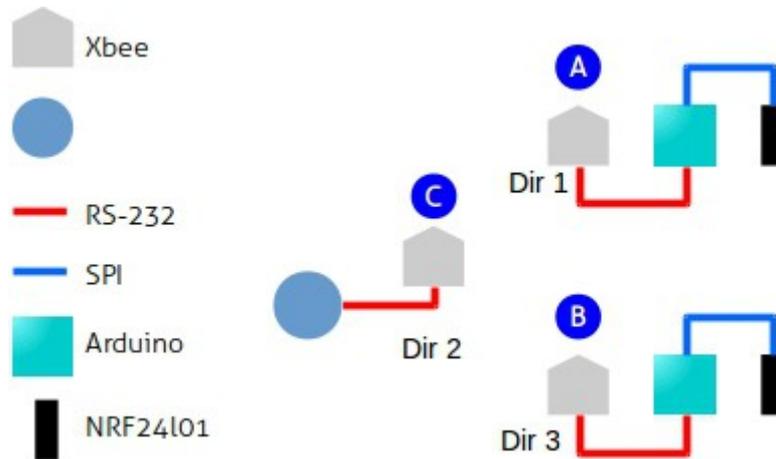


Figura 7.1

-Para el **nodo A** el programa es el siguiente:

```
//NODO_A_XBEE_A1
#include <RHReliableDatagram.h>
#include <RH_NRF24.h>
#include <SPI.h>

#define NODO_A_XBEE_A1 1
#define FOCO_CUARTO 2
#define FOCO_SALA 3//
//#define Manta 4
#define CORTINA 5//
#define VENTANA 6

//RH_NRF24 driver(CE, CS);
RH_NRF24 driver(9, 10); // Para Arduino
RHReliableDatagram manager(driver, NODO_A_XBEE_A1);

char var;
String programa="NODO_A_XBEE_A1";
int canal=90;

//Variables nuevas
char cadena[30]; //Creamos un array que almacenará los caracteres que escribiremos en la consola del PC.
Le asignamos un tope de caracteres, en este caso 30
```

```

byte posicion=0; //Variable para cambiar la posición de los caracteres del array
String texto;// variable para convertir lo que se recibe por el puerto en un texto.
String cortado;//Variable para extraer parte del texto substring

void setup() {
Serial.begin(9600);
if (!manager.init())
    Serial.println("inicializacion fallida =(");
// Defaults after init are 2.402 GHz (channel 2), 2Mbps, 0dBm
if (!driver.setChannel(canal) //Cambia el canal de transmision
    Serial.println("seleccion de canal fallida");
//Cambia la velocidad de transmision pone la ganancia en 0dBms
if (!driver.setRF(RH_NRF24::DataRate250kbps,RH_NRF24::TransmitPower0dBm))
    Serial.println("configuracion de NRF fallida");
}

uint8_t buf[RH_NRF24_MAX_MESSAGE_LEN];

void loop(){
if (Serial.available())
{
    memset(cadena, 0,sizeof(cadena));//memset borra el contenido del array "cadena" desde la posición
0 hasta el final sizeof

    //var = Serial.read();
//Guardamos la cadena que lleca en el array cadena[]
while(Serial.available(>0) //Mientras haya datos en el buffer ejecuta la función
{
    delay(5); //Poner un pequeño delay para mejorar la recepción de datos
    cadena[posicion]=Serial.read();//Lee un carácter del string "cadena" de la "posicion", luego lee
el siguiente carácter con "posicion++"
    posicion++;
}
    texto=String(cadena);//Convierte la cadena en un texto
/*
Serial.print("Llego: ");
Serial.println(cadena);
Serial.print("Longitud de cadena: ");
Serial.println(texto.length());
*/
//Cuando se envia un caracter desde el puerto serial directamente al arduino, la longitud de la
cadena es 3.
//Pero si se recibe desde el xbee la longitud de la cadena es 1.
if(texto.length()==1)//Si llega solo un caracter ponemos el valor recibido en var
{
    var=cadena[0];
}
}
}

```

```

}

if(texto.length()>1)//Si llega mas de un caracter decomponemos la cadena
{
    cortado=texto.substring(1,texto.length());
    var=' ';
    if(cadena[0]=='c')//Si la primera letra es 'c' el mensaje es para la cortina
    {
        uint8_t aux[5];
        cortado.getBytes(aux, 5);
        manager.sendtoWait(aux, sizeof(aux), CORTINA);//Envia el dato al nodo cortina
        //delay(100);
        recibedato();
    }
}

//*****
//Codigo para nodos con NRF24L01
/*    if(var=='l')

    {//Cambio de estado
        uint8_t data[] = "l";
        manager.sendtoWait(data, sizeof(data), FOCO_CUARTO);
        //delay(100);
        recibedato();
    }

    if(var=='r')

    {//Comprueba el estado
        uint8_t data[] = "r";
        manager.sendtoWait(data, sizeof(data), FOCO_CUARTO);
        //delay(100);
        recibedato();
    }
}
*/

if(var=='m')

{//Cambio de estado
    uint8_t data[] = "m";
    manager.sendtoWait(data, sizeof(data), FOCO_SALA);
    //delay(100);
    recibedato();
}
}

```

```
if(var=='s')

{
    //Comprueba el estado
    uint8_t data[] = "s";
    manager.sendtoWait(data, sizeof(data), FOCO_SALA);
    //delay(100);
    recibedato();
}

//Sentencias para la cortina.
if(var=='u')

{
    //Comprueba el estado de cortina
    uint8_t data[] = "u";
    manager.sendtoWait(data, sizeof(data), CORTINA);
    //delay(100);
    recibedato();
}

if(var=='h')

{
    //Comprueba el estado de la manta
    uint8_t data[] = "h";
    manager.sendtoWait(data, sizeof(data), CORTINA);
    //delay(100);
    recibedato();
}

if(var=='q')

{
    uint8_t data[] = "q";
    manager.sendtoWait(data, sizeof(data), VENTANA);
    //delay(100);
    recibedato();
}

if(var=='p')

{
    uint8_t data[] = "p";
    manager.sendtoWait(data, sizeof(data), VENTANA);
    //delay(100);
}
```

```

        recibedato();
    }

    if(var=='o')

    {
        uint8_t data[] = "o";
        manager.sendtoWait(data, sizeof(data), VENTANA);
        //delay(100);
        recibedato();
    }

    //Comprueba la comunicacin con el xbee
    if (var == 'x') {Serial.println(2);}

    if(var=='i')

    {
        //Imprime informacion del nodo

        informacion();
    }

    posicion=0;//Ponemos la posicion a 0
}

}

void recibedato(){
    uint8_t len = sizeof(buf);
    uint8_t from;
    if (manager.recvfromAckTimeout(buf, &len, 2000, &from))
    {
        //Serial.print("Respuesta desde : 0x");
        //Serial.print(from, HEX);
        //Serial.print(": ");
        Serial.println((char*)buf);
    }
    else
    {
        Serial.println("3");//Significa que no ha una respuesta
    }
}
}

```

```

void informacion()
{
  Serial.println("*****");
  Serial.println("NODO: NODO_A_XBEE_A1");
  Serial.print("Nombre de programa: ");
  Serial.println(programa);
  Serial.print("Canal utilizado: ");
  Serial.println(canal);
  Serial.println("Velocidad NRF24L01 250kbps");
  Serial.println("Potencia NRF24L01 0bBm");
  Serial.println("Elementos de red");
  /* Serial.println("Nodo: FOCO_CUARTO <----> NODO_A_XBEE_A1");
  Serial.println(" -Interactue con:");
  Serial.println(" -Letra 'l' <----> cambia estado");
  Serial.println(" -Letra 'r' <----> reporta el estado");*/
  // Serial.println(" ");
  Serial.println("Nodo: FOCO_SALA <----> NODO_A_XBEE_A1");
  Serial.println(" -Interactue con:");
  Serial.println(" -Letra 'm' <----> cambia estado");
  Serial.println(" -Letra 's' <----> reporta el estado");
  Serial.println(" ");
  Serial.println("Nodo: CORTINA <----> NODO_A_XBEE_A1");
  Serial.println(" -Interactue con:");
  Serial.println("          u          -Devuelve la posiscion de la cortina");
  Serial.println(" -Cadena cXX donde XX es la poscion que se desea en la cortina");
  Serial.println(" ");
  Serial.println("Nodo: VENTANA <----> NODO_A_XBEE_A1");
  Serial.println("Letras para interaccion  Respuesta  significado");
  Serial.println("          q          h  Abre la ventana");
  Serial.println("          p          c  Cierra la ventana");
  Serial.println("          o          Comprueba el estado");
  Serial.println("          e  La ventana esta entreabierta");
  Serial.println("          c  La ventana esta cerrada");
  Serial.println("          a  La ventana esta abierta");
  Serial.println("          k  La ventana esta asegurada");
  Serial.println("*****");
}

```

-Para el **nodo A** el programa es el siguiente:

```
//NODO B XBEE B3 ..... canal 2
#include <RHReliableDatagram.h>
#include <RH_NRF24.h>
#include <SPI.h>

#define NODO_B_XBEE_B3 1
#define Manta 2
#define PUERTA 3
#define FOCO_AUX 4
//#define B 3

//RH_NRF24 driver(CE, CS);
RH_NRF24 driver(9, 10); // Para Arduino
RHReliableDatagram manager(driver, NODO_B_XBEE_B3);

int var;
String programa="NODO_B_XBEE_B3";
int canal=95;
void setup() {
  Serial.begin(9600);
  if (!manager.init())
    Serial.println("inicializacion fallida =(");
  // Defaults after init are 2.402 GHz (channel 2), 2Mbps, 0dBm
  if (!driver.setChannel(canal)) //Cambia el canal de transmision
    Serial.println("seleccion de canal fallida");
  //Cambia la velocidad de transmision pone la ganancia en 0dBm
  if (!driver.setRF(RH_NRF24::DataRate250kbps,RH_NRF24::TransmitPower0dBm))
    Serial.println("configuracion de NRF fallida");
}

uint8_t buf[RH_NRF24_MAX_MESSAGE_LEN];

void loop(){
  if (Serial.available())
  {
    var = Serial.read();
    //*****
    //Codigo para nodos con NRF24L01
    if(var=='a')

      {//Cambio de estado
        uint8_t data[] = "c";
        manager.sendtoWait(data, sizeof(data), FOCO_AUX);
        // delay(100);
        recibedato();
      }

    if (var == 'z')
      {//Comprueba el estado
        uint8_t data[] = "w";
        manager.sendtoWait(data, sizeof(data), FOCO_AUX);
        //delay(100);
        recibedato();
      }

    //Sentencias para la manta.
    if(var=='e')

      {//Activa manata
        uint8_t data[] = "e";
        manager.sendtoWait(data, sizeof(data), Manta);
        //delay(100);
        recibedato();
      }

    if(var=='d')

      {//Comprueba el estado y retorna el tiempo faltante
        uint8_t data[] = "d";
        manager.sendtoWait(data, sizeof(data), Manta);
        //delay(100);
      }
  }
}
```

```

        recibedato();
    }

    if(var=='f')
    {
        //Desactiva la manta en caso de error
        uint8_t data[] = "f";
        manager.sendtoWait(data, sizeof(data), Manta);
        //delay(100);
        recibedato();
    }

    if(var=='g')
    {
        //Comprueba el estado de la manta
        uint8_t data[] = "g";
        manager.sendtoWait(data, sizeof(data), Manta);
        //delay(100);
        recibedato();
    }

    //Sentencias para puerta
    if(var=='q')
    {
        //Cambio de estado
        uint8_t data[] = "q";
        manager.sendtoWait(data, sizeof(data), PUERTA);
        //delay(100);
        recibedato();
    }

    if(var=='p')
    {
        //Cambio de estado
        uint8_t data[] = "p";
        manager.sendtoWait(data, sizeof(data), PUERTA);
        //delay(100);
        recibedato();
    }

    if(var=='o')
    {
        //Cambio de estado
        uint8_t data[] = "o";
        manager.sendtoWait(data, sizeof(data), PUERTA);
        //delay(100);
        recibedato();
    }

    //Comprueba la comunicacin con el xbee
    if (var == 'x') {Serial.println(2);}

    if(var=='i')
    {
        //Imprime informacion del nodo

        informacion();
    }
}

}

void recibedato(){
    uint8_t len = sizeof(buf);
    uint8_t from;
    if (manager.recvfromAckTimeout(buf, &len, 2000, &from))
    {
        //Serial.print("Respuesta desde : 0x");
        //Serial.print(from, HEX);
        //Serial.print(": ");
        Serial.println((char*)buf);
    }
    else
    {
        Serial.println("3"); //Significa que no ha una respuesta
    }
}

```

```

}

void informacion(){
  Serial.println("NODO_B_XBEE_B3");
  Serial.print("Nombre de programa: ");
  Serial.println(programa);
  Serial.print("Canal utilizado: ");
  Serial.println(canal);
  Serial.println("Velocidad NRF24L01 250kbps");
  Serial.println("Potencia NRF24L01 0bBm");
  Serial.println("Elementos de red");
  Serial.println("Nodo: Manta <----> NODO_B_XBEE_B3");
  Serial.println("Interactue con");
  Serial.println("  -Letra      Funcion          Devuelve");
  Serial.println("    e          activa          1 si se activo");
  Serial.println("    f          desactiva       0 si se desactivo");
  Serial.println("    d          tiempo          tiempo restante ms");
  Serial.println("    g          comprueba       1 si esta activa o 0 si no esta activa");
  Serial.println(" ");
  Serial.println("Nodo: Puerta <----> NODO_B_XBEE_B3");
  Serial.println("Interaccion      Respuesta      Significado");
  Serial.println("    q             h      Abre la puerta");
  Serial.println("    p             c      Cierra la puerta");
  Serial.println("    o             o      Comprueba el estado");
  Serial.println("                   e      La puerta esta entreabierta");
  Serial.println("                   c      La puerta esta cerrada");
  Serial.println("                   a      La puerta esta abierta");
  Serial.println("                   k      La puerta esta asegurada");
  Serial.println(" ");
  Serial.println("Nodo: Foco_aux falta configurar <----> NODO_B_XBEE_B3");
  Serial.println("  -Interactue con");
  Serial.println("  -Letra 'c' <----> cambia estado");
  Serial.println("  -Letra 'w' <----> reprota el estado");
  Serial.println("*****");
}

```

# Anexo 8

## Diagrama electrónico nodos intermedios.

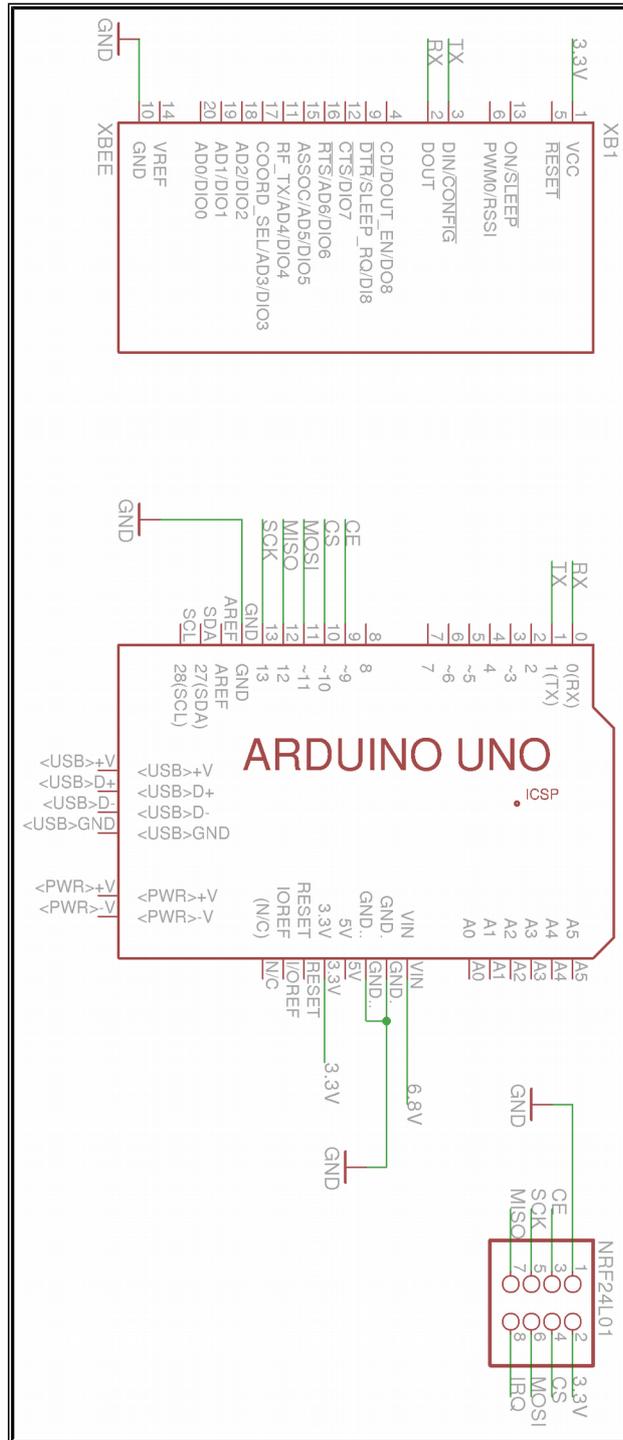


Figura 8.1

## Anexo 9

### Programa de nodo lámpara.

```
//LIBRERIAS
#include <RHReliableDatagram.h>
#include <RH_NRF24.h>
#include <SPI.h>

#define NODO_A_XBEE_A1 1
#define FOCO_SALA 3
RH_NRF24 driver(9, 10); // Pines CE y CS para el NRF24L01

// ASIGNACIÓN DE NOMBRE PARA ESTE NODO
RHReliableDatagram manager(driver, FOCO_SALA);

//Variables
const int foco =7;//pin de salida a la lámpara
const int in =8; //pin para conexión de interruptor convencional
char *dato; //vector de caracteres para la recepción de datos
int bandera;
int y;//Variable para entrada interruptor
int z;//Variable para entrada inalámbrica

String programa="FOCO_SALA";//Nombre del programa

void setup()
{
  pinMode(foco, OUTPUT);
  pinMode(in, INPUT_PULLUP);
  Serial.begin(9600);
  Serial.println("FOCO_SALA");
  Serial.print("Nombre de programa: ");
  Serial.println(programa);
  if (!manager.init()){//Inicialización NRF24L01
    Serial.println("Inicializacion fallida");
  }
  else{
    Serial.println("Inicializacion correcta.");
  }
}

if (!driver.setChannel(16)){ //Configuracion de canal
  Serial.println("Configuracion de canal fallida");
}
else{
  Serial.println("Canal utilizado: 16");
} //Velocidad y potencia de salida
if (!driver.setRF(RH_NRF24::DataRate250kbps,RH_NRF24::TransmitPower0dBm)) {
  Serial.println("Configuración de velocidad y potencia de salida fallida");
}
else{
  Serial.println("Velocidad NRF24L01 250kbps");
  Serial.println("Potencia NRF24L01 0dBm");
  Serial.println("Conectado a NODO_A_XBEE_A1");
  Serial.println("Letras");
  Serial.println("Letra 'm' <----> cambia estado");
  Serial.println("Letra 's' <----> reporta el estado");
}
}

uint8_t buf[RH_NRF24_MAX_MESSAGE_LEN];//Buffer de datos

void loop()
{
  //Comprueba el estado del interruptor
  if(digitalRead(in))
  {
    y=1;
  }
  else
  {
    y=0;
  }
  //*****

  //Sentencia para la recepción de las variables via inalámbrica
```

```

if (manager.available()) //Hay algún mensaje para este nodo??
{
  uint8_t len = sizeof(buf); // Variable que contiene la longitud del buffer
  uint8_t from; // Variable para obtener la dirección del nodo que envía el mensaje

  if (manager.recvfromAck(buf, &len, &from)) //Envía un ACK de confirmación
  {
    dato = (char*)buf; //Guarda el dato recibido en la primera posición del vector de caracteres

    if(dato[0]=='m') //Comprueba la variable
    {
      bandera=bandera+1; //Aumenta el valor de la variable "bandera" hasta 10 y luego la iguala a cero
      if(bandera==10)
      {
        bandera=0;
      }

      if(bandera%2==0) // Si el número es par pone 0 en la variable z caso.
      {
        z=0;
      }
      else
      {
        z=1;
      }
      estado(); // Va a la subrutina para comprobar el estado de la lámpara.
    }

    // Si llega la variable "s" salta a la subrutina reporte() la cual devuelve el valor actual de la
    lámpara
    if(dato[0]=='s')
    {
      reporte();
    }
  }
}

if(z ^ y){ //Realiza una XOR con las variables "z" y "y" y actúa directamente en el pin donde
esta conectado el circuito de acoplamiento a AC.

  digitalWrite(foco, HIGH);
}
else
{
  digitalWrite(foco, LOW);
}
}

//Envía el estado de la lámpara al nodo que pregunto

int estado(){
int lectural= digitalRead(foco);
String str;
uint8_t aux[4];
str=String(lectural);
str.getBytes(aux, 4);
manager.sendtoWait(aux, sizeof(aux), 1);
}

//Envía el reporte del estado de la lámpara al nodo que pregunto
int reporte(){
int lectural=digitalRead(foco);
String str;
uint8_t aux[4];
str=String(lectural);
str.getBytes(aux, 4);
manager.sendtoWait(aux, sizeof(aux), 1);
}

```

# Anexo 10

## Diagrama electrónico del nodo lámpara.

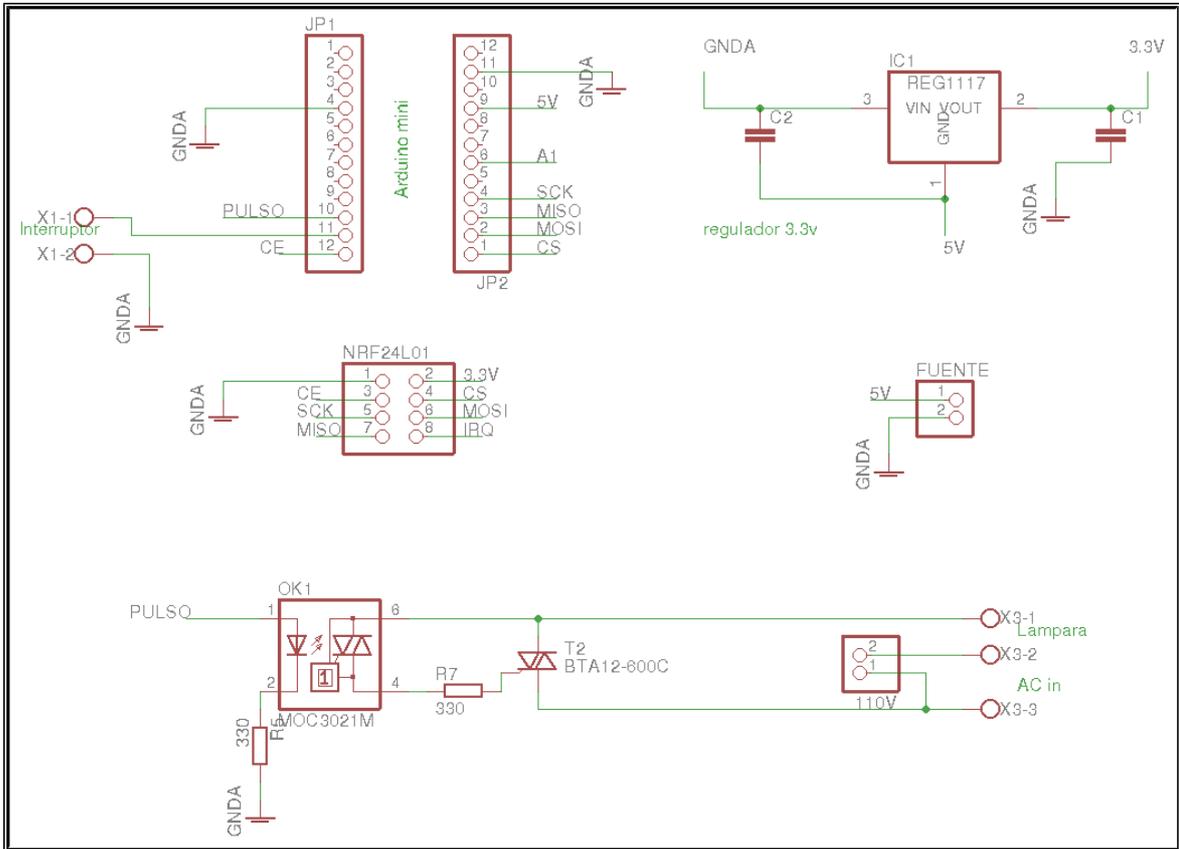


Figura 10.1

## PCB

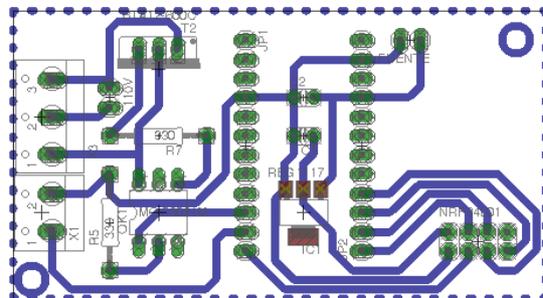


Figura 10.2

# Anexo 11

## Programa nodo manata.

```
//MANTA
//Programa para Manta Termoelectrica
/*
Indicaciones.
-Nodo asociado al SERVIDOR 2 (XBEE B3)
-Letras para interaccion desde la Raspberry Pi

Letra      Funcion              Devuelve
e          activa              1 si se activo
f          desactiva             0 si se desactivo
d          tiempo              tiempo restante ms
g          comprueba           1 si esta activa o 0 si no esta activa
*/

#include <RHReliableDatagram.h>
#include <RH_NRF24.h>
#include <SPI.h>
/*****
#define NODO_B_XBEE_B3 1
#define Manta 2
//Pines (CE, CS) para NRF24L01
RH_NRF24 driver(9, 10);

RHReliableDatagram manager(driver, Manta); //Identifica ha este nodo como el nodo MANTA
/*****
String programa="MANTA"; //Nombre del programa
const int pin_led = 8; //Pin ha donde se conecta el opto-acoplador
const int led_indicador = 7; //Pin indicador
const int botonPin1 = 6; //Boton para activar la manata manualmente
const int botonPin2 = 5; //Boton para desactivar la manata manualmente
long millis_anteriores = 0; // Variable para almacenar milisegundos
long intervalo = 900000; //tiempo en milisegundos que la manta se mantendra conectada 15 minutos
/*
Cambiar el valor segun el tiempo necesitado
60000=1min
*/
char *dato; // Variable para recibir los caracteres desde los NRF24L01
int aux; //Variable auxiliar
int aux2=0; //Variable auxiliar
int estado=0;
int canal=95; //canal utilizado
int estadoBoton1=0;
int estadoBoton2=0;
int boton_activa;
int boton_desactiva;

void setup()
{
  Serial.begin(9600);
  //Configuración de pines
  pinMode(pin_led, OUTPUT);
  pinMode(led_indicador, OUTPUT);
  pinMode(botonPin1, INPUT_PULLUP);
  pinMode(botonPin2, INPUT_PULLUP);
  if (!manager.init()){//inicialización de configuración
    Serial.println("Inicializacion fallida");
  }
  else{
    Serial.println("Inicializacion correcta.");
  }
  if (!driver.setChannel(canal)){//Cambia el canal de transmision
    Serial.println("Seleccion de canal fallido");
  }
  else{
    Serial.print("Canal utilizado: "); //Imprieme el canal utilizado
    Serial.println(canal);
  }
  //Cambia la velocidad de transmision pone la ganancia en 0dBm
  if (!driver.setRF(RH_NRF24::DataRate250kbps, RH_NRF24::TransmitPower0dBm)) {
    Serial.println("Configuracim NRF fallida");
  }
}
```

```

else{
//Imprime información del nodo
Serial.println("NODO: MANTA");
Serial.print("Nombre de programa: ");
Serial.println(programa);
Serial.println("Velocidad NRF24L01 250kbps");
Serial.println("Potencia NRF24L01 0bBm");
Serial.println("Conectado con NODO_A_XBEE_B3");
Serial.println("Letras");
Serial.println("Letra      Funcion          Devuelve");
Serial.println("e          activa            1 si se activo");
Serial.println("f          desactiva           0 si se desactivo");
Serial.println("d          tiempo             tiempo restante ms");
Serial.println("g          comprueba         1 si esta activa o 0 si no esta activa");
}
}

uint8_t buf[RH_NRF24_MAX_MESSAGE_LEN];

void loop()
{
Botones();

/*****
*****
//Lineas para temporizacion manta
unsigned long millis_actuales = millis(); //Toma el valor actual de millis
// Si se presiona el boton activar o llega la variable "e"pone la variable
millis_anteriores=millis_actuales para empezar la cuenta desde ese instante
if(aux2==1 ||boton_activa==1){
millis_anteriores = millis_actuales;
estado=1;
digitalWrite(pin_led,HIGH); //activa el triac de la manta
digitalWrite(led_indicador,HIGH);
aux2=0; // Cambia el estado de la variable para que no entre a esta condicion hasta que reciba
nuevamente la letra 'e'
}

if(aux2==0){ //Entra a esta condicion por el anterior if
/* todo empieza con las variables iguales
millis_actuales y millis_anteriores iguales por ejemplo
millis_actuales=10
millis_anteriores=10
intervalo =30
La variable millis_actuales va incrementando si valor hasta que la diferencia entre
millis_actuales y millis_anteriores sea mayor que 30,
esto hace que exacatamente transcurra el valor puesto en la variable intervalo; luego desactiva
el pin del triac y cambiamos el estado de la variable aux2 para que no entre
a ninguna de las condiciones de la manta hasta que se reciba nuevamente la letra 'e'
*/
if(millis_actuales - millis_anteriores > intervalo || boton_desactiva==1) {

millis_anteriores = millis_actuales;

digitalWrite(pin_led,LOW);
digitalWrite(led_indicador,LOW);
estado=0;
aux2=2;
}
}

/*****
*****

if (manager.available())//Hay algun mensaje?
{
uint8_t len = sizeof(buf);
uint8_t from;

if (manager.recvfromAck(buf, &len, &from)//Responda con un ACK
{
dato = (char*)buf;//Ponga el contenido del mensaje en la variable dato
//Compruebe
if(dato[0]=='e'){

```

```

    aux2=1;
    uint8_t data[] = "1";//Respuesta para la amta activa
    manager.sendtoWait(data, sizeof(data), from);
}

if(dato[0]=='d') // Cuando llega la letra 'd' envia el tiempo restante.
{
    if(estado==1)
    {
        String str;
        uint8_t aux[10];
        long a=millis_actuales - millis_anteriores;
        long b=intervalo-a;
        str=String(b);
        str.getBytes(aux, 9);
        manager.sendtoWait(aux, sizeof(aux), from);//En via el tiempo en milisegundos
    }
    else
    {
        uint8_t data[] = "0";//Si no esta activa envia cero.
        manager.sendtoWait(data, sizeof(data), from);
    }
}

if(dato[0]=='f') // Cuando llega la letra 'f' desactiva la lámpara si esta fue activada por error
{
    if(estado==1)
    {
        millis_anteriores = millis_actuales;
        digitalWrite(pin_led,LOW);
        digitalWrite(led_indicador,LOW);
        estado=0;
        aux2=2;
        uint8_t data[] = "0";//Envia cero para la manta desactivada
        manager.sendtoWait(data, sizeof(data), from);
    }
}

if(dato[0]=='g') // Cuando llega la letra 'g' envia el estado de la manta
{
    if(estado==1)
    {
        uint8_t data[] = "1";//Para manta activa
        manager.sendtoWait(data, sizeof(data), from);
    }
    else
    {
        uint8_t data[] = "0";//Para manta desactivada
        manager.sendtoWait(data, sizeof(data), from);
    }
}
}
}

//Funcion para leer los estados de los botones
void Botones(){
    estadoBoton1 = digitalRead(botonPin1);
    if (estadoBoton1 == HIGH) {
        boton_activa=0;
    }
    else {
        boton_activa=1;
    }

    estadoBoton2 = digitalRead(botonPin2);
    if (estadoBoton2 == HIGH) {
        boton_desactiva=0;
    }
    else {
        boton_desactiva=1;
    }
}
}

```

# Anexo 12

## Diagrama electrónico nodo manta.

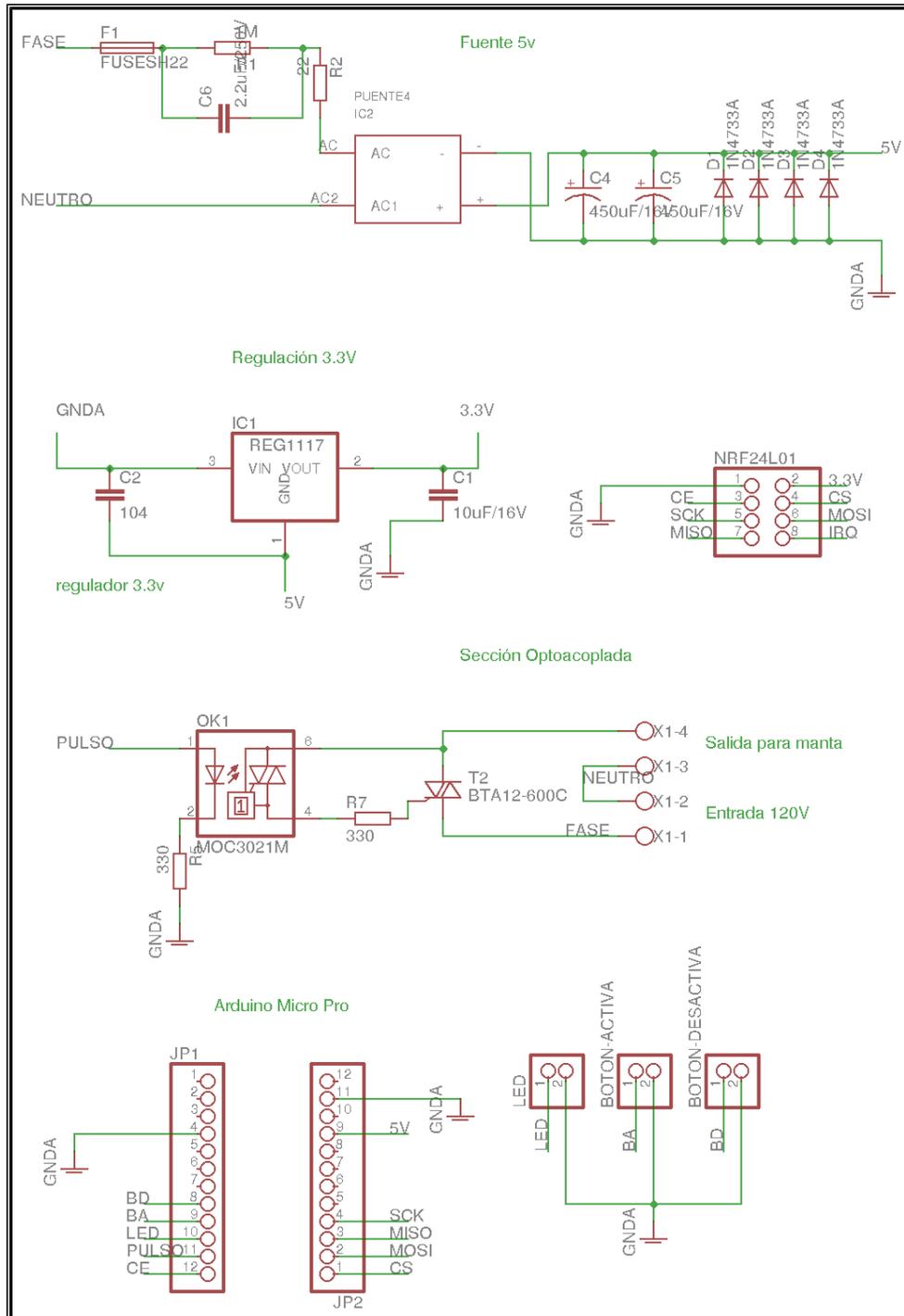


Figura 12.1

### PCB

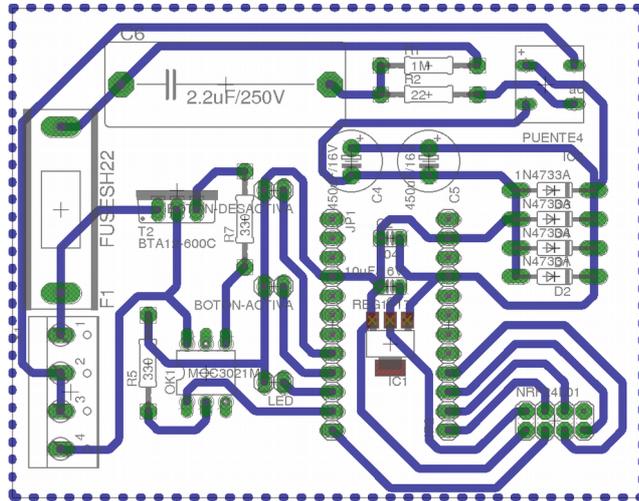


Figura 12.2

# Anexo 13

## Diagrama electrónico nodo cortina.

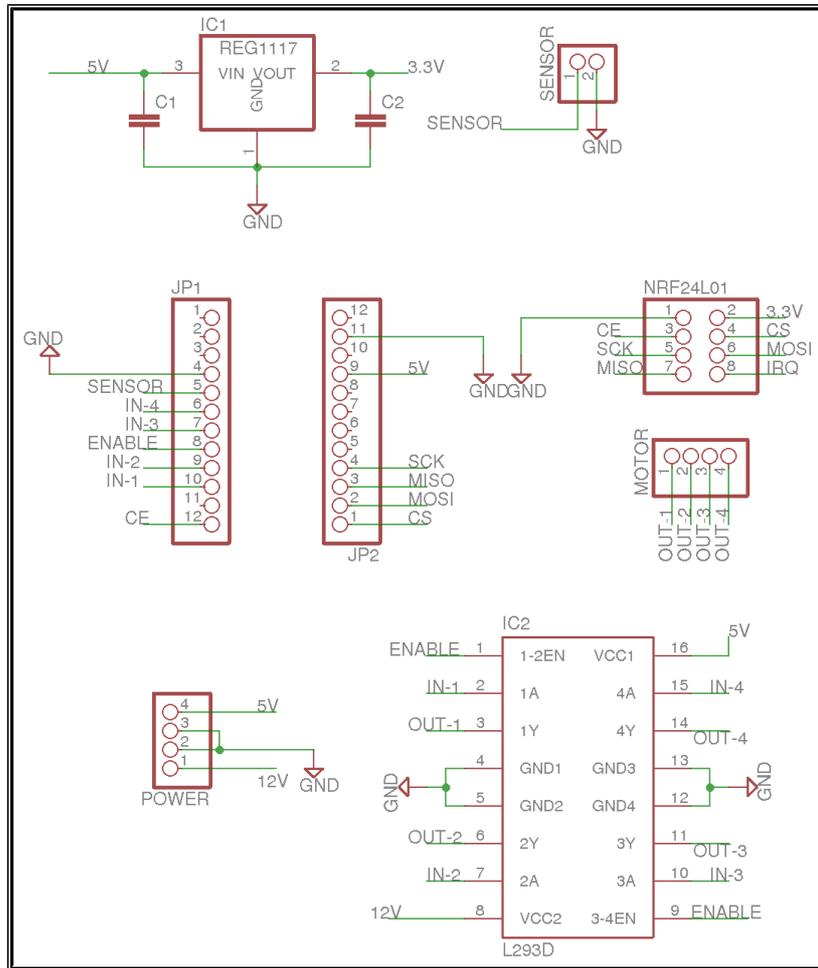


Figura 13.1

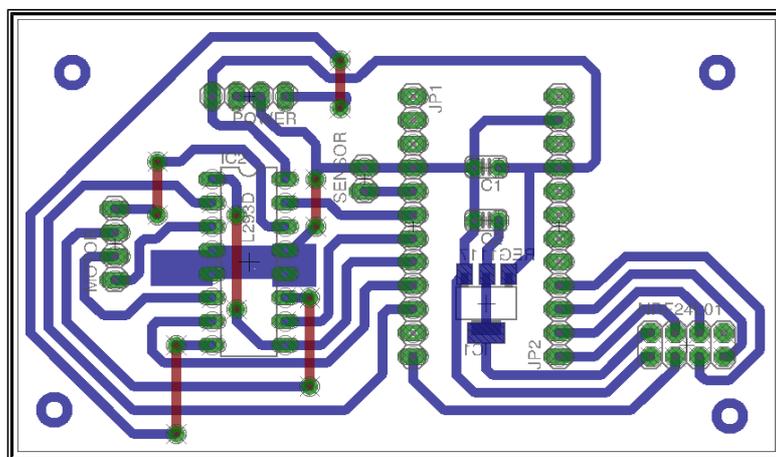


Figura 13.2

## Anexo 14

### Programa nodo cortina.

```
/*
Letra      Devuelve      Accion
u          -la posicion  -Devuelve el estado de la cortina
Cadena
cxx .- es una cadena que se recibe del nodo central del tipo (c1, c20,c33). La letra c indica que se va
ha realizar un cambio de posicion y el numero
siguiente indica la posicion deseada en la que se quiere poner la cortina; esta cadena se construye en
la Raspberry Pi corespondientemente con lo
que el usuario haya indicado en la interfaz.

//INFORMACION ACERCA DEL NODO
-El nodo se llama CORTINA y se conecta la XBEE A.
-La direccion del nodo es la numero 5.

*/
#include <Stepper.h> //Importamos la libreria para controlar motores paso a paso
#include <RHReliableDatagram.h>
#include <RH_NRF24.h>
#include <SPI.h>
#define STEPS 49//Ponemos el número de pasos que necesita para dar una vuelta.
//#define STEPS 96//Ponemos el número de pasos que necesita para dar una vuelta.

#define NODO_A_XBEE_A1 1
#define CORTINA 5

RH_NRF24 driver(9, 10); // Para Arduino
RHReliableDatagram manager(driver, CORTINA);

Stepper stepper(STEPS, 7, 6, 4, 3); //para pbc con shield L298n

const int sensor=2;
const int ENA =5;//enable
const int vuelta=STEPS;//numero de pasos para completar una vuelta
int var;
int velocidad=80;//Velocidad en rpm del motor

//Variables para recibir cadenas completas del puerto serial
char cadena[30]; //Creamos un array que almacenará los caracteres que escribiremos en la consola del PC.
Le asignamos un tope de caracteres, en este caso 30
byte posicion=0; //Variable para cambiar la posición de los caracteres del array
int valor; //Variable del valor entero

String programa="prueba con sensor y pasos cortos 1293d doble";
int canal=90;
char *dato; //vector de caracteres para la recepcin de datos

//Variables para calculo de posicion
int ultima_pos=0;
int nueva_pos;
int resultado;
int aux;

void setup()
{
  Serial.begin(9600);
  stepper.setSpeed(velocidad); //velocidad motor(200)
  pinMode(ENA, OUTPUT);
  // pinMode(ENB, OUTPUT);
  pinMode(sensor, INPUT_PULLUP);
  digitalWrite(ENA, LOW);
  //digitalWrite(ENB, LOW);
  // Serial.println("Configuracion de motor correcta");

  //Configuracion nrf24101
  if (!manager.init())//Inicializacin de la configuracion
    Serial.println("Inicializacion fallida");//En caso fallido
  if (!driver.setChannel(canal)) //Cambia el canal de transmision
    Serial.println("Seleccion de canal fallida");//En caso fallido
  //Cambia la velocidad de transmision pone la ganancia en 0dBms
  if (!driver.setRF(RH_NRF24::DataRate250kbps,RH_NRF24::TransmitPower0dBm))
    Serial.println("Configuracion de NRF fallida"); //En caso fallido

  //Impresion de informacion via puerto serial.
  Serial.println("Nodo: CORTINA ");
  Serial.print("Programa: ");
  Serial.println(programa);
  Serial.println("Conectado a NODO_A_XBEE_A1");
}
```

```

    Serial.print("Canal utilizado: ");

    Serial.println(canal);
    Serial.println("Letras para interaccion");
    Serial.println("          u          -Devuelve la posicion de la cortina");
    Serial.println("Cadena cXX donde XX es la poscion que se desea en la cortina");
}
uint8_t buf[RH_NRF24_MAX_MESSAGE_LEN];

void loop()
{
    valor=digitalRead(sensor);// Si esta variable esta en cero significa que ha habido intervencion
    manual.
    //Serial.println(valor);
    if (manager.available()) //Hay algun mensaje para este nodo??
    {
        uint8_t len = sizeof(buf);// Variable que contiene la longitud del buffer
        uint8_t from; // Variable para obtener la direccion del nodo que envia el mensaje

        if (manager.recvfromAck(buf, &len, &from)//Envia un ACK de confirmacion
        {
            dato = (char*)buf;//Guarda el dato recibido el la primera posicion del vector de caracteres

            if(dato[0]=='u')
            {
                if(valor==0)
                {
                    respuesta_manual();
                    Girar(-6);//para encerar la cortina
                }
                else
                {
                    respuesta();
                }
            }
            else
            {
                if(valor==0)
                {
                    respuesta_manual();
                    Girar(-6);//para encerar la cortina
                }

                else{

                    int n = atoi((char*)dato);//trandorma a un entero

                    nueva_pos=n;
                    resultado=nueva_pos-ultima_pos;
                    ultima_pos=nueva_pos;
                    /*Serial.print("Estado final: ");
                    Serial.println(ultima_pos);
                    Serial.print("Numero de vuletas: ");
                    Serial.println(resultado);*/
                    respuesta();
                    Girar(resultado);

                }
            }
        }
    }

    //Serial.println(dato[0]);
}

void Activar()
{
    digitalWrite(ENA, HIGH);
    //digitalWrite(ENB,HIGH);
}

void Desactivar()
{
    digitalWrite(ENA, LOW);
    //digitalWrite(ENB,LOW);
}

```

```

void Girar(int num)
{
    aux;
    aux=num*vuelta;
    Activar();
    stepper.step(aux);
    Desactivar();
    Pasos();
}

void Pasos()
{int x;
  if(aux>=0){x=1;}else{x=-1;}
  valor=digitalRead(sensor);

  while(valor==0)
  {
    stepper.setSpeed(20);
    Activar();
    stepper.step(x);
    //delay(100);
    valor=digitalRead(sensor);
    Serial.println(valor);
  }
  Desactivar();
  stepper.setSpeed(velocidad);
}

int respuesta(){
//int lectural=!digitalRead(foco);
String str;
uint8_t aux[10];
str=String(ultima_pos);
str.getBytes(aux, 9);
manager.sendtoWait(aux, sizeof(aux), 1);
// manager.sendtoWait(dato, sizeof(dato), 1);
}

int respuesta_manual(){
  ultima_pos=0;
  nueva_pos=0;
  String str;
  uint8_t aux[10];
  str=String(ultima_pos);
  str.getBytes(aux, 9);
  manager.sendtoWait(aux, sizeof(aux), 1);
}

```

# Anexo 15

## Diagrama electrónico nodo puerta y ventana.

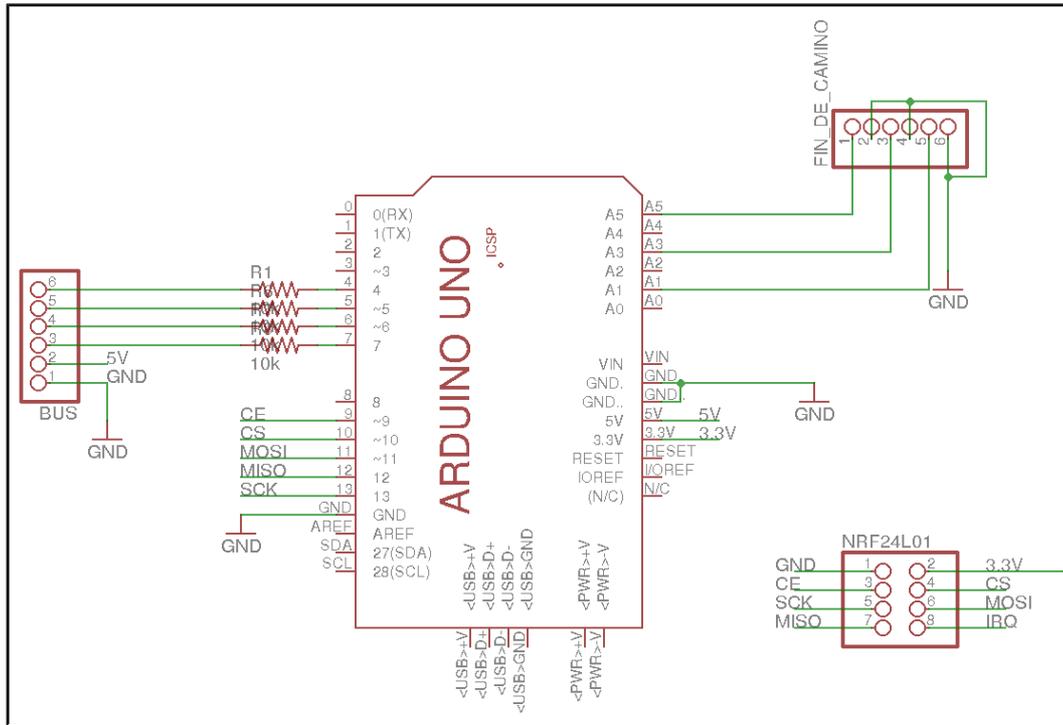


Figura 15.1

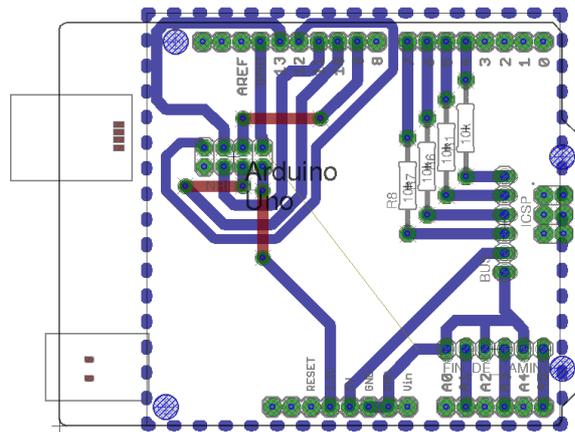


Figura 15.2

### Circuito controlado de motor

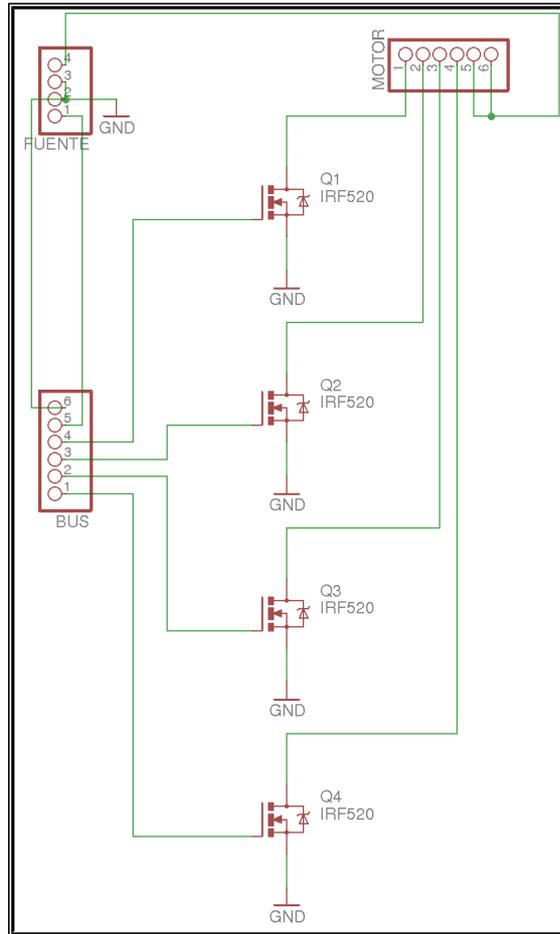


Figura 15.3

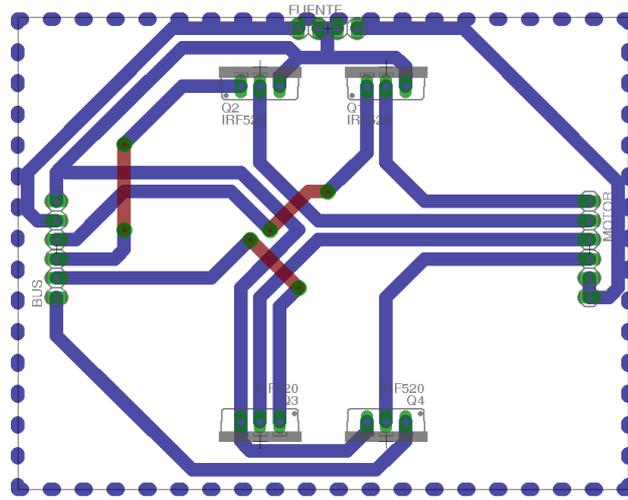


Figura 15.4

## Anexo 16

### Programa nodo Puerta.

```
#include <Stepper.h> //Importamos la libreria para controlar motores paso a paso
#include <RHReliableDatagram.h>
#include <RH_NRF24.h>
#include <SPI.h>
#define STEPS 200//Ponemos el número de pasos que necesita para dar una vuelta.
//Direcciones
#define NODO_B_XBEE_B3 1
#define PUERTA 3

RH_NRF24 driver(9, 10); // Para Arduino
RHReliableDatagram manager(driver, PUERTA);
Stepper stepper(STEPS, 7, 6, 5, 4);
int velocidad=120;//Velocidad en rpm del motor
const int vuelta=STEPS;//numero de pasos para completar una vuelta
int var;
int pasos_derecha;
const int finB=A5;
const int finA=A3;
const int abierto=A1;

String programa="PUERTA";
int canal=95;
char *dato; //vector de caracteres para la recepcin de datos

void setup()
{
digitalWrite(7, LOW);
digitalWrite(6, LOW);
digitalWrite(5, LOW);
digitalWrite(4, LOW);
Serial.begin(9600);
stepper.setSpeed(velocidad);//velocidad motor
pinMode(finB, INPUT_PULLUP);
pinMode(finA, INPUT_PULLUP);
pinMode(abierto, INPUT_PULLUP);
//Configuracion nrf24101
if (!manager.init())//Inicializacin de la configuracion
Serial.println("Inicializacion fallida");//En caso fallido
if (!driver.setChannel(canal) //Cambia el canal de transmision
Serial.println("Seleccion de canal fallida");//En caso fallido
//Cambia la velocidad de transmision pone la ganancia en 0dBms
if (!driver.setRF(RH_NRF24::DataRate250kbps,RH_NRF24::TransmitPower0dBm))
Serial.println("Configuracion de NRF fallida"); //En caso fallido
//Impresion de informacion via puerto serial.
Serial.println("Nodo: PUERTA ");
Serial.print("Programa: ");
Serial.println(programa);
Serial.println("Conectado a NODO_B_XBEE_B3");
Serial.print("Canal utilizado: ");
Serial.println(canal);
Serial.println("Letras para interaccion Respuesta significado");
Serial.println(" q h Abre la puerta");
Serial.println(" p c Cierra la puerta");
Serial.println(" o Comprueba el estado");
Serial.println(" e La puerta esta entreabierta");
Serial.println(" c La puerta esta cerrada");
Serial.println(" a La puerta esta abierta");
Serial.println(" k La puerta esta asegurada");

}

uint8_t buf[RH_NRF24_MAX_MESSAGE_LEN];
uint8_t respuesta[RH_NRF24_MAX_MESSAGE_LEN];

void loop()
{
if (manager.available())
{// Serial.println("mensaje recibido ");
uint8_t len = sizeof(buf);// Variable que contiene la longitud del buffer
uint8_t from; // Variable para obtener la direccion del nodo que envia el mensaje
```

```

    if (manager.recvfromAck(buf, &len, &from))//Envia un ACK de confirmacion
    {
        if(!digitalRead(abierto))
        {
            dato = (char*)buf;//Guarda el dato recibido en la primera posicion del vector de
caracteres
            if(dato[0]=='q')
            {
                respuesta_abierto();
                GirarDerecha();
            }
            if(dato[0]=='p')
            {
                respuesta_cerrado();
                GirarIzquierda();
            }
            if(dato[0]=='o')
            {
                respuesta_estado();
            }
        }
        else
            respuesta_asegurado();
    }
}

}

void Desactivar()
{
digitalWrite(7, LOW);
digitalWrite(6, LOW);
digitalWrite(5, LOW);
digitalWrite(4, LOW);
}

void GirarDerecha()
{
    int aux=1;
    int valor=digitalRead(finA);
    Serial.println(valor);
    while(valor==1)
    {
        stepper.step(aux);
        valor=digitalRead(finA);
    }
    Desactivar();
}

void GirarIzquierda()
{
    int aux=-1;
    int valor=digitalRead(finB);

    while(valor==1)
    {
        stepper.step(aux);
        valor=digitalRead(finB);
    }
    Desactivar();
}
}

```

# Anexo 17

## Programa nodo ventana.

```
#include <Stepper.h> //Importamos la libreria para controlar motores paso a paso
#include <RHReliableDatagram.h>
#include <RH_NRF24.h>
#include <SPI.h>
#define STEPS 200//Ponemos el número de pasos que necesita para dar una vuelta.
//#define STEPS 96//Ponemos el número de pasos que necesita para dar una vuelta. M42SP-4

//Direcciones
#define NODO_A_XBEE_A1 1
#define VENTANA 6

RH_NRF24 driver(9, 10); // Para Arduino
RHReliableDatagram manager(driver, VENTANA);
Stepper stepper(STEPS, 7, 6, 5, 4);
int velocidad=120;//Velocidad en rpm del motor

//Variables
const int vuelta=STEPS;//numero de pasos para completar una vuelta
int var;
int pasos_derecha;
const int finB=A5;
const int finA=A3;
const int abierto=A1;

String programa="VENTANA";
int canal=90;
char *dato; //vector de caracteres para la recepcin de datos

void setup()
{
digitalWrite(7, LOW);
digitalWrite(6, LOW);
digitalWrite(5, LOW);
digitalWrite(4, LOW);
Serial.begin(9600);
stepper.setSpeed(velocidad);//velocidad motor
pinMode(finB, INPUT_PULLUP);
pinMode(finA, INPUT_PULLUP);
pinMode(abierto, INPUT_PULLUP);
//Configuracion nrf24l01
if (!manager.init())//Inicializacin de la configuracion
Serial.println("Inicializacion fallida");//En caso fallido
if (!driver.setChannel(canal)) //Cambia el canal de transmision
Serial.println("Seleccion de canal fallida");//En caso fallido
//Cambia la velocidad de transmision pone la ganancia en 0dBms
if (!driver.setRF(RH_NRF24::DataRate250kbps,RH_NRF24::TransmitPower0dBm))
Serial.println("Configuracion de NRF fallida"); //En caso fallido
//Impresion de informacion via puerto serial.
Serial.println("Nodo: VENTANA");
Serial.print("Programa: ");
Serial.println(programa);
Serial.println("Conectado a NODO_A_XBEE_A1");
Serial.print("Canal utilizado: ");
Serial.println(canal);
Serial.println("Letras para interaccion Respuesta significado");
Serial.println(" q h Abre la ventana");
Serial.println(" p c Cierra la ventana");
Serial.println(" o Comprueba el estado");
Serial.println(" e La ventana esta entreabierta");
Serial.println(" c La ventana esta cerrada");
Serial.println(" a La ventana esta abierta");
Serial.println(" k La ventana esta asegurada");
}
uint8_t buf[RH_NRF24_MAX_MESSAGE_LEN];
uint8_t respuesta[RH_NRF24_MAX_MESSAGE_LEN];

void loop()
{
if (manager.available())
```

```

    // Serial.println("mensaje recibido ");
    uint8_t len = sizeof(buf); // Variable que contiene la longitud del buffer
    uint8_t from; // Variable para obtener la direccion del nodo que envia el mensaje

    if (manager.recvfromAck(buf, &len, &from) //Envia un ACK de confirmacion
    {
        if (!digitalRead(abierto))
        {
            dato = (char*)buf; //Guarda el dato recibido en la primera posicion del vector de
caracteres
            if (dato[0]=='q')
            {
                respuesta_abierto();
                GirarDerecha();
            }
            if (dato[0]=='p')
            {
                respuesta_cerrado();
                GirarIzquierda();
            }
            if (dato[0]=='o')
            {
                respuesta_estado();
            }
        }
        else
            respuesta_asegurado();
    }
}

void Desactivar()
{
    digitalWrite(7, LOW);
    digitalWrite(6, LOW);
    digitalWrite(5, LOW);
    digitalWrite(4, LOW);
}

void GirarDerecha()
{
    int aux=1;
    int valor=digitalRead(finA);
    Serial.println(valor);
    while(valor==1)
    {
        stepper.step(aux);
        valor=digitalRead(finA);
    }
    Desactivar();
}

void GirarIzquierda()
{
    int aux=-1;
    int valor=digitalRead(finB);

    while(valor==1)
    {
        stepper.step(aux);
        valor=digitalRead(finB);
    }
    Desactivar();
}

int respuesta_abierto(){
uint8_t dato[] = "h";
manager.sendtoWait(dato, sizeof(dato), 1);
}

int respuesta_cerrado(){
uint8_t dato[] = "j";
manager.sendtoWait(dato, sizeof(dato), 1);
}

int respuesta_asegurado(){
uint8_t dato[] = "k";
manager.sendtoWait(dato, sizeof(dato), 1);
}

```

```
}  
int respuesta_estado(){  
    if(digitalRead(finA)&& digitalRead(finB)){//Si ambos estan sin presionar la ventana esta entreabierta  
        //uint8_t respuesta[] = "entre_abierta";  
        uint8_t respuesta[] = "e";  
        manager.sendtoWait(respuesta, sizeof(respuesta), 1);  
        //Serial.println("abierto");  
    }  
    if(!digitalRead(finA)){//Si esta presionado la ventana esta abierta  
        //uint8_t respuesta[] = "abierta";  
        uint8_t respuesta[] = "a";  
        manager.sendtoWait(respuesta, sizeof(respuesta), 1);  
        //Serial.println("abierto");  
    }  
    if(!digitalRead(finB)){//Si esta presionado la ventana esta cerrada  
        //uint8_t respuesta[] = "cerrada";  
        uint8_t respuesta[] = "c";  
        manager.sendtoWait(respuesta, sizeof(respuesta), 1);  
        //Serial.println("cerrado");  
    }  
}
```

# Anexo 18

## Diagrama electrónico nodo cortina.

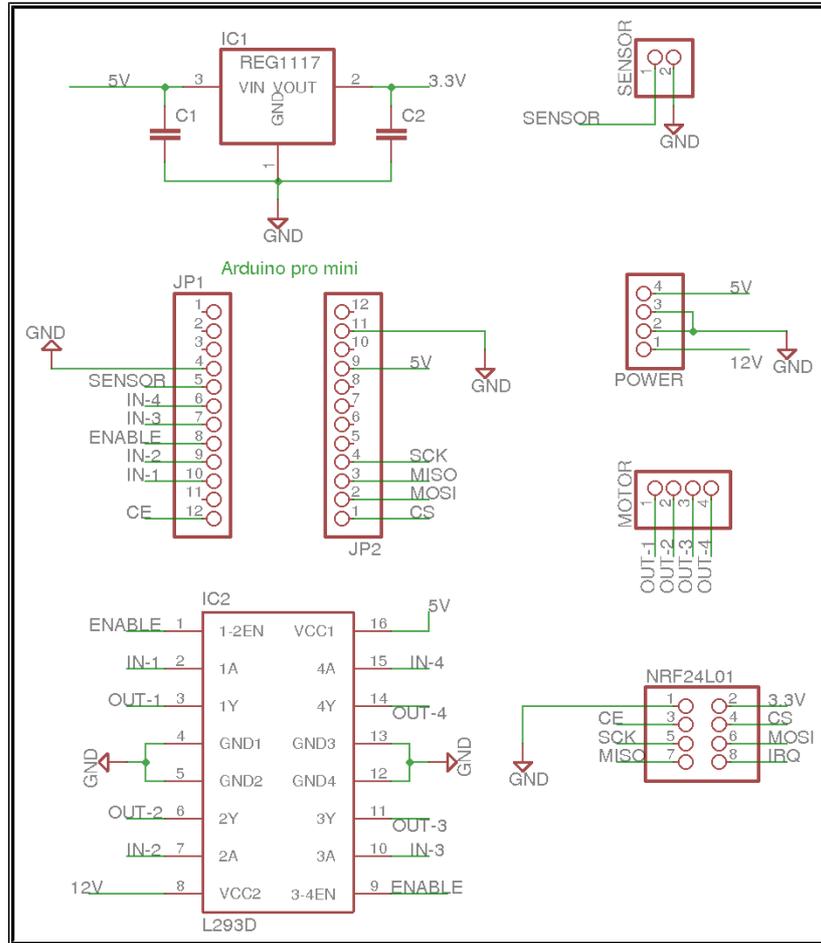


Figura 18.1

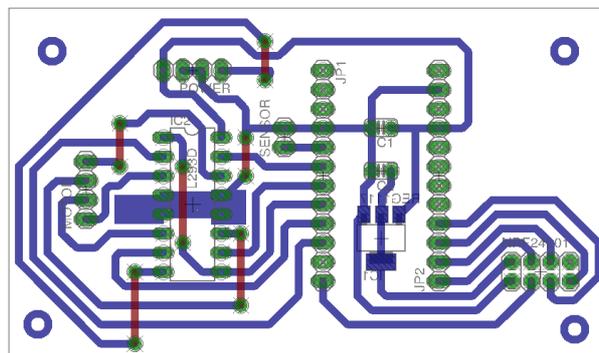


Figura 18.2

# Anexo 19

## Programa nodo cortina.

```
//LETRAS PARA LA INTERACCIÓN
/*
Letra      Devuelve      Acción
u          -la posición  -Devuelve el estado de la cortina

Cadena
cxx .- es una cadena que se recibe del nodo central del tipo (c1, c20,c33). La letra c indica que se va
ha realizar un cambio de posición y el numero
siguiente indica la posición deseada en la que se quiere poner la cortina; esta cadena se construye en
la Raspberry Pi correspondientemente con lo
que el usuario haya indicado en la interfaz.

//INFORMACIÓN ACERCA DEL NODO
-El nodo se llama CORTINA y se conecta la XBEE A.
-La dirección del nodo es la numero 5.
*/
#include <Stepper.h> //Importamos la librería para controlar motores paso a paso
#include <RHReliableDatagram.h>
#include <RH_NRF24.h>
#include <SPI.h>
#define STEPS 49//Ponemos el número de pasos que necesita para dar una vuelta.
//#define STEPS 96//Ponemos el número de pasos que necesita para dar una vuelta.

#define NODO_A_XBEE_A1 1
#define CORTINA 5

RH_NRF24 driver(9, 10); // Para Arduino
RHReliableDatagram manager(driver, CORTINA);

//Stepper stepper(STEPS, 8, 7, 6, 5); //Stepper nombre motor (número de pasos por vuelta, pins de
control)
//Stepper stepper(STEPS, 5, 6, 3, 4); //para protoboard
Stepper stepper(STEPS, 7, 6, 4, 3); //para pbc con shield L298n

const int sensor=2;
// const int ENA =4;
// const int ENB =3;
// const int ENA =7;//para protoboard
// const int ENB =8;//para protoboard

const int ENA =5;//para protoboard
//const int ENB =3;//para protoboard

const int vuelta=STEPS;//numero de pasos para completar una vuelta
int var;
int velocidad=80;//Velocidad en rpm del motor

//Variables para recibir cadenas completas del puerto serial
char cadena[30]; //Creamos un array que almacenará los caracteres que escribiremos en la consola del PC.
Le asignamos un tope de caracteres, en este caso 30
byte posicion=0; //Variable para cambiar la posición de los caracteres del array
int valor; //Variable del valor entero

String programa="prueba con sensor y pasos cortos l293d doble";
int canal=90;
char *dato; //vector de caracteres para la recepción de datos

//Variables para calculo de posición
int ultima_pos=0;
int nueva_pos;
int resultado;
int aux;

void setup()
{
  Serial.begin(9600);
  stepper.setSpeed(velocidad);//velocidad motor(200)
  pinMode(ENA, OUTPUT);
  // pinMode(ENB, OUTPUT);
  pinMode(sensor, INPUT_PULLUP);
  digitalWrite(ENA, LOW);
  //digitalWrite(ENB,LOW);
  // Serial.println("Configuracion de motor correcta");
}
```

```

//Configuracion nrf24l01
if (!manager.init())//Inicializacin de la configuración
  Serial.println("Inicialización fallida");//En caso fallido
if (!driver.setChannel(canal)) //Cambia el canal de transmisión
  Serial.println("Selección de canal fallida");//En caso fallido
//Cambia la velocidad de transmisión pone la ganancia en 0dBms
if (!driver.setRF(RH_NRF24::DataRate250kbps,RH_NRF24::TransmitPower0dBm))
  Serial.println("Configuracion de NRF fallida"); //En caso fallido

//Impresion de informacion via puerto serial.
Serial.println("Nodo: CORTINA ");
Serial.print("Programa: ");
Serial.println(programa);
Serial.println("Conectado a NODO_A_XBEE_A1");//LETRAS PARA LA INTERACCIÓN
/*
Letra      Devuelve      Acción
u          -la posición    -Devuelve el estado de la cortina

Cadena
cxx .- es una cadena que se recibe del nodo central del tipo (c1, c20,c33). La letra c indica que se va
ha realizar un cambio de posición y el numero
siguiente indica la posición deseada en la que se quiere poner la cortina; esta cadena se construye en
la Raspberry Pi correspondientemente con lo
que el usuario haya indicado en la interfaz.

//INFORMACIÓN ACERCA DEL NODO
-El nodo se llama CORTINA y se conecta la XBEE A.
-La dirección del nodo es la numero 5.

*/
#include <Stepper.h> //Importamos la librería para controlar motores paso a paso
#include <RHReliableDatagram.h>
#include <RH_NRF24.h>
#include <SPI.h>
#define STEPS 49//Ponemos el número de pasos que necesita para dar una vuelta.
//#define STEPS 96//Ponemos el número de pasos que necesita para dar una vuelta.

#define NODO_A_XBEE_A1 1
#define CORTINA 5

RH_NRF24 driver(9, 10); // Para Arduino
RHReliableDatagram manager(driver, CORTINA);

//Stepper stepper(STEPS, 8, 7, 6, 5); //Stepper nombre motor (número de pasos por vuelta, pins de
control)
//Stepper stepper(STEPS, 5, 6, 3, 4); //para protoboard
Stepper stepper(STEPS, 7, 6, 4, //LETRAS PARA LA INTERACCIÓN
/*
Letra      Devuelve      Acción
u          -la posición    -Devuelve el estado de la cortina

Cadena
cxx .- es una cadena que se recibe del nodo central del tipo (c1, c20,c33). La letra c indica que se va
ha realizar un cambio de posición y el numero
siguiente indica la posición deseada en la que se quiere poner la cortina; esta cadena se construye en
la Raspberry Pi correspondientemente con lo
que el usuario haya indicado en la interfaz.

//INFORMACIÓN ACERCA DEL NODO
-El nodo se llama CORTINA y se conecta la XBEE A.
-La dirección del nodo es la numero 5.

*/
#include <Stepper.h> //Importamos la librería para controlar motores paso a paso
#include <RHReliableDatagram.h>
#include <RH_NRF24.h>
#include <SPI.h>
#define STEPS 49//Ponemos el número de pasos que necesita para dar una vuelta.
//#define STEPS 96//Ponemos el número de pasos que necesita para dar una vuelta.

#define NODO_A_XBEE_A1 1
#define CORTINA 5

RH_NRF24 driver(9, 10); // Para Arduino
RHReliableDatagram manager(driver, CORTINA);

//Stepper stepper(STEPS, 8, 7, 6, 5); //Stepper nombre motor (número de pasos por vuelta, pins de
control)
//Stepper stepper(STEPS, 5, 6, 3, 4); //para protoboard

```

```

Stepper stepper(STEPS, 7, 6, 4, 3); //para pbc con shield L298n

const int sensor=2;
// const int ENA =4;
// const int ENB =3;
// const int ENA =7;//para protoboard3); //para pbc con shield L298n

const int sensor=2;
// const int ENA =4;
// const int ENB =3;
// const int ENA =7;//para protoboard
Serial.print("Canal utilizado: ");
Serial.println(canal);
Serial.println("Letras para interaccion");
Serial.println("          u          -Devuelve la posicion de la cortina");
Serial.println("Cadena cXX donde XX es la poscion que se desea en la cortina");
}
uint8_t buf[RH_NRF24_MAX_MESSAGE_LEN];

void loop()
{
  valor=digitalRead(sensor);// Si esta variable esta en cero significa que ha habido intervencion
manual.
  //Serial.println(valor);
  if (manager.available()) //Hay algun mensaje para este nodo??
  {
    uint8_t len = sizeof(buf);// Variable que contiene la longitud del buffer
    uint8_t from; // Variable para obtener la direccion del nodo que envia el mensaje

    if (manager.recvfromAck(buf, &len, &from)//Envia un ACK de confirmacion
    {
      dato = (char*)buf;//Guarda el dato recibido el la primera posicion del vector de caracteres

      if(dato[0]=='u')
      {
        if(valor==0)
        {
          respuesta_manual();
          Girar(-6);//para encerrar la cortina
        }
        else
        {
          respuesta();
        }
      }
    }
    else
    {
      if(valor==0)
      {
        respuesta_manual();
        Girar(-6);//para encerrar la cortina
      }
      else{
        int n = atoi((char*)dato);//trandorma a un entero

        nueva_pos=n;

        resultado=nueva_pos-ultima_pos;
        ultima_pos=nueva_pos;
        /*Serial.print("Estado final: ");
        Serial.println(ultima_pos);
        Serial.print("Numero de vuletas: ");
        Serial.println(resultado);*/
        respuesta();
        Girar(resultado);
      }
    }
  }
}
//Serial.println(dato[0]);
}

void Activar()

```

```

{
digitalWrite(ENA, HIGH);
//digitalWrite(ENB,HIGH);
}

void Desactivar()
{
digitalWrite(ENA, LOW);
//digitalWrite(ENB,LOW);
}

void Girar(int num)
{
aux;
aux=num*vuelta;
Activar();
stepper.step(aux);
Desactivar();
Pasos();
}

void Pasos()
{int x;
if(aux>=0){x=1;}else{x=-1;}
valor=digitalRead(sensor);

while(valor==0)
{
stepper.setSpeed(20);
Activar();
stepper.step(x);
//delay(100);
valor=digitalRead(sensor);
Serial.println(valor);
}
Desactivar();
stepper.setSpeed(velocidad);
}

int respuesta(){
//int lectural=!digitalRead(foco);
String str;
uint8_t aux[10];
str=String(ultima_pos);
str.getBytes(aux, 9);
manager.sendtoWait(aux, sizeof(aux), 1);
// manager.sendtoWait(dato, sizeof(dato), 1);
}

int respuesta_manual(){
ultima_pos=0;
nueva_pos=0;
String str;
uint8_t aux[10];
str=String(ultima_pos);
str.getBytes(aux, 9);
manager.sendtoWait(aux, sizeof(aux), 1);
}

```

## Anexo 20

### Página login.

#### Código Fuente *loginjqr.php*

```
<?php
session_start(); //crea la cookie... sesion
if(isset($_POST['login']))
{
$completa = $_POST['pass'];
$comparar = md5 ($completa); //codifica lo escrito en md5
$clave = '****'; //contraseña
$final = md5 ( $clave ); //la codifica md5
if ( $comparar == $final ) { //Condición para comprobar la contraseña
$_SESSION['listo'] = true;
header('Location: lugares.php'); //redirige a la pagina despues de iniciar sesion
exit;
} else
{
?>
<script type="text/javascript">
<!--
alert('Password Incorrecto') //Mensaje en caso de contraseña incorrecta
/-->
</script>
<?php
}
}
?>

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Domótica Web</title>
<link rel="stylesheet" type="text/css" href="jquery.mobile-1.3.2.css">
<script type="text/javascript" src="jquery-1.10.2.min.js"></script>
<script type="text/javascript" src="jquery.mobile-1.3.2.js"></script>
</head>
<body>
<div data-role="page" data-theme="c">
<!--cabecera-->
<div data-role="header" data-theme="b">
<h2>Bienvenido</h2>
</div>
<!--contenido-->
<div data-role="content" data-theme="c">
<h2>Bienvenido</h2>
<form action="" method="post" data-ajax="false">
<label for="contra">Contraseña:</label>
<input type="password" name="pass">
<input type="submit" value="Entrar" name="login">
</form>
</div>
<!--pie de pagina-->
<div data-role="footer" data-position="fixed" data-theme="c">
<p>&copy; Domótica Web</p>
</div>
</div>
</body>
</html>
```

# Anexo 21

## Página áreas.

### Código Fuente *lugares.php*

```
<?php session_start(); require 'cookie.php';?>
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Domótica Web</title>
  <link rel="stylesheet" type="text/css" href="jquery.mobile-1.3.2.css">
  <script type="text/javascript" src="jquery-1.10.2.min.js"></script>
  <script type="text/javascript" src="jquery.mobile-1.3.2.js"></script>

</head>
<body>
<div data-role="page" data-theme="c">
  <!--cabecera-->
  <div data-role="header" data-theme="b">
    <h2>Áreas</h2>
    <a href="logout.php" data-icon="delete" data-theme="a" data-transition="Pop" data-ajax="false"
class="ui-btn-right">Salir</a>
  </div>
  <!--contenido-->
  <div data-role="content" data-theme="c">
<p>Seleccione un área a controlar.</p>
    <a href="sala.php" data-role="button" data-icon="gear" data-transition="Pop" data-ajax="false"
data-theme="c">Sala</a>
    <a href="manta-termica.php" data-role="button" data-icon="gear" data-transition="Pop" data-
ajax="false" data-theme="c">Manta térmica</a>
    <a href="puerta.php" data-role="button" data-icon="gear" data-transition="Pop" data-
ajax="false" data-theme="c">Puerta</a>
  </div>

  <!--pie de pagina-->
  <div data-role="footer" data-theme="d" data-position="fixed">
    <p>&copy; Domótica Web</p>
  </div>

</div>

</body>
</html>
```

## Anexo 22

### Archivos PHP para direccionamiento.

#### Código Fuente direccionA.php

```
<?php

$direccion = file_get_contents("direccion.txt");
//echo $direccion;
// si el archivo contiene 1 significa que ya esta en la dir B
// si el archivo contiene 2 significa que ya esta en la dir A
if($direccion == 2){
echo " OK";
//echo "Ya esta en la dir A";
}
else{
    $fp = fopen('/dev/ttyUSB0','r+'); //Selecciona el puerto serial
    fwrite($fp, "+++"); //Entra al modo de configuracion en el xbee
    $a = fread($fp,2); //Recibe el primer ok
    // echo $a."\n";
    fwrite($fp, "atdl1."\r"); //Envia la nueva direccion
    $b = fread($fp,3); //Recibe el segundo ok
    // echo $b."\n";
    fwrite($fp, "atwr."\r");
    $d = fread($fp,3);
    // echo $d."\n";
    fwrite($fp, "atcn."\r"); //Sale del modo de configuracion
    $c = fread($fp,3); //Recibe el tercer ok
    echo $c."\n"; //Imprime el tercer ok
    fclose($fp); //Cierra el puerto serial
    exec("echo 2 >./direccion.txt"); // pone la bandera indicando que hac el cambio
}
?>
```

Como se observa este archivo inicia comprobando el contenido de un archivo de texto sencillo(direccion.txt), si el contenido es 1 significa que el xbee esta direccionado con destino hacia el xbee B, pero si el contenido es 2 quiere decir que el destino es el xbee A, en este caso solo devuelve un “OK”.

#### Código Fuente direccionB.php

```
<?php

$direccion = file_get_contents("direccion.txt");
//echo $direccion;
// si el archivo contiene 1 significa que ya esta en la dir B
// si el archivo contiene 2 significa que ya esta en la dir A
if($direccion == 1){
echo " OK";
//echo "Ya esta en la dir B";
}
else{
    $fp = fopen('/dev/ttyUSB0','r+'); //Selecciona el puerto serial
    fwrite($fp, "+++"); //Entra al modo de configuracion en el xbee
    $a = fread($fp,2); //Recibe el primer ok
    //echo $a."\n";
    fwrite($fp, "atdl3."\r");//Envia la nueva direccion
    $b = fread($fp,3);//Recibe el segundo ok
    //echo $b."\n";
    fwrite($fp, "atwr."\r");
    $d = fread($fp,3);
    fwrite($fp, "atcn."\r");//Sale del modo de configuracion
    $c = fread($fp,3); //Recibe el tercer ok
    echo $c."\n";//Imprime el tercer ok
    fclose($fp);//Cierra el puerto serial
    exec("echo 1 >./direccion.txt");// pone la bandera indicando que hac el cambio
    //echo "NO estaba en B";
}
```

?> }

De manera similar se comprueba el mismo archivo texto direccion.txt y si el contenido es 1 significa que el xbee esta direccionado con destino hacia el xbee B, pero si el contenido es 2 quiere decir que el destino es el xbee A. Si el caso es el primero entonces solo se envía un "OK".

## Anexo 23

### Archivos de la carpeta control-sala.

#### Código fuente archivo estado-foco-sala.php

```
<?php
$puerto = fopen('/dev/ttyUSB0','r+'); //Selecciona el puerto serial
fwrite($puerto, "x");
//$var= fgets($fp);
$var = fread($puerto,1);
//echo $var."\n";
if($var==2){
    fwrite($puerto, "s");
    $r = fread($puerto,3);
    //echo $r;
    $array = array($r); // pone en un vector los resultados
    $aux = implode("", $array); //Transforma el vector en una cadena
    $estados = preg_replace("/\r\n+|\r+|\n+|\t+i", "", $aux); // elimina saltos de linea y espacios
entre el texto
    echo $estados."\n"; // imprime
}
else{
    $read="x";
    echo $read;
}

fclose($puerto);
?>
```

#### Código fuente archivo foco-sala.php

```
<?php
$fp = fopen('/dev/ttyUSB0','r+');
fwrite($fp, "m");
$x = fread($fp,1);
    $array = array($x); // pone en un vector los resultados
    $aux = implode("", $array); //Transforma el vector en una cadena
    $estados = preg_replace("/\r\n+|\r+|\n+|\t+i", "", $aux); // elimina saltos de linea y espacios
entre el texto
    echo $estados."\n";// imprime

//echo $x."\n";
fclose($fp);
?>
```

## Anexo 24

### Código fuente página sala.php

```
<?php session_start(); require 'cookie.php';?>
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Domótica Web</title>
  <link rel="stylesheet" type="text/css" href="jquery.mobile-1.3.2.css">
  <script type="text/javascript" src="jquery-1.10.2.min.js"></script>
  <script type="text/javascript" src="jquery.mobile-1.3.2.js"></script>

  <!--Script para ejecutar php sin recargar la pagina
  Esta es la pagina pricipal
  -->
</script>
function focol(){
//Controla el foco l
$.ajax({
  url: 'control/control-sala/foco_sala.php',
  type: 'post',
  beforeSend: function () {
    $("#estadofoco0").html("");
  },
  success: function (response) {
    if( x[0]==1){
      var fts = $('#flag0');
      fts.val('on');
      fts.slider('refresh');
    }
    if( x[0]==0){
      var fts = $('#flag0');
      fts.val('off');
      fts.slider('refresh');
    }
    if( x[0]==3){
      $("#estadofoco0").html("Problemas de comunicación, intente nuevamente.");
    }
  }
});
}

function configurar(){
//Direcciona el XBEE A
$.ajax({
  url: 'control/direccionA.php',
  type: 'post',
  beforeSend: function () {
    $("#Carga").html("Cargando...");
  },
  success: function (response) {
    $("#Carga").html(response);
    var estado = $("#Carga").html();
    if(estado[1]=='O'){
      if(estado[2]=='K'){
        $("#Carga").html("Cargado");
      }
    }
  },
  complete : function (){
    estado();
  }
});
}

function estado(){
$.ajax({
  url: 'control/control-sala/estado-foco-sala.php',
  type: 'post',
  beforeSend: function () {
    $("#estadofoco0").html("");
  },
  success: function (response) {
    var x=response;
  }
});
}
```

```

        if( x[0]==1){
            var fts = $('#flag0');
            fts.val('on');
            fts.slider('refresh');
        }
        if( x[0]==0){
            var fts = $('#flag0');
            fts.val('off');
            fts.slider('refresh');
        }
        if( x[0]==3){
            $('#estadofoco0').html("Problemas de comunicación, intente nuevamente.");
        }
    }
});
}
</script>
<!--Script para comprobar el estado de los puertos de arduino-->
<script language="javascript" type="text/javascript">
    $(function() {
        configurar();
    });
</script>
</head>
<body onload=>
    <div data-role="page" data-theme="c">
        <!--cabecera-->
        <div data-role="header" data-fullscreen="true" data-theme="b">
            <h2>Sala</h2>
            <a href="lugares.php" data-theme="c" data-transition="Pop" data-ajax="false">Áreas</a>
            <a href="logout.php" data-theme="c" data-transition="Pop" data-ajax="false">Salir</a>
        </div>
        <!--contenido-->
        <div data-role="content" data-theme="c">
            <!--aviso estado actual-->
            <div>
                <label data-inline ="True">Área Sala:</label>
                <label id="Carga" data-inline ="true"></label>
            </div>
            <div data-role="collapsible" data-theme="c" data-content-theme="d" data-collapsed="false">
                <h4>Luz</h4>
                <!--switch-Luz-->
                <div>
                    <label data-inline ="True">Estado:</label>
                    <label id="estadofoco0" data-inline ="true"></label>
                </div>
                <!--slider Luz-->
                <select name="flag0" id="flag0" data-role="slider">
                    <option value="off">off</option>
                    <option value="on">on</option>
                </select>
            </div>
            <a href="cortina.php" data-role="button" data-icon="gear" data-transition="Pop" data-ajax="false" data-theme="c">Cortina</a>
            <a href="ventana.php" data-role="button" data-icon="gear" data-transition="Pop" data-ajax="false" data-theme="c">Ventana</a>
        </div>
        <!--pie de pagina-->
        <div data-role="footer" data-theme="c" data-mini="true" data-position="fixed">
            <button id="button" type="button">Actualizar </button>
        </div>
    </div>
    <script type="text/javascript">
        $('select#flag0').change(function() {foco1()});
    </script>
    <script type="text/javascript">
        $('#button').click(function() {estado()});
    </script>

```

```
</script>  
</body>  
</html>
```

## Anexo 25

### Archivos directorio control-cortina

#### Código fuente estrado.php

```
<?php
$fp = fopen('/dev/ttyUSB0','r+'); //Abre el puerto serial
fwrite($fp, "u"); //Envía u
$x = fread($fp,3); //Recibe la respuesta
echo $x."\n"; //Imprime la respuesta
fclose($fp); //Cierra el puerto serial
?>
```

#### Código fuente mover.php

```
<?php
$valor = $_GET['valor']; //recupera el valor ingresado por el usuario
$fp = fopen('/dev/ttyUSB0','r+'); //abre el puerto serial
fwrite($fp, "c".$valor); //une la letra c al numero ingresado CXX
$x = fread($fp,3); //Recibe la respuesta del nodo
echo $x."\n"; //Imprime la respuesta
fclose($fp); //Cierra el puerto serial.
?>
```

## Anexo 26

### Código fuente página cortina.php

```
<?php session_start(); require 'cookie.php';?>
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Domótica Web</title>
  <link rel="stylesheet" type="text/css" href="jquery.mobile-1.3.2.css">
  <script type="text/javascript" src="jquery-1.10.2.min.js"></script>
  <script type="text/javascript" src="jquery.mobile-1.3.2.js"></script>

  <!--Script para ejecutar php sin recargar la pagina
  Esta es la pagina principal
  -->
<script>
function cortina(){
  var valor = $('#slider-10').val();
  $.ajax({
    url: 'control/control-cortina/mover.php?valor='+valor,
    type: 'post',
    beforeSend: function () {
    },
    success: function (response) {
      $('#Estado').html(response);
    }
  });
}

function Estado_cortina(){
  $.ajax({
    url: 'control/control-cortina/estado.php',
    type: 'post',
    beforeSend: function () {
    },
    success: function (response) {
      $('#Estado').html(response);
      var estado = $('#Estado').html();
      var fts = $('#slider-10');
      fts.val(parseInt(estado));
      fts.slider('refresh');
    }
  });
}

function configurar(){
  //Direcciona el XBEE A1
  $.ajax({
    url: 'control/direccionA.php',
    type: 'post',
    beforeSend: function () {
      $('#Carga').html("Cargando...");
    },
    success: function (response) {
      $('#Carga').html(response);
      var estado = $('#Carga').html();
      if(estado[7]=='0'){
        estado[8]=='K'){
          $('#Carga').html("Cargado");
        }
      }
    },
    complete : function (){
      Estado_cortina();
    }
  });
}

</script>

<!--Script para comprobar el estado de los puertos de arduino-->
<script language="javascript" type="text/javascript">

$(function() {
  configurar();
});
```

```

    });
</script>

</head>
<body onload=>

<div data-role="page" data-theme="c">
  <!--cabecera-->
  <div data-role="header" data-theme="b">
    <h2>Cortina</h2>
    <a href="lugares.php" data-theme="c" data-transition="Pop" data-ajax="false">Áreas</a>
    <a href="logout.php" data-theme="c" data-transition="Pop" data-ajax="false">Salir</a>
  </div>

  <!--contenido-->
  <div data-role="content" data-theme="c">
    <!--aviso estado actual-->

    <div>
      <label data-inline ="True">Cortina:</label>
      <label id="Carga" data-inline ="true"></label>
    </div>

    <div data-role="collapsible" data-theme="c" data-content-theme="d" data-collapsed="false">
      <h4>Estado</h4>
      <form>
        <div>
          <label for="slider-10">Slider:</label>
          <label id="Estado" data-inline ="true"></label>
        </div>
        <input name="slider-10" id="slider-10" min="0" max="100" step="1" value="0" type="range">
        <button id="button" type="button">Enviar</button>
      </form>
    </div>

  </div>
  <!--pie de pagina-->
  <div data-role="footer" data-theme="d" data-mini="true" data-position="fixed" >
    <p>&copy; Domótica Web</p>
  </div>
</div>

<script type="text/javascript">
  $('#button').click(function() {cortina()});
</script>

</body>
</html>

```

## Anexo 27

### Archivos de las capetas control-ventana y control-puerta

#### estado.php

```
<?php
$fp = fopen('/dev/ttyUSB0','r+'); //abre el puerto serial
fwrite($fp, "o"); //escribe o en el puerto serial
$x = fread($fp,1); //recibe la respuesta
echo $x."\n"; //imprime
fclose($fp);
?>
```

#### abrir.php

```
<?php
$fp = fopen('/dev/ttyUSB0','r+'); //abre el puerto serial
fwrite($fp, "q"); //escribe en el puerto serial
$x = fread($fp,1); //recibe la respuesta
echo $x."\n"; //imprime
fclose($fp); //cierra el puerto serial
?>
```

#### cerrar.php

```
<?php
$fp = fopen('/dev/ttyUSB0','r+'); //abre el puerto serial
fwrite($fp, "p"); //escribe p en el puerto serial
$x = fread($fp,1); //recibe la respuesta
echo $x."\n"; //imprime
fclose($fp); //cierra el puerto serial
?>
```

## Anexo 28

### Código fuente páginas puerta y ventana.

#### Código fuente ventana

```
<?php session_start(); require 'cookie.php';?>
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Domótica Web</title>
<link rel="stylesheet" type="text/css" href="jquery.mobile-1.3.2.css">
<script type="text/javascript" src="jquery-1.10.2.min.js"></script>
<script type="text/javascript" src="jquery.mobile-1.3.2.js"></script>

<!--Script para ejecutar php sin recargar la pagina
Esta es la pagina principal
-->
<script>

function configurar()
{
//Direcciona el XBEE A
$.ajax({
url: 'control/direccionA.php',
type: 'post',
beforeSend: function () {
$("#Carga").html("Cargando...");
},
success: function (response) {
$("#Carga").html(response);
var estado = $("#Carga").html();
//alert(estado[1]);
//alert(estado);
if(estado[1]=='O'){
if(estado[2]=='K'){
$("#Carga").html("Cargado");
}
}
},
complete : function (){
//alert("configuracion correcta");
estado();
}
});
}

function estado(){

$.ajax({
url: 'control/control-ventana/estado.php',
type: 'post',
beforeSend: function () {
//$("#Estado").html("");
},
success: function (response) {
//$("#Estado").html(response);
var x=response;
//alert(x[0]);
if( x[0]=='k'){
$("#button1").button('disable');
$("#button2").button('disable');
$("#Estado").html("La ventana esta ASEGURADA.");
}
if( x[0]=='c'){
$("#button2").button('disable');
$("#button1").button('enable');
$("#Estado").html("La ventana esta CERRADA.");
}
}

if( x[0]=='a'){
$("#button2").button('enable');
}
```

```

$('#button1').button('disable');
$('#Estado').html("La ventana esta ABIERTA.");

}
if( x[0]=='e'){
$('#Estado').html("Ventana entreabierta");
$('#button1').button('enable');
$('#button2').button('enable');
}
if( x[0]==3){
$('#Estado').html("Problemas de comunicación, intente nuevamente.");
}

},
complete : function (){
//miTiempo();

}
});
}

function miAbrir()
{
//alert("Activaste la funcion");
$.ajax({
url: 'control/control-ventana/abrir.php',
type: 'post',
beforeSend: function () {
//$('#Estado').html("");
},
success: function (response) {
//$('#Estado').html(response);
//var x = $('#Estado').html();
//alert(x);
var x=response;
if( x[0]=='k'){
$('#button1').button('disable');
$('#button2').button('disable');
$('#Estado').html("La ventana esta ASEGURADA.");
}
if( x[0]=='h'){
$('#button1').button('disable');
$('#button2').button('enable');
$('#Estado').html("Abriendo ventana.");
}
if( x[0]==3){
$('#Estado').html("Problemas de comunicación, intente nuevamente.");
}

},
complete : function (){

}

});
}

function miCerrar()
{
//alert("Activaste la funcion");
$.ajax({
url: 'control/control-ventana/cerrar.php',
type: 'post',
beforeSend: function () {
//$('#Estado').html("");
},
success: function (response) {
//$('#Estado').html(response);
//var x = $('#Estado').html();
//alert(x);
var x=response;
if( x[0]=='k'){
$('#button1').button('disable');
$('#button2').button('disable');
$('#Estado').html("La ventana esta ASEGURADA.");
}

```

```

}
if( x[0]=='j'){
$('#button1').button('enable');
$('#button2').button('disable');
$("#Estado").html("Cerrando ventana.");
}
if( x[0]==3){
$("#Estado").html("Problemas de comunicación, intente nuevamente.");
}
},
complete : function (){

}

});
}

</script>

<script language="javascript" type="text/javascript">

$(function() {
configurar();
});
</script>

</head>
<body>

<div data-role="page" data-theme="c">
<!--cabecera-->
<div data-role="header">
<h2>Ventana</h2>
<a href="lugares.php" data-theme="a" data-transition="Pop" data-ajax="false">Áreas</a>
<a href="logout.php" data-theme="a" data-transition="Pop" data-ajax="false">Salir</a>
</div>

<!--contenido-->
<div data-role="content" data-theme="c">
<div>
<label data-inline ="True">Ventana:</label>
<label id="Carga" data-inline ="true"></label>
</div>
<div>
<label data-inline ="True">Estado:</label>
<label id="Estado" data-inline ="true"></label>
</div>

<!--Menu colapsable opciones manta-->
<div data-role="collapsible" data-theme="c" data-content-theme="d" data-collapsed="false">
<h4>Configuración Ventana</h4>

<!--<form action="" method="post" data-ajax="false"> -->

<button id="button1" type="button">Abrir </button>
<button id="button2" type="button">Cerrar </button>

<!--</form> -->

</div>
</div>

<!--pie de pagina-->
<div data-role="footer" data-theme="d" data-position="fixed">
<p>&copy; Domótica Web</p>
<button id="button3" type="button" >Actualizar </button>

</div>

```

```

</div>

<script type="text/javascript">
$( '#button1' ).click(function() {miAbrir()});
$( '#button2' ).click(function() {miCerrar()});
$( '#button3' ).click(function() {estado()});
</script>

</body>
</html>

```

## Código fuente puerta

```

<?php session_start(); require 'cookie.php';?>
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Domótica Web</title>
<link rel="stylesheet" type="text/css" href="jquery.mobile-1.3.2.css">
<script type="text/javascript" src="jquery-1.10.2.min.js"></script>
<script type="text/javascript" src="jquery.mobile-1.3.2.js"></script>

<!--Script para ejecutar php sin recargar la pagina
Esta es la pagina principal
-->
<script>

function configurar()
{
//Direcciona el XBEE B
$.ajax({
url: 'control/direccionB.php',
type: 'post',
beforeSend: function () {
$("#Carga").html("Cargando...");
},
success: function (response) {
$("#Carga").html(response);
var estado = $("#Carga").html();
//alert(estado[1]);
//alert(estado);
if(estado[1]=='O'){
if(estado[2]=='K'){
$("#Carga").html("Cargado");
}
}
},
complete : function (){
//alert("configuracion correcta");
estado();
}

});

}

function estado(){

$.ajax({
url: 'control/control-puerta/estado.php',
type: 'post',
beforeSend: function () {
//$("#Estado").html("");
},
success: function (response) {
//$("#Estado").html(response);
var x=response;
//alert(x[0]);
if ( x[0]=='k'){
$( '#button1' ).button('disable');

```

```

$('#button2').button('disable');
$("#Estado").html("La puerta esta ASEGURADA.");
}
if( x[0]=='c'){
$('#button2').button('disable');
$('#button1').button('enable');
$("#Estado").html("La puerta esta CERRADA.");
}

if( x[0]=='a'){
$('#button2').button('enable');
$('#button1').button('disable');
$("#Estado").html("La puerta esta ABIERTA.");
}

if( x[0]=='e'){
$("#Estado").html("Puerta entreabierta");
$('#button1').button('enable');
$('#button2').button('enable');
}
if( x[0]==3){
$("#Estado").html("Problemas de comunicación, intente nuevamente.");
}

},
complete : function (){
//miTiempo();

}
});
}

function miAbrir()
{
//alert("Activaste la funcion");
$.ajax({
url: 'control/control-puerta/abrir.php',
type: 'post',
beforeSend: function () {
//$("#Estado").html("");
},
success: function (response) {
//$("#Estado").html(response);
//var x = $("#Estado").html();
//alert(x);
var x=response;
if( x[0]=='k'){
$('#button1').button('disable');
$('#button2').button('disable');
$("#Estado").html("La puerta esta ASEGURADA.");
}
if( x[0]=='h'){
$('#button1').button('disable');
$('#button2').button('enable');
$("#Estado").html("Abriendo puerta.");
}
if( x[0]==3){
$("#Estado").html("Problemas de comunicación, intente nuevamente.");
}

},
complete : function (){

}

});
}

function miCerrar()
{
//alert("Activaste la funcion");
$.ajax({

```

```

url: 'control/control-puerta/cerrar.php',
type: 'post',
beforeSend: function () {
//$("#Estado").html("");
},
success: function (response) {
//$("#Estado").html(response);
//var x = $("#Estado").html();
//alert(x);
var x=response;
if ( x[0]=='k'){
$("#button1").button('disable');
$("#button2").button('disable');
$("#Estado").html("La puerta esta ASEGURADA.");
}
if ( x[0]=='j'){
$("#button1").button('enable');
$("#button2").button('disable');
$("#Estado").html("Cerrando puerta.");
}
if ( x[0]==3){
$("#Estado").html("Problemas de comunicación, intente nuevamente.");
}

},
complete : function (){

}

});
}

</script>

<script language="javascript" type="text/javascript">

$(function() {
configurar();
});
</script>

</head>
<body>

<div data-role="page" data-theme="c">
<!--cabecera-->
<div data-role="header">
<h2>Puerta</h2>
<a href="lugares.php" data-theme="a" data-transition="Pop" data-ajax="false">Áreas</a>
<a href="logout.php" data-theme="a" data-transition="Pop" data-ajax="false">Salir</a>
</div>

<!--contenido-->
<div data-role="content" data-theme="c">
<div>
<label data-inline ="True">Puerta:</label>
<label id="Carga" data-inline ="true"></label>
</div>
<div>
<label data-inline ="True">Estado:</label>
<label id="Estado" data-inline ="true"></label>
</div>

<!--Menu colapsable opciones manta-->
<div data-role="collapsible" data-theme="c" data-content-theme="d" data-collapsed="false">
<h4>Configuración Puerta</h4>

<!--<form action="" method="post" data-ajax="false"> -->

<button id="button1" type="button">Abrir </button>
<button id="button2" type="button">Cerrar </button>

```

```
<!--</form> -->

</div>
</div>

<!--pie de pagina-->
<div data-role="footer" data-theme="d" data-position="fixed">
<p>&copy; Domótica Web</p>
<button id="button3" type="button" >Actualizar </button>

</div>

</div>

<script type="text/javascript">
$( '#button1' ).click(function() {miAbrir()});
$( '#button2' ).click(function() {miCerrar()});
$( '#button3' ).click(function() {estado()});
</script>

</body>
</html>
```

## Anexo 29

### Archivos directorio control-manta

#### estado.php

```
<?php
//Comprueba el estado de la manta
$fp = fopen('/dev/ttyUSB0','r+'); //use this for Linux
fwrite($fp, "g"); //write string to serial
$x = fread($fp,1);
echo $x."\n";
fclose($fp);
?>
```

#### desactivar.php

```
<?php
$fp = fopen('/dev/ttyUSB0','r+'); //use this for Linux
fwrite($fp, "f"); //write string to serial
$x = fread($fp,1);
echo $x."\n";
fclose($fp);
?>
```

#### tiempo.php

```
<?php
$fp = fopen('/dev/ttyUSB0','r+'); //use this for Linux
fwrite($fp, "d"); //write string to serial
$x= fgets($fp);//Recibe toda la cadena
//echo $x."\n";
if($x==3)
{
    echo "Problemas de comunicación, intente nuevamente.";
}
else
{
    $segundos=$x/1000;
    //$seconds = 8525;
    if($segundos < 60)
    {
        echo "Manta activa y se apagará en: ".gmdate('i:s', $segundos) ." segundos";
    }
    else
    {
        echo "Manta activa y se apagará en: ".gmdate('i:s', $segundos) ." minutos";
    }
}

fclose($fp);
?>
```

## Anexo 30

### Código fuente página manta-termica.php

```
<?php session_start(); require 'cookie.php';?>
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>Domótica Web</title>
<link rel="stylesheet" type="text/css" href="jquery.mobile-1.3.2.css">
<script type="text/javascript" src="jquery-1.10.2.min.js"></script>
<script type="text/javascript" src="jquery.mobile-1.3.2.js"></script>

<!--Script para ejecutar php sin recargar la pagina
Esta es la pagina principal
-->
<script>

function configurar()
{
//Direcciona el XBEE B
$.ajax({
url: 'control/direccionB.php',
type: 'post',
beforeSend: function () {
$("#Carga").html("Cargando...");
},
success: function (response) {
$("#Carga").html(response);
var estado = $("#Carga").html();
//alert(estado[1]);
//alert(estado);
if(estado[1]=='O'){
if(estado[2]=='K'){
$("#Carga").html("Cargado");
}
}
},
complete : function (){
//alert("configuracion correcta");
estado();
}
});
}

function estado(){

$.ajax({
url: 'control/control-manta/estado.php',
type: 'post',
beforeSend: function () {
//$("#Estado").html("");
},
success: function (response) {
//$("#Estado").html(response);
var x=response;
//alert(x[0]);
if( x[0]==1){
$("#button1").button('disable');
$("#button2").button('enable');
miTiempo();
}

if( x[0]==0){
$("#button1").button('enable');
$("#button2").button('disable');
$("#Estado").html("La manta esta desactivada.");
}
if( x[0]==3){
$("#Estado").html("Problemas de comunicación, intente nuevamente.");
}

},
complete : function (){
```

```

//miTiempo();

}
});
}

function miTiempo()
{
$.ajax({
url: 'control/control-manta/tiempo.php',
type: 'post',
beforeSend: function () {
$("#Estado").html("");
},
success: function (response) {
$("#Estado").html(response);
//var x = $("#Estado").html();
//alert(x);

},
complete : function (){

}

});
}

function miActivar()
{
//alert("Activaste la funcion");
$.ajax({
url: 'control/control-manta/activar.php',
type: 'post',
beforeSend: function () {
$("#Estado").html("");
},
success: function (response) {
//$("#Estado").html(response);
//var x = $("#Estado").html();
//alert(x);
var x=response;
if ( x[0]==1){
$('#button1').button('disable');
$('#button2').button('enable');
miTiempo();
}
if ( x[0]==3){
$("#Estado").html("Problemas de comunicación, intente nuevamente.");
}

},
complete : function (){

}

});
}

function miDesactivar()
{
//alert("Activaste la funcion");
$.ajax({
url: 'control/control-manta/desactivar.php',
type: 'post',
beforeSend: function () {
$("#Estado").html("");
},
success: function (response) {
//$("#Estado").html(response);
//var x = $("#Estado").html();
//alert(x);
var x=response;
if ( x[0]==0){
$('#button1').button('enable');
$('#button2').button('disable');
$("#Estado").html("La manta esta desactivada.");
}
}
}

```

```

if( x[0]==3){
$("#Estado").html("Problemas de comunicación, intente nuevamente.");
}

},
complete : function (){

}

});
}

</script>

<script language="javascript" type="text/javascript">

$(function() {
configurar();
});
</script>

</head>
<body>

<div data-role="page" data-theme="c">
<!--cabecera-->
<div data-role="header">
<h2>Manta térmica</h2>
<a href="lugares.php" data-theme="a" data-transition="Pop" data-ajax="false">Áreas</a>
<a href="logout.php" data-theme="a" data-transition="Pop" data-ajax="false">Salir</a>
</div>

<!--contenido-->
<div data-role="content" data-theme="c">
<div>
<label data-inline ="True">Manta:</label>
<label id="Carga" data-inline ="true"></label>
</div>
<div>
<label data-inline ="True">Estado:</label>
<label id="Estado" data-inline ="true"></label>
</div>

<!--Menu colapsable opciones manta-->
<div data-role="collapsible" data-theme="c" data-content-theme="d" data-collapsed="false">
<h4>Configuración Manta</h4>

<!--<form action="" method="post" data-ajax="false"> -->

<button id="button1" type="button">Activar </button>
<button id="button2" type="button">Desactivar </button>

<!--</form> -->

</div>
</div>

<!--pie de pagina-->
<div data-role="footer" data-theme="d" data-position="fixed">
<p>&copy; Domótica Web</p>
<button id="button3" type="button" >Actualizar </button>

</div>

</div>

<script type="text/javascript">
$('#button1').click(function() {miActivar()});
$('#button2').click(function() {miDesactivar()});

```

```
$('#button3').click(function() {estado()});  
</script>
```

```
</body>  
</html>
```

## Anexo 31

### Materiales y costos.

#### Nodo central

Material	Precio
-Raspberry pi	40\$
-Fuente 5v 1A	3\$
-Módulo xbee s1	32\$
-Cable USB	1\$
-Base Xbee USB	5\$
-Materiales varios	3\$
Total: <b>84\$</b>	

#### Nodo intermedio

Material	Precio
-Arduino uno	5,30\$
-Shield xbee	2,56\$
-Xbee s1	32\$
-Fuente 6.8v	5,40\$
-NRF24L01	1,61\$
-Materiales varios	2\$
Total: <b>48,87\$ x 2 nodos = 97,74\$</b>	

#### Nodo lamparas

Materiales	Precio
-Arduino pro mini	1,71\$
-NRF24L01	1,61\$
-Triack bta12600	1\$
-Moc3021	1,20\$
- C 2,2uF	2\$
-Materiales varios	5\$
Total: <b>12,52\$ x 11 nodos = 137,72\$</b>	

### Nodo manta

<b>Materiales</b>	<b>Precio</b>
-Arduino pro mini	1,71\$
-NRF24L01	1,61\$
-Triack bta12600	1\$
-Moc3021	1,20\$
- C 2,2uF	2\$
-Materiales varios	10\$
<b>Total:17,52\$</b>	

### Nodo cortina

<b>Materiales</b>	<b>Precio</b>
-Arduino pro mini	1,71\$
-NRF24L01	1,61\$
-2 drivers L293D	1\$
-motor M49SP-2K	10\$
-Rodamientos	2\$
-Piezas mecánicas extras	10\$
-Materiales varios electrónicos	5\$
<b>Total:31,82\$ x 3 nodos = 95,46\$</b>	

### Nodo Puerta

<b>Materiales</b>	<b>Precio</b>
-Arduino UNO	5,30\$
-NRF24L01	1,61
-4 irfz44n	1.20\$
-Modulo lm2596s	2\$
-Fuente 19.5V 3A	30\$
-Motor nema 17	8\$
-Materiales varios	15\$
<b>Total:63,11\$ x 2 nodos = 126,22\$</b>	

**Nodo Ventana**

<b>Materiales</b>	<b>Precio</b>
-Arduino UNO	5,30\$
-NRF24L01	1,61
-4 irfz44n	1.20\$
-Fuente ATX	12\$
-Motor nema 17	8\$
-Materiales varios	15\$
<b>Total: 43,11\$ x 3 nodos = 43,11\$</b>	

Materiales varios para la instalación de los nodos: 30\$

**Costo total para la instalación de las áreas propuestas:**

Nodo central	84\$
Nodos intermedios	97,74\$
Nodo manta	17,52\$
Nodos lámpara	137,72\$
Nodos puerta	126,22\$
Nodos ventana	129,33
Nodos cortinas	95,46\$

**Sumatorio total del sistema domótico instalado: 687,99\$**

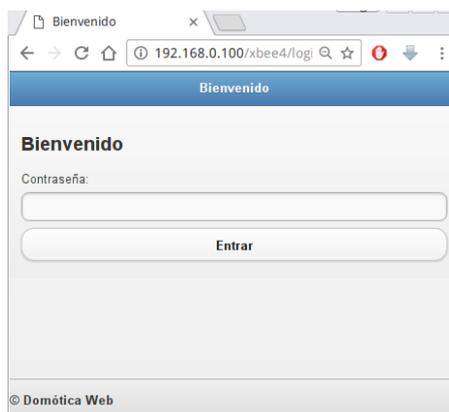
## Anexo 32

### Manuales de usuario y administrador.

#### -Manual de usuario.

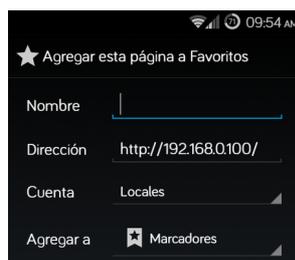
**Caso 1:** El usuario se encuentra en el lugar donde están los dispositivos.

1. Proceda a conectarse a la red wifi llamada TOTOLINK N300RH.
2. Abra el navegador de su preferencia y ponga en la barra de direcciones el siguiente número: 192.168.0.100 a continuación presione enter o ir. Esto abrirá la interfaz de usuario.

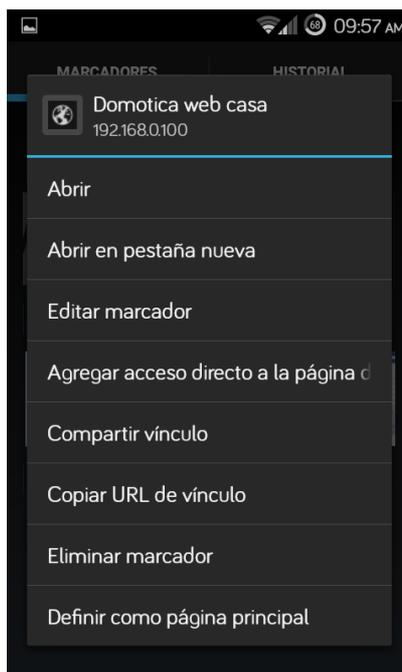


Aquí se recomienda crear un acceso directo hacia la página web en el escritorio del dispositivo. En el caso de Android siga la siguiente secuencia de pasos.

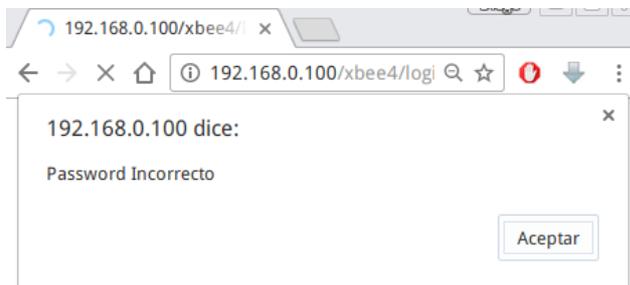
- Toque el botón de opciones y vaya a Guardar en marcadores, ponga un nombre distintivo para este marcador.



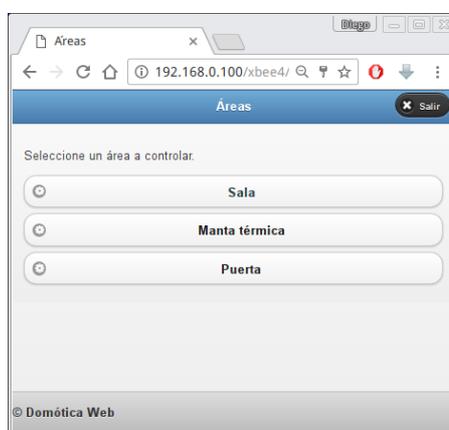
- Salga, ahora nuevamente toque el botón opciones y vaya a Marcadores.
- Elija el marcador deseado y sostenga brevemente el icono del marcador anteriormente creado.



- Ahora busque la opción de Agregar acceso directo a la página de inicio. Con esto se habrá agregado un icono en la pantalla de inicio que conducirá rápidamente a la página de la aplicación.
3. Ingrese su clave de seguridad. Si aparece el siguiente mensaje, quiere decir que la clave ha sido errónea.

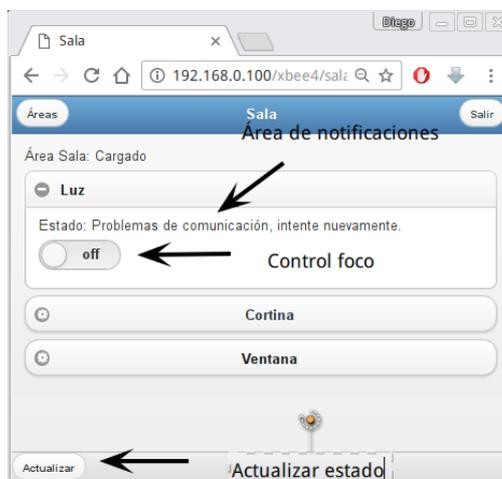


4. Si la clave es correcta aparecerá la siguiente pantalla.



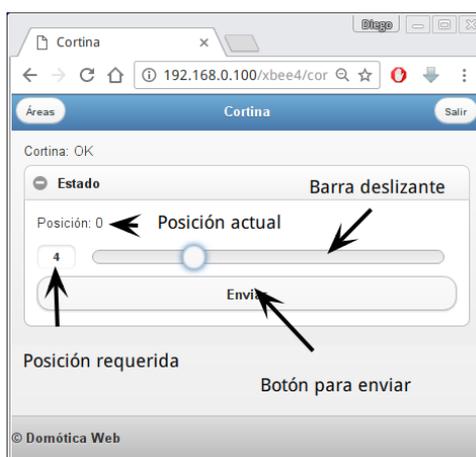
Esta pantalla contiene las áreas y los dispositivos a los que se pueden acceder, para acceder a cualquier dispositivo o área solo presiones el botón correspondiente.

5. En el caso de entrar en la sala, aparecerá el siguiente contenido.



Esta es la pantalla correspondiente a el área sala, en la ilustración se indican las partes que conforman esta página, como los dispositivos que se encuentran instalados en ella. La ventana y cortina, pueden ser accedidos desde aquí.

6. Continuando, si se entra en la cortina aparecerá la siguiente pantalla:



Aquí se observan los controles correspondientes a la cortina, lo mas destacable de esta pantalla es el control deslizable que permite mover la cortina a voluntad, cuando se haya seleccionado el valor deseado se debe presionar el botón enviar para que los cambios surtan efecto.

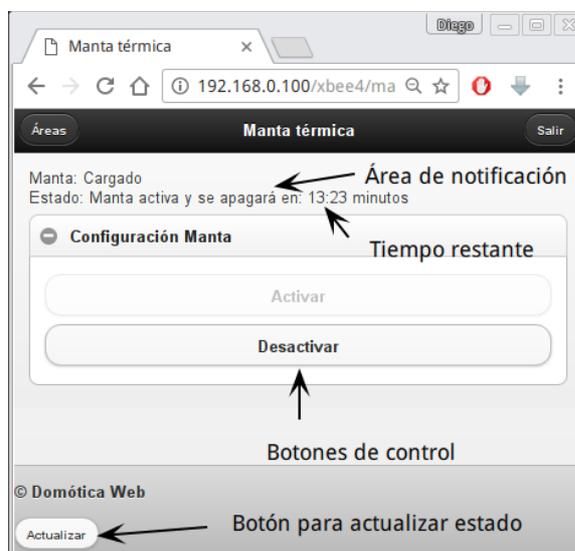
7. En el caso de entrar hacia la ventana, aparece la siguiente pantalla



Como se observa aquí se puede observar en uso el área de notificación, en este caso indica que la ventana esta cerrada, motivo por el cual el botón cerrar se encuentra deshabilitado, finalmente en la parte inferior se puede ver el botón actualizar que permite recuperar el estado actual de la ventana.

*NOTA: El usuario puede regresar a la pantalla áreas o salir en cualquier momento de la interfaz, haciendo uso de los botones ubicados en la parte superior Salir y Áreas.*

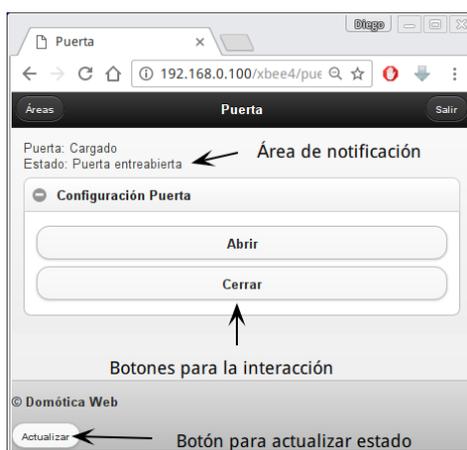
8. Si el usuario regresa a la pantalla Áreas puede acceder a los últimos dos dispositivos que aún no se describen, estos son la manta eléctrica y la puerta.
9. En el caso de la manta eléctrica aparecerá la siguiente pantalla:



Aquí se observan los controles correspondientes con el dispositivo que controla a la manta, en este caso el área de notificación muestra un mensaje indicando que la manta esta activa y seguido muestra el tiempo en el cual se desactivará. De manera similar a otros dispositivos, depende el estado actual de la manta para que los controles se habiliten, en

este caso el botón Activar se encuentra deshabilitado por que la manta ya está activa. Finalmente el tiempo de desconexión de la manta se puede ir actualizando al presionar el botón actualizar, ubicado en la parte inferior.

10. Por último se tiene la pantalla puerta, que se presenta de la siguiente manera:



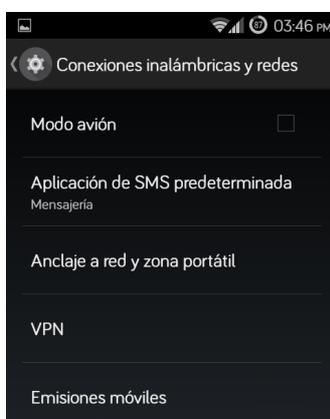
Como se observa esta pantalla es similar a la pantalla ventana, teniendo el mismo funcionamiento con controles idénticos. Aquí se observa que la puerta esta entreabierta, por lo cual se encuentran activos los dos botones Abrir y Cerrar.

*NOTA FINAL: Si el usuario recibe un mensaje en el área de notificación indicando que no se puede establecer conexión o un error similar, por favor actualizar la página, si le problema persiste seguramente el nodo se encuentra desconectado o el nodo intermedio correspondiente también esta desconectado. Si este es el caso proceda a verificar que todos los dispositivos estén conectados a la red eléctrica.*

**Caso 1:** El usuario desea conectarse desde Internet.

Para la configuración de cualquier dispositivo que requiera conexión externa por favor revise el Anexo 5 correspondiente a la conexión mediante hamachi o en su defecto comuníquese con el administrador del sistema.

- Una vez que el dispositivo se haya ingresado a la VPN se procede a ir a la configuración del equipo, en el apartado de conexiones se busca la la opción VPN.



- En esta opción se elige la red VPN correspondiente a hamachi.



Cuando la conexión se haya establecido aparecerá una llave en el barra de notificaciones, al pulsar en esta llave parecen la red a la que se esta conectado y las estadísticas de conexión.

- Finalmente ya se puede acceder a la interfaz para el control de los dispositivos apuntando a la siguiente dirección 25.64.10.104.9.

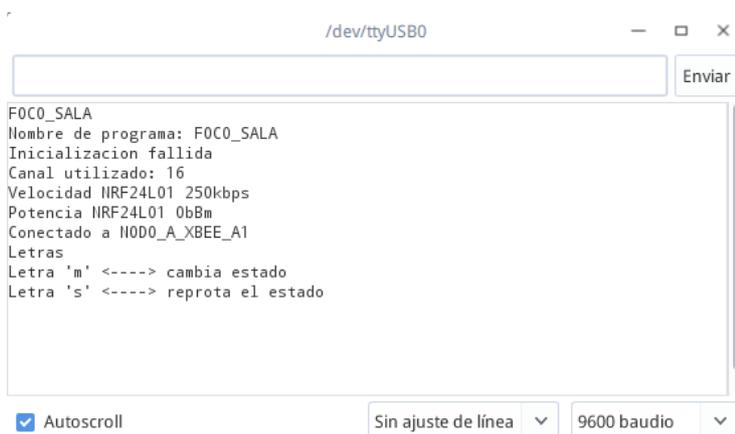


- Para poder acceder rápidamente a esta interfaz se recomienda hacer lo mismo que en el caso anterior, es decir crear un acceso directo en la pantalla de inicio.

## Manual de administrador.

Las indicaciones generales para el administrador son las siguientes.

- El sistema al estar conformado por muchos dispositivos, necesita que todos estén conectados a la red eléctrica. Muchas fallas se pueden mal interpretar como una falla del sistema pero en realidad se debe a que algún dispositivo está desconectado.
- Los dispositivos están ubicados de tal manera que no se superan los alcances de los módulos inalámbricos, por lo que los errores de comunicación son muy poco comunes.
- Los programas grabados en las placas Arduino de los dispositivos cuentan con una sección, que permite saber de qué dispositivo se trata al conectarlo por un puerto serie a una computadora. De esta manera cada vez que las placas son reiniciadas siempre envían esta información para que el administrador sepa de qué dispositivo se trata. Aquí se muestra un ejemplo.



```
 /dev/ttyUSB0
FOCO_SALA
Nombre de programa: FOCO_SALA
Inicializacion fallida
Canal utilizado: 16
Velocidad NRF24L01 250kbps
Potencia NRF24L01 0dBm
Conectado a H0D0_A_XBEE_A1
Letras
Letra 'm' <----> cambia estado
Letra 's' <----> reprot el estado
```

Autoscroll  Sin ajuste de línea 9600 baudio

- Los programas y las respectivas librerías que utiliza arduino se encuentran en el disco adjunto a este documento.
- Los archivos que utiliza la placa Rapberry pi se encuentran en el directorio

**/var/www/html/xbee4**

Dentro de este directorio se encuentra la totalidad de los archivos utilizados por el sistema, aquí se encuentran los archivos de JQuery mobile que utiliza el sistema para visualizar las ventanas, así también los archivos PHP que se ejecutan al presionar los botones y controles de la interfaz. El directorio **/var/www/html/xbee4/control** posee permisos de ejecución y todos los archivos dentro del directorio xbee4 pertenecen al usuario www-data.

- Para realizar cambios en la interfaz de una manera rápida se utilizan scripts contruidos que permiten el fácil reemplazo y la correcta configuración de permisos de cada directorio. Los scripts son los siguientes y siempre se deben ejecutar como súper usuario y fuera del directorio xbee4.
  - editar.sh .- este script copia la carpeta xbee4 al directorio personal del usuario y luego cambia el propietario de la carpeta al del usuario sin privilegios.

Aquí el código:

```
#!/bin/bash
echo "Copiando archivos para ser editados"
rm -r /home/pi/xbee4
cp -r /var/www/html/xbee4 /home/pi
chown -R pi:pi /home/pi/xbee4
echo "Archivos copiados con éxito"
```

- importar.sh .- este script hace un respaldo de la carpeta **/var/www/html/xbee4/**, borra la carpeta **/var/www/html/xbee4/** ,copia los nuevos archivos ya

modificados desde la carpeta personal y finalmente configura el nuevo propietario y los permisos respectivos.

Aquí el código:

```
echo "Copiando respaldo"
rm -r /var/www/html/xbee-resp
cp -r /var/www/html/xbee4 /var/www/html/xbee-resp

echo "Borrando archivos anteriores"
rm -r /var/www/html/xbee4

echo "Copiando archivos nuevos"
cp -r /home/pi/xbee4 /var/www/html

echo "Configurando permisos"
sudo chown -R www-data:www-data /var/www/html/xbee4
chmod 775 -R /var/www/html/xbee4/control/

echo "Archivos copiados con éxito"
```

- restaurar.sh.- este archivo permite recuperar la copia anterior y restaurarla en caso de fallo o un error mayor. Si se ejecuta este script y se desea editar los archivos nuevamente se debe ejecutar antes el script editar.sh.

Aquí el código:

```
#!/bin/bash
echo "Borrando archivos defectuosos"
rm -r /var/www/html/xbee4

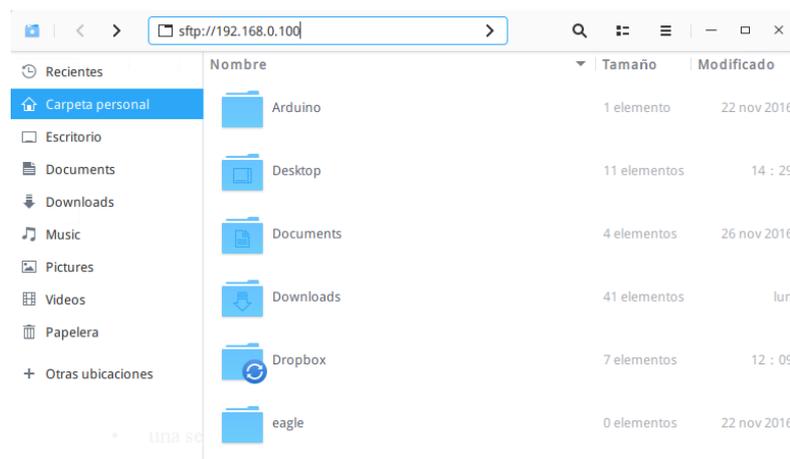
echo "Restaurando archivos"
cp -r /var/www/html/xbee-resp /var/www/html/xbee4

echo "Configurando permisos"
sudo chown -R www-data:www-data /var/www/html/xbee4
chmod 775 -R /var/www/html/xbee4/control/

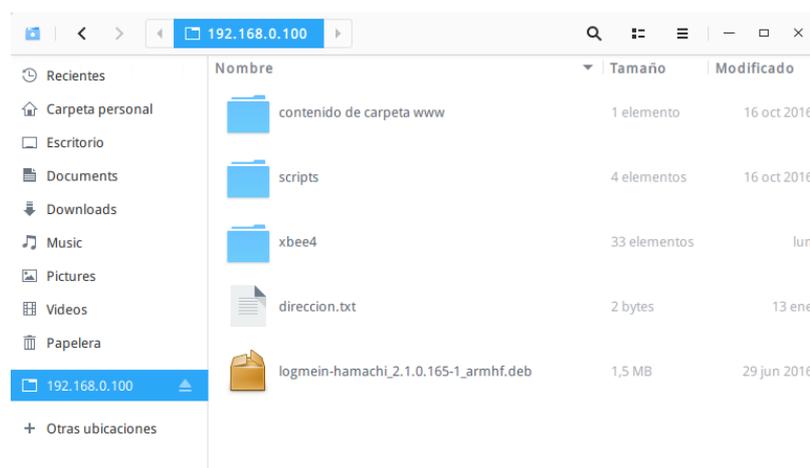
echo "Restauración completa"
```

Con estos scripts se pueden hacer cambios de manera rápida en la interfaz mediante la edición de los archivos. Aquí se indica como.

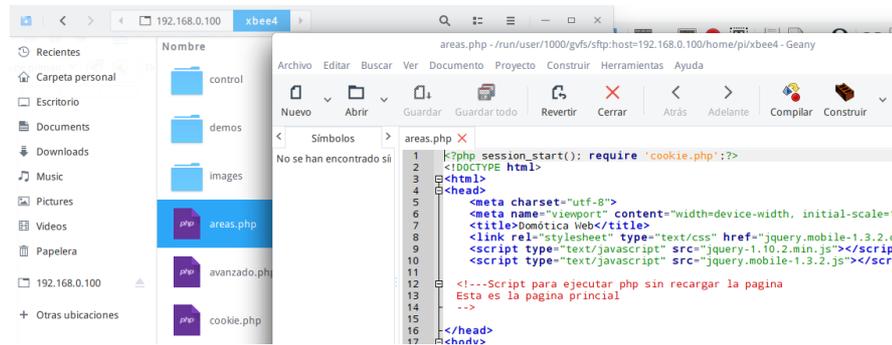
1. Conectese a la red, inicie una sesión SSH en un terminal como root y ejecute el script editar.sh (los scripts pueden ejecutarse desde cualquier directorio a excepción de el indicado anteriormente), ahora abra el navegador de archivos que soporte el protocolo sftp y ponga la esto:



2. El programa pedirá los credenciales ssh y mostrará los archivos alojados en la Raspberry pi en el navegador así:



3. Cuando ya se tienen los archivos disponibles ya se pueden editar con cualquier editor, personalmente se recomienda utilizar el programa GEANY .



4. Cuando los archivos hayan sido editados se guarda los cambios y se ejecuta el script `importar.sh` desde la sesión SSH anteriormente abierta. Para visualizar los cambios basta con actualizar la pagina en cuestión en un navegador.

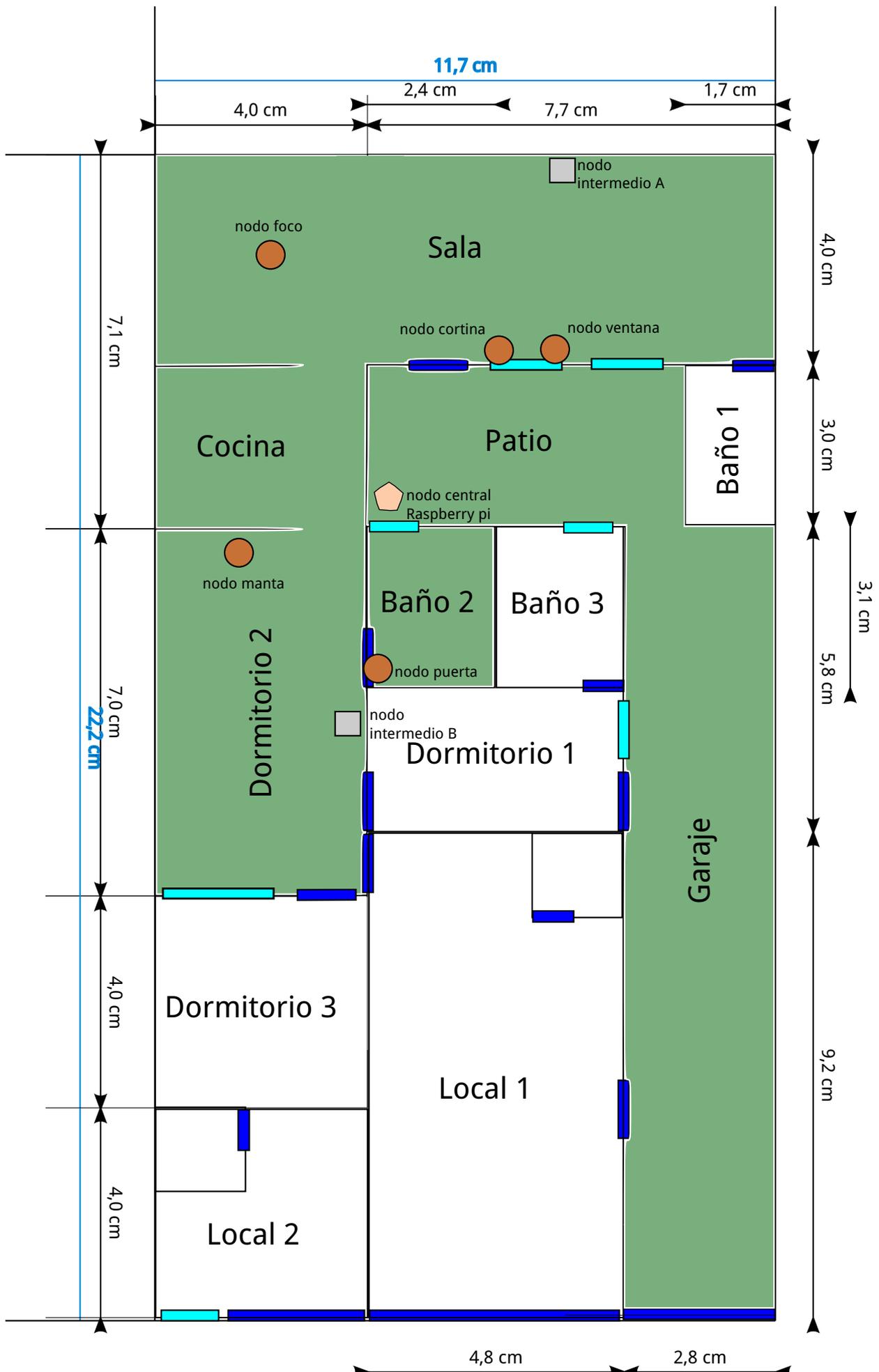
De esta manera se puede editar cualquier archivo contenido en el directorio `/var/www/html/xbee4/`.

Se recomienda encarecidamente tener cuidado con los parámetros que se mueven en los diferentes archivos, siempre es bueno tener un respaldo aparte en caso de algún error grave.

- Finalmente se recomienda leer detenidamente el Anexo 5, en caso de que el usuario requiera conectar algún dispositivo nuevo desde Internet.

## **Anexo 33**

**Distribución de dispositivos en el domicilio.**



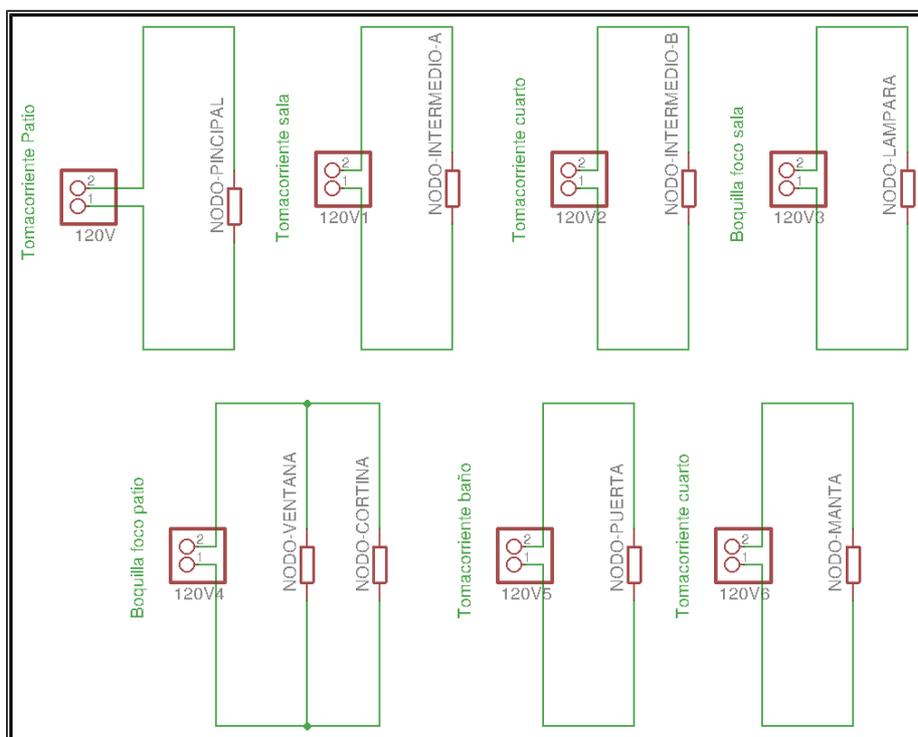
- Accesos o puertas
- Áreas más utilizadas
- Ventanas

- Nodo central Raspberry pi
- Nodos intermedios
- Dispositivos finales

Escala 1cm : 1m

## Anexo 34

### Diagrama eléctrico de las cargas aumentadas en el domicilio.



El diagrama eléctrico muestra las cargas que se aumentaron al domicilio luego de la instalación del sistema, a continuación la lista de cargas en vatios.

Nodo Principal	3,5W
Nodo intermedio A	0,6W
Nodo intermedio B	0,6W
Nodo lámpara	0,25W
Nodo ventana	30W
Nodo cortina	14,4W
Nodo puerta	47,5W
Nodo manta	0,25W
Total:	97,1W

## **Anexo 35**

**Fuentes de voltaje resistivas y capacitivas.**

---

---

## Fuente de Alim. Sin-Transformador: Capacitiva y Resistiva

---

---

<i>Author: Reston Condit Microchip Technology Inc.</i>
--

### INTRODUCCIÓN

Hay varias maneras de convertir un voltaje de CA de la pared en el voltaje de C.C. requerido por un microcontrolador. Usualmente, esto se ha hecho con un transformador y circuito de rectificador. Hay también soluciones de fuente de alimentación de conmutación, sin embargo, en aplicaciones que involucran suministrar un voltaje de CC al microcontrolador y otros dispositivos de bajas corrientes, fuentes de alimentación basados en transformador o conmutación pueden no ser rentables. La razón es que los transformadores en soluciones en base al transformador, y el (inductor / MOSFET/controlador) en soluciones en base a la conmutación, es caro y ocupa mucho espacio. Esto es especialmente verdad en el mercado del electrodoméstico, donde el costo y el tamaño de los componentes que rodean la fuente de alimentación puede ser mucho menor que el costo de sólo la fuente de alimentación.

Las fuentes de alimentación de Sin-Transformador proporcionan una alternativa económica de fuentes de alimentación basado en el transformador y basado en la conmutación. Los dos tipos básicos de fuentes de alimentación sin transformador son resistivos y capacitivos. Esta nota de la aplicación discutirá los dos con un enfoque en lo siguiente:

1. Un análisis del circuito de alimentación.
2. Las ventajas y desventajas de cada fuente de alimentación.
3. Consideraciones adicionales incluso los requisitos de seguridades e intercambios asociados con la rectificación de media onda contra la de onda completa.

<b>Advertencia:</b> <i>Un riesgo de la electrocución existe al experimentar con circuitos sin transformador conectados a la pared. No hay el transformador para el aislamiento de la energía de línea en los circuitos siguiente, así el usuario será muy cuidadoso y evaluará los riesgos de transitorios de línea en la aplicación del usuario. Un transformador de aislamiento debe usarse al examinar los circuitos siguientes.</i>
---

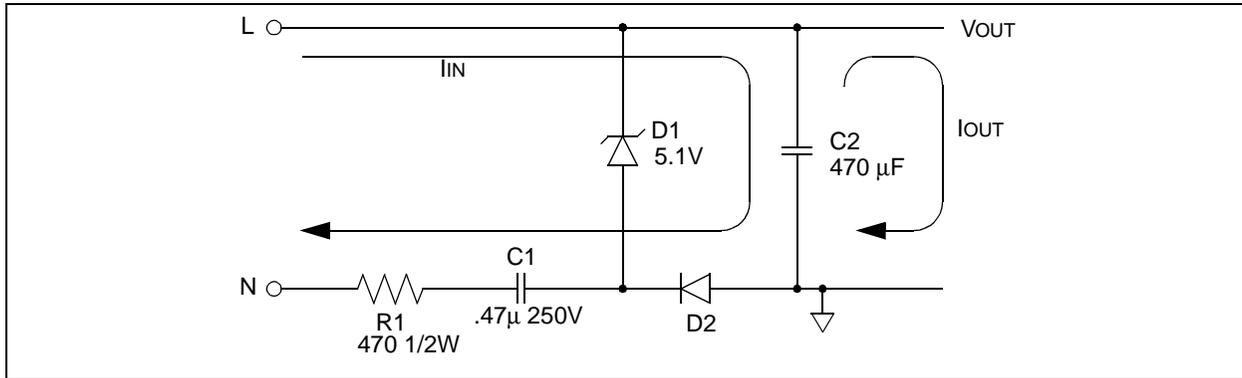
# AN954

## FUENTE DE ALIMENTACIÓN CAPACITIVA SIN TRANSFORMADOR

Una fuente de alimentación sin transformador capacitiva se muestra en la Figura 1. El voltaje en la carga permanecerá constante siempre y cuando la corriente de (IOUT) es menor o igual a la corriente de entrada (IIN). IIN está limitado por R1 y la reactancia de C1.

**Note:** R1 limita la corriente de irrupción. El valor de R1 es escogido que para no disipe mucha potencia, es bastante grande para limitar corriente de la irrupción.

**FIGURA 1: FUENTE DE ALIMENTACIÓN CAPACITIVA**



IIN es dada por:

**ECUACIÓN 1:**

$$I_{IN} = \frac{V_{HFRMS}}{X_{C1} + R1} \geq I_{OUT}$$

VHFRMS es el voltaje RMS de una media onda de una onda seno de CA y X<sub>C1</sub> es la reactancia de C1.

**ECUACIÓN 2:**

$$V_{HFRMS} = \frac{V_{PEAK} - VZ}{2} = \frac{\sqrt{2}V_{RMS} - VZ}{2}$$

VPEAK es el voltaje pico del tomacorriente, VRMS es el voltaje del tomacorriente (p.e., Estados Unidos: 115 VCA, Europa: 220 VCA) y Vz es la caída de voltaje en D1.

**ECUACIÓN 3:**

$$X_{C1} = \frac{1}{2\pi f C1}$$

f es la frecuencia (p.e., Estados Unidos : 60 Hz, otros países : 50 Hz).

Sustituyendo Ecuación 2 y Ecuación 3 en Ecuación 1 resulta en:

**ECUACIÓN 4:**

$$I_{IN} = \frac{\sqrt{2}V_{RMS} - VZ}{2 \left( \frac{1}{2\pi f C1} + R1 \right)}$$

El valor mínimo de  $I_{IN}$  se calcula para la aplicación, mientras el valor máximo de  $I_{IN}$  se calcula para los requisitos de potencia de los componentes individuales.

## EJEMPLO 1: CÁLCULO de $I_{IN}$ MÍNIMO POSIBLE

Asumir valores mínimos de todos componentes excepto  $V_Z$ ,  $R_1$ . Tome valores máximos de  $V_Z$ ,  $R_1$ .

- $V_{RMS} = 110 \text{ VAC}$
- $V_Z = 5.1 \text{ V}$
- $f = 59.5 \text{ Hz}$
- $C = C_1 = 0.47 \mu\text{F} \times 0.8 = 0.38 \mu\text{F}$   
(asumir el capacitor de  $\pm 20\%$ )
- $R = R_1 = 470 \times 1.1 = 517$  (asumir el resistor  $\pm 10\%$ )
- $I_{INMIN} = 10.4 \text{ mA}$

## EJEMPLO 1: CÁLCULO de $I_{IN}$ MÁXIMO POSIBLE

Asumir valores máximos de todos componentes excepto  $V_Z$ ,  $R_1$ . Tome valores mínimos de  $V_Z$ ,  $R_1$ .

- $V_{RMS} = 120 \text{ VAC}$
- $V_Z = 5 \text{ V}$
- $f = 60.1 \text{ Hz}$
- $C = C_1 = 0.47 \mu\text{F} \times 1.20 = 0.56 \mu\text{F}$   
(asumir el capacitor  $\pm 20\%$ )
- $R = R_1 = 470 \times 0.9 = 423$  (asumir el resistor  $\pm 10\%$ )

$I_{INMAX} = 16.0 \text{ mA}$

$V_{OUT}$  está dado po:

### ECUACIÓN 5:

$$V_{OUT} = V_Z - V_D$$

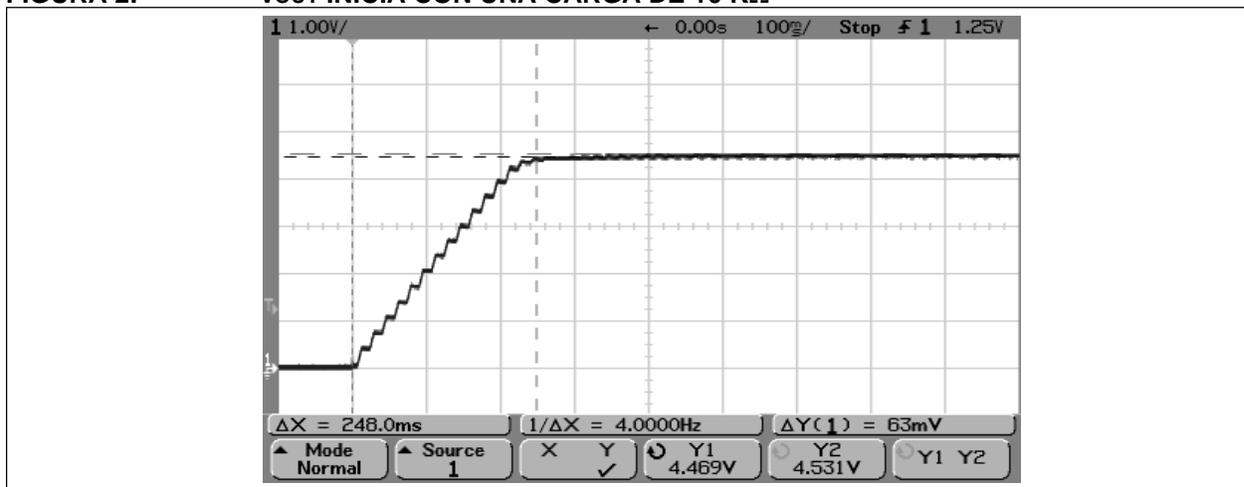
Donde  $V_D$  es la caída de voltaje continuo en  $D_2$ .

Se asume un diodo zéner de 5.1V y una caída de 0.6V en  $D_2$ , el voltaje de salida estará cerca de 4.5V. Esto está bien dentro de la especificación de voltaje para los microcontroladores PIC®.

### OBSERVACIONES

Figure 2 muestra una trama de osciloscopio de  $V_{OUT}$  inicia con una carga 10 kΩ en la salida (entre  $V_{OUT}$  y tierra.) La carga 10 kΩ delinea sólo 0.45 mA. Como resultando, el tiempo de subida de  $V_{OUT}$  es 280 ms (tan veloz como se puede para  $I_{IN}$  y  $C_2$  dados), el rizo es mínimo cuando  $V_{OUT}$  estabiliza al voltaje calculado en la Ecuación 5, aproximadamente 4.5V.

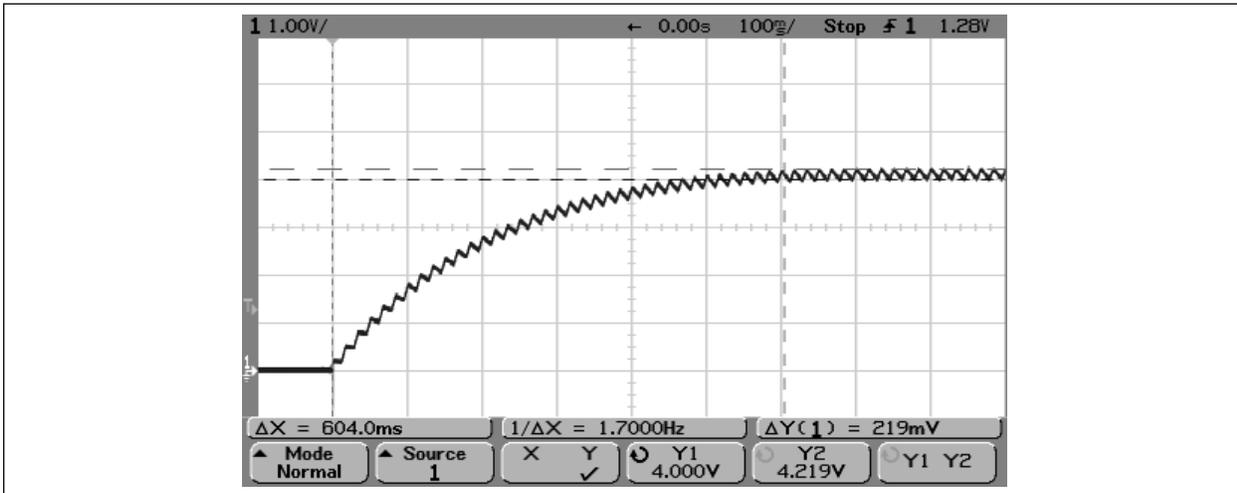
**FIGURA 2:  $V_{OUT}$  INICIA CON UNA CARGA DE 10 KΩ**



# AN954

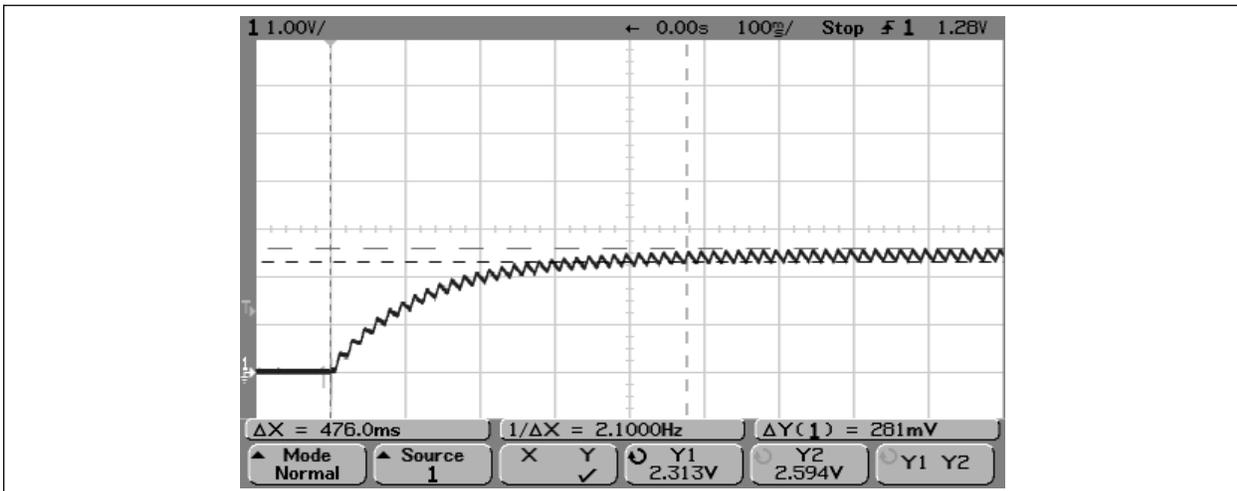
Si la carga se incrementa, el comportamiento del circuito cambia de varias maneras. Figura 3 muestra una trama del osciloscopio de  $V_{OUT}$  durante el mismo marco de tiempo para una carga de  $500\Omega$ . Una carga de  $500\Omega$  dibuja  $9\text{ mA}$  a  $4.5\text{V}$ . Esto está cerca del límite calculado en Ejemplo 1 de  $10.4\text{ mA}$ . El tiempo de subida de  $V_{OUT}$  es más largo ( $680\text{ ms}$ ) según lo esperado porque no sólo es  $I_{OUT}$  que carga  $C_2$ , sino una cantidad significativa de corriente está siendo arrastrado por la carga.  $V_{OUT}$  estabiliza a aproximadamente  $4.1\text{V}$ , cerca a cuatro décimo de  $1$  voltio debajo del voltaje de salida calculado en la Ecuación 5. El rizado en  $V_{OUT}$  es más pronunciado con el incremento de corriente.

**FIGURA 3:  $V_{OUT}$  INICIA CON UNA CARGA DE  $500\Omega$**



Si aún más la corriente es exigida desde el circuito, la alimentación estabilizará a un voltaje debajo del nivel deseado. La Figura 4, muestra una trama del osciloscopio de  $V_{OUT}$  durante el marco de tiempo igual para una carga de  $270\Omega$ . Una carga de  $270\Omega$  consumirá aproximadamente  $16\text{ mA}$  con un voltaje de salida de  $4.5\text{V}$ . Esta corriente no puede ser proporcionada por el circuito, por lo tanto, el voltaje de salida se compromete.

**FIGURE 4:  $V_{OUT}$  INICIA CON UNA CARGA DE  $270\Omega$**



## CONSIDERACIONES DE POTENCIA

Cálculo de la disipación de potencia de los componentes en el circuito es una consideración crítica. La regla, es seleccionar los componentes con potencias de por lo menos dos veces la potencia máxima calculada para cada parte. Para los componentes de CA, los valores RMS máximos de voltaje y corriente se usan para calcular los requisitos de la potencia.

### Valor de R1:

La corriente sobre R1 es la corriente onda completa. Esta corriente es equivalente al voltaje de la línea dividido por la impedancia de C1.

### ECUACIÓN 6:

$$P_{R1} = I^2 R = (V_{RMS} * 2\pi f C)^2 R I$$

$$= (21.3 \text{ mA})^2 (470 \Omega \times 1.1) = 0.23 \text{ W}$$

(asuma el resistor  $\pm 10\%$ )

Duplicando esto da 0.46W, así un resistor de 1/2W es suficiente.

### Valor de C1:

Asuma un voltaje de línea máximo de 120 VCA, el doble es 240V. Un capacitor de clase X2 250V es OK.

**Note:** La capacitor clase de X2 se usa en aplicaciones definidas por la categoría de la instalación II de IEC664. Esta categoría cubre aplicaciones de voltajes de 150 a 250 CA (nominal).

### Valor de D1:

D1 soportará más corriente sin carga presente. Asuma present. Asuma la condición del peor de los casos, D1 soportará casi la corriente de onda completa una vez que se carga C2. Esta corriente se calculó al hallar R1 (vea arriba).

### ECUACIÓN 7:

$$P_{D1} = I \times V = (21.3 \text{ mA})(5.1 \text{ V}) = 0.089 \text{ W}$$

Duplicando esto excede 1/4W, un diodo Zener de 5.1V a 1/2W es una buena opción.

### Valor de D2:

La corriente RMS máxima que fluirá sobre D2 se calculó en Ejemplo 2. Asuma una caída de 0.7V en el resistor para la mitad de la onda, la ecuación siguiente (arriba) aproxima la potencia disipada de D2.

### ECUACIÓN 8:

$$P_{D2} = I \times V = (16.0 \text{ mA})(0.7 \text{ V}) = 0.011 \text{ W}$$

Un rectificador de 1/8 W es OK para D2.

### Valor de C2:

C2 debe ser dos veces el voltaje del diodo Zener. En este caso, un capacitor electrolítico de 16V es OK. C2 almacena la corriente y la libera en la carga. Se halla según el rizado que es aceptable para VOUT. VOUT con decaimiento según Ecuación 9.

### ECUACIÓN 9:

$$V_{out} = V_d e^{-\frac{t}{RC}}$$

V<sub>D</sub> se calculó en Ecuación 5

## Ventajas y Desventajas

Ventajas de la Fuente de Alimentación Capacitiva:

1. Mucho más pequeña que una fuente de alimentación basada en transformador.
2. Más rentable que una fuente de alimentación basado en transformador o en conmutación.
3. Es más eficiente que una fuente de alimentación sin transformador resistiva (se discutirá después).

Desventajas de la Fuente de Alimentación Capacitiva:

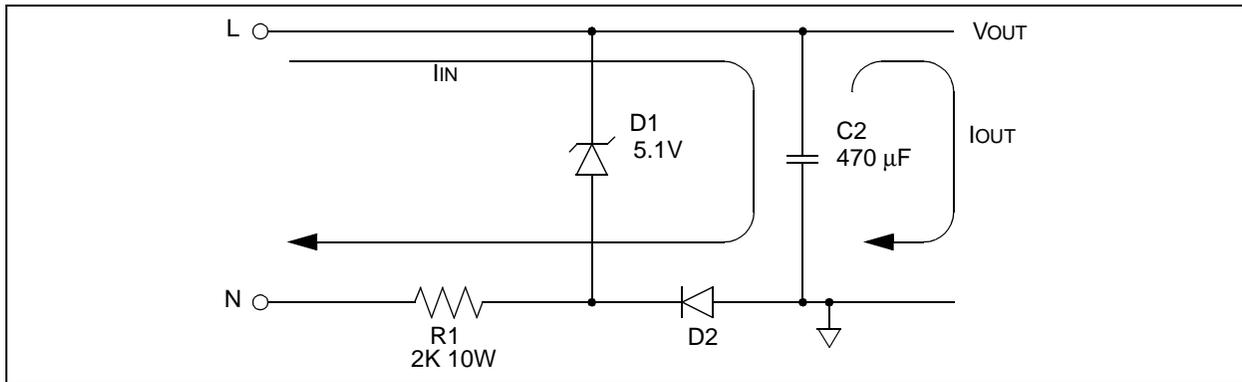
1. No aislado del voltaje de línea de CA que trae problemas de seguridad.
2. Mayor costo que una fuente de alim. resistiva.

# AN954

## FUENTE DE ALIMENTACIÓN SIN-TRANSFORMADOR RESISTIVA

Una fuente de alimentación del sin-transformador resistiva básica se muestra en la Figura 5. En lugar de usar una reactancia para limitar corriente, esta fuente de alimentación sólo usa un resistencia. Como con la fuente de alimentación capacitiva,  $V_{OUT}$  seguirá estable si la corriente de salida ( $I_{OUT}$ ) es menor o igual a la corriente de ( $I_{IN}$ .)

**FIGURA 5: FUENTE DE ALIMENTACIÓN RESISTIVA**



$I_{IN}$  es dada por:

**ECUACIÓN 10:**

$$I_{IN} = \frac{V_{HFRMS}}{R1} \geq I_{OUT}$$

$V_{HFRMS}$  es el voltaje RMS de una media onda de una onda seno de CA.

**ECUACIÓN 11:**

$$V_{HFRMS} = \frac{V_{PEAK} - V_Z}{2} = \frac{\sqrt{2}V_{RMS} - V_Z}{2}$$

$V_{PEAK}$  es el voltaje pico del tomacorriente,  $V_{RMS}$  es el voltaje del tomacorriente (p.e., Estados Unidos: 115VCA, Europa: 220VCA) y  $V_Z$  es la caída de voltaje en D1.

Substituting Equation 11 into Equation 10 results in:

**ECUACIÓN 12:**

$$I_{IN} = \frac{\sqrt{2}V_{RMS} - V_Z}{2R1}$$

El valor mínimo de  $I_{IN}$  se debe calcular para la aplicación mientras que el valor máximo de  $I_{IN}$  se debe calcular para los requisitos de potencia.

**EJEMPLO 3: CALCULO DEL MÍNIMO POSIBLE DE  $I_{IN}$**

Asuma valor mínimo de  $V_{RMS}$ . Assume maximum value of  $V_Z$  and R.

- $V_{RMS} = 110 \text{ VAC}$
- $V_Z = 5.1 \text{ V}$
- $R = R1 = 2 \text{ k}\Omega \times 1.1 = 2.2 \text{ k}\Omega$  (asuma el resistor  $\pm 10\%$ )

$I_{INMIN} = 34.2 \text{ mA}$

**EJEMPLO 4: CALCULO DEL MÁXIMO POSIBLE DE  $I_{IN}$**

Assume maximum value of  $V_{RMS}$ . Assume minimum value of  $V_Z$  and R.

- $V_{RMS} = 120 \text{ VAC}$
- $V_Z = 5 \text{ V}$
- $R = R1 = 2 \text{ k}\Omega \times 0.9 = 1.8 \text{ k}\Omega$  (asuma el resistor  $\pm 10\%$ )

$I_{INMIN} = 45.8 \text{ mA}$

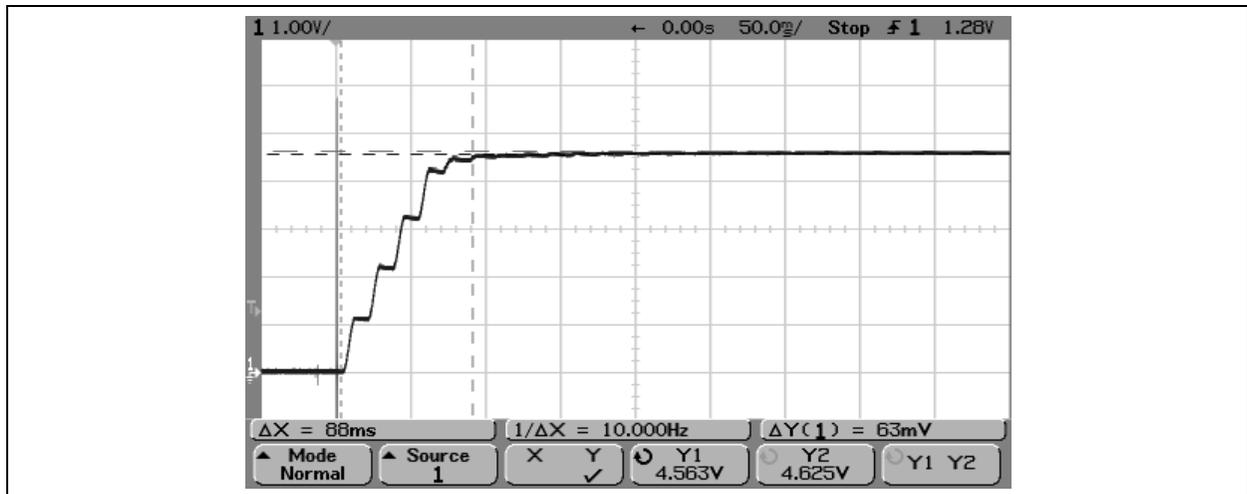
$V_{OUT}$  es igual que el dado para la fuente de alimentación capacitiva (vea Ecuación 5).

## OBSERVACIONES

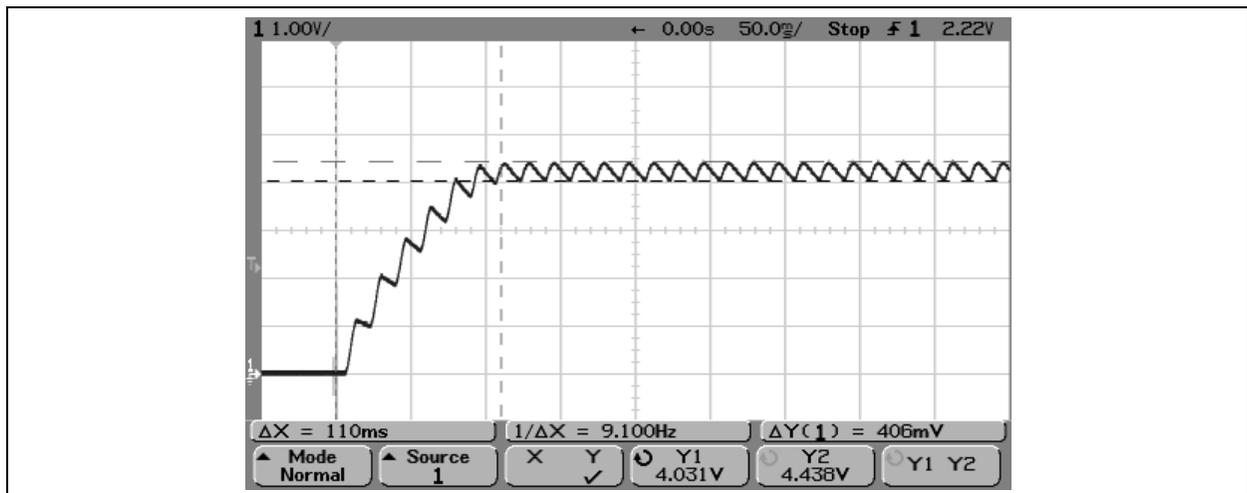
Las observaciones para la fuente de alimentación resistiva son muy similares a la fuente de alimentación capacitiva. Por favor refiérase a las "Observaciones" en la Sección "**Fuente de Alimentación Sin-Transformador Capacitiva**" para más detalles.

La Figura 6, la Figura 7 y la Figura 8 muestran  $V_{OUT}$  inicia para la fuente de alimentación resistiva con las cargas de  $10\text{ k}\Omega$ ,  $270\Omega$  y  $100\Omega$ , respectivamente. Estas cargas corresponden a las corrientes de salida de  $0.45\text{ mA}$ ,  $16\text{ mA}$  y  $45\text{ mA}$ , respectivamente, asumiendo un voltaje de salida de  $4.5\text{V}$ .  $V_{OUT}$  claramente no es  $4.5\text{V}$  en la Figura 6 porque la demanda de corriente puesta en la fuente de alimentación es demasiado alta.

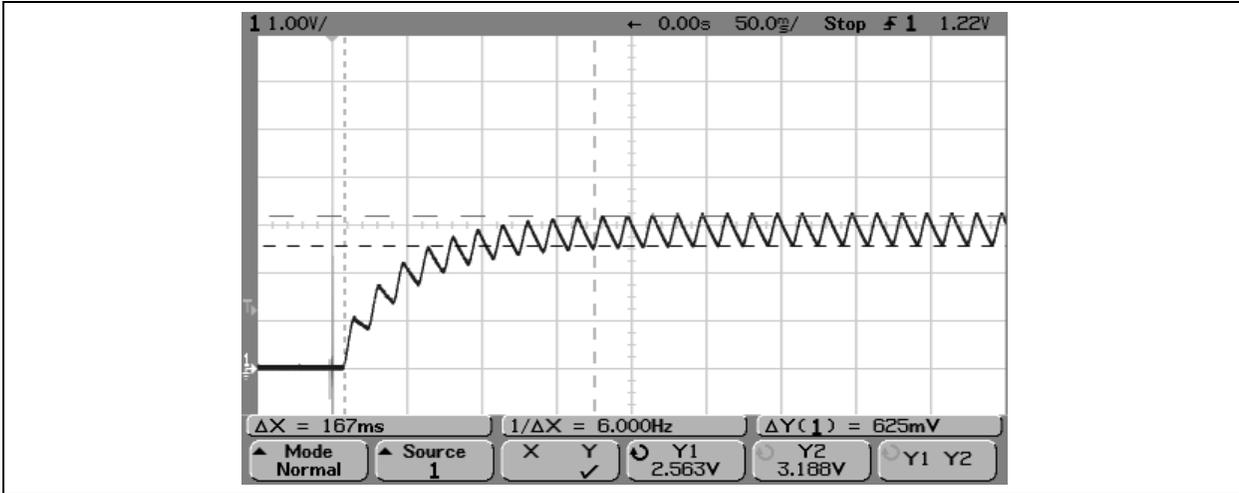
**FIGURA 6:  $V_{OUT}$  INICIA CON UNA CARGA DE  $10\text{ k}\Omega$**



**FIGURA 7:  $V_{OUT}$  INICIA CON UNA CARGA DE  $270\Omega$**



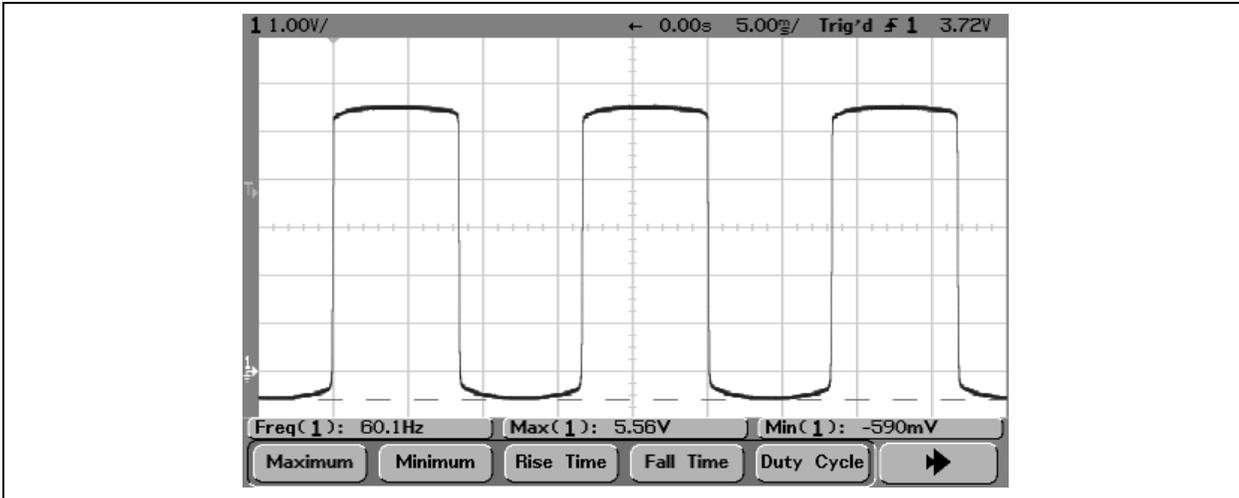
**FIGURA 8:  $V_{OUT}$  INICIA CON UNA CARGA DE  $100\Omega$**



Al trabajar con una fuente de CA a 60Hz, es a menudo deseable saber cuando el voltaje de línea cruza Neutro. El cruce, conocido como cruce por cero, puede ser capturado fácilmente conectando el nodo formado por D1, C1 y D2 a una entrada en el microcontrolador. La forma de onda observada a este nodo se muestra en la Figura 9.

Para la fuente de alimentación resistiva, la transición en esta forma de onda ocurre en el cruce por cero. Para las alimentaciones capacitivas, un poco de retraso está presente debido al capacitor en serie (C1 en la Figura 1).

**FIGURA 9: FIGURA A: FORMA DE ONDA EN EL NODE DE CRUCE POR CERO**



## CONSIDERACIONES DE POTENCIA

Elegir la potencia del componente en el circuito es una consideración crítica. Una regla general, el valor del componente debe ser de dos veces la potencia máxima calculada para cada dispositivo. Para los componentes de CA, se usan los valores de voltaje y corriente RMS para calcular los requisitos de potencia.

### Valor de R1:

#### ECUACIÓN 13:

$$P_{R1} = I^2 R = \frac{V^2}{R}$$

$$\left( \frac{120^2}{2 \text{ k}\Omega \times 0.9} \right) = 8W$$

(asuma un resistor  $\pm 10\%$ )

A Un resistor de 10W incorpora en 2 vatios la seguridad así que será usado.

### Valor de D1:

Sin carga, la corriente en D1 será casi igual a la corriente de la onda completa en R1.

#### ECUACIÓN 14:

$$P_{D1} = V_x I = V_z \frac{V_{RMS}}{R1}$$

$$5.1V \left( \frac{120}{2 \text{ k}\Omega \times 0.9} \right) = 0.34W$$

Un diodo zéner de 5.1V 1W debe usarse.

### Valor de D2:

La corriente RMS máxima que fluirá en D2 se calculó en Ejemplo 4. Asumiendo una caída de 0.7V en el resistor para la mitad la onda, la ecuación siguiente (arriba) aproxima la potencia disipada en D2.

#### ECUACIÓN 15:

$$P_{D2} = I_x V = (45.8 \text{ mA})(0.7V) = 0.032W$$

Un diodo de 1/8W es un suficiente para D2.

### Valor de C2:

C2 debe tener dos veces el voltaje del diodo zéner. En este caso, un capacitor electrolítico de 16V servirá. C2 acumula la corriente sólo para cederla a la carga. El valor se basa en las fluctuaciones de voltaje que son aceptable en VOUT. VOUT se deteriora según la Ecuación 9.

## Ventajas y Desventajas

Advantages of Resistive Power Supply:

1. Mucho más chica que una fuente de alimentación basada en transformador.
2. Costo más bajo que una fuente de alimentación basada en transformador.
3. Menor costo que una fuente de alim. capacitiva.

Desventajas de Fuente de Alimentación Resistiva:

1. No aislado del voltaje de línea de CA que introduce problemas de seguridad.
2. La energía es menos eficiente que una fuente de alimentación capacitiva.
3. La pérdida de energía es disipada como calor en R1.

## OTRAS CONSIDERACIONES

### Consideraciones de seguridad

**Disclaimer:** This section does not provide all the information needed to meet UL requirements. UL requirements are application specific and are not exclusive to the circuit design itself. Some of the other characteristics that are factors in meeting UL requirements are trace width, trace proximity to one another, and (but not limited to) other layout requirements. Visit the Underwriters Laboratories Inc. Web page at [www.ul.com](http://www.ul.com) for more information.

**FIGURA 10: FUENTE DE ALIMENTACIÓN CAPACITIVA CON CONSIDERACIONES DE SEGURIDAD**

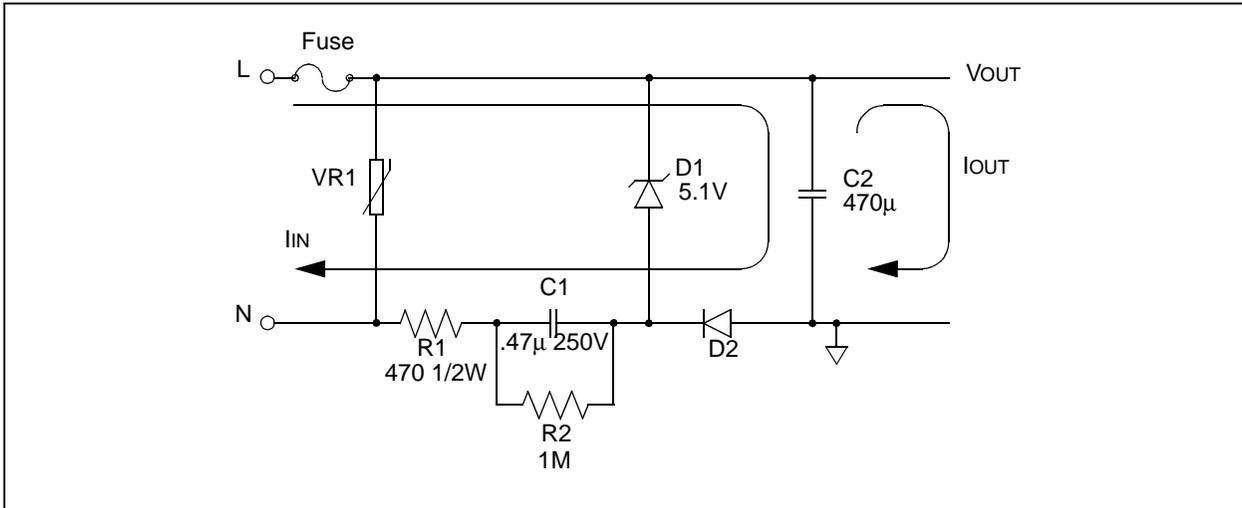
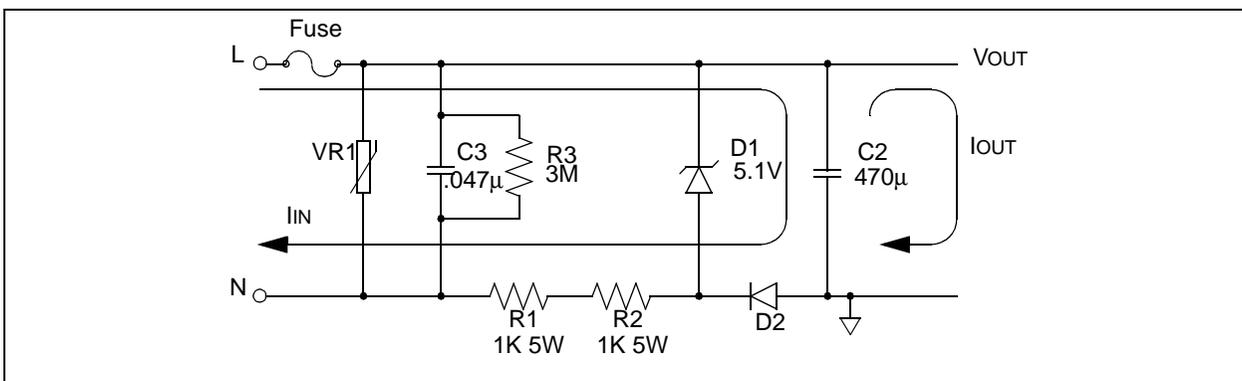


Figure 10 muestra una fuente de alim. capacitiva con varias consideraciones de UL en el diseño. Un fusible se pone para proteger el circuito durante una condición de sobre-corriente. R2 paralelamente a C1 crea un filtro que atenúe la EMI de retorno sobre la línea. Un varistor o MOV, proporciona protección transitoria.

Figura 11 muestra una fuente de alim. resistiva con varias consideraciones (1) de UL de diseño.

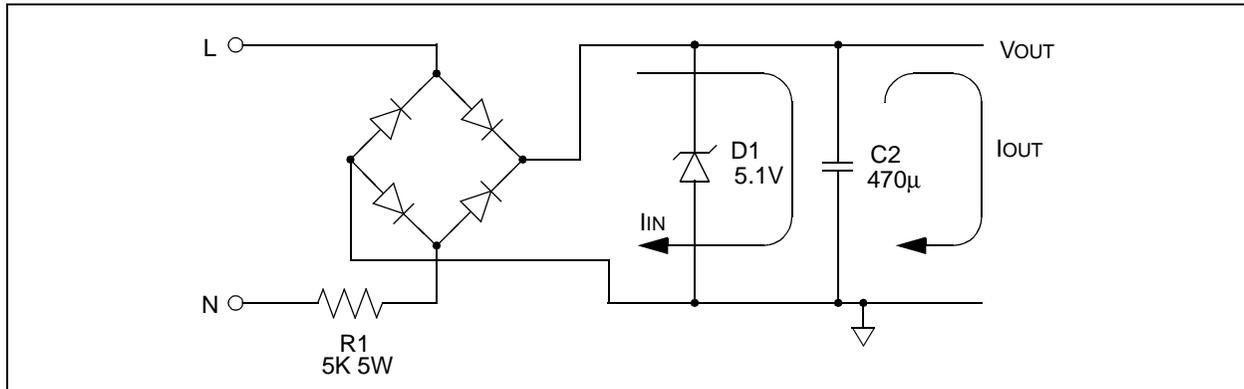
**Nota 1:** Se debe investigar las datos aplicables de UL que se ponen al producto específico del usuario. Los productos se probarán en un laboratorio certificado para ver los requisitos de UL se cumplen.

**FIGURA 11: FUENTE DE ALIMENTACIÓN RESISTIVA CON LAS CONSIDERACIONES DE SEGURIDAD**



Como con la fuente de alimentación capacitiva, se han puesto un fusible y el varistor para dar protección de sobre corriente y transitorio respectivamente. El resistor de 2k se separa en dos resistores en serie de 1k. El resistor se debe dividir en dos resistores de modo que un transitorio de alto voltaje no puentee el resistor. El uso de los dos resistores también reduce el voltaje en los resistores, reduciendo la posibilidad de formar arco. C3 y R3 crean un filtro que evite que la EMI creada por el circuito emigre sobre la vía de línea o neutral.

**FIGURE 12: RESISTIVE POWER SUPPLY WITH BRIDGE RECTIFIER**



## Rectificación del Puente

La corriente de salida de los circuitos descritos se puede incrementar en 141% con poner un económico rectificador puente. La Figura 12 muestra como la fuente de alimentación resistiva se ve con esta adición.

En vez de dar corriente durante sólo una mitad del periodo de la forma de onda de CA, la corriente es dada por la fuente durante ambas mitades. La Ecuación 16 da el voltaje RMS del voltaje RMS de onda completa en R1.

### ECUACIÓN 16:

$$V_{FLRMS} = \frac{\sqrt{2}V_{RMS} - V_Z}{\sqrt{2}}$$

Sustituyendo en la Ecuación 10 da la ecuación para IIN:

### ECUACIÓN 17:

$$I_{IN} = \frac{\sqrt{2}V_{RMS} - V_Z}{\sqrt{2}R}$$

Ventajas del rectificador puente sobre el de media onda:

1. Proporciona 141% más de corriente.
2. Más eficiente.
3. VOUT es más estable.

Desventajas del rectificador puente comparado al rectificador de media onda:

1. Más caro.
2. VOUT no está referido a la línea o al neutral haciendo el control del triac imposible.

# AN954

---

## CONCLUSIÓN

Las fuentes de alimentación sin-transformador son instrumentales en tener costos bajos en aplicaciones basadas en microcontroladores alimentados de la pared. Las fuentes de alimentaciones resistivas y capacitivas ofrecen un ahorro sustancial de costo y de espacio sobre las alimentaciones basados en transformador y conmutación. Las fuentes de alimentación capacitivas ofrecen una solución eficiente de energía, mientras las fuentes de alimentación resistivos ofrece un incremento del ahorro en costo.

## REFERENCIAS

"*Transformerless Power Supply*" D'Souza, Stan, TB008, Microchip Technology Inc.

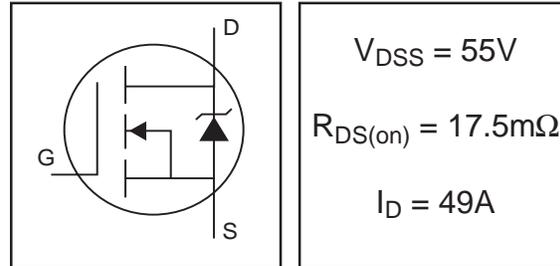
## **Anexo 36**

**Hojas de datos de los elementos electrónicos utilizados.**

# IRFZ44N

HEXFET® Power MOSFET

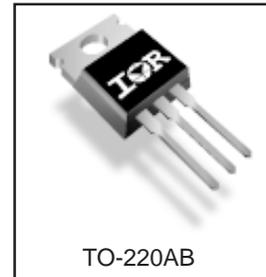
- Advanced Process Technology
- Ultra Low On-Resistance
- Dynamic dv/dt Rating
- 175°C Operating Temperature
- Fast Switching
- Fully Avalanche Rated



## Description

Advanced HEXFET® Power MOSFETs from International Rectifier utilize advanced processing techniques to achieve extremely low on-resistance per silicon area. This benefit, combined with the fast switching speed and ruggedized device design that HEXFET power MOSFETs are well known for, provides the designer with an extremely efficient and reliable device for use in a wide variety of applications.

The TO-220 package is universally preferred for all commercial-industrial applications at power dissipation levels to approximately 50 watts. The low thermal resistance and low package cost of the TO-220 contribute to its wide acceptance throughout the industry.



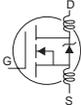
## Absolute Maximum Ratings

	Parameter	Max.	Units
$I_D @ T_C = 25^\circ\text{C}$	Continuous Drain Current, $V_{GS} @ 10\text{V}$	49	A
$I_D @ T_C = 100^\circ\text{C}$	Continuous Drain Current, $V_{GS} @ 10\text{V}$	35	
$I_{DM}$	Pulsed Drain Current ①	160	
$P_D @ T_C = 25^\circ\text{C}$	Power Dissipation	94	W
	Linear Derating Factor	0.63	W/°C
$V_{GS}$	Gate-to-Source Voltage	$\pm 20$	V
$I_{AR}$	Avalanche Current ①	25	A
$E_{AR}$	Repetitive Avalanche Energy ①	9.4	mJ
dv/dt	Peak Diode Recovery dv/dt ③	5.0	V/ns
$T_J$	Operating Junction and	-55 to + 175	°C
$T_{STG}$	Storage Temperature Range		
	Soldering Temperature, for 10 seconds		
	Mounting torque, 6-32 or M3 screw	10 lbf•in (1.1N•m)	

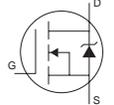
## Thermal Resistance

	Parameter	Typ.	Max.	Units
$R_{\theta JC}$	Junction-to-Case	—	1.5	°C/W
$R_{\theta CS}$	Case-to-Sink, Flat, Greased Surface	0.50	—	
$R_{\theta JA}$	Junction-to-Ambient	—	62	

## Electrical Characteristics @ $T_J = 25^\circ\text{C}$ (unless otherwise specified)

	Parameter	Min.	Typ.	Max.	Units	Conditions
$V_{(BR)DSS}$	Drain-to-Source Breakdown Voltage	55	—	—	V	$V_{GS} = 0V, I_D = 250\mu A$
$\Delta V_{(BR)DSS}/\Delta T_J$	Breakdown Voltage Temp. Coefficient	—	0.058	—	V/°C	Reference to $25^\circ\text{C}, I_D = 1\text{mA}$
$R_{DS(on)}$	Static Drain-to-Source On-Resistance	—	—	17.5	mΩ	$V_{GS} = 10V, I_D = 25A$ ④
$V_{GS(th)}$	Gate Threshold Voltage	2.0	—	4.0	V	$V_{DS} = V_{GS}, I_D = 250\mu A$
$g_{fs}$	Forward Transconductance	19	—	—	S	$V_{DS} = 25V, I_D = 25A$ ④
$I_{DSS}$	Drain-to-Source Leakage Current	—	—	25	μA	$V_{DS} = 55V, V_{GS} = 0V$
		—	—	250		$V_{DS} = 44V, V_{GS} = 0V, T_J = 150^\circ\text{C}$
$I_{GSS}$	Gate-to-Source Forward Leakage	—	—	100	nA	$V_{GS} = 20V$
	Gate-to-Source Reverse Leakage	—	—	-100		$V_{GS} = -20V$
$Q_g$	Total Gate Charge	—	—	63	nC	$I_D = 25A$
$Q_{gs}$	Gate-to-Source Charge	—	—	14		$V_{DS} = 44V$
$Q_{gd}$	Gate-to-Drain ("Miller") Charge	—	—	23		$V_{GS} = 10V$ , See Fig. 6 and 13
$t_{d(on)}$	Turn-On Delay Time	—	12	—	ns	$V_{DD} = 28V$
$t_r$	Rise Time	—	60	—		$I_D = 25A$
$t_{d(off)}$	Turn-Off Delay Time	—	44	—		$R_G = 12\Omega$
$t_f$	Fall Time	—	45	—		$V_{GS} = 10V$ , See Fig. 10 ④
$L_D$	Internal Drain Inductance	—	4.5	—	nH	Between lead, 6mm (0.25in.) from package and center of die contact
$L_S$	Internal Source Inductance	—	7.5	—		
$C_{iss}$	Input Capacitance	—	1470	—	pF	$V_{GS} = 0V$
$C_{oss}$	Output Capacitance	—	360	—		$V_{DS} = 25V$
$C_{rss}$	Reverse Transfer Capacitance	—	88	—		$f = 1.0\text{MHz}$ , See Fig. 5
$E_{AS}$	Single Pulse Avalanche Energy ②	—	530 ⑤	150 ⑥		mJ

## Source-Drain Ratings and Characteristics

	Parameter	Min.	Typ.	Max.	Units	Conditions
$I_S$	Continuous Source Current (Body Diode)	—	—	49	A	MOSFET symbol showing the integral reverse p-n junction diode. 
$I_{SM}$	Pulsed Source Current (Body Diode) ①	—	—	160		
$V_{SD}$	Diode Forward Voltage	—	—	1.3	V	$T_J = 25^\circ\text{C}, I_S = 25A, V_{GS} = 0V$ ④
$t_{rr}$	Reverse Recovery Time	—	63	95	ns	$T_J = 25^\circ\text{C}, I_F = 25A$
$Q_{rr}$	Reverse Recovery Charge	—	170	260	nC	$di/dt = 100A/\mu s$ ④
$t_{on}$	Forward Turn-On Time	Intrinsic turn-on time is negligible (turn-on is dominated by $L_S + L_D$ )				

### Notes:

- ① Repetitive rating; pulse width limited by max. junction temperature. (See fig. 11)
- ② Starting  $T_J = 25^\circ\text{C}$ ,  $L = 0.48\text{mH}$   
 $R_G = 25\Omega, I_{AS} = 25A$ . (See Figure 12)
- ③  $I_{SD} \leq 25A, di/dt \leq 230A/\mu s, V_{DD} \leq V_{(BR)DSS}, T_J \leq 175^\circ\text{C}$
- ④ Pulse width  $\leq 400\mu s$ ; duty cycle  $\leq 2\%$ .
- ⑤ This is a typical value at device destruction and represents operation outside rated limits.
- ⑥ This is a calculated value limited to  $T_J = 175^\circ\text{C}$ .

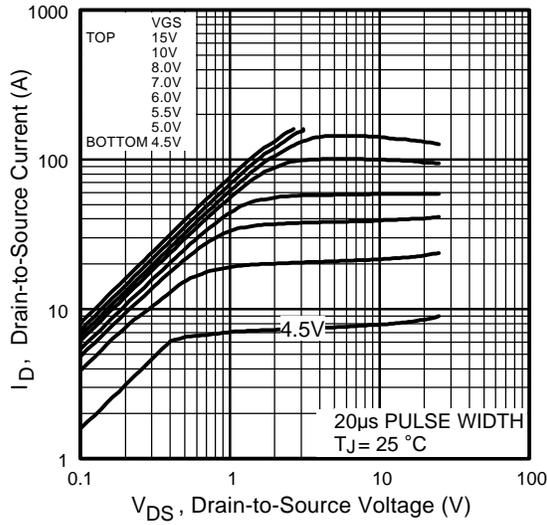


Fig 1. Typical Output Characteristics

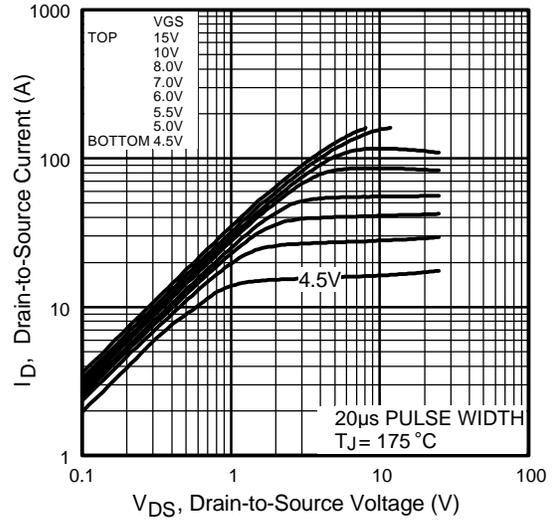


Fig 2. Typical Output Characteristics

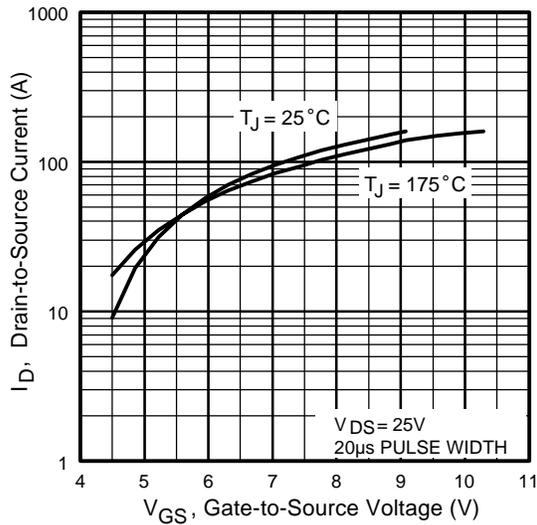


Fig 3. Typical Transfer Characteristics

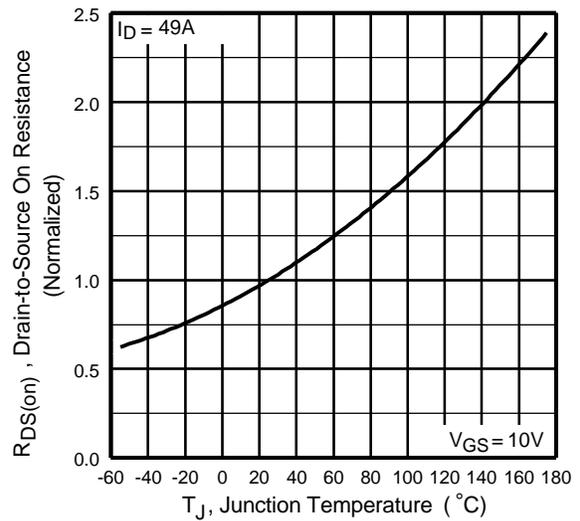
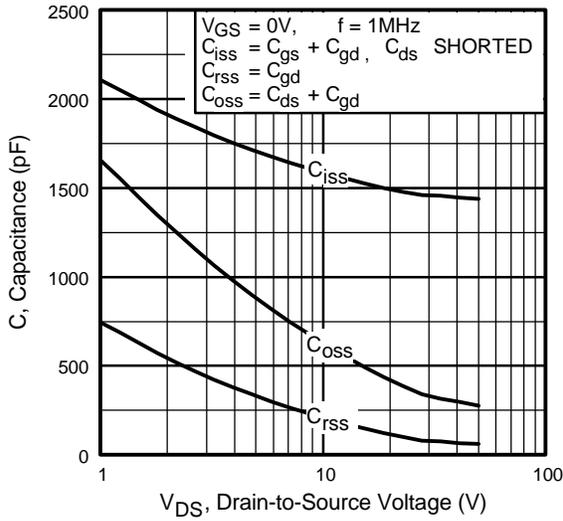
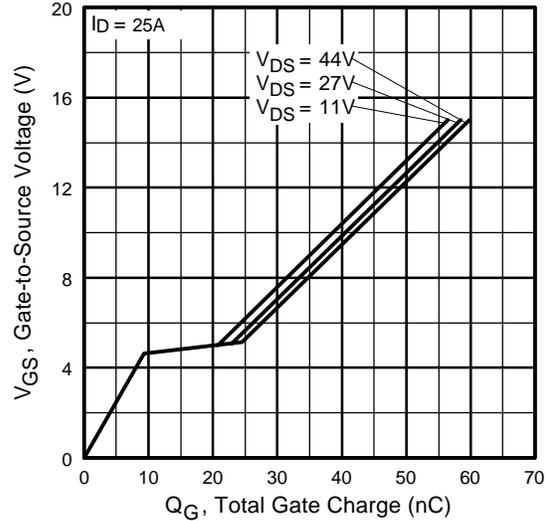


Fig 4. Normalized On-Resistance Vs. Temperature

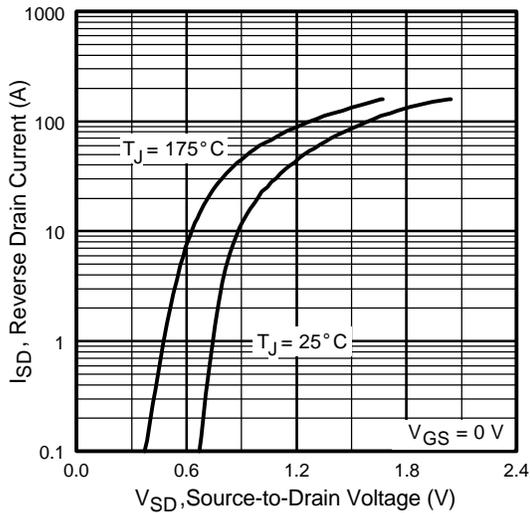
# IRFZ44N



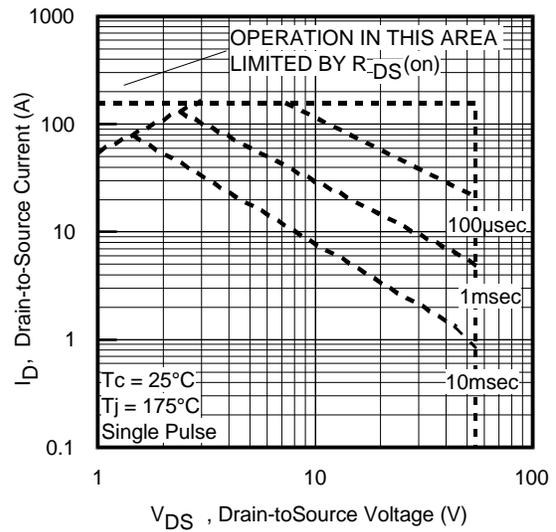
**Fig 5.** Typical Capacitance Vs. Drain-to-Source Voltage



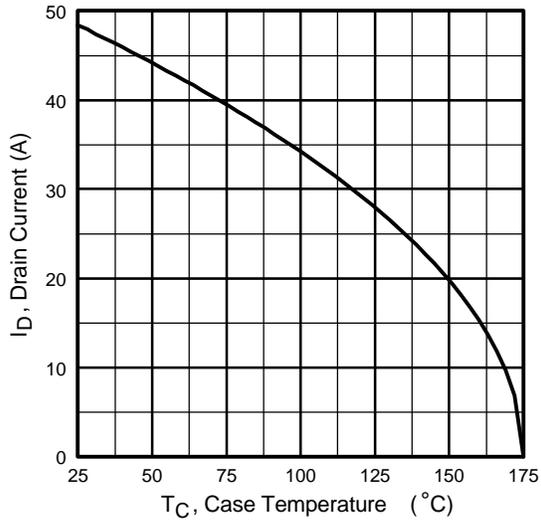
**Fig 6.** Typical Gate Charge Vs. Gate-to-Source Voltage



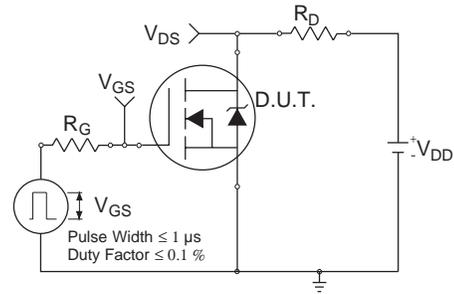
**Fig 7.** Typical Source-Drain Diode Forward Voltage



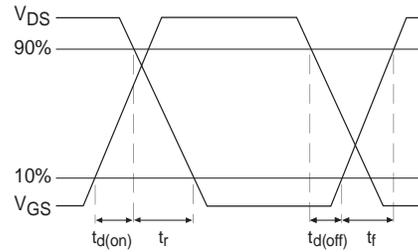
**Fig 8.** Maximum Safe Operating Area



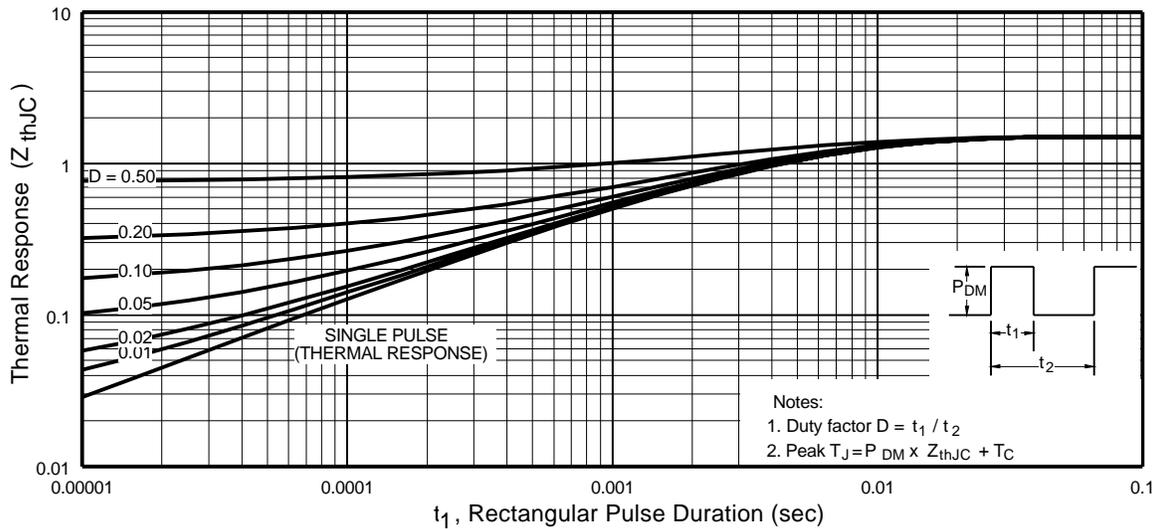
**Fig 9.** Maximum Drain Current Vs. Case Temperature



**Fig 10a.** Switching Time Test Circuit



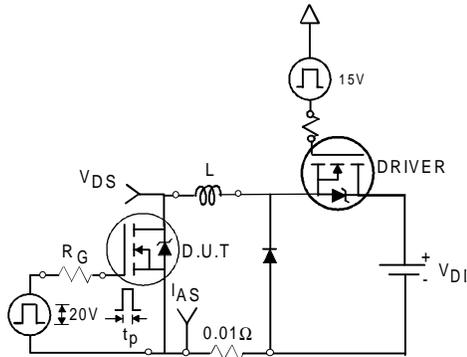
**Fig 10b.** Switching Time Waveforms



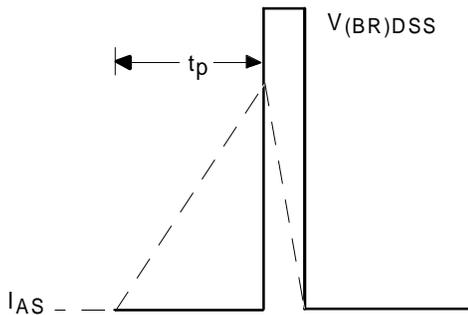
**Fig 11.** Maximum Effective Transient Thermal Impedance, Junction-to-Case

# IRFZ44N

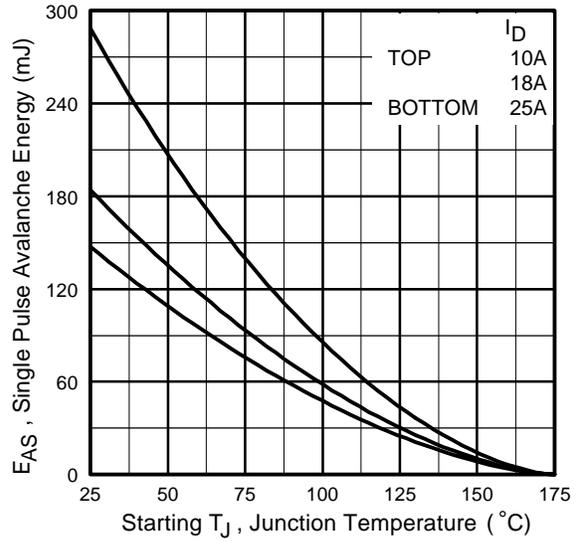
International  
**IR** Rectifier



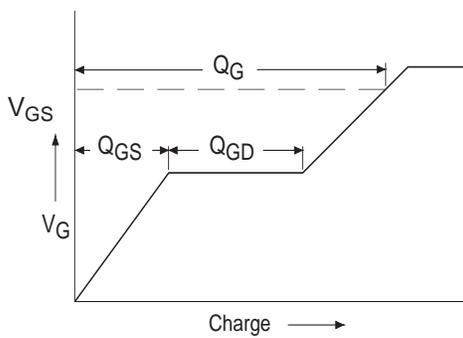
**Fig 12a.** Unclamped Inductive Test Circuit



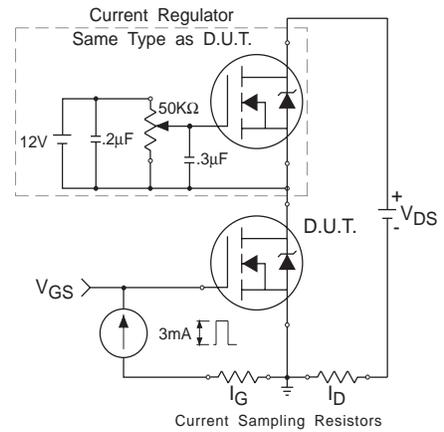
**Fig 12b.** Unclamped Inductive Waveforms



**Fig 12c.** Maximum Avalanche Energy Vs. Drain Current

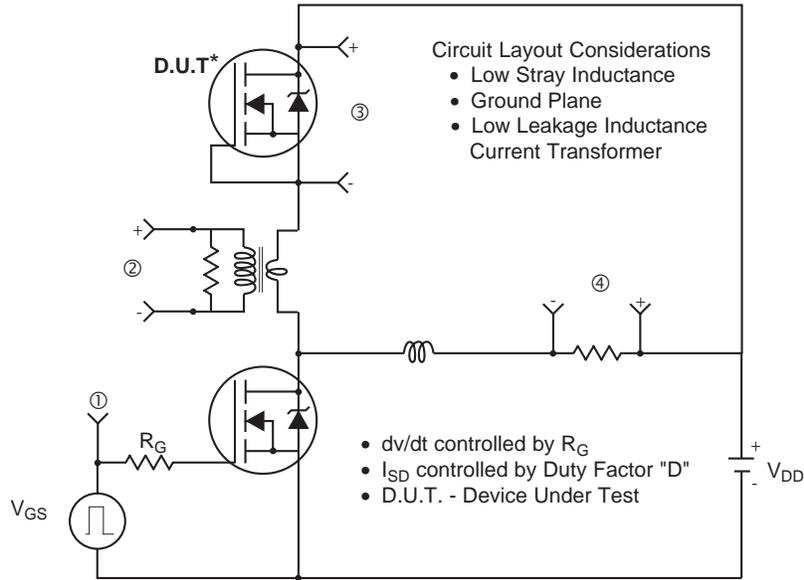


**Fig 13a.** Basic Gate Charge Waveform

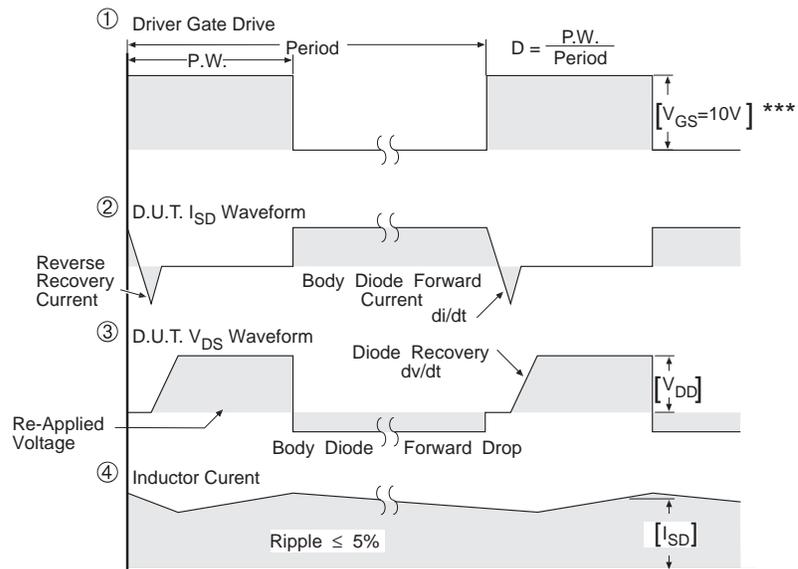


**Fig 13b.** Gate Charge Test Circuit

**Peak Diode Recovery dv/dt Test Circuit**



\* Reverse Polarity of D.U.T for P-Channel



\*\*\*  $V_{GS} = 5.0V$  for Logic Level and 3V Drive Devices

**Fig 14.** For N-channel HEXFET® power MOSFETs





MOC3020X, MOC3021X, MOC3022X, MOC3023X  
MOC3020, MOC3021, MOC3022, MOC3023



**OPTICALLY COUPLED BILATERAL SWITCH NON-ZERO CROSSING TRIAC**

**APPROVALS**

- UL recognised, File No. E91231

**'X' SPECIFICATION APPROVALS**

- VDE 0884 in 3 available lead forms : -  
- STD  
- G form  
- SMD approved to CECC 00802

**DESCRIPTION**

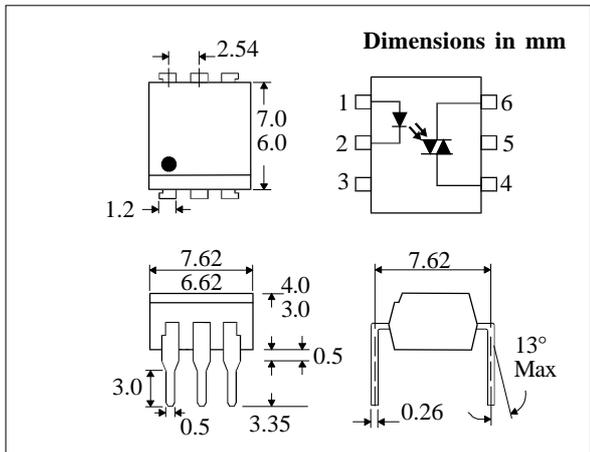
The MOC302\_ series are optically coupled isolators consisting of a Gallium Arsenide infrared emitting diode coupled with a light activated silicon bilateral switch performing the functions of a triac mounted in a standard 6 pin dual-in-line package.

**FEATURE**

- Options :-  
10mm lead spread - add G after part no.  
Surface mount - add SM after part no.  
Tape&reel - add SMT&R after part no.
- High Isolation Voltage (5.3kV<sub>RMS</sub>, 7.5kV<sub>PK</sub>)
- 400V Peak Blocking Voltage
- All electrical parameters 100% tested
- Custom electrical selections available

**APPLICATIONS**

- CRTs
- Power Triac Driver
- Motors
- Consumer appliances
- Printers



**ABSOLUTE MAXIMUM RATINGS (25 °C unless otherwise noted)**

Storage Temperature \_\_\_\_\_ -55°C-+150°C  
Operating Temperature \_\_\_\_\_ -40°C-+100°C  
Lead Soldering Temperature \_\_\_\_\_ 260°C  
(1.6mm from case for 10 seconds)  
Input-to-output Isolation Voltage (Pk) \_7500 Vac  
(60 Hz , 1sec. duration)

**INPUT DIODE**

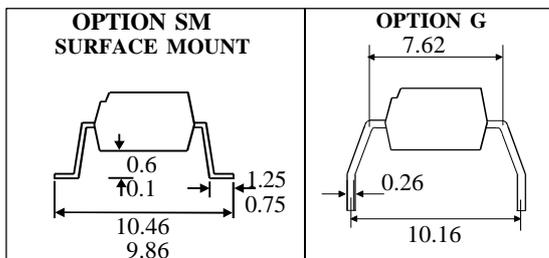
Forward Current \_\_\_\_\_ 50mA  
Reverse Voltage \_\_\_\_\_ 6V  
Power Dissipation \_\_\_\_\_ 70mW  
(derate linearly 0.93mW/°C above 25°C)

**OUTPUT PHOTO TRIAC**

Off-State Output Terminal Voltage \_\_\_ 400V  
RMS Forward Current \_\_\_\_\_ 100mA  
Forward Current (Peak) \_\_\_\_\_ 1A  
Power Dissipation \_\_\_\_\_ 300mW  
(derate linearly 4.0mW/°C above 25°C)

**POWER DISSIPATION**

Total Power Dissipation \_\_\_\_\_ 330mW  
(derate linearly 4.4mW/°C above 25°C)



**ISOCOM COMPONENTS LTD**  
Unit 25B, Park View Road West,  
Park View Industrial Estate, Brenda Road  
Hartlepool, Cleveland, TS25 1YD  
Tel: (01429) 863609 Fax :(01429) 863581

**ISOCOM INC**  
1024 S. Greenville Ave, Suite 240,  
Allen, TX 75002 USA  
Tel: (214) 495-0755 Fax: (214) 495-0901  
e-mail info@isocom.com  
http://www.isocom.com

**ELECTRICAL CHARACTERISTICS (  $T_A = 25^\circ\text{C}$  Unless otherwise noted )**

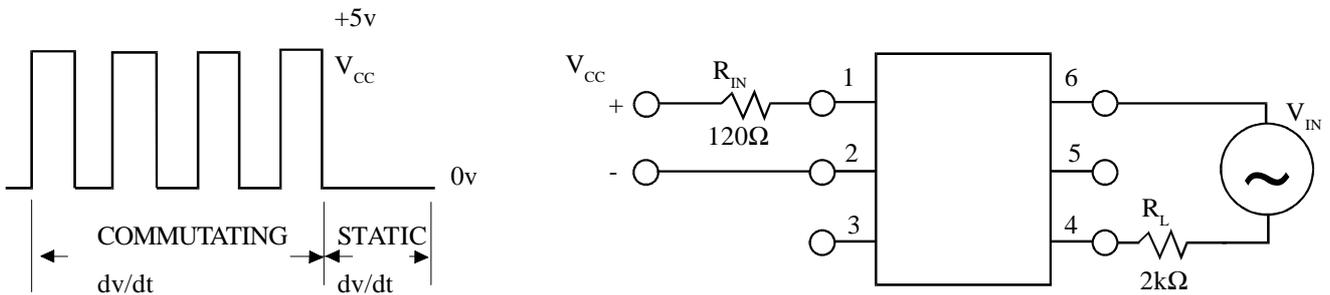
PARAMETER		MIN	TYP	MAX	UNITS	TEST CONDITION
Input	Forward Voltage ( $V_F$ ) Reverse Current ( $I_R$ )		1.2	1.5	V $\mu\text{A}$	$I_F = 10\text{mA}$ $V_R = 6\text{V}$
Output	Peak Off-state Current ( $I_{\text{DRM}}$ ) Peak Blocking Voltage ( $V_{\text{DRM}}$ ) On-state Voltage ( $V_{\text{TM}}$ ) Critical rate of rise of off-state Voltage ( $dv/dt$ ) ( note 1 ) Critical rate of rise of commutating Voltage ( $dv/dt$ ) ( note 1 )	400		100 3.0	nA V V $\text{V}/\mu\text{s}$ $\text{V}/\mu\text{s}$	$V_{\text{DRM}} = 400\text{V}$ (note 1) $I_{\text{DRM}} = 100\text{nA}$ $I_{\text{TM}} = 100\text{mA}$ ( peak )  $I_{\text{load}} = 15\text{mA}$ , $V_{\text{IN}} = 30\text{V}$ ( fig 1. )
Coupled	Input Current to Trigger ( $I_{\text{FT}}$ ) (note 2 ) MOC3020 MOC3021 MOC3022 MOC3023  Holding Current , either direction ( $I_H$ )  Input to Output Isolation Voltage $V_{\text{ISO}}$			30 15 10 5	mA mA mA mA  $\mu\text{A}$  $\text{V}_{\text{RMS}}$ $\text{V}_{\text{PK}}$	$V_D = 3\text{V}$ ( note 2 )      See note 3 See note 3

Note 1. Test voltage must be applied within  $dv/dt$  rating.

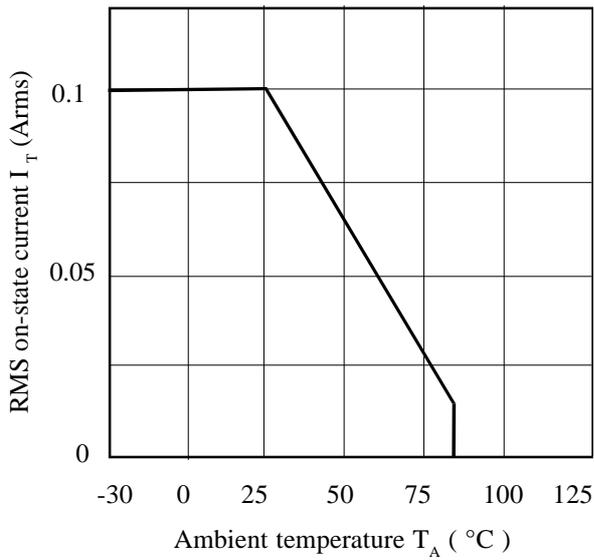
Note 2. Guaranteed to trigger at an  $I_F$  value less than or equal to max.  $I_{\text{FT}}$ , recommended  $I_F$  lies between Rated  $I_{\text{FT}}$  and absolute max.  $I_{\text{FT}}$ .

Note 3. Measured with input leads shorted together and output leads shorted together.

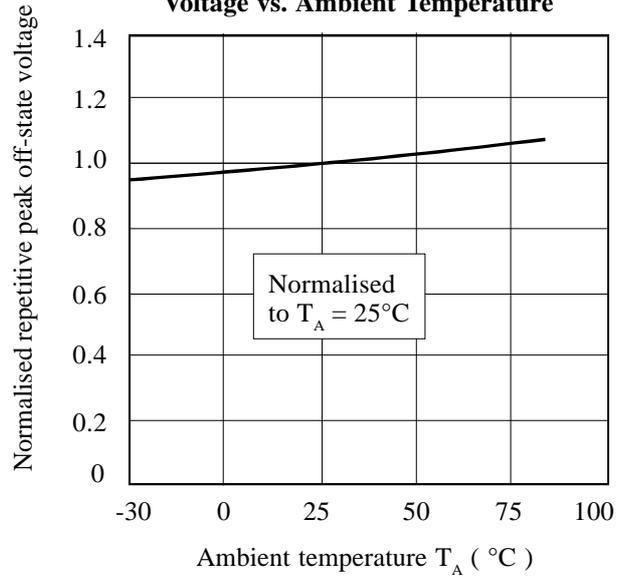
**FIGURE 1**



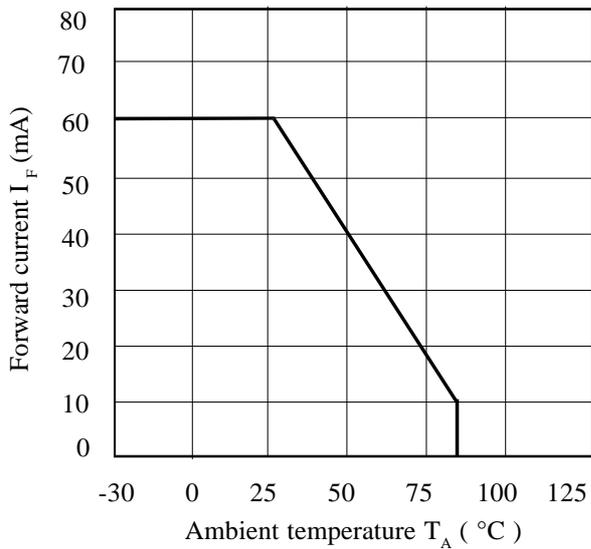
**RMS On-state Current vs. Ambient Temperature**



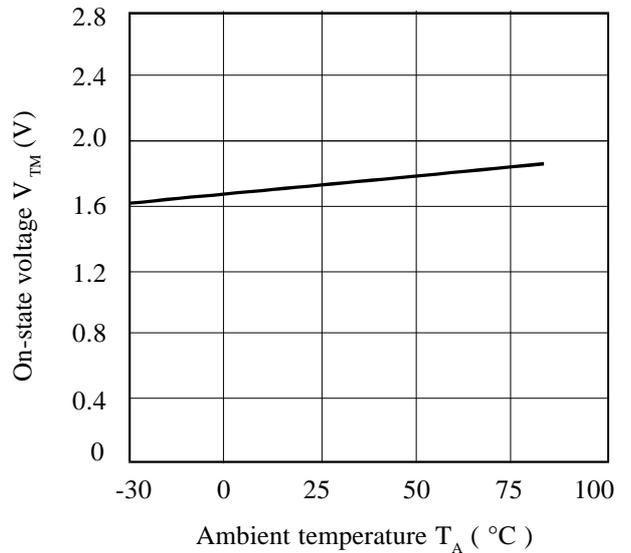
**Normalised Repetitive Peak Off-state Voltage vs. Ambient Temperature**



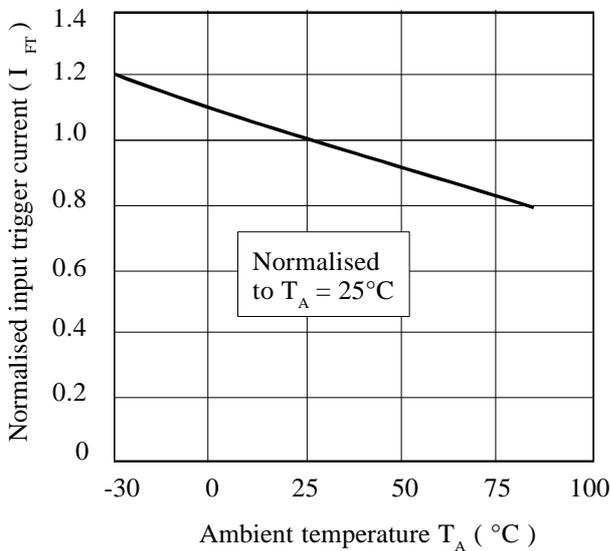
**Forward Current vs. Ambient Temperature**



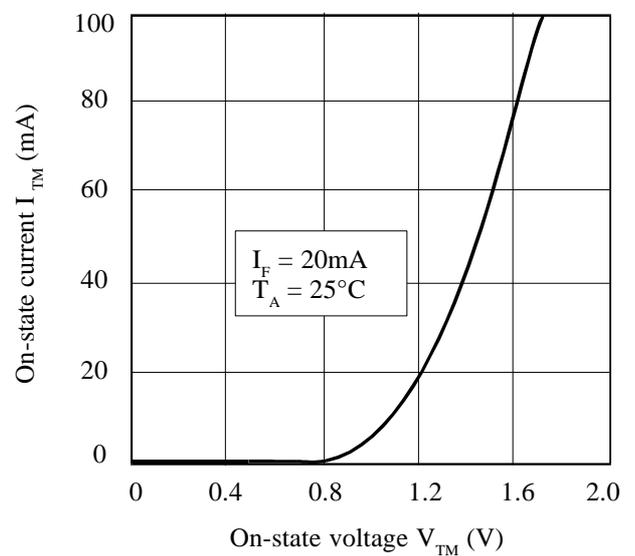
**On-state Voltage vs. Ambient Temperature**



**Normalised Input Trigger Current vs. Ambient Temperature**



**On-state Current vs. On-state Voltage**





### HAOPIN MICROELECTRONICS CO.,LTD.

#### Description

Glass passivated triacs in a plastic envelope, intended for use in applications requiring high bidirectional transient and blocking voltage capability and high thermal cycling performance. Typical applications include motor control, industrial and domestic lighting, heating and static switching.

<p>Symbol</p> 		<p>Simplified outline</p>  <p>TO-220</p>	
Pin	Description		
1	Main terminal 1 (T1)		
2	Main terminal 2 (T2)		
3	gate (G)		
TAB	Main terminal		

#### Applications:

- ◆ Motor control
- ◆ Industrial and domestic lighting
- ◆ Heating
- ◆ Static switching

#### Features

- ◆ Blocking voltage to 600 V
- ◆ On-state RMS current to 12 A

SYMBOL	PARAMETER	Value	Unit
$V_{DRM}$	Repetitive peak off-state voltages	600	V
$I_T (RMS)$	RMS on-state current (full sine wave)	12	A
$I_{TSM}$	Non-repetitive peak on-state current (full cycle, $T_j$ initial=25°C)	126	A

SYMBOL	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNIT
$R_{th(j-c)}$	Junction to case(AC)		-	2.3	-	°C/W
$R_{th(j-a)}$	Junction to ambient		-	60	-	°C/W

### HAOPIN MICROELECTRONICS CO.,LTD.

Limiting values in accordance with the Maximum system(IEC 134)

SYMBOL	PARAMETER	CONDITIONS		MIN	Value	UNIT
$V_{DRM}$	Repetitive peak off-state Voltages			-	600	V
$I_{T(RMS)}$	RMS on-state current	$T_c=90^\circ\text{C}$		-	12	A
$I_{TSM}$	Non repetitive surge peak on-state current	$T_j$ initial =25 $^\circ\text{C}$	F=50Hz t=20ms	-	120	A
			F=60Hz t=16.7ms	-	126	A
$I^2t$	$I^2t$ value for fusing	$T_p=10\text{ms}$		-	78	A <sup>2</sup> S
di/dt	Critical rate of rise of on-state current	$I_G=2 \times I_{GT}$ , tr $\leq$ 100ns	F=120Hz Tj=125 $^\circ\text{C}$	-	50	A/ $\mu$ s
$I_{GM}$	Peak gate current	$T_p=20 \mu$ s		-	4	A
$I_{DRM}$	$V_{DRM}=V_{RRM}$			-	5	$\mu$ A
$I_{RRM}$	$V_{DRM}=V_{RRM}$			-	1	mA
$P_{G(AV)}$	Average gate power			-	1	W
$T_{stg}$	Storage temperature range			-40	150	$^\circ\text{C}$
$T_j$	Operating junction Temperature range			-40	125	$^\circ\text{C}$

$T_j=25^\circ\text{C}$  unless otherwise stated

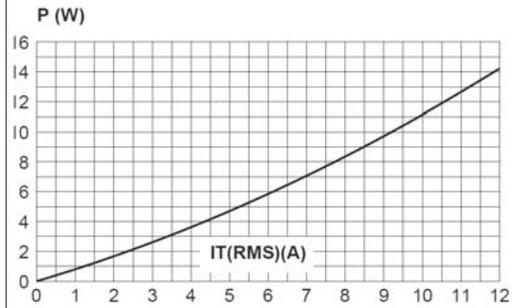
SYMBOL	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNIT
Static characteristics						
$I_{GT}$		$V_D=12\text{V}; R_L=30 \Omega$	I-II-III	-	-	25 mA
			IV	-	-	50 mA
$I_L$		$I_G=1.2 I_{GT}$	I-III-IV	-	-	40 mA
			II	-	-	80 mA
$I_H$		$I_T=500\text{mA}$	-	-	25	mA
$V_{GT}$		$V_D=12\text{V}; R_L=30 \Omega$	-	-	1.3	V
$V_{GD}$		$V_D=V_{DRM} R_L=3.3\text{K} \Omega T_j=125^\circ\text{C}$	0.2	-	-	V
dV/dt		$V_D=67\%V_{DRM}$ gate open; $T_j=125^\circ\text{C}$	200	-	-	V/ $\mu$ s
(dV/dt)c	(di/dt)c=5.3A/ms	$T_j=125^\circ\text{C}$	5	-	-	V/ $\mu$ s

#### Dynamic Characteristics

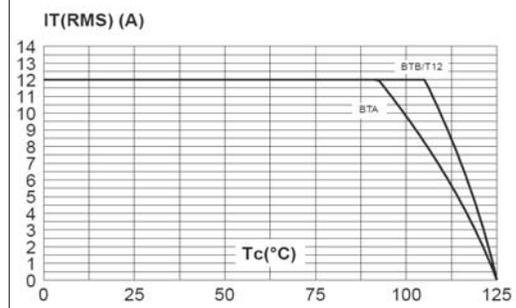
$V_{TM}$	$I_{TM}=17\text{A}$ tp=380 $\mu$ s	$T_j=25^\circ\text{C}$	-	-	1.55	V
$V_{to}$	Threshold voltage	$T_j=125^\circ\text{C}$	-	-	0.85	V
$R_d$	Dynamic resistance	$T_j=125^\circ\text{C}$	-	-	35	m $\Omega$

### Description

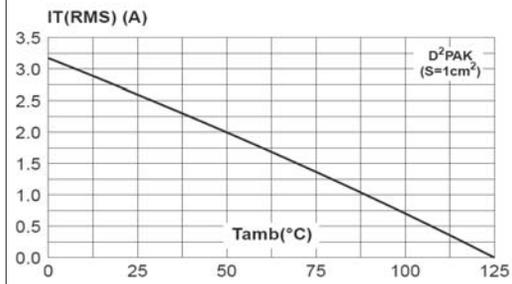
**Fig. 1:** Maximum power dissipation versus RMS on-state current (full cycle).



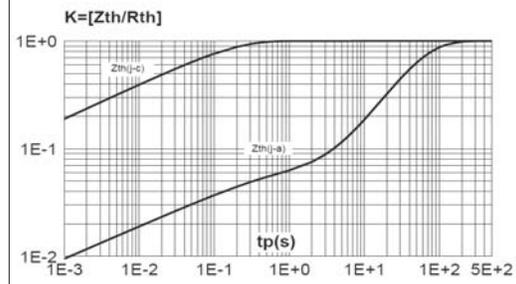
**Fig. 2-1:** RMS on-state current versus case temperature (full cycle).



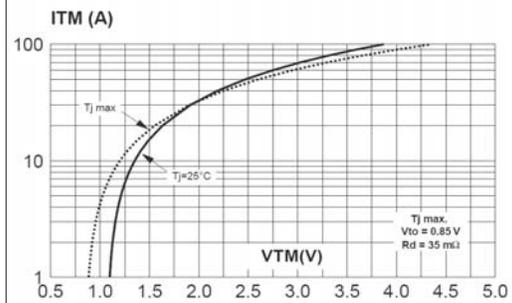
**Fig. 2-2:** RMS on-state current versus ambient temperature (printed circuit board FR4, copper thickness: 35µm), full cycle.



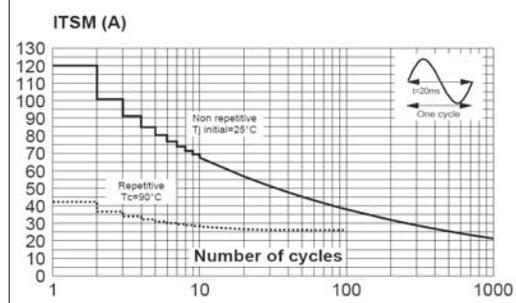
**Fig. 3:** Relative variation of thermal impedance versus pulse duration.



**Fig. 4:** On-state characteristics (maximum values).

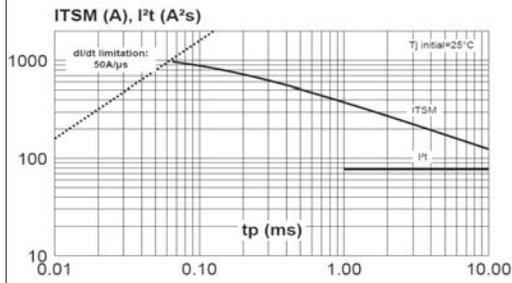


**Fig. 5:** Surge peak on-state current versus number of cycles.

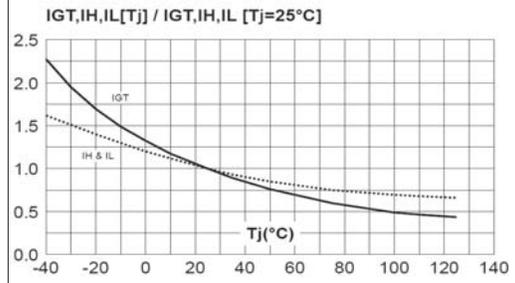


Description

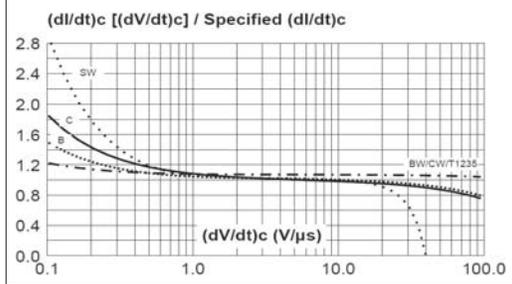
**Fig. 6:** Non-repetitive surge peak on-state current for a sinusoidal pulse with width  $t_p < 10\text{ms}$ , and corresponding value of  $I^2t$ .



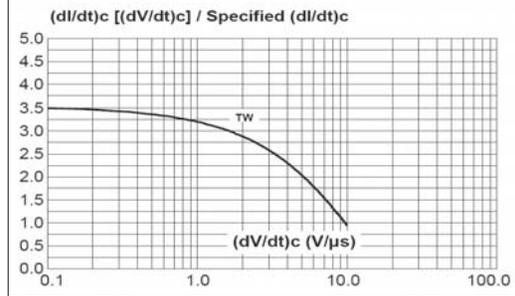
**Fig. 7:** Relative variation of gate trigger current, holding current and latching current versus junction temperature (typical values).



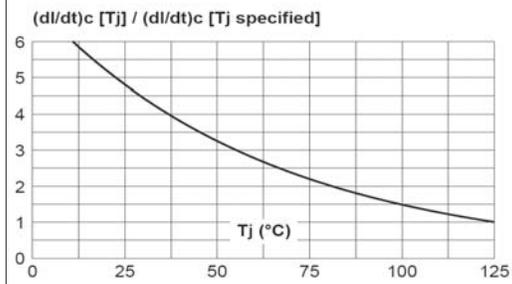
**Fig. 8-1:** Relative variation of critical rate of decrease of main current versus  $(dV/dt)_c$  (typical values) (BW/CW/T1235).



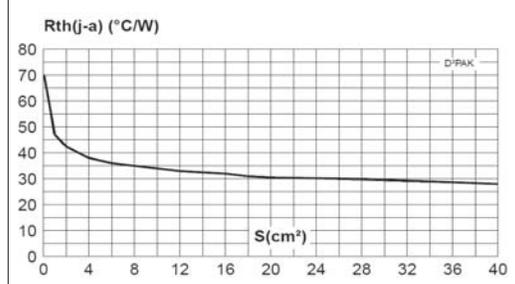
**Fig. 8-2:** Relative variation of critical rate of decrease of main current versus  $(dV/dt)_c$  (typical values) (TW).



**Fig. 9:** Relative variation of critical rate of decrease of main current versus junction temperature.



**Fig. 10:** D<sup>2</sup>PAK Thermal resistance junction to ambient versus copper surface under tab (printed circuit board FR4, copper thickness: 35 µm).

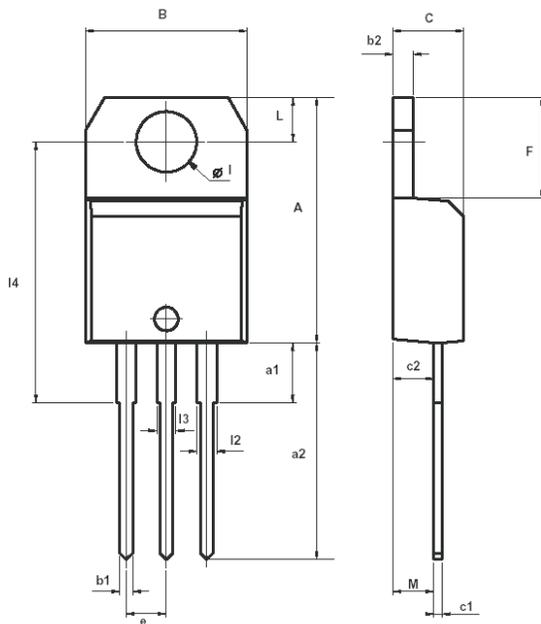


### MECHANICAL DATA

Dimensions in mm

Net Mass: 2g

TO-220AB



REF.	DIMENSIONS		
	Millimeters		
	Min.	Typ.	Max.
A	15.2		15.9
a1	3.2		3.95
a2	13.00		14.00
B	10.00		10.40
b1	0.61		0.88
b2	1.20		1.32
C	4.40		4.6
c1	0.45		0.7
c2	2.20		2.72
e	2.30		2.70
f	6.20		6.60
l	3.70		3.85
l4	15.80		16.8
L	2.65		2.95
l2	1.14		1.70
l3	1.14		1.70
M	2.47		2.73



## L293x Quadruple Half-H Drivers

### 1 Features

- Wide Supply-Voltage Range: 4.5 V to 36 V
- Separate Input-Logic Supply
- Internal ESD Protection
- High-Noise-Immunity Inputs
- Output Current 1 A Per Channel (600 mA for L293D)
- Peak Output Current 2 A Per Channel (1.2 A for L293D)
- Output Clamp Diodes for Inductive Transient Suppression (L293D)

### 2 Applications

- Stepper Motor Drivers
- DC Motor Drivers
- Latching Relay Drivers

### 3 Description

The L293 and L293D devices are quadruple high-current half-H drivers. The L293 is designed to provide bidirectional drive currents of up to 1 A at voltages from 4.5 V to 36 V. The L293D is designed to provide bidirectional drive currents of up to 600-mA at voltages from 4.5 V to 36 V. Both devices are designed to drive inductive loads such as relays, solenoids, DC and bipolar stepping motors, as well as other high-current/high-voltage loads in positive-supply applications.

Each output is a complete totem-pole drive circuit, with a Darlington transistor sink and a pseudo-Darlington source. Drivers are enabled in pairs, with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4EN.

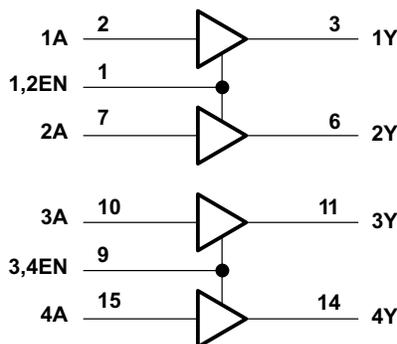
The L293 and L293D are characterized for operation from 0°C to 70°C.

#### Device Information<sup>(1)</sup>

PART NUMBER	PACKAGE	BODY SIZE (NOM)
L293NE	PDIP (16)	19.80 mm x 6.35 mm
L293DNE	PDIP (16)	19.80 mm x 6.35 mm

(1) For all available packages, see the orderable addendum at the end of the data sheet.

#### Logic Diagram



## Table of Contents

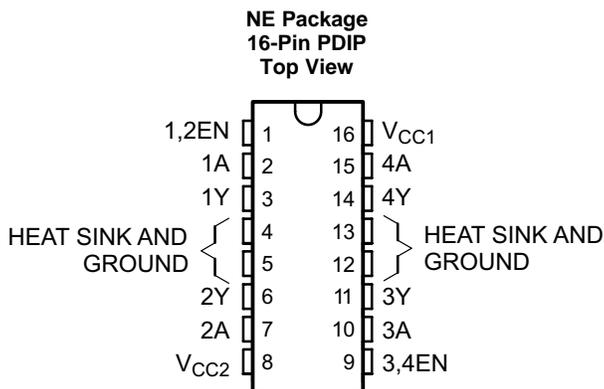
<b>1 Features</b> .....	<b>1</b>	8.3 Feature Description .....	<b>7</b>
<b>2 Applications</b> .....	<b>1</b>	8.4 Device Functional Modes .....	<b>8</b>
<b>3 Description</b> .....	<b>1</b>	<b>9 Application and Implementation</b> .....	<b>9</b>
<b>4 Revision History</b> .....	<b>2</b>	9.1 Application Information .....	<b>9</b>
<b>5 Pin Configuration and Functions</b> .....	<b>3</b>	9.2 Typical Application .....	<b>9</b>
<b>6 Specifications</b> .....	<b>4</b>	9.3 System Examples .....	<b>10</b>
6.1 Absolute Maximum Ratings .....	<b>4</b>	<b>10 Power Supply Recommendations</b> .....	<b>13</b>
6.2 ESD Ratings .....	<b>4</b>	<b>11 Layout</b> .....	<b>14</b>
6.3 Recommended Operating Conditions .....	<b>4</b>	11.1 Layout Guidelines .....	<b>14</b>
6.4 Thermal Information .....	<b>4</b>	11.2 Layout Example .....	<b>14</b>
6.5 Electrical Characteristics .....	<b>5</b>	<b>12 Device and Documentation Support</b> .....	<b>15</b>
6.6 Switching Characteristics .....	<b>5</b>	12.1 Related Links .....	<b>15</b>
6.7 Typical Characteristics .....	<b>5</b>	12.2 Community Resources .....	<b>15</b>
<b>7 Parameter Measurement Information</b> .....	<b>6</b>	12.3 Trademarks .....	<b>15</b>
<b>8 Detailed Description</b> .....	<b>7</b>	12.4 Electrostatic Discharge Caution .....	<b>15</b>
8.1 Overview .....	<b>7</b>	12.5 Glossary .....	<b>15</b>
8.2 Functional Block Diagram .....	<b>7</b>	<b>13 Mechanical, Packaging, and Orderable Information</b> .....	<b>15</b>

## 4 Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

<b>Changes from Revision C (November 2004) to Revision D</b>	<b>Page</b>
• Removed <i>Ordering Information</i> table .....	<b>1</b>
• Added <i>ESD Ratings</i> and <i>Thermal Information</i> tables, <i>Feature Description</i> section, <i>Device Functional Modes</i> , <i>Application and Implementation</i> section, <i>Power Supply Recommendations</i> section, <i>Layout</i> section, <i>Device and Documentation Support</i> section, and <i>Mechanical, Packaging, and Orderable Information</i> section. ....	<b>1</b>

## 5 Pin Configuration and Functions



### Pin Functions

PIN		TYPE	DESCRIPTION
NAME	NO.		
1,2EN	1	I	Enable driver channels 1 and 2 (active high input)
<1:4>A	2, 7, 10, 15	I	Driver inputs, noninverting
<1:4>Y	3, 6, 11, 14	O	Driver outputs
3,4EN	9	I	Enable driver channels 3 and 4 (active high input)
GROUND	4, 5, 12, 13	—	Device ground and heat sink pin. Connect to printed-circuit-board ground plane with multiple solid vias
V <sub>CC1</sub>	16	—	5-V supply for internal logic translation
V <sub>CC2</sub>	8	—	Power VCC for drivers 4.5 V to 36 V

## 6 Specifications

### 6.1 Absolute Maximum Ratings

 over operating free-air temperature range (unless otherwise noted)<sup>(1)</sup>

	MIN	MAX	UNIT
Supply voltage, $V_{CC1}$ <sup>(2)</sup>		36	V
Output supply voltage, $V_{CC2}$		36	V
Input voltage, $V_I$		7	V
Output voltage, $V_O$	-3	$V_{CC2} + 3$	V
Peak output current, $I_O$ (nonrepetitive, $t \leq 5$ ms): L293	-2	2	A
Peak output current, $I_O$ (nonrepetitive, $t \leq 100$ $\mu$ s): L293D	-1.2	1.2	A
Continuous output current, $I_O$ : L293	-1	1	A
Continuous output current, $I_O$ : L293D	-600	600	mA
Maximum junction temperature, $T_J$		150	°C
Storage temperature, $T_{stg}$	-65	150	°C

- (1) Stresses beyond those listed under *Absolute Maximum Ratings* may cause permanent damage to the device. These are stress ratings only, which do not imply functional operation of the device at these or any other conditions beyond those indicated under *Recommended Operating Conditions*. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.
- (2) All voltage values are with respect to the network ground terminal.

### 6.2 ESD Ratings

		VALUE	UNIT
$V_{(ESD)}$	Electrostatic discharge	Human-body model (HBM), per ANSI/ESDA/JEDEC JS-001 <sup>(1)</sup>	V
		Charged-device model (CDM), per JEDEC specification JESD22-C101 <sup>(2)</sup>	
		$\pm 2000$	
		$\pm 1000$	

- (1) JEDEC document JEP155 states that 500-V HBM allows safe manufacturing with a standard ESD control process.
- (2) JEDEC document JEP157 states that 250-V CDM allows safe manufacturing with a standard ESD control process.

### 6.3 Recommended Operating Conditions

over operating free-air temperature range (unless otherwise noted)

		MIN	NOM	MAX	UNIT
Supply voltage	$V_{CC1}$	4.5		7	V
	$V_{CC2}$	$V_{CC1}$		36	
$V_{IH}$	High-level input voltage	$V_{CC1} \leq 7$ V		$V_{CC1}$	V
		$V_{CC1} \geq 7$ V	2.3	7	V
$V_{IL}$	Low-level output voltage	-0.3 <sup>(1)</sup>		1.5	V
$T_A$	Operating free-air temperature	0		70	°C

- (1) The algebraic convention, in which the least positive (most negative) designated minimum, is used in this data sheet for logic voltage levels.

### 6.4 Thermal Information

THERMAL METRIC <sup>(1)</sup>		L293, L293D	UNIT
		NE (PDIP)	
		16 PINS	
$R_{\theta JA}$	Junction-to-ambient thermal resistance <sup>(2)</sup>	36.4	°C/W
$R_{\theta JC(top)}$	Junction-to-case (top) thermal resistance	22.5	°C/W
$R_{\theta JB}$	Junction-to-board thermal resistance	16.5	°C/W
$\Psi_{JT}$	Junction-to-top characterization parameter	7.1	°C/W
$\Psi_{JB}$	Junction-to-board characterization parameter	16.3	°C/W

- (1) For more information about traditional and new thermal metrics, see the *Semiconductor and IC Package Thermal Metrics* application report, [SPRA953](#).
- (2) The package thermal impedance is calculated in accordance with JESD 51-7.

### 6.5 Electrical Characteristics

over operating free-air temperature range (unless otherwise noted)

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
V <sub>OH</sub>	High-level output voltage	L293: I <sub>OH</sub> = -1 A		V <sub>CC2</sub> - 1.8	V <sub>CC2</sub> - 1.4		V
		L293D: I <sub>OH</sub> = -0.6 A					
V <sub>OL</sub>	Low-level output voltage	L293: I <sub>OL</sub> = 1 A			1.2	1.8	V
		L293D: I <sub>OL</sub> = 0.6 A					
V <sub>OKH</sub>	High-level output clamp voltage	L293D: I <sub>OK</sub> = -0.6 A		V <sub>CC2</sub> + 1.3			V
V <sub>OKL</sub>	Low-level output clamp voltage	L293D: I <sub>OK</sub> = 0.6 A		1.3			V
I <sub>IH</sub>	High-level input current	A	V <sub>I</sub> = 7 V	0.2		100	μA
		EN		0.2		10	
I <sub>IL</sub>	Low-level input current	A	V <sub>I</sub> = 0	-3		-10	μA
		EN		-2		-100	
I <sub>CC1</sub>	Logic supply current	I <sub>O</sub> = 0	All outputs at high level	13		22	mA
			All outputs at low level	35		60	
			All outputs at high impedance	8		24	
I <sub>CC2</sub>	Output supply current	I <sub>O</sub> = 0	All outputs at high level	14		24	mA
			All outputs at low level	2		6	
			All outputs at high impedance	2		4	

### 6.6 Switching Characteristics

over operating free-air temperature range (unless otherwise noted) V<sub>CC1</sub> = 5 V, V<sub>CC2</sub> = 24 V, T<sub>A</sub> = 25°C

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
t <sub>PLH</sub>	Propagation delay time, low-to-high-level output from A input	L293NE, L293DNE		800			ns
		L293DWP, L293N L293DN		750			
t <sub>PHL</sub>	Propagation delay time, high-to-low-level output from A input	L293NE, L293DNE		400			ns
		L293DWP, L293N L293DN		200			
t <sub>TLH</sub>	Transition time, low-to-high-level output	L293NE, L293DNE		300			ns
		L293DWP, L293N L293DN		100			
t <sub>THL</sub>	Transition time, high-to-low-level output	L293NE, L293DNE		300			ns
		L293DWP, L293N L293DN		350			

C<sub>L</sub> = 30 pF, See Figure 2

### 6.7 Typical Characteristics

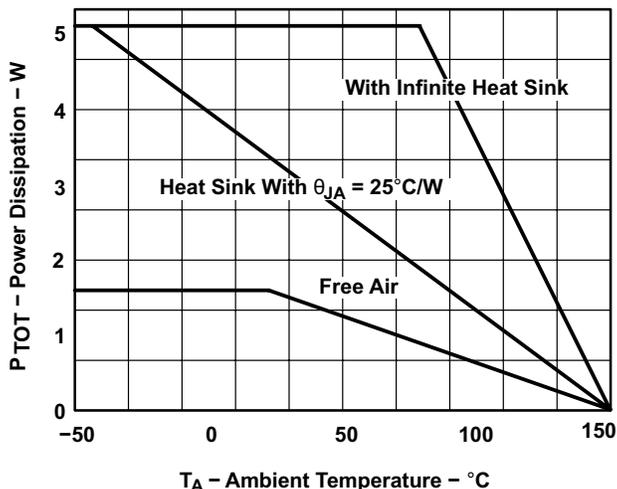
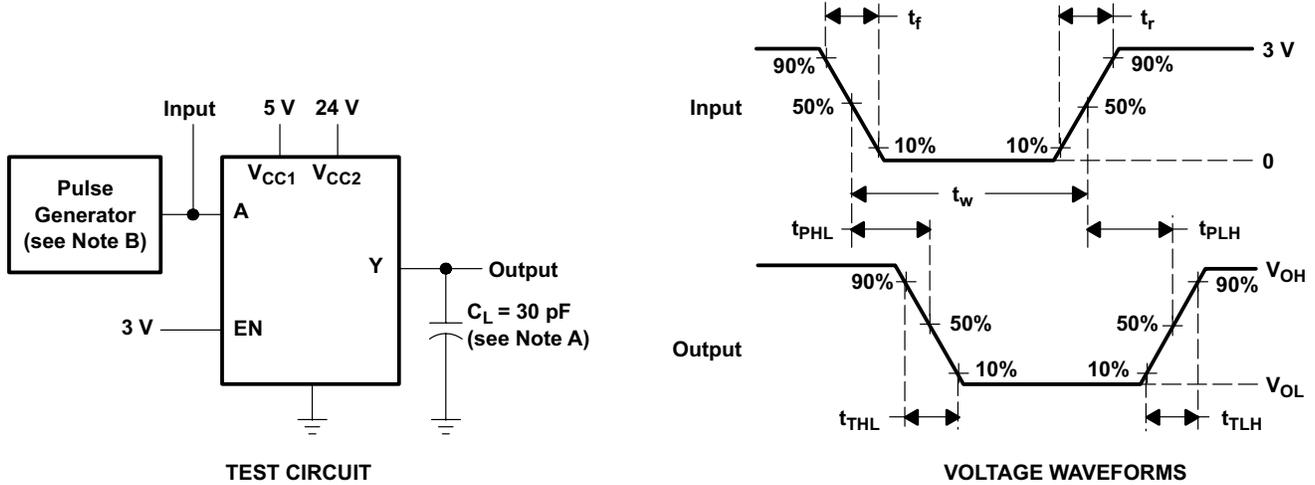


Figure 1. Maximum Power Dissipation vs Ambient Temperature

## 7 Parameter Measurement Information



NOTES: A.  $C_L$  includes probe and jig capacitance.

B. The pulse generator has the following characteristics:  $t_r \leq 10$  ns,  $t_f \leq 10$  ns,  $t_w = 10$   $\mu$ s, PRR = 5 kHz,  $Z_O = 50$   $\Omega$ .

**Figure 2. Test Circuit and Voltage Waveforms**

## 8 Detailed Description

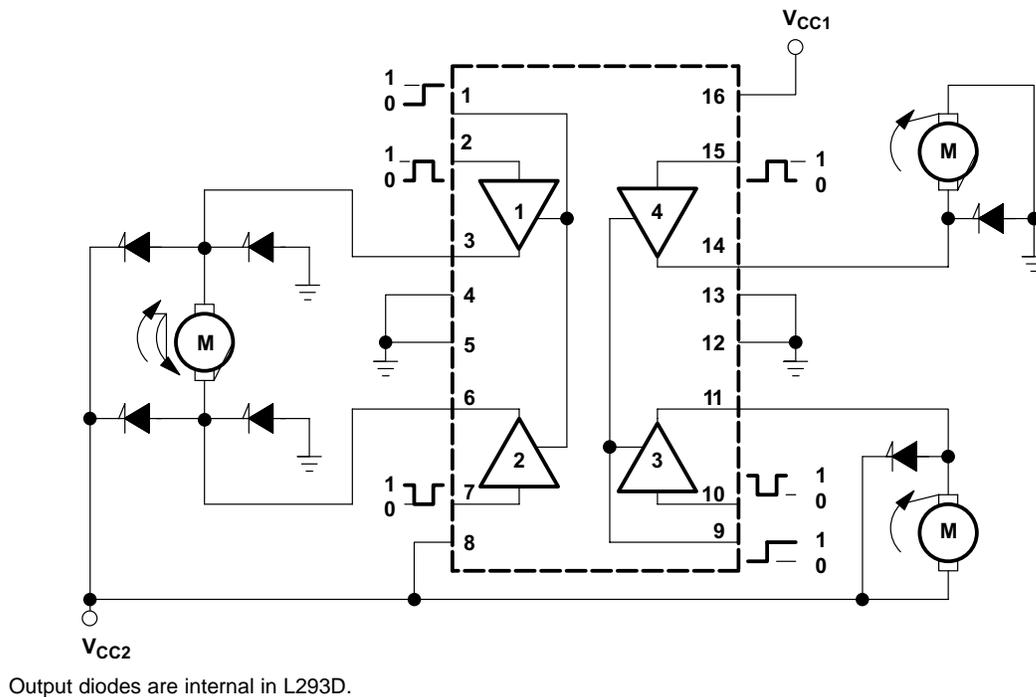
### 8.1 Overview

The L293 and L293D are quadruple high-current half-H drivers. These devices are designed to drive a wide array of inductive loads such as relays, solenoids, DC and bipolar stepping motors, as well as other high-current and high-voltage loads. All inputs are TTL compatible and tolerant up to 7 V.

Each output is a complete totem-pole drive circuit, with a Darlington transistor sink and a pseudo-Darlington source. Drivers are enabled in pairs, with drivers 1 and 2 enabled by 1,2EN and drivers 3 and 4 enabled by 3,4EN. When an enable input is high, the associated drivers are enabled, and their outputs are active and in phase with their inputs. When the enable input is low, those drivers are disabled, and their outputs are off and in the high-impedance state. With the proper data inputs, each pair of drivers forms a full-H (or bridge) reversible drive suitable for solenoid or motor applications.

On the L293, external high-speed output clamp diodes should be used for inductive transient suppression. On the L293D, these diodes are integrated to reduce system complexity and overall system size. A  $V_{CC1}$  terminal, separate from  $V_{CC2}$ , is provided for the logic inputs to minimize device power dissipation. The L293 and L293D are characterized for operation from 0°C to 70°C.

### 8.2 Functional Block Diagram



### 8.3 Feature Description

The L293x has TTL-compatible inputs and high voltage outputs for inductive load driving. Current outputs can get up to 2 A using the L293.

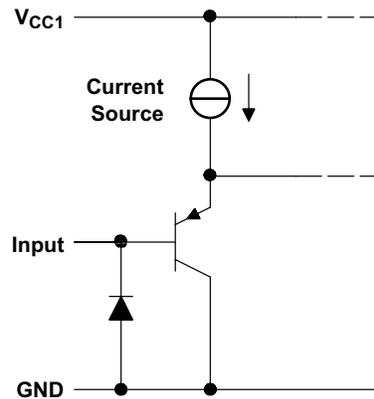
### 8.4 Device Functional Modes

Table 1 lists the functional modes of the L293x.

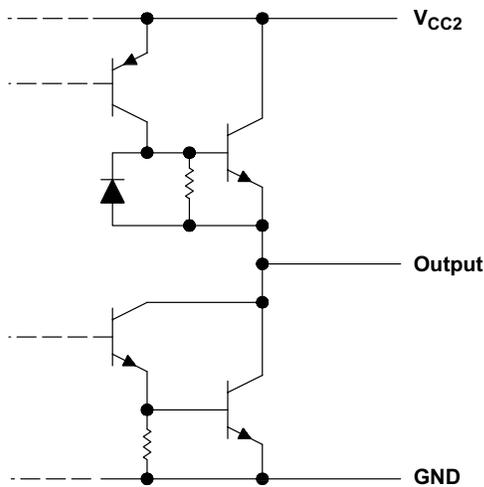
**Table 1. Function Table (Each Driver)<sup>(1)</sup>**

INPUTS <sup>(2)</sup>		OUTPUT (Y)
A	EN	
H	H	H
L	H	L
X	L	Z

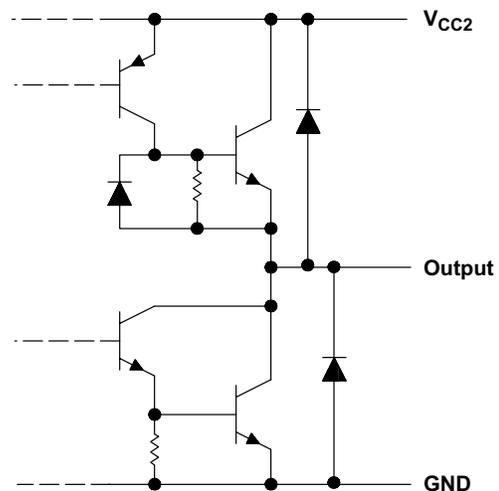
- (1) H = high level, L = low level, X = irrelevant, Z = high impedance (off)
- (2) In the thermal shutdown mode, the output is in the high-impedance state, regardless of the input levels.



**Figure 3. Schematic of Inputs for the L293x**



**Figure 4. Schematic of Outputs for the L293**



**Figure 5. Schematic of Outputs for the L293D**

## 9 Application and Implementation

### NOTE

Information in the following applications sections is not part of the TI component specification, and TI does not warrant its accuracy or completeness. TI's customers are responsible for determining suitability of components for their purposes. Customers should validate and test their design implementation to confirm system functionality.

### 9.1 Application Information

A typical application for the L293 device is driving a two-phase motor. Below is an example schematic displaying how to properly connect a two-phase motor to the L293 device.

Provide a 5-V supply to  $V_{CC1}$  and valid logic input levels to data and enable inputs.  $V_{CC2}$  must be connected to a power supply capable of supplying the needed current and voltage demand for the loads connected to the outputs.

### 9.2 Typical Application

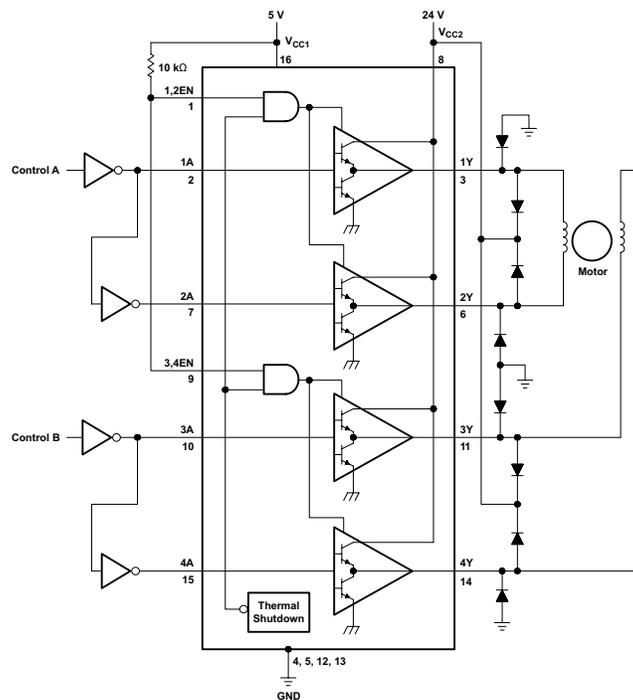


Figure 6. Two-Phase Motor Driver (L293)

#### 9.2.1 Design Requirements

The design techniques in the application above as well as the applications below should fall within the following design requirements.

1.  $V_{CC1}$  should fall within the limits described in the [Recommended Operating Conditions](#).
2.  $V_{CC2}$  should fall within the limits described in the [Recommended Operating Conditions](#).
3. The current per channel should not exceed 1 A for the L293 (600mA for the L293D).

#### 9.2.2 Detailed Design Procedure

When designing with the L293 or L293D, careful consideration should be made to ensure the device does not exceed the operating temperature of the device. Proper heatsinking will allow for operation over a larger range of current per channel. Refer to the [Power Supply Recommendations](#) as well as the [Layout Example](#).

## Typical Application (continued)

### 9.2.3 Application Curve

Refer to [Power Supply Recommendations](#) for additional information with regards to appropriate power dissipation. [Figure 7](#) describes thermal dissipation based on [Figure 14](#).

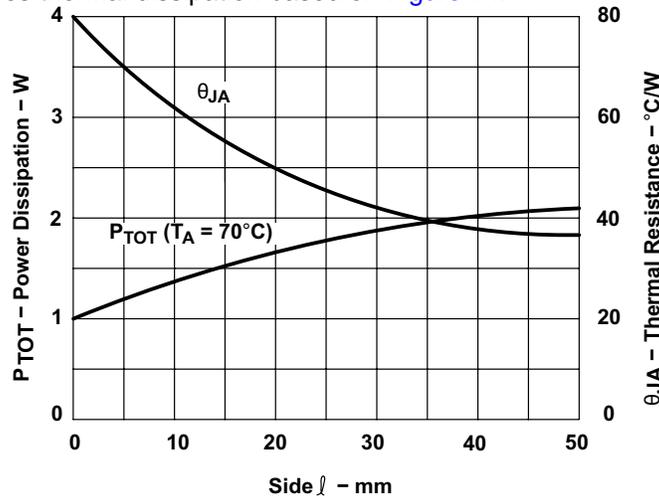


Figure 7. Maximum Power and Junction vs Thermal Resistance

## 9.3 System Examples

### 9.3.1 L293D as a Two-Phase Motor Driver

[Figure 8](#) below depicts a typical setup for using the L293D as a two-phase motor driver. Refer to the [Recommended Operating Conditions](#) when considering the appropriate input high and input low voltage levels to enable each channel of the device.

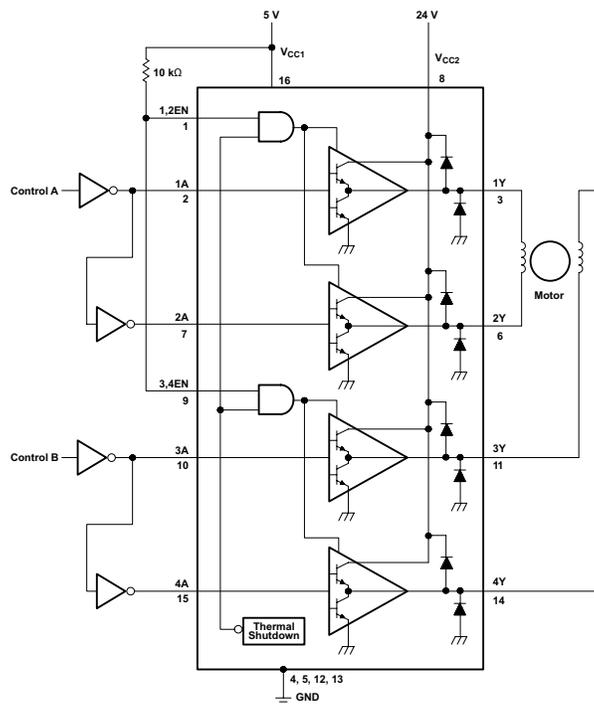
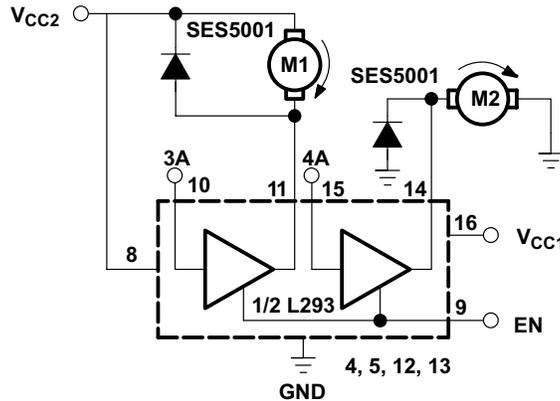


Figure 8. Two-Phase Motor Driver (L293D)

System Examples (continued)

9.3.2 DC Motor Controls

Figure 9 and Figure 10 below depict a typical setup for using the L293 device as a controller for DC motors. Note that the L293 device can be used as a simple driver for a motor to turn on and off in one direction, and can also be used to drive a motor in both directions. Refer to the function tables below to understand unidirectional vs bidirectional motor control. Refer to the *Recommended Operating Conditions* when considering the appropriate input high and input low voltage levels to enable each channel of the device.



Connections to ground and to supply voltage

Figure 9. DC Motor Controls

Table 2. Unidirectional DC Motor Control

EN	3A	M1 <sup>(1)</sup>	4A	M2
H	H	Fast motor stop	H	Run
H	L	run	L	Fast motor stop
L	X	Free-running motor stop	X	Free-running motor stop

(1) L = low, H = high, X = don't care

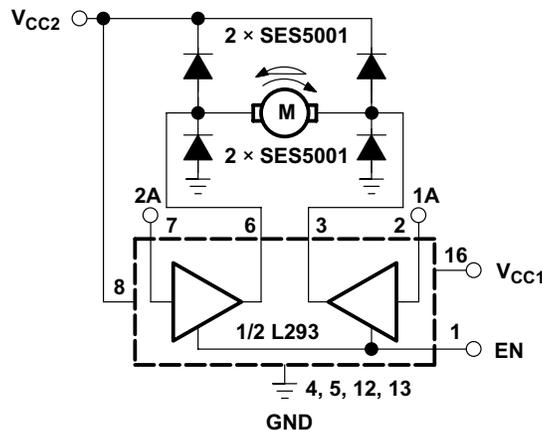


Figure 10. Bidirectional DC Motor Control

Table 3. Bidirectional DC Motor Control

EN	1A	2A	FUNCTION <sup>(1)</sup>
H	L	H	Turn right
H	H	L	Turn left

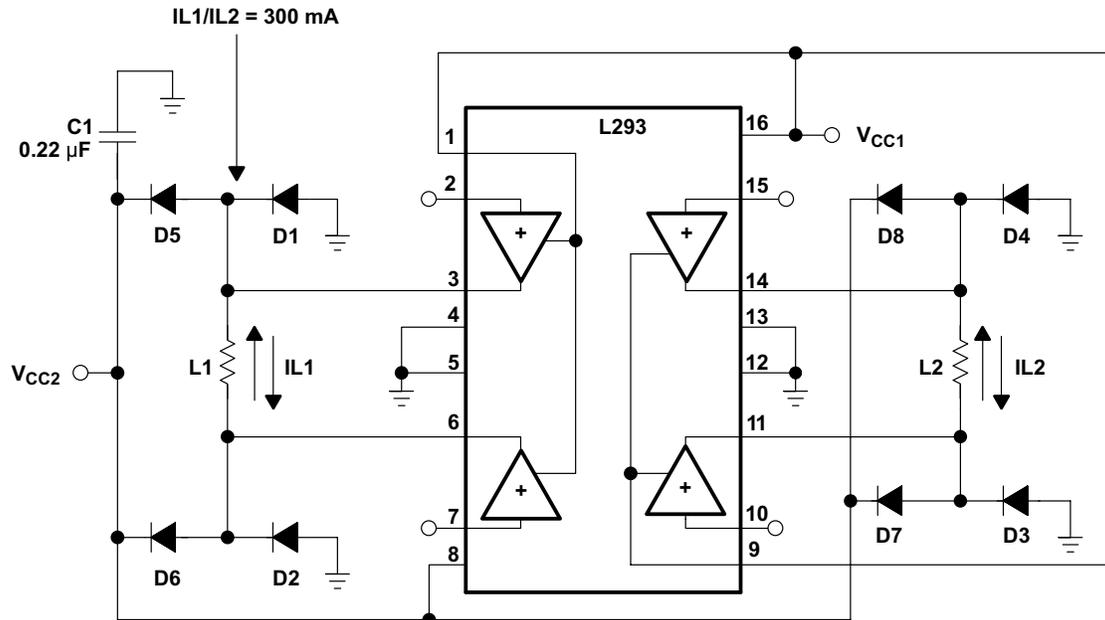
(1) L = low, H = high, X = don't care

**Table 3. Bidirectional DC Motor Control (continued)**

EN	1A	2A	FUNCTION <sup>(1)</sup>
H	L	L	Fast motor stop
H	H	H	Fast motor stop
L	X	X	Free-running motor stop

### 9.3.3 Bipolar Stepping-Motor Control

Figure 11 below depicts a typical setup for using the L293D as a two-phase motor driver. Refer to the [Recommended Operating Conditions](#) when considering the appropriate input high and input low voltage levels to enable each channel of the device.



D1–D8 = SES5001

**Figure 11. Bipolar Stepping-Motor Control**

## 10 Power Supply Recommendations

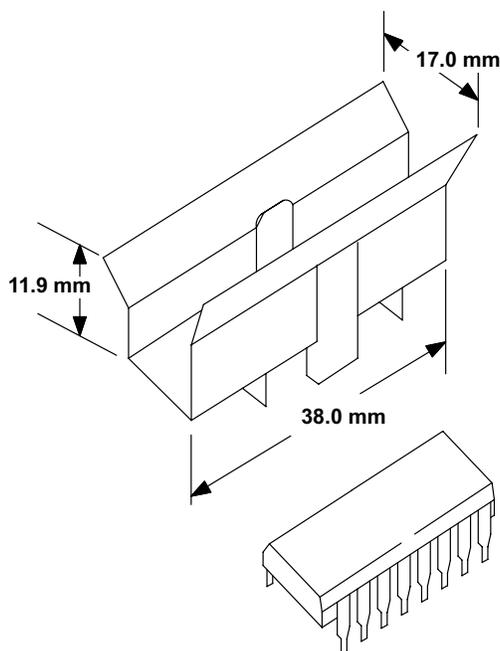
$V_{CC1}$  is  $5\text{ V} \pm 0.5\text{ V}$  and  $V_{CC2}$  can be same supply as  $V_{CC1}$  or a higher voltage supply with peak voltage up to 36 V. Bypass capacitors of 0.1  $\mu\text{F}$  or greater should be used at  $V_{CC1}$  and  $V_{CC2}$  pins. There are no power up or power down supply sequence order requirements.

Properly heatsinking the L293 when driving high-current is critical to design. The  $R_{thj-amp}$  of the L293 can be reduced by soldering the GND pins to a suitable copper area of the printed circuit board or to an external heat sink.

Figure 14 shows the maximum package power  $PTOT$  and the  $\theta_{JA}$  as a function of the side of two equal square copper areas having a thickness of 35  $\mu\text{m}$  (see Figure 14). In addition, an external heat sink can be used (see Figure 12).

During soldering, the pin temperature must not exceed 260°C, and the soldering time must not exceed 12 seconds.

The external heatsink or printed circuit copper area must be connected to electrical ground.



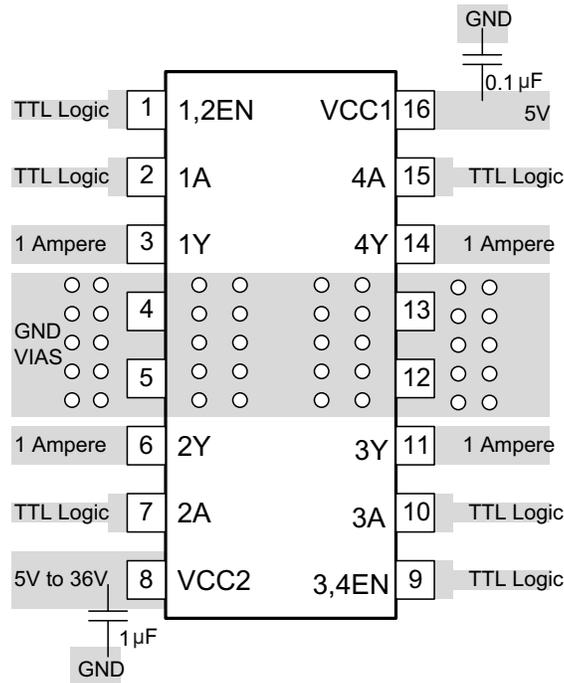
**Figure 12. External Heat Sink Mounting Example ( $\theta_{JA} = 25^\circ\text{C/W}$ )**

## 11 Layout

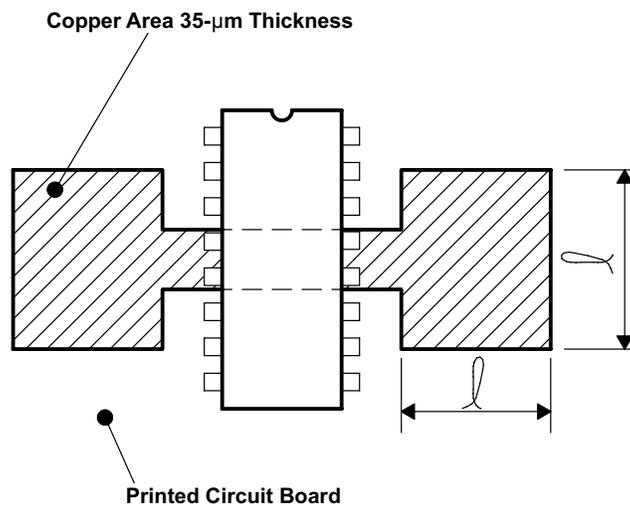
### 11.1 Layout Guidelines

Place the device near the load to keep output traces short to reduce EMI. Use solid vias to transfer heat from ground pins to ground plane of the printed-circuit-board.

### 11.2 Layout Example



**Figure 13. Layout Diagram**



**Figure 14. Example of Printed-Circuit-Board Copper Area (Used as Heat Sink)**

## 12 Device and Documentation Support

### 12.1 Related Links

The table below lists quick access links. Categories include technical documents, support and community resources, tools and software, and quick access to sample or buy.

**Table 4. Related Links**

PARTS	PRODUCT FOLDER	SAMPLE & BUY	TECHNICAL DOCUMENTS	TOOLS & SOFTWARE	SUPPORT & COMMUNITY
L293	<a href="#">Click here</a>				
L293D	<a href="#">Click here</a>				

### 12.2 Community Resources

The following links connect to TI community resources. Linked contents are provided "AS IS" by the respective contributors. They do not constitute TI specifications and do not necessarily reflect TI's views; see TI's [Terms of Use](#).

**TI E2E™ Online Community** *TI's Engineer-to-Engineer (E2E) Community*. Created to foster collaboration among engineers. At [e2e.ti.com](http://e2e.ti.com), you can ask questions, share knowledge, explore ideas and help solve problems with fellow engineers.

**Design Support** *TI's Design Support* Quickly find helpful E2E forums along with design support tools and contact information for technical support.

### 12.3 Trademarks

E2E is a trademark of Texas Instruments.  
All other trademarks are the property of their respective owners.

### 12.4 Electrostatic Discharge Caution



These devices have limited built-in ESD protection. The leads should be shorted together or the device placed in conductive foam during storage or handling to prevent electrostatic damage to the MOS gates.

### 12.5 Glossary

**SLYZ022** — *TI Glossary*.

This glossary lists and explains terms, acronyms, and definitions.

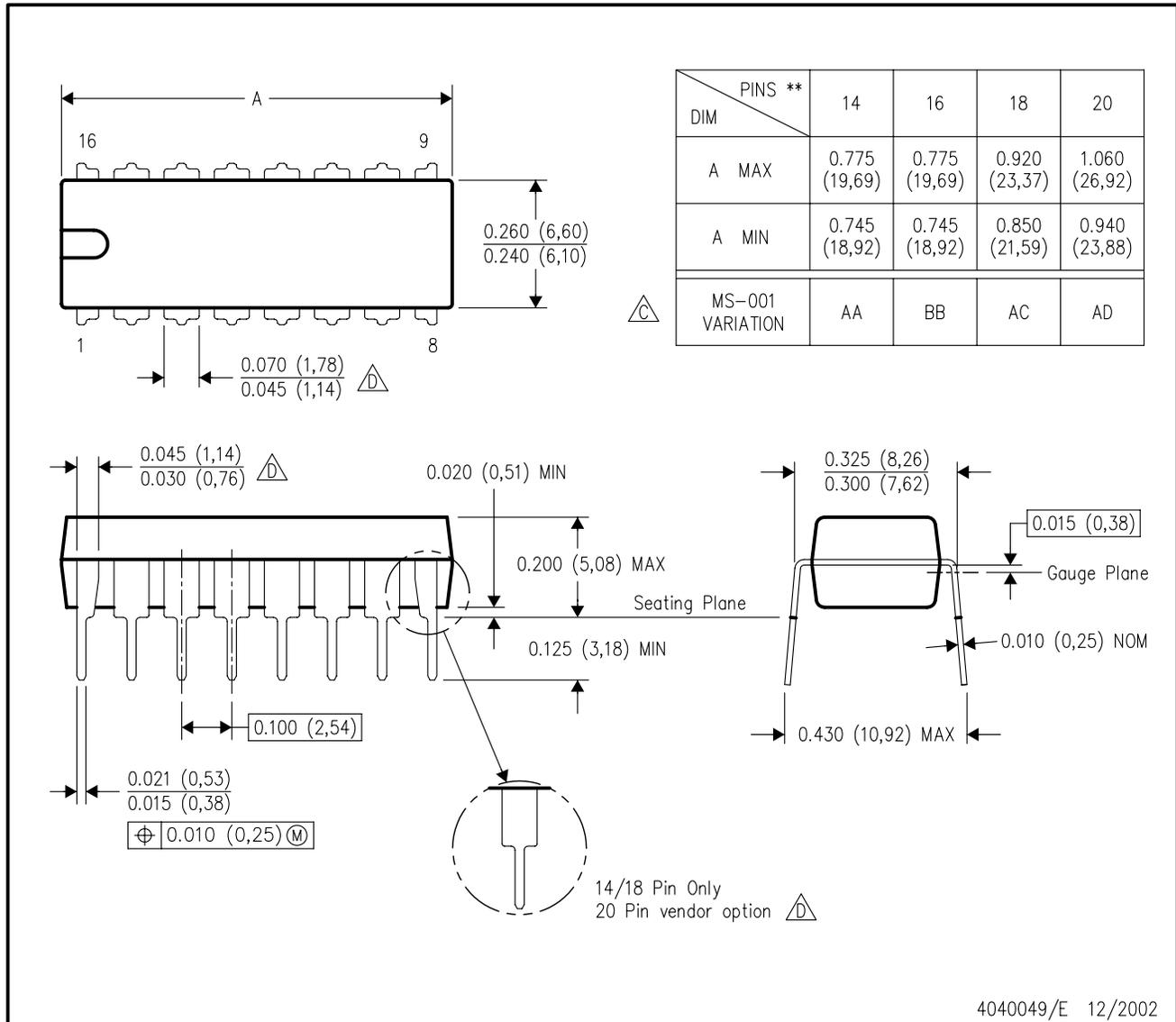
## 13 Mechanical, Packaging, and Orderable Information

The following pages include mechanical, packaging, and orderable information. This information is the most current data available for the designated devices. This data is subject to change without notice and revision of this document. For browser-based versions of this data sheet, refer to the left-hand navigation.

N (R-PDIP-T\*\*)

PLASTIC DUAL-IN-LINE PACKAGE

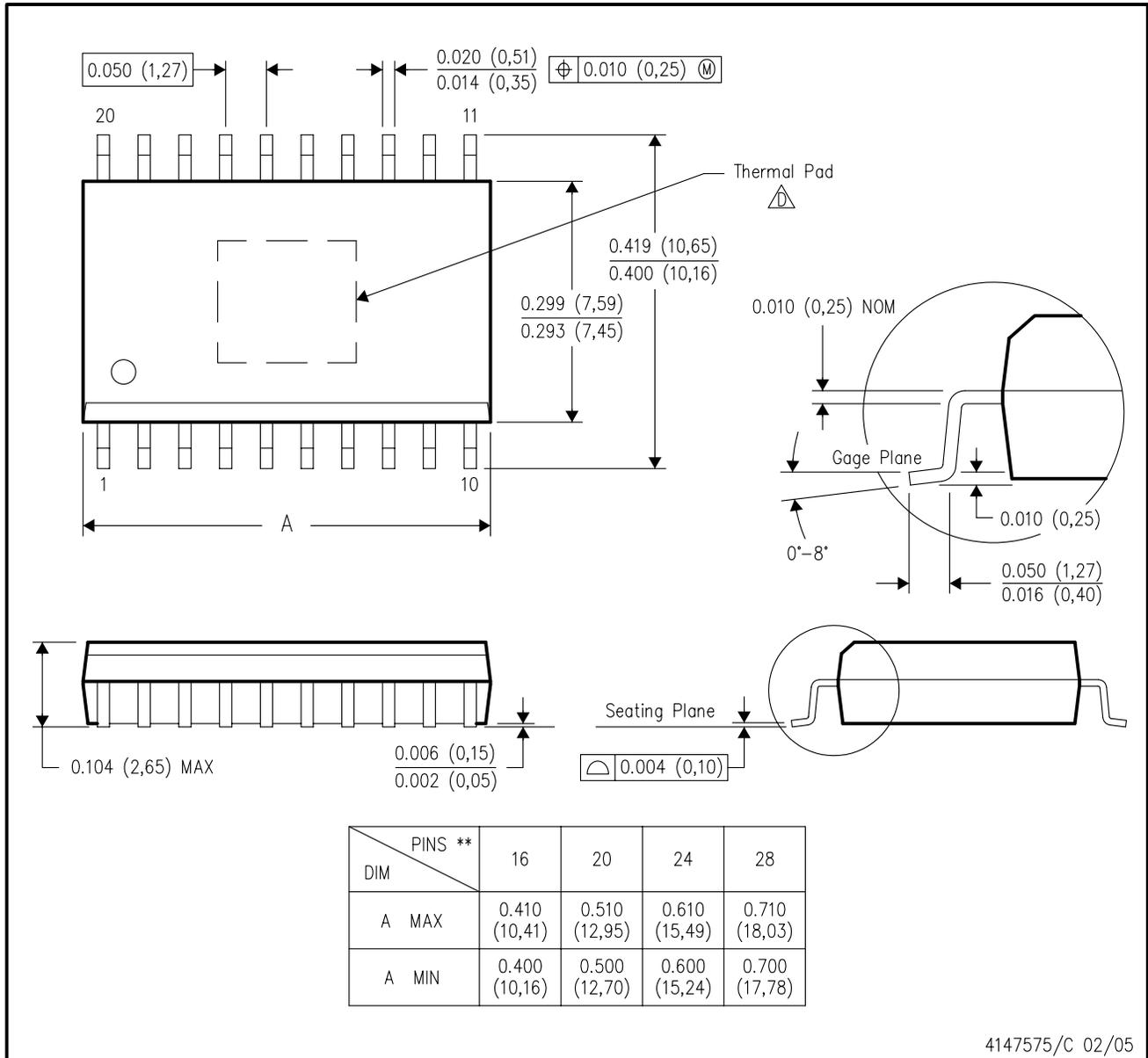
16 PINS SHOWN



- NOTES:
- A. All linear dimensions are in inches (millimeters).
  - B. This drawing is subject to change without notice.
  - Falls within JEDEC MS-001, except 18 and 20 pin minimum body length (Dim A).
  - The 20 pin end lead shoulder width is a vendor option, either half or full width.

DWP (R-PDSO-G\*\*) 20 PINS SHOWN

PowerPAD™ PLASTIC SMALL-OUTLINE PACKAGE

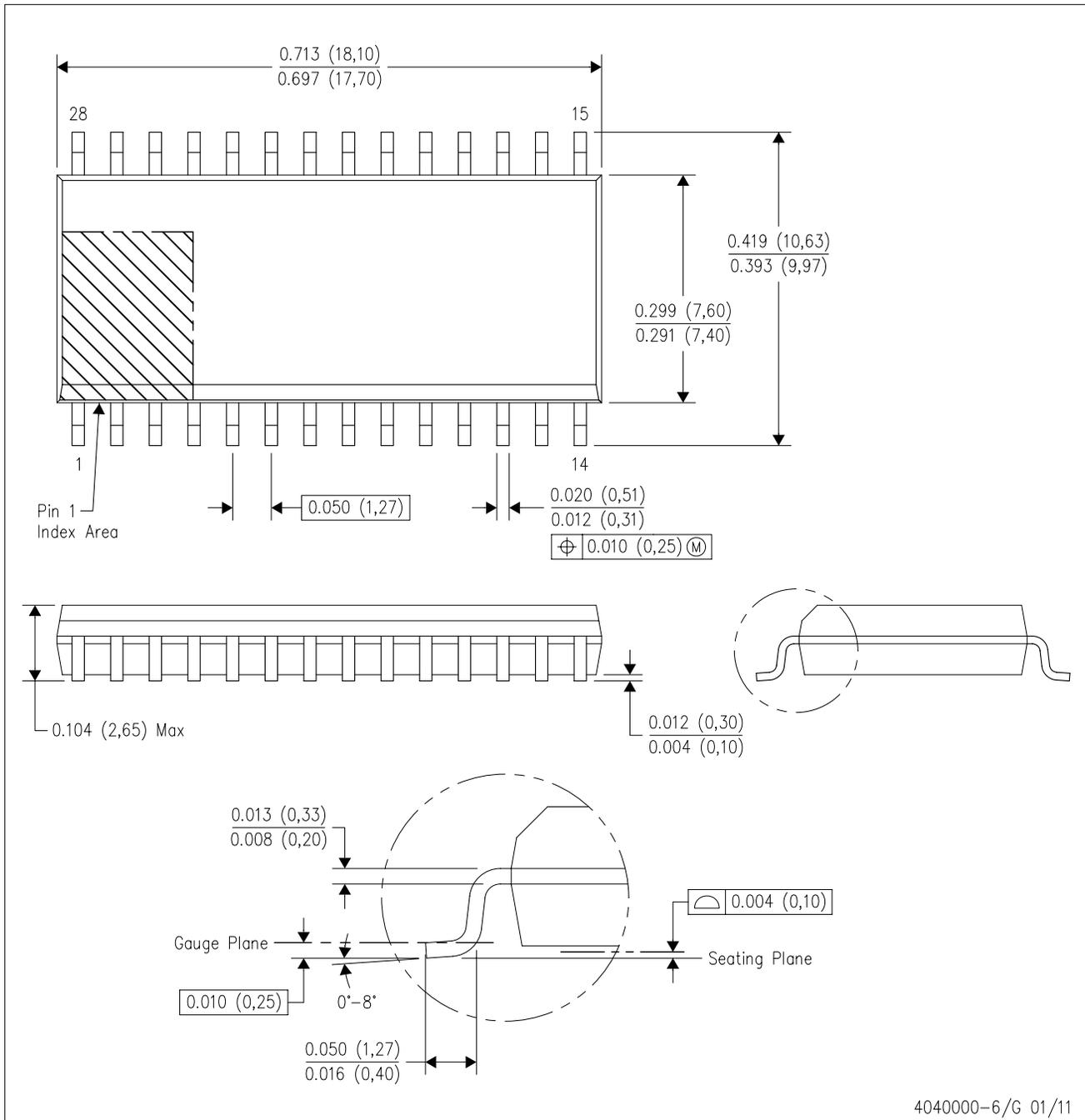


- NOTES:
- A. All linear dimensions are in inches (millimeters).
  - B. This drawing is subject to change without notice.
  - C. Body dimensions do not include mold flash or protrusion not to exceed 0.006 (0,15).
- △ This package is designed to be soldered to a thermal pad on the board. Refer to Technical Brief, PowerPAD Thermally Enhanced Package, Texas Instruments Literature No. SLMA002 for information regarding recommended board layout. This document is available at [www.ti.com](http://www.ti.com) <<http://www.ti.com>>. See the product data sheet for details regarding the exposed thermal pad dimensions.

PowerPAD is a trademark of Texas Instruments.

DW (R-PDSO-G28)

PLASTIC SMALL OUTLINE



4040000-6/G 01/11

- NOTES:
- All linear dimensions are in inches (millimeters). Dimensioning and tolerancing per ASME Y14.5M-1994.
  - This drawing is subject to change without notice.
  - Body dimensions do not include mold flash or protrusion not to exceed 0.006 (0,15).
  - Falls within JEDEC MS-013 variation AE.

## IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as "components") are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI's terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers' products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers' products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI's goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or "enhanced plastic" are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have **not** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

### Products

Audio	<a href="http://www.ti.com/audio">www.ti.com/audio</a>
Amplifiers	<a href="http://amplifier.ti.com">amplifier.ti.com</a>
Data Converters	<a href="http://dataconverter.ti.com">dataconverter.ti.com</a>
DLP® Products	<a href="http://www.dlp.com">www.dlp.com</a>
DSP	<a href="http://dsp.ti.com">dsp.ti.com</a>
Clocks and Timers	<a href="http://www.ti.com/clocks">www.ti.com/clocks</a>
Interface	<a href="http://interface.ti.com">interface.ti.com</a>
Logic	<a href="http://logic.ti.com">logic.ti.com</a>
Power Mgmt	<a href="http://power.ti.com">power.ti.com</a>
Microcontrollers	<a href="http://microcontroller.ti.com">microcontroller.ti.com</a>
RFID	<a href="http://www.ti-rfid.com">www.ti-rfid.com</a>
OMAP Applications Processors	<a href="http://www.ti.com/omap">www.ti.com/omap</a>
Wireless Connectivity	<a href="http://www.ti.com/wirelessconnectivity">www.ti.com/wirelessconnectivity</a>

### Applications

Automotive and Transportation	<a href="http://www.ti.com/automotive">www.ti.com/automotive</a>
Communications and Telecom	<a href="http://www.ti.com/communications">www.ti.com/communications</a>
Computers and Peripherals	<a href="http://www.ti.com/computers">www.ti.com/computers</a>
Consumer Electronics	<a href="http://www.ti.com/consumer-apps">www.ti.com/consumer-apps</a>
Energy and Lighting	<a href="http://www.ti.com/energy">www.ti.com/energy</a>
Industrial	<a href="http://www.ti.com/industrial">www.ti.com/industrial</a>
Medical	<a href="http://www.ti.com/medical">www.ti.com/medical</a>
Security	<a href="http://www.ti.com/security">www.ti.com/security</a>
Space, Avionics and Defense	<a href="http://www.ti.com/space-avionics-defense">www.ti.com/space-avionics-defense</a>
Video and Imaging	<a href="http://www.ti.com/video">www.ti.com/video</a>

### TI E2E Community

[e2e.ti.com](http://e2e.ti.com)