

# **ESCUELA POLITÉCNICA NACIONAL**

## **ESCUELA DE INGENIERÍA**

### **DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE AUTOMATIZACIÓN PARA EL HOGAR SOBRE EL PROTOCOLO X- 10 PRO Y CON INTERFAZ PARA PC**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN  
ELECTRÓNICA Y CONTROL**

**JORGE ANÍBAL SALAZAR FLORES**

**DIRECTOR: Dr. LUIS CORRALES**

**Quito, Marzo 2007**

## DECLARACIÓN

Yo, Jorge Aníbal Salazar Flores, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

---

Jorge Aníbal Salazar Flores

## **CERTIFICACIÓN**

Certifico que el presente trabajo fue realizado por Jorge Aníbal Salazar Flores, bajo mi supervisión.

---

Dr. Luis Corrales  
DIRECTOR DE PROYECTO

<b>CONTENIDO</b>	
<b>DECLARACIÓN.....</b>	<b>II</b>
<b>CERTIFICACIÓN.....</b>	<b>III</b>
<b>CONTENIDO.....</b>	<b>IV</b>
<b>RESUMEN.....</b>	<b>IX</b>
<b>PRESENTACIÓN.....</b>	<b>X</b>
<b>1. CAPITULO 1: REVISIÓN DE LOS FUNDAMENTOS X-10.</b>	
.....	<b>Pág. 2</b>
<b>1.1. PRINCIPIOS DEL PROTOCOLO X-10. ....</b>	<b>Pág. 3</b>
<b>1.1.1. FUNCIONAMIENTO GENERAL DEL PROTOCOLO X-10. ....</b>	<b>Pág. 4</b>
<b>1.1.2. TRAMA X-10. ....</b>	<b>Pág. 7</b>
<b>1.1.2.1.Encabezado. ....</b>	<b>Pág. 7</b>
<b>1.1.2.2.Código de casa. ....</b>	<b>Pág. 7</b>
<b>1.1.2.3.Código Clave. ....</b>	<b>Pág. 8</b>
<b>1.1.2.4.Consideraciones para enviar códigos X-10. ....</b>	<b>Pág. 9</b>
<b>1.2. JUSTIFICACIÓN DE PROYECTO. ....</b>	<b>Pág. 11</b>
<b>1.3. PROPUESTA DE DISEÑO. ....</b>	<b>Pág. 11</b>
<b>2. CAPÍTULO 2: DISEÑO DEL HARDWARE. ....</b>	<b>Pág. 16</b>
<b>2.1. DESCRIPCIÓN DEL MODULADOR DE SEÑALES X-10 :</b>	
<b>MÓDULO PSC04. ....</b>	<b>Pág. 17</b>
<b>2.1.1. ELEMENTOS DEL MÓDULO PSC04. ....</b>	<b>Pág. 18</b>
<b>2.1.1.1. Fuente de poder. ....</b>	<b>Pág. 19</b>
<b>2.1.1.2.Oscilador de 120kHz. ....</b>	<b>Pág. 20</b>
<b>2.1.1.3.Circuito de detección de cruce por cero. ....</b>	<b>Pág. 21</b>
<b>2.2. MÓDULO DE CONTROL. ....</b>	<b>Pág. 24</b>
<b>2.2.1. SELECCIÓN DEL MICROCONTROLADOR. ....</b>	<b>Pág. 24</b>
<b>2.2.2. DETECCIÓN DE CRUCE POR CERO. ....</b>	<b>Pág. 25</b>
<b>2.2.3. MODULADOR DE SEÑALES X-10. ....</b>	<b>Pág. 26</b>



3.1.5. ALMACENAMIENTO, LECTURA Y EJECUCIÓN DE COMANDOS GUARDADOS EN MEMORIA EEPROM. ....	Pág. 98
3.1.5.1. Almacenamiento de datos en memoria EEPROM. ....	Pág. 99
3.1.5.2. Algoritmo de lectura y procedimientos de ejecución de comandos almacenados en memoria EEPROM. ....	Pág. 102
3.1.6. ALGORITMO DE IGUALACIÓN DEL RELOJ. ....	Pág. 105
3.1.7. PRUEBAS DE FUNCIONAMIENTO DEL MÓDULO DE CONTROL.. .....	Pág. 107
<b>3.2. SOFTWARE PARA LA COMPUTADORA. ....</b>	<b>Pág. 113</b>
3.2.1. SUBVIS PARA ENVÍO DE COMANDOS X-10. ....	Pág. 114
3.2.1.1. Comandos de ambiente. ....	Pág. 115
3.2.1.1.1. <i>Comando Encender Todas las Luces.</i> ....	Pág. 115
3.2.1.1.2. <i>Comando Apagar Todas las Luces.</i> ....	Pág. 118
3.2.1.1.3. <i>Comando Apagar Todo.</i> ....	Pág. 120
3.2.1.2. Comandos X-10 individuales. ....	Pág. 122
3.2.1.2.1. <i>Comando Encender.</i> ....	Pág. 123
3.2.1.2.2. <i>Comando Apagar.</i> ....	Pág. 126
3.2.1.2.3. <i>Comandos Dim y Bright.</i> ....	Pág. 129
3.2.1.2.4. <i>Botón genérico.</i> ....	Pág. 135
3.2.1.2.5. <i>Generador de pulsos.</i> ....	Pág. 136
3.2.1.3. Comando para Igualar el reloj del Módulo de Control. ....	Pág. 139
3.2.1.4. Comando para escribir en la memoria EEPROM. ....	Pág. 142
3.2.2. CONSTRUCCIÓN DEL PANEL DE CONTROL DE LA INTERFAZ DE USUARIO. ....	Pág. 146
<b>3.3. SOFTWARE PARA LOS MÓDULOS RECEPTORES. .....</b>	<b>Pág. 151</b>
3.3.1. DISEÑO DEL PROGRAMA PARA LOS RECEPTORES. .....	Pág. 153
3.3.1.1. Lector De Bits X-10. ....	Pág. 153
3.3.1.2. Identificador De Encabezados. ....	Pág. 154
3.3.1.3. Receptor De Tramas X-10. ....	Pág. 156

3.3.1.4. Interprete De Comandos X-10. ....	Pág. 157
<b>4. CAPÍTULO 4: PRUEBAS Y RESULTADOS. ....</b>	<b>Pág. 162</b>
<b>4.1. PRUEBAS DE FUNCIONAMIENTO GENERAL. ....</b>	<b>Pág. 162</b>
4.1.1. DISPOSITIVOS DE ILUMINACIÓN TIPO ON / OFF. ....	Pág. 163
4.1.2. DISPOSITIVOS DE ILUMINACIÓN ATENUABLES. ....	Pág. 165
4.1.3. DISPOSITIVOS DE POTENCIA. ....	Pág. 168
<b>4.2. PRUEBAS DE COMPATIBILIDAD. ....</b>	<b>Pág. 170</b>
4.2.1. ENVÍO DE COMANDOS DESDE EL MÓDULO DE CONTROL DISEÑADO HACIA ACTUADORES COMERCIALES. ....	Pág. 170
4.2.2. ENVÍO DE COMANDOS DESDE MÓDULOS COMERCIALES HACIA LOS ACTUADORES DISEÑADOS. ....	Pág. 172
<b>4.3. PRUEBAS DE FUNCIONAMIENTO DE LA         PROGRAMACIÓN DE SECUENCIAS. ....</b>	<b>Pág. 174</b>
4.3.1. SECUENCIA 1: RUTINA PARA AUTOMATIZAR LAS TAREAS MATUTINAS Y REALIZAR EL DESAYUNO. ....	Pág. 174
4.3.2. SECUENCIA 2: SIMULACIÓN DE PRESENCIA EN EL HOGAR. .....	Pág. 177
<b>4.4. PRUEBAS DE FUNCIONAMIENTO DE LAS RUTINAS DE         AHORRO DE ENERGÍA. ....</b>	<b>Pág. 180</b>
4.4.1. PRINCIPALES CAUSAS DEL DESPERDICIO DE ENERGÍA. .....	Pág. 180
4.4.2. DISEÑO Y PRUEBA DE LAS RUTINAS DE AHORRO DE ENERGÍA. ....	Pág. 181
<b>5. CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES     .....</b>	<b>Pág. 185</b>
<b>5.1. CONCLUSIONES. ....</b>	<b>Pág. 185</b>
<b>5.2. RECOMENDACIONES. ....</b>	<b>Pág. 187</b>
<b>REFERENCIAS BIBLIOGRÁFICAS. ....</b>	<b>Pág. 189</b>

**ANEXOS**

<b>ANEXO 1: DIAGRAMA DEL CIRCUITO DE CONTROL DEL MÓDULO DE CONTROL .....</b>	<b>Pág. 192</b>
<b>ANEXO 2: DIAGRAMA DEL CIRCUITO DE CONTROL DEL MÓDULO RECEPTOR.....</b>	<b>Pág. 194</b>
<b>ANEXO 3: HOJA DE DATOS DEL CIRCUITO INTEGRADO MAX232 .....</b>	<b>Pág. 196</b>
<b>ANEXO 4: HOJA DE DATOS DEL CIRCUITO INTEGRADO 4N25 (NTE3040) .....</b>	<b>Pág. 200</b>
<b>ANEXO 5: HOJA DE DATOS DEL CIRCUITO INTEGRADO LM324N (NTE987).. .....</b>	<b>Pág. 203</b>
<b>ANEXO 6: HOJA DE DATOS DEL CIRCUITO INTEGRADO TL084 (NTE859).. .....</b>	<b>Pág. 207</b>

## RESUMEN

En este trabajo se desarrolla un sistema de automatización para el hogar utilizando el Protocolo X-10 Pro para enviar las señales de control por el cableado eléctrico ya existente. Esto permite implementar un sistema de domótica de bajo costo y de fácil instalación, el cual está diseñado para facilitar la realización de las tareas diarias dentro del hogar y permitir el ahorro de energía mediante la implementación de una rutina que apague los electrodomésticos que se quedan encendidos innecesariamente.

El sistema de domótica es controlado a través de una interfaz para PC “amigable” diseñada utilizando LabView. Ésta interfaz permite el control de cada uno de los elementos de la Red X-10 en tiempo real, ya sea desde la PC conectada al Módulo de Control X-10 o desde una ubicación remota utilizando una conexión de Internet. La interfaz también permite la programación de comandos en la memoria no volátil del Módulo de Control para que sean ejecutados automáticamente.

El Módulo de Control utiliza el módulo comercial PSC04 como modulador de señales X-10 y requiere de la interfaz con la PC para ser manejado. El Módulo de Control permite recibir la información de los comandos X-10 desde la PC en formato comprimido, para optimizar tiempos de transmisión y para ahorrar espacio en la memoria no volátil en el caso de que se le envíe un comando que se debe ejecutar posteriormente. Éste módulo también tiene implementado un reloj en tiempo real y los algoritmos necesarios para descomprimir la información enviada desde la PC y convertirla en paquetes X-10.

Los Módulos actuadores están diseñados para filtrar y demodular las señales X-10 que detecten viajando por la Red. Luego éstos módulos determinan si son los destinatarios del paquete y toman las acciones de control apropiadas.

Todos los módulos diseñados están basados en sistemas micro procesados para permitir que se realice fácilmente cualquier mejora de seguridad o de transmisión.

## PRESENTACIÓN

En la actualidad, la tecnología se está involucrando en todos los aspectos de la vida de los seres humanos. La Domótica pretende introducir al control automático a los hogares para brindar mayor confort y facilitar las tareas cotidianas de la vida.

El protocolo X-10 es el sistema de domótica de más fácil instalación y de menor precio, en comparación a sus principales competidores que utilizan la tecnología Bluetooth para implementar sistemas de domótica. Las ventajas que presenta el protocolo X-10 es que no requiere cableado adicional al ya existente en los hogares y prácticamente no recibe interferencias electromagnéticas provenientes de las señales de radiofrecuencia que en la actualidad inundan nuestros hogares.

El presente proyecto ofrece una manera de utilizar el protocolo X-10 como columna vertebral de un sistema de domótica que pueda integrar no solo dispositivos X-10 sino que permita la interacción con dispositivos inalámbricos. Este proyecto se enfoca en la digitalización de los elementos básicos de una red X-10 con dispositivos electrónicos fáciles de conseguir en el mercado nacional. También desarrolla el software necesario para manejar los recursos de una manera eficiente y eficaz.

En el Capítulo 1 se presentan los fundamentos del protocolo X-10 desde una breve historia de sus orígenes y el enfoque que le dieron sus creadores hasta una explicación detallada de las tramas X-10, la manera en que se construyen los paquetes X-10 y las consideraciones que se deben tomar en cuenta para poder enviar estos datos a través del medio físico.

El diseño del Hardware tanto para el Módulo de Control como para los Módulos Actuadores se aborda en el Capítulo 2. Para la modulación de comandos X-10 se utiliza el Módulo comercial PSC04 ya que su adquisición también otorga permisos legales para utilizar el protocolo X-10 PRO para construir sistemas de domótica. Para el caso de la construcción de los Módulos Actuadores se pone énfasis en los

circuitos utilizados para la demodulación X-10 ya que fue el proceso más difícil de lograr en lo que se refiere al diseño del Hardware.

El Capítulo 3 describe el funcionamiento del Software, cuya construcción se la realizó poniendo énfasis en el diseño modular de sus subrutinas. Dichas subrutinas permitirán que las mejoras recomendadas para éste proyecto sean fáciles de implementar.

Las pruebas y los resultados obtenidos se tratan en el Capítulo 4. La información obtenida se presenta en forma de tablas para su fácil comprensión.

En el Capítulo 5 se presentan las principales conclusiones a las que se llegó luego del estudio e implementación del Protocolo X-10. En el mismo capítulo se presentan varias recomendaciones que permitirán al lector evitar algunos errores al momento de implementar una red X-10.

Finalmente se presentan las fuentes bibliográficas que, junto con los Anexos, brindaron la información necesaria para la realización de este proyecto.

# **CAPÍTULO 1**

## **REVISIÓN DE LOS FUNDAMENTOS X-10**

## **CAPÍTULO 1 : REVISIÓN DE LOS FUNDAMENTOS X-10**

El objetivo global del presente proyecto de titulación es incursionar en el campo de la Domótica utilizando el protocolo X-10, un protocolo que a nivel mundial está ampliamente difundido pero en el Ecuador es todavía una tecnología en desarrollo. La Domótica permite tener control de luces, electrodomésticos y alarmas de una manera integral, incorporando tecnologías digitales para permitir un control con algoritmos mas eficientes y eficaces. La digitalización del sistema abre la posibilidad una comunicación no solo entre los dispositivos del sistema X-10 sino que adicionalmente permite conectarse a través de una PC, desde cualquier lugar del mundo a través de la Internet.

La ventaja de utilizar el protocolo X-10 para hacer automatización en hogares es que no se requiere cableado adicional para las señales de control. Igualmente, los costos de los dispositivos X-10 son bajos en comparación con los de otros sistemas, y su instalación es rápida y sencilla. Gracias a estas ventajas se puede ampliar el alcance de éstas tecnologías que facilitan las tareas cotidianas y permiten ahorrar tiempo y dinero. Por otro lado es importante estudiar tecnologías emergentes ya que solo así se podrá intentar acortar la brecha de desarrollo tecnológico entre nuestro país y los países desarrollados.

En forma más específica, el objetivo principal del presente proyecto es digitalizar completamente los transmisores y receptores de códigos X-10, para que en el futuro se puedan crear redes más inteligentes, capaces de ejecutar tareas de mayor complejidad que las tareas que actualmente cumplen los dispositivos análogos que se hallan al momento en el mercado.

La digitalización de los dispositivos X-10 también permitirá una mejor integración de la tecnología X-10 con otras tecnologías que utilizan sistemas microprocesados, lo cual permitirá reducir muchas de las debilidades que el protocolo X-10 presenta en la actualidad.

## **1.1. PRINCIPIOS DEL PROTOCOLO X-10.**

El protocolo X-10 está diseñado para enviar y recibir información a través de señales que viajan por las redes eléctricas domésticas ya instaladas y pueden ser combinadas con señales de radiofrecuencia para permitir control inalámbrico.

La utilización de las redes eléctricas ya instaladas evita la necesidad de un cableado de control adicional por lo que este método de transmisión da la posibilidad de automatizar hogares y oficinas de una manera sencilla y económica. Otra ventaja es que permite una compatibilidad casi total entre los dispositivos X-10 de diferente fabricante.

Esta tecnología fue creada entre 1976 y 1978 por Pico Electronics Ltd. en Glenrothes, Escocia y fue parte de los estudios realizados en los proyectos llamados "series X". Luego de observar el gran potencial que ofrecían los sistemas de transmisión de datos a través de las redes eléctricas, se fue perfeccionando el protocolo X-10, pero siempre manteniendo la compatibilidad hacia atrás.

En la actualidad, la versión más usada de éste protocolo es el X-10 PRO, el cual fue introducido en 1978 por Sears Home Control System y Radio Shack Plug'n Power System. Esta nueva versión brinda la facilidad de poder utilizar códigos extendidos y datos extendidos, lo cual permite crear códigos adicionales que permitan cubrir las necesidades crecientes de los usuarios. En la actualidad se ha combinado el envío de señales de control, a través de las redes eléctricas, con radiotransmisores que envían señales de control a radiorreceptores que transforman la señal de radio en señal X-10 y la inyectan en la red para controlar los dispositivos X-10 instalados. El más reciente avance en la automatización de hogares combina todas las tecnologías ya mencionadas con Internet. De esta manera se puede monitorear ciertos elementos del sistema X-10, aunque esta tecnología todavía está en desarrollo.

El protocolo X-10 se encuentra en auge en países como: Estados Unidos, Canadá, Alemania, Francia, Suiza, Holanda, Japón y Australia entre otros. En la actualidad muchas empresas fabrican equipos compatibles con el protocolo X-10 PRO, por ejemplo: G.E., Radio Shack (Plug-N-Power), Sears, Stanley (Lightmaker), IBM (Home Director), Zenith, X10 Pro, X10 Powerhouse, ActiveHome, Smart Home, entre otras.

### **1.1.1 FUNCIONAMIENTO GENERAL DEL PROTOCOLO X-10**

Ya se indicó que el protocolo de comunicación X-10 es un sistema de transmisión de datos digitales que usa las instalaciones eléctricas de hogares y oficinas. La versión original de este protocolo permitía realizar seis funciones: encender, apagar, “dim” que significa atenuar, “bright” que significa iluminar, encender todo y apagar todo. Las versiones actuales permiten ejecutar otras nueve funciones además de las tradicionales. Entre las nueve funciones adicionales se encuentran dos que permiten que el protocolo pueda realizar una gran cantidad de tareas nuevas. Estos comandos permiten enviar códigos extendidos (normalmente hasta 256 códigos) que a su vez permiten enviar datos extendidos, normalmente hasta dos bytes adicionales, aunque, dependiendo de la tecnología usada, pueden ser ilimitados si no hay conflictos entre los dispositivos instalados.

Para realizar la transmisión de datos se utilizan señales de radiofrecuencia que se inyectan a la red eléctrica, sincronizándolas con los cruces por cero de la señal de poder ( 60 Hz). Esta técnica es llamada control por corriente portadora (“carrier current” control)<sup>1</sup>. Por ejemplo, para transmitir un uno lógico es necesario inyectar señales de radiofrecuencia de 120kHz, dentro de los 200 microsegundos posteriores al cruce por cero de la señal de poder. La Figura 1.1 muestra un diagrama de los tiempos máximos y mínimos que se deben respetar para la transmisión de un código X-10.

---

<sup>1</sup> Intro to X-10

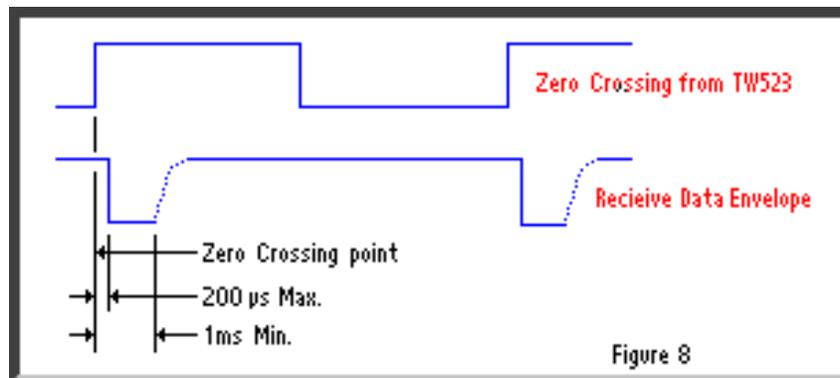


Figura 1.1 Diagrama de tiempos<sup>2</sup>

La presencia de las señales de radiofrecuencia en la red debe ser de 1 milisegundo para que el uno lógico sea válido. Un cero lógico es representado por la ausencia de las señales de radiofrecuencia.

Otra consideración importante es que en un sistema trifásico los dispositivos X-10 que se encuentren conectados en fases distintas no pueden comunicarse, a menos de que se coloque un puente de señal que permita que la señal X-10 viaje por toda la red sin importar la fase. Por este motivo es necesario que los pulsos de radiofrecuencia se retransmitan al  $1/3$  y a los  $2/3$  del semiperiodo. La ilustración de la transmisión de un uno lógico en X-10 para un sistema trifásico de 60Hz se puede ver en la Figura 1.2.

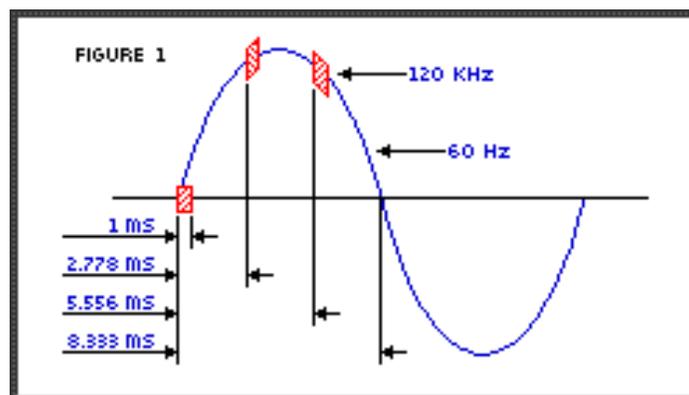


Figura 1.2. Envío de un 1 lógico<sup>3</sup>

<sup>2</sup> X-10 Transmission theory

<sup>3</sup> X-10 Transmission theory

Un paquete de datos X-10 o byte se transmite utilizando 11 ciclos de la señal de poder. El paquete consta en primer lugar de un encabezado, el cual es transmitido durante dos ciclos; en segundo lugar un código denominado de casa, para el cual se necesitan cuatro ciclos; y finalmente el Código Clave, el cual puede ser un Código de Dirección o un Código de Función. El Código Clave requiere de cinco ciclos para ser transmitido

El Código de Casa junto con el código de dirección sirven para identificar a cualquiera de los 256 dispositivos X-10 admitidos dentro de una red X-10. El Código de Casa fue pensado originalmente para no tener interferencias entre vecinos que utilicen este protocolo en la automatización de sus hogares. Cada vecino tomaría uno de los 16 códigos de casa y dentro de su hogar distribuiría los 16 códigos de dirección que corresponden a su Código de Casa asignado. En el presente proyecto se prevé utilizar el Código de Casa para agrupar las distintas áreas dentro de un hogar para aprovechar al máximo las características de éste protocolo. Las interferencias con los hogares vecinos se pueden evitar instalando un filtro a la entrada del cable de alimentación eléctrica del hogar.

Para el envío de un código X-10 completo es necesario transmitir dos bytes con un intervalo entre ellos de tres ciclos completos de la línea de poder. En el primer byte se transmite el Encabezado, el Código de Casa y el Código de Dirección. En el segundo byte se transmite el Encabezado, el Código de Casa y el Código de Función.

La repetición del Código de Casa en los dos paquetes y el envío del complemento de todos los bits, a excepción de los bits del encabezado, da cierta seguridad a la transmisión; es decir, la única manera de verificación de errores que posee este sistema es el envío duplicado de información. De lo indicado se puede deducir que un código X-10 completo se transmite en 25 ciclos de la línea de poder. Debido a que la tasa de transferencia en un sistema a 60 Hz es de 60 bps, la transmisión del código completo tarda 417 milisegundos. Por esta razón, este protocolo solo es usado para tareas en las que la velocidad de transmisión de datos no es vital.

Adicionalmente, se debe considerar que para mantener la compatibilidad hacia atrás es necesario dejar una pausa de tres ciclos de línea entre el fin del primer comando completo enviado y el siguiente. Esto es necesario ya que los dispositivos análogos que se encuentran en el mercado están diseñados para esperar tres ciclos de línea antes de comenzar a leer el siguiente paquete.

## 1.1.2 TRAMA X-10

### 1.1.2.1 Encabezado

El encabezado siempre es el código "1110". Los bits se transmiten cada cruce por cero, es decir, un bit por cada semiciclo de la línea de poder.

### 1.1.2.2 Código de casa

El Código de Casa permite 16 diferentes combinaciones, las cuales son identificadas por las letras de la "A" a la "P". La Tabla 1.1 permite visualizar las diferentes combinaciones para el código de casa.

Tabla 1.1. Códigos de Casa

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

Para transmitir el Código de Casa se debe transmitir un bit durante el primer semiciclo y su complemento lógico durante el segundo semiciclo, por lo que la transmisión de cada bit se la realiza en un ciclo completo de la línea de poder. Por ejemplo, para transmitir el Código de Casa "A" (0110) se debe transmitir la secuencia:

0	1	1	0	1	0	0	1
---	---	---	---	---	---	---	---

Lo que implica ocupar 4 ciclos de la línea de poder.

### 1.1.2.3 Código Clave

Este código permite las combinaciones siguientes. Las 16 primeras corresponden a la dirección del dispositivo y las otras 16 corresponden a una función a ser ejecutada. Las direcciones de los dispositivos son identificadas por los números del 1 al 16. Existen 15 comandos diferentes dentro del protocolo X-10 Pro. La Tabla 1.2 explica las combinaciones posibles.

Tabla 1.2. Explicación del Código Clave – Códigos de dirección

		D1	D2	D4	D8	D16
Código de Dirección	1	0	1	1	0	0
	2	1	1	1	0	0
	3	0	0	1	0	0
	4	1	0	1	0	0
	5	0	0	0	1	0
	6	1	0	0	1	0
	7	0	1	0	1	0
	8	1	1	0	1	0
	9	0	1	1	1	0
	10	1	1	1	1	0
	11	0	0	1	1	0
	12	1	0	1	1	0
	13	0	0	0	0	0
	14	1	0	0	0	0
	15	0	1	0	0	0
	16	1	1	0	0	0

Tabla 1.2. Explicación del Código Clave – Códigos de función

		D1	D2	D4	D8	D16
Código de Función	Apagar todos los dispositivos	0	0	0	0	1
	Encender todas las luces	0	0	0	1	1
	Encender	0	0	1	0	1
	Apagar	0	0	1	1	1
	Dim	0	1	0	0	1
	Brigth	0	1	0	1	1
	Apagar todas las luces	0	1	1	0	1
	Código extendido	0	1	1	1	1
	Pedir saludo (Hail request)	1	0	0	0	1
	Conceder saludo	1	0	0	1	1
	Dim pre seteado	1	0	1	X	1
	Dato extendido	1	1	0	0	1
	Estado encendido	1	1	0	1	1
	Estado apagado	1	1	1	0	1
	Petición de estado	1	1	1	1	1

#### 1.1.2.4 Consideraciones para enviar códigos X-10

No todos los comandos se envían de la misma manera, las excepciones y consideraciones especiales son las siguientes.

En primer lugar, se mencionó que se debe dejar una pausa de tres ciclos de la línea de poder entre un comando X-10 completo y otro para poder mantener la compatibilidad hacia atrás con los módulos antiguos de X-10 PRO.

En el caso de que el comando sea para toda una casa (es decir, todos los dispositivos tienen el mismo Código de Casa) es suficiente con el envío de un solo byte, el cual contendrá: encabezado, Código de Casa y Código Clave. Los comandos que se aplican a todos los dispositivos con el mismo Código de Casa son: encender todas las luces, apagar todas las luces y apagar todos los dispositivos.

Cuando se ejecuta un código de Dim o Brigth se aumenta o disminuye el voltaje de alimentación en un paso. Si se desea variar más que un solo paso se envían más comandos de Dim o Bright. Para ello se envía la dirección en el primer byte y

el primer código de Dim en el segundo byte. Los dos primeros Bytes se pueden transmitir normalmente pero los siguientes comandos de Dim o Bright se envían consecutivamente, sin ninguna pausa entre los bytes. Generalmente son 20 el número de pasos de voltaje que se pueden dar para apagar o encender completamente un módulo.

Para el envío de una instrucción de Código Extendido se envían normalmente los dos primeros bytes y, sin ninguna pausa, se envía un byte de ocho bits el cual representa un comando distinto de los 15 comandos estándar del protocolo X-10.

El código “Pedir Saludo” sirve para buscar otros dispositivos capaces de enviar códigos X-10 PRO dentro del rango de comunicación del dispositivo que lo envía. Esta petición de saludo será respondida por cualquier otro transmisor/receptor de señales X-10 que reciba el comando “Pedir Saludo” y tenga el Código de Casa recibido en dicho comando. La petición de saludo se responde con un comando de “Contestar Saludo”. Estos comandos permiten evitar que dos transmisores X-10 tengan la misma Dirección de Casa e impedir que algún dispositivo de otro propietario de controladores X-10 interfiera con el funcionamiento del sistema a instalarse. Lo más aconsejable para evitar estas interferencias es colocar un filtro contra radiofrecuencia en el cable de alimentación del hogar donde se van a instalar dispositivos X-10.

Para una instrucción de Dim conteniendo un valor pre seteado, el bit D8 es el bit mas significativo del nivel de luminosidad y los bits H1, H2, H4 y H8 son los menos significativos de los 5 bits que contienen la información del “set point” de luminosidad. La combinación de estos 5 bits permiten 32 combinaciones distintas, pero por lo general los actuadores solo permiten valores entre 0 y 20. Para enviar esta instrucción el contenido del segundo byte es el único que se debe alterar.

El código de Dato Extendido se usa para recibir información de cualquier tipo, como por ejemplo enviar pequeños archivos de eventos ocurridos durante el día desde un sensor hasta el Módulo de Control. La transmisión es normal para los dos primeros bytes (como en el caso de Dim o bright) pero los siguientes bytes

serán de 8 bits y sin pausas entre ellos. Se recomienda que el primer byte adicional contenga el número de bytes de 8 bits que se van a transmitir, aunque eso depende del protocolo fijado por el programador para el envío de datos extendidos, porque el protocolo X-10 no ha normalizado esto.

Los comandos para petición de estado y reporte de estado se usan para dispositivos con capacidad de transmisión bidireccional y sirven para monitorear el funcionamiento de los módulos X-10.

## **1.2. JUSTIFICACIÓN DE PROYECTO**

En el mercado existen una gran variedad de equipos para Domótica que utilizan el protocolo X-10 debido a las ventajas económicas, que presenta frente a sus competidores, como el ahorro en la instalación, por dar un ejemplo. Pero sus grandes desventajas son la velocidad de transmisión y la falta de comunicación bidireccional entre todos los componentes del sistema. Este trabajo busca dar el primer paso para solucionar las deficiencias de los sistemas tradicionales de Domótica basados en el protocolo X-10: Se incorporan microprocesadores tanto para la transmisión como para la recepción de los códigos X-10. Éste avance permitirá que en el futuro se incremente la velocidad de transmisión mediante técnicas de modulación y transmisión digital, manteniendo la compatibilidad hacia atrás para aprovechar la amplia difusión de dispositivos que utilizan este protocolo. La migración de sistemas análogos a digitales también permitirá la evolución del protocolo X-10 a un sistema bidireccional de comunicación con seguridades en la transmisión y recepción de datos, para poder diseñar sistemas de control y de seguridad que no tengan las debilidades que tienen los sistemas actuales que utilizan el protocolo X-10.

## **1.3. PROPUESTA DE DISEÑO**

El protocolo X-10 PRO está patentado por sus creadores sin embargo al adquirir un módulo PSC04 o PSC05 se obtiene el permiso de realizar cualquier tipo de aplicación que use el protocolo X-10 PRO. Adquirir estos equipos es la mejor

alternativa para familiarizarse con la tecnología y desarrollar las primeras aplicaciones sin la necesidad de obtener permisos para la utilización de este protocolo ni de desarrollar un codificador de señales X-10 (el desarrollo del codificador requiere de conocimientos de alta frecuencia y de materiales de difícil adquisición en nuestro medio). Por estos motivos se eligió comprar el módulo PSC04 para inyectar la señal X10 en la red. Como interfase entre la PC y el prototipo del Módulo de Control se utilizará el microcontrolador PIC 16F877A debido a que cumple fácilmente con los requerimientos de la aplicación y en el mercado se tiene una buena oferta de este IC y por lo tanto también su precio es conveniente. En el capítulo correspondiente se detallarán los requerimientos de procesamiento para realizar esta interfaz.

Este proyecto busca diseñar un sistema de automatización para hogares y oficinas pequeñas que permita el control de encendido y apagado de aparatos eléctricos así como el control del encendido, apagado y nivel de luminosidad de bombillas incandescentes. Para ello se ha pensado en un microcontrolador como elemento de control. El programa del microcontrolador será diseñado en base a las consideraciones necesarias de tiempos de sincronización y transmisión de datos para mantener la compatibilidad con cualquier otro dispositivo X-10 que se encuentre en el mercado. El diagrama de bloques propuesto se muestra en la Figura 1.3.

El programa del microcontrolador deberá permitir comunicación serial mediante protocolo RS232 con una computadora personal, la cual servirá de interfaz entre el usuario y el módulo controlador. El controlador estará en capacidad de ejecutar comandos X-10 en determinado momento o podrá almacenar comandos en su memoria no volátil para ejecutarlos en otro instante, incluso podrá almacenar instrucciones repetitivas que serán ejecutadas diariamente o semanalmente a una hora programada y en la secuencia determinada por el usuario.

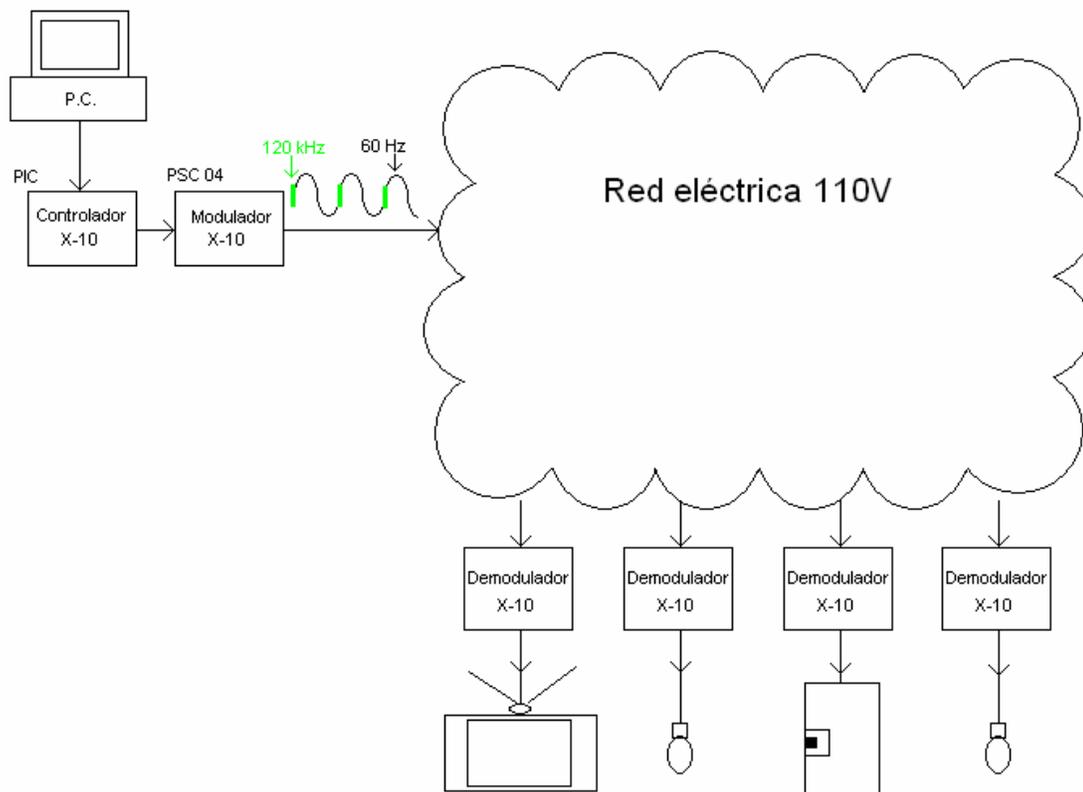


Figura 1.3 Diagrama de bloques propuesto

La comunicación entre la computadora y el Módulo de Control permitirá la verificación de errores y la petición de reenvío de datos en caso de que exista alguna falla en la comunicación. El envío de datos de la computadora al Módulo de Control se lo realizará mediante reglas de comunicación que permiten la diferenciación entre comandos inmediatos y comandos programados. El microprocesador realizará la codificación a protocolo X-10 al momento de enviar el comando por la red eléctrica, lo que permitirá compresión de datos para optimizar el uso de memoria y una transmisión de datos más rápida entre la computadora y el Módulo de Control. Para la ejecución de comandos programados el microcontrolador tendrá un reloj generado por software que se igualará cada vez que se inicia la comunicación con la computadora.

El programa de la computadora se lo realizará en Labview debido a las herramientas que posee para la utilización de puertos y por las facilidades y seguridades que brinda para la comunicación vía Internet. El programa estará

diseñado para ser de fácil manejo para el usuario con una organización por habitaciones. De esta manera resultará fácil encontrar el dispositivo que se quiere controlar y dar el comando deseado. La ventana para programación de comandos permitirá elaborar secuencias de encendido que serán ejecutadas una o varias veces según la necesidad del usuario.

El envío del comando X-10 a los dispositivos conectados a la red eléctrica se lo realizará a través del módulo PSC04, el cual inyectará una señal de 120kHz a la red cada vez que se coloca un uno lógico en su entrada. Este módulo también entregará una señal de detección de cruce por cero.

Los módulos actuadores permiten extraer la señal X-10 de la red eléctrica y la acondicionan para poder ser utilizada por un microcontrolador, el cual valida los bits según el protocolo X-10, realiza la comprobación de dirección y en caso de que la dirección recibida coincida con la suya decodifica y ejecuta el comando X-10.

# **CAPÍTULO 2**

## **DISEÑO DEL HARDWARE**

## **CAPÍTULO 2: DISEÑO DEL HARDWARE**

En este capítulo se describen los pasos seguidos para el diseño del hardware del presente proyecto. El hardware a diseñarse está constituido por tres módulos que deben interactuar entre si.

Un modulador de señales X-10, con capacidad de generar señales de 120kHz y proveer aislamiento entre la red y el Módulo de Control.

Un Módulo de Control, el cual se basa en un microcontrolador que debe codificar y generar señales X-10, comunicarse con una PC, guardar comandos X-10 en memoria no volátil, y decodificarlos para ejecutar los comandos programados a la hora especificada.

Un módulo receptor, el cual debe poder demodular las señales X-10 que viajan en la red eléctrica, identificar y validar comandos para ejecutarlos.

Antes de poder diseñar los moduladores y demoduladores, fue necesario hacer varios estudios previos de dispositivos comerciales, para comprender su funcionamiento a nivel de Hardware y como interactúa con el Software que es provisto como interfase entre el usuario y el sistema. Solo de esta manera se pudo comprender con la profundidad necesaria el modo de operación de los dispositivos X-10 para identificar los métodos de transmisión de los dispositivos comerciales análogos y así tener una guía para el diseño de los dispositivos digitales. Gracias a este trabajo previo fue posible identificar la mayoría de componentes básicos que deberían tener los dispositivos a diseñarse.

El principal problema que hubo al momento de diseñar y construir los módulos receptores es que no es posible utilizar los circuitos integrados especiales con los que cuentan los módulos comerciales. Por esta razón, fue necesario analizar varias soluciones alternativas usando circuitos integrados que se puedan encontrar en el mercado.

## 2.1 DESCRIPCIÓN DEL MODULADOR DE SEÑALES X-10 : MÓDULO PSC04

Este módulo es el encargado de convertir los pulsos TTL que entrega el Módulo de Control a señales X-10; es decir, éste módulo es el encargado de transformar los pulsos de control entregados por el microcontrolador en señales con una frecuencia de 120kHz e inyectarlos en la red eléctrica.

Para el caso del modulador se decidió comprar el módulo PSC04 debido a que luego de realizar un análisis detallado de sus circuitos internos, se observó que se requeriría de algunos elementos difíciles de conseguir en el mercado local. Adicionalmente, al comprar este módulo también se obtiene el permiso de transmitir señales en codificación X-10 Pro. Éste módulo también contiene un detector de cruce por cero. Para inyectar la señal de 120kHz se debe enviar una señal de control desde el Módulo de Control hacia el módulo PSC04 que debe estar entre 4 y 20 voltios de corriente continua, lo que provocará una corriente de 2.5 a 18 mili amperios en la entrada del módulo PSC04. Cuando se transmita un cero, el voltaje de control debe ser menor de 0.8 voltios<sup>4</sup>. Para adquirir la señal de cruce por cero que el módulo PSC04 produce, se requiere construir un circuito que utilice el opto transistor de salida de dicho módulo para generar una onda cuadrada. El opto transistor se comporta como un interruptor cerrado durante el semiciclo positivo y como un interruptor abierto durante el semiciclo negativo. El módulo PSC04 se lo presenta en la Figura 2.1.



Figura 2.1. Módulo PSC04

---

<sup>4</sup> Tech Note X-10 communications protocol and power line interface PSC04 & PSC05

Como se puede observar en la Figura 2.1, la conexión de este módulo con el controlador se la realiza mediante un conector telefónico RJ11. La distribución de los cables de conexión es la siguiente:

1. B (negro).- Salida de detección de cruce por cero.
2. R (rojo).- Tierra para señal de cruce por cero.
3. G (verde).- Tierra para la señal de control de envío X-10
4. Y (amarillo).- Entrada de la señal de control de envío X-10

### 2.1.1 ELEMENTOS DEL MÓDULO PSC04

El módulo PSC04 ha sido diseñado para transmitir señales de radiofrecuencia de 120kHz con una amplitud de hasta 5V pico – pico (en promedio entrega a la red una potencia de 60 mW sobre una carga de 5  $\Omega$ ) dependiendo del estado de la entrada: 0L o 1L. Éste módulo también está en capacidad de detectar los cruces por cero de la línea de poder y notificarlos a través de un transistor de colector abierto. Éste transistor puede estar abierto o cerrado dependiendo del semiciclo en el que se encuentre la línea de poder. La última función del módulo PSC04 es la de brindar aislamiento óptico entre la red eléctrica y el dispositivo de control. La Figura 2.2 contiene el diagrama de bloques del módulo PSC04.

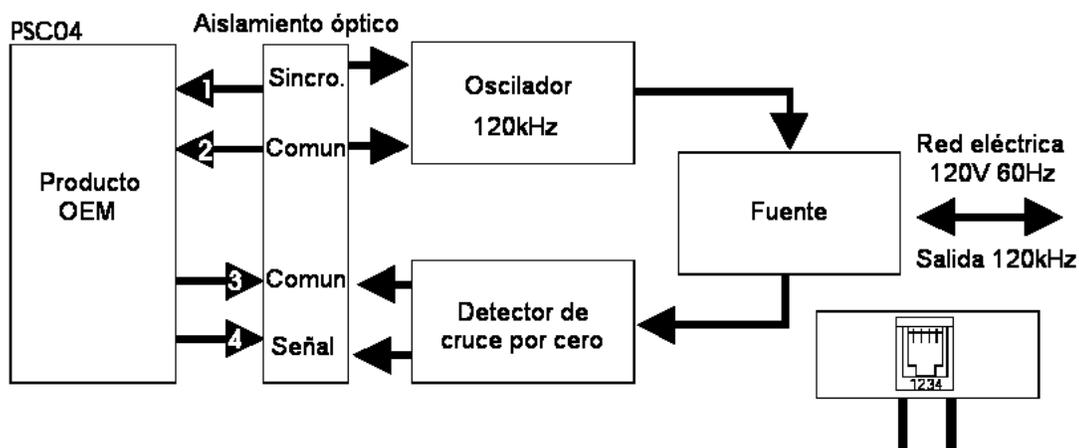


Figura 2.2. Diagrama de bloques del módulo PSC04

En el diagrama se puede observar que el Módulo PSC04 consta de tres submódulos (más el aislamiento óptico para la entrada que controla el oscilador de

120kHz y la salida los cruces por cero). Los módulos son: Fuente de poder, Oscilador de 120kHz y Circuito de detección de cruce por cero.

### 2.1.1.1 Fuente de poder

Esta fuente de poder toma la energía directamente de la red pero no utiliza ningún transformador para reducir el voltaje de línea. Para la reducción de voltaje utiliza un divisor de voltaje construido con capacitores y diodos zener. Tiene una resistencia de bajo valor en serie con los capacitores, la cual funciona como limitadora de corriente. El circuito de la fuente de poder del módulo PSC04 se lo puede observar en la Figura 2.3.

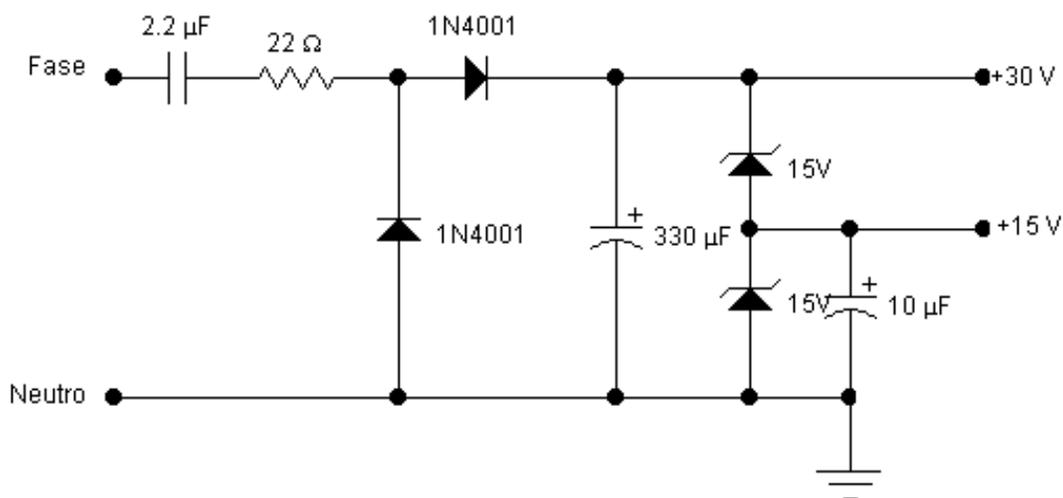


Figura 2.3. Fuente de poder del módulo PSC04<sup>5</sup>

El divisor de voltaje de capacitores se lo calcula tomando en cuenta las impedancias de los capacitores a 60Hz. La capacitancia se la calculó en base a la cantidad de energía que se necesita almacenar. La rectificación se la realiza por medio de un diodo conectado entre los capacitores. Éste módulo contiene una fuente de 30V y una de 15V. Para garantizar que el capacitor se cargue hasta los 30V se utilizan dos diodos Zener de 15V conectados en serie, y es necesario que el divisor de voltaje se calcule para un valor mayor que 30V para que la regulación de los zener sea mejor. Para la fuente de 15V se conecta un capacitor en paralelo

<sup>5</sup> <http://www.x10pro.com/pro/pdf/technote.pdf>

a uno de los zener de 15V (en paralelo al zener que esta conectado a tierra), el cual sirve simplemente para almacenar energía.

### 2.1.1.2 Oscilador de 120kHz

Éste submódulo esta construido en base a transformadores regulables y capacitores sintonizados para oscilar a una frecuencia de 120kHz. Esta señal permanece aislada (oscilando continuamente) hasta que se recibe un uno lógico desde el controlador externo, en este caso el Módulo de Control. Cuando se recibe el uno lógico desde el Módulo de Control, ésta señal es utilizada para alimentar un arreglo de transistores que permiten el paso de la salida de la señal alterna de 120kHz a la red por medio de acoplamiento capacitivo. Éste submódulo también contiene un circuito para manejar el led de indicación. El circuito del Oscilador de 10kHz se lo puede observar en la Figura 2.4.

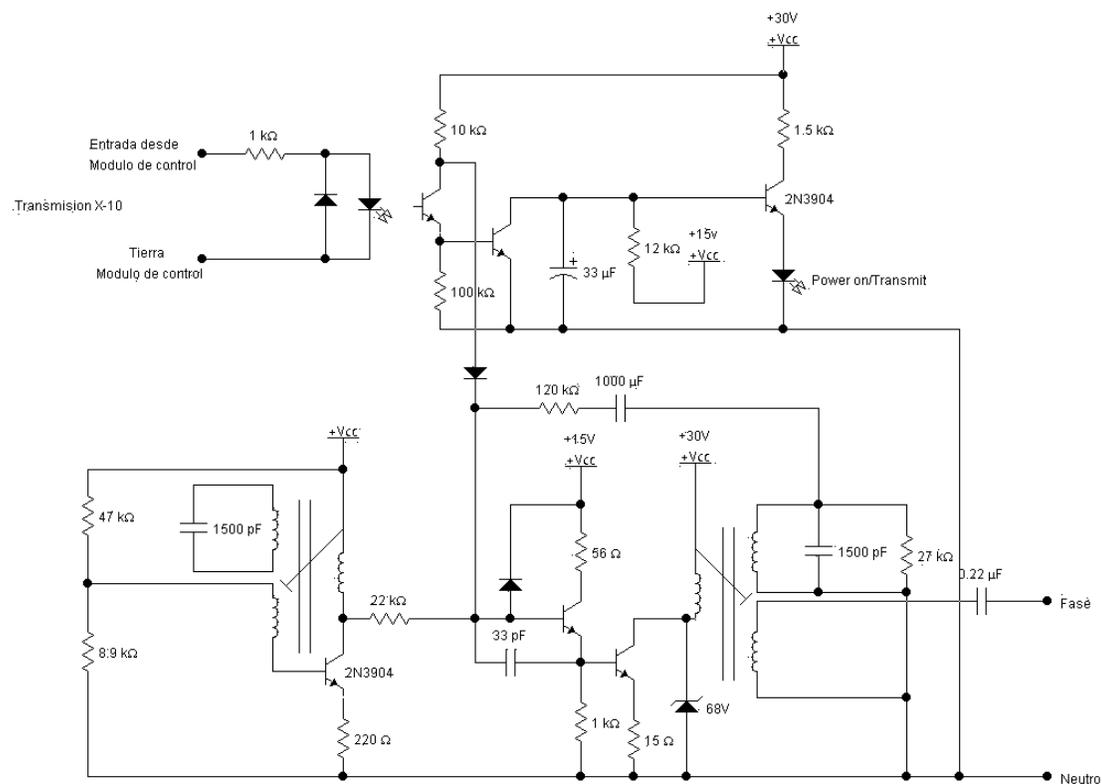


Figura 2.4. Oscilador de 120kHz del módulo PSC04<sup>6</sup>

<sup>6</sup> <http://www.x10pro.com/pro/pdf/technote.pdf>

El led se enciende si el módulo está encendido y la señal de control es cero lógico, y se apaga cuando la señal de control es un uno lógico o cuando se desconecta el módulo. Esto se hace tomando la señal de habilitación del emisor del opto transistor. Ésta señal se la lleva a la base de otro transistor, el transistor de habilitación, el cual está activado cuando existe un uno lógico y se apaga con un cero lógico de la señal de control externa. El último transistor de este arreglo tiene el led conectado al emisor, la fuente de 30V al colector y a su base están conectados el colector del transistor anterior junto con la fuente de 15V. De esta manera, cuando el transistor de habilitación esta apagado provoca que la fuente de 15V encienda al transistor de visualización y se encienda el led. Cuando el transistor de habilitación está encendido, la corriente fluye por éste provocando una caída de tensión sobre la resistencia por lo que el voltaje a la base del transistor de visualización es el voltaje colector-emisor del transistor de habilitación, lo cual provoca que el led de visualización se apague.

### 2.1.1.3 Circuito de detección de cruce por cero

Este circuito toma la señal directamente de la red y la lleva a la base de un transistor a través de una resistencia limitadora de corriente como se puede observar en la Figura 2.5.

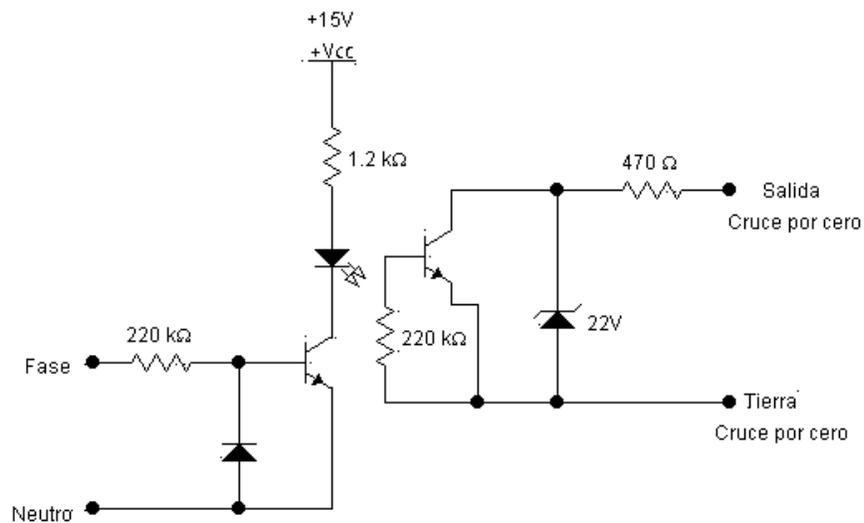


Figura 2.5. Circuito de detección de cruce por cero del módulo PSC04<sup>7</sup>.

<sup>7</sup> <http://www.x10pro.com/pro/pdf/technote.pdf>

El emisor de este transistor está conectado a tierra mientras que el colector está conectado a la fuente de 15V a través de un led (acoplado a un foto transistor) y su resistencia limitadora de corriente. De esta manera, durante el semiciclo positivo se obtiene un uno lógico mientras que durante el semiciclo negativo la señal de la red es puesta a tierra por medio de un diodo, provocando una salida de cero lógico. El resultado es una señal cuadrada que se la puede observar en la Figura 2.6.

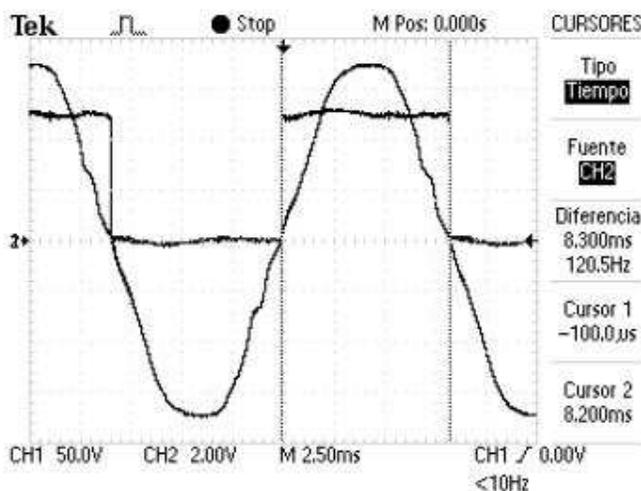


Figura 2.6. Salida del circuito detector de cruce por cero

Es recomendable que cualquier controlador X-10 que vaya a ser utilizado con el módulo PSC04 esté aislado de la red eléctrica y para ello el módulo provee aislamiento óptico en las entradas y salidas. Este aislamiento permite evitar que los elementos capacitivos e inductivos provoquen picos de voltaje o corriente que afecten al controlador. Este aislamiento también evita que las altas frecuencias generadas causen ruido que se retroalimente hacia el controlador y pueda causar su mal funcionamiento. Otra gran ventaja de este módulo es que con su compra también se reciben los permisos legales para desarrollar hardware o software para aplicaciones X-10, ya que, como se mencionó anteriormente, el protocolo X-10 PRO está patentado y ampliamente difundido en otras partes del mundo. Esto evita muchas complicaciones legales en caso de querer comercializar los

dispositivos diseñados y libera al diseñador de cualquier consideración de UL, CSA o FCC<sup>8</sup>.

---

<sup>8</sup> X-10 Home Automation

## **2.2. MÓDULO DE CONTROL**

El Módulo de Control es el encargado de manejar al Módulo PSC04 (modulador X-10) y de realizar la comunicación con la PC. El Módulo de Control debe ser capaz de codificar comandos X-10 almacenados en su memoria o comandos recibidos vía comunicación serial desde la PC e inyectarlos en la red a través del módulo PSC04. Éste Módulo de Control debe tener la capacidad de establecer comunicación bidireccional con la PC. También debe ejecutar comandos en tiempo real o guardarlos en memoria no volátil para su ejecución futura. Se debe implementar un reloj para ejecutar los comandos en la hora y fecha programadas. Para la realización de todas estas tareas se selecciona el microcontrolador PIC que más se ajuste a los requerimientos del proceso, ya que este tipo de microcontroladores son de fácil uso y se pueden conseguir fácilmente en el mercado.

### **2.2.1 SELECCIÓN DEL MICROCONTROLADOR**

Para seleccionar el microcontrolador más apto para el trabajo que debe realizar el módulo controlador se tomaron en cuenta los siguientes aspectos:

1. Los requerimientos de procesamiento aritmético y lógico de datos no son muy elevados así que basta con un procesador de 8 bits.
2. Debido a que se requiere almacenar datos, es necesario que el microcontrolador tenga memoria no volátil de tipo EEPROM.
3. Se requieren al menos dos timers debido a que se debe implementar un reloj en tiempo real y cumplir las especificaciones de tiempo del protocolo X-10.
4. Debe estar en capacidad de atender interrupciones externas, preferentemente por cambio de estado, para hacer que la rutina de sincronización con el cruce por cero sea más eficiente (debido a las características del módulo PSC04),.
5. Es preferible que el microcontrolador tenga en su hardware incorporada la capacidad de comunicación serial mediante protocolo RS 232.

En base a las consideraciones anteriores, el microcontrolador que más se ajusta a éstos requerimientos es el PIC 16F688 porque tiene un procesador de 8 bits con 256 bytes de memoria EEPROM; tiene dos timers, uno de 8 bits y otro de 16 bits; tiene interrupciones por cambio de estado e interfaz USART.

Para la construcción del prototipo se seleccionó el PIC 16F877A debido a que cumple con los requerimientos mínimos y adicionalmente tiene 5 puertos, los cuales pueden ser usados para colocar LEDs de visualización y para el seguimiento de los diferentes procesos. Estos LEDs permitirán depurar más fácilmente el programa del microcontrolador ya que facilitarán el seguimiento de los diferentes procesos y eventos involucrados.

### **2.2.2 DETECCIÓN DE CRUCE POR CERO**

Debido a que el módulo PSC04 tiene un circuito de detección de cruce por cero incorporado, se debe construir un pequeño circuito que permita utilizar esta señal.

Para adquirir la señal de cruce por cero se utilizará la interrupción por cambio de estado del puerto B, debido a que la señal entregada por el módulo PSC04 es cuadrada con un uno lógico durante el semiciclo positivo y un cero lógico durante el semiciclo negativo. De esta manera se genera una interrupción que permite sincronizar la escritura de señales X-10 tanto al semiciclo positivo como al negativo.

El circuito propuesto por el fabricante del módulo PSC04 se lo puede encontrar en la Figura 2.7.

Para la implementación del circuito se reemplazó la resistencia de  $5k\Omega$  por una de  $5,6 k\Omega$ . Los dos diodos son necesarios para proteger los pines del microcontrolador frente a voltajes negativos o superiores a 5 voltios.

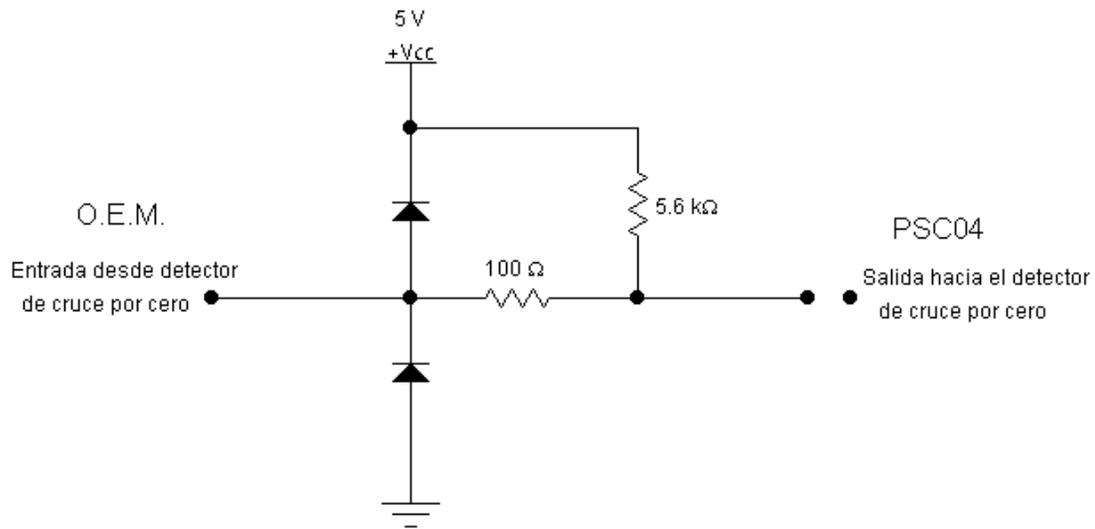


Figura 2.7. Circuito para adquirir la señal de cruce por cero del módulo PSC04

La resistencia de  $100\Omega$  sirve para limitar la corriente que pueda surgir debido a voltajes negativos o superiores a  $5\text{ V}$  (provenientes de alguna falla del Módulo PSC04) para proteger los diodos. La resistencia de  $5,6\text{k}\Omega$  permite tener un 1 lógico en el canal de entrada del microcontrolador cuando el opto transistor de salida del módulo PSC04 se encuentra abierto, y tener un 0 lógico en el canal de entrada del microcontrolador cuando el opto transistor de salida del módulo PSC04 se encuentra cerrado.

### 2.2.3 MODULADOR DE SEÑALES X-10

Para que el módulo PSC04 inyecte una señal alterna de  $120\text{kHz}$  en la red es necesario que el Módulo de Control le proporcione una señal de  $4$  a  $20\text{V}$  de voltaje continuo a su entrada de transmisión. El fabricante provee un circuito sugerido para realizar esta tarea, el cual permite realizar una amplificación de corriente de la señal de transmisión del O.E.M. para que la entrada del módulo PSC04 funcione apropiadamente. Como la capacidad de corriente del microcontrolador que se está usando es de  $25\text{ mA}$  por canal, no es necesario amplificar esta corriente porque la corriente máxima que puede demandar el módulo PSC04 es de  $18\text{mA}$  (a  $20\text{V}$ ). Por esta razón, el circuito que se utilizó para enviar la señal de transmisión es el de la Figura 2.8.

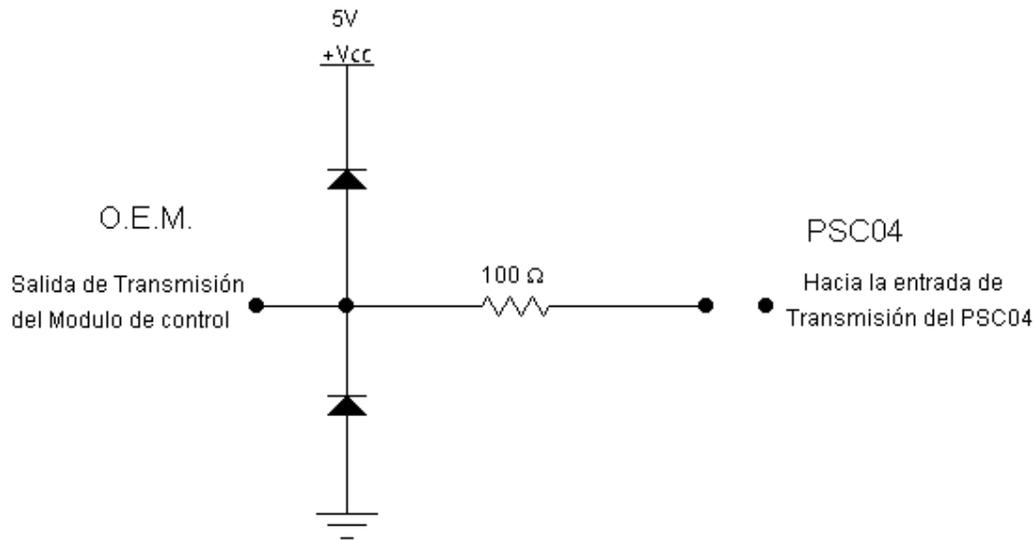


Figura 2.8. Circuito para enviar la señal de transmisión hacia el módulo PSC04

En este circuito los diodos sirven como protección contra voltajes negativos o superiores a 5V. La señal viaja del microcontrolador hacia la entrada de transmisión del módulo PSC 04 a través de una resistencia de 100Ω, la cual está presente para limitar cualquier posible corriente de falla que provenga desde el módulo PSC04 y así proteger a los diodos y al dispositivo OEM.

#### 2.2.4 COMUNICACIÓN SERIAL CON LA PC.

Debido a las diferencias entre los niveles de voltaje del microcontrolador y la PC, se necesita de un convertidor de niveles de voltaje TTL a RS232 y viceversa. El dispositivo que realiza esta conversión es el MAX232. Éste circuito integrado permite doblar su voltaje de alimentación para poder obtener los niveles de voltaje requeridos para la transmisión RS232. Al ser alimentado con 5 V (TTL) este doblador de voltaje permite obtener 10V y gracias a un inversor de voltaje también se pueden obtener -10V, obteniéndose así niveles de voltaje compatibles con el protocolo RS232.

La transmisión de datos se la realiza utilizando solamente tres cables; es decir, transmisión, recepción y tierra. Las conexión entre la PC y MAX 232 se la realiza

con un cable multipar en cuyos extremos tiene instalados dos conectores DB9 macho. La computadora posee un conector DB9 hembra cuyas características físicas<sup>9</sup> y su distribución de pines se puede observar en la Figura 2.9

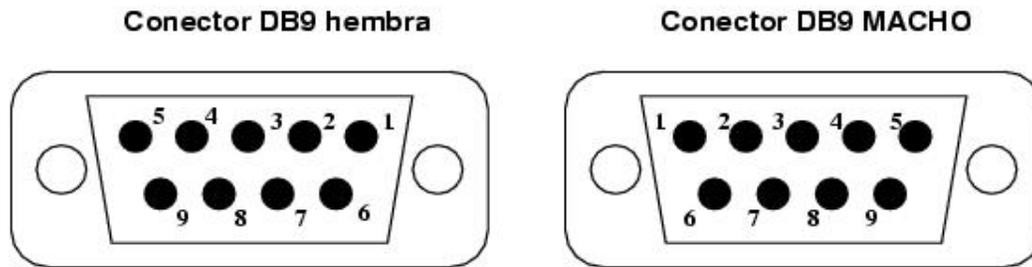


Figura 2.9 Distribución de pines de los Conectores DB9 macho y hembra<sup>10</sup>

La distribución de pines de acuerdo con su propósito del conector DB9 del puerto serial de una computadora se encuentra en la Tabla 2.1

Tabla 2.1 Distribución y propósito de los pines del puerto serial de una PC.

Pin	Propósito	Nombre de la señal
1	Data carrier detect	DCD
2	Received data	RxDData
3	Transmitted data	TxDData
4	Data terminal ready	DTR
5	Signal ground	Gnd
6	Data set ready	DSR
7	Request to send	RTS
8	Clear to send	CTS
9	Ring indicator	RI

#### 2.2.4.1 Circuito integrado MAX232

Este circuito integrado pertenece a una familia de manejadores/receptores hechos con el propósito de cumplir con los requerimientos de las interfases de

<sup>9</sup> <http://www.computerhope.com/help/serial.htm>

<sup>10</sup> <http://www.learobotics.com/proyectos/cuadernos/ct1/ct1.html>

comunicación EIA/TIA-232E y V.28/V.24<sup>11</sup>, especialmente donde no se dispone de voltajes de  $\pm 12$  voltios. Para ello requiere de una alimentación de 5V, aunque el circuito admite voltajes de alimentación de entre 3 a 5 voltios. Este circuito integrado requiere 5 capacitores externos de  $1\mu\text{F}$ . La conexión de los capacitores se muestra en la Figura 2.10.

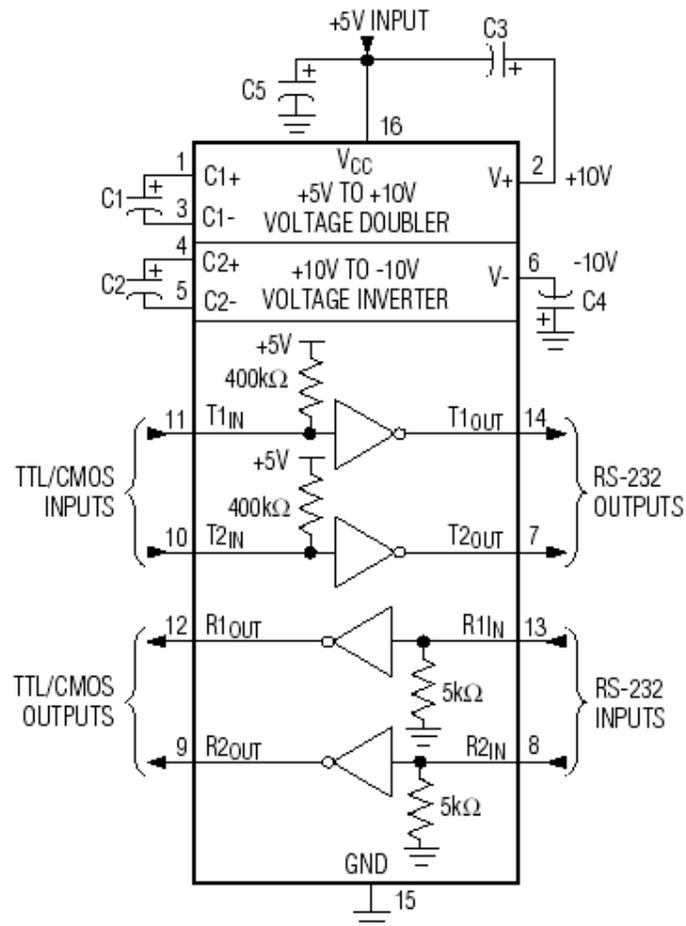


Figura 2.10 Conexiones necesarias para el funcionamiento del Max232

Para la transmisión de datos del microcontrolador a la computadora se conectará el canal del PIC16F877A correspondiente a la transmisión RS232 (puerto C, 6) con el canal del Max232 correspondiente a la entrada TTL. Luego el canal del Max 232 correspondiente a la salida RS232 con la entrada de recepción de la computadora, pin2 del conector DB9. Para la transmisión de datos de la

<sup>11</sup> Data Sheet MAX220 – MAX249

computadora al microcontrolador se debe conectar el pin3 del conector DB9 del puerto serial de la computadora con el canal del Max232 correspondiente a una entrada RS232. Finalmente se debe conectar el canal del Max232 correspondiente a la salida TTL al canal del PIC16F877A correspondiente a la recepción del puerto serial.

### **2.2.5 DISEÑO DE LA FUENTE DE ENERGÍA**

La fuente de energía para el Módulo de Control debe ser capaz de entregar un voltaje constante, con el menor rizado posible y, adicionalmente, es necesario que tenga la capacidad de entregar la corriente necesaria para el funcionamiento de todos los circuitos del Módulo de Control. La familia de microcontroladores seleccionada requiere de un voltaje de entre 2,0 a 5,5V para funcionar apropiadamente. Debido a que se va a realizar comunicación serial con una computadora se eligió trabajar con un voltaje de alimentación de 5V para permitir que el circuito integrado MAX232 funcione adecuadamente. El circuito elegido como fuente de alimentación es un rectificador de onda completa que tiene a su entrada un voltaje alterno de 12V. Éste voltaje de 12V se lo obtiene mediante un transformador reductor de 110V a 12V. Luego de que la corriente alterna es rectificadas se utiliza un filtro capacitivo para tener una corriente continua mucho mas pura. Finalmente para garantizar que el voltaje de alimentación sea siempre de 5V se utiliza un circuito integrado regulador de voltaje NTE960 o su equivalente genérico el CI 7805. Éste circuito integrado requiere como entrada un voltaje de 7V a 20V y entrega hasta 1A de corriente (incluso más si se utiliza disipadores de calor). Como la corriente máxima que consume el microcontrolador seleccionado es de 0.2A, éste circuito integrado es el apropiado para esta aplicación.

#### **2.2.5.1 Diseño del circuito de respaldo de energía**

Debido a que el Módulo de Control utiliza un reloj para ejecutar las tareas programadas en su memoria EEPROM, es necesario que dicho reloj siempre este igualado. Por este motivo es necesario garantizar un suministro constante de

energía para que el reloj interno siga funcionando si hay algún apagón o si se desea reubicar dicho dispositivo. Para permitir que el Módulo de Control siga trabajando apropiadamente, luego de cualquiera de estos sucesos, se diseñó un circuito de respaldo de energía que consiste en tres pilas de 1,5 V conectadas en serie para obtener un voltaje de 4,5 V. Para lograr que la energía de las baterías no se consuma cuando no sea necesario se las colocará en serie con un diodo rectificador y todo este circuito se conectará en paralelo con la fuente principal de energía como se puede observar en la Figura 2.11. De esta manera, cuando la fuente de energía principal esta funcionando se tiene 5 V mientras que en el circuito de las pilas se tiene 4,5V lo que provoca que el diodo se polarice inversamente y no permita la circulación de corriente de las pilas. Si la fuente de energía es desconectada el diodo se polariza directamente y permite el paso de corriente desde las pilas hasta los circuitos del Módulo de Control, evitando que se pierda la cuenta del reloj. Con este tipo de conmutación entre las fuentes energía se logra evitar que el voltaje de alimentación caiga a valores menores que 4,5V y así se garantiza el funcionamiento continuo y confiable del Módulo de Control.

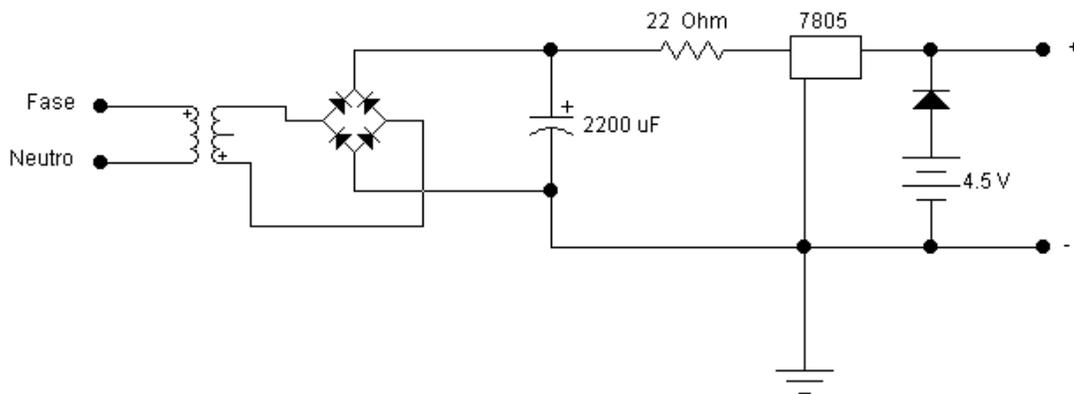


Figura 2.11 Circuito de la fuente de energía

## 2.2.6 CIRCUITO COMPLETO

El circuito del Módulo de Control se encuentra en la Figura 2.12. En dicha figura se puede observar como se conectan las tres partes del módulo con el microcontrolador.

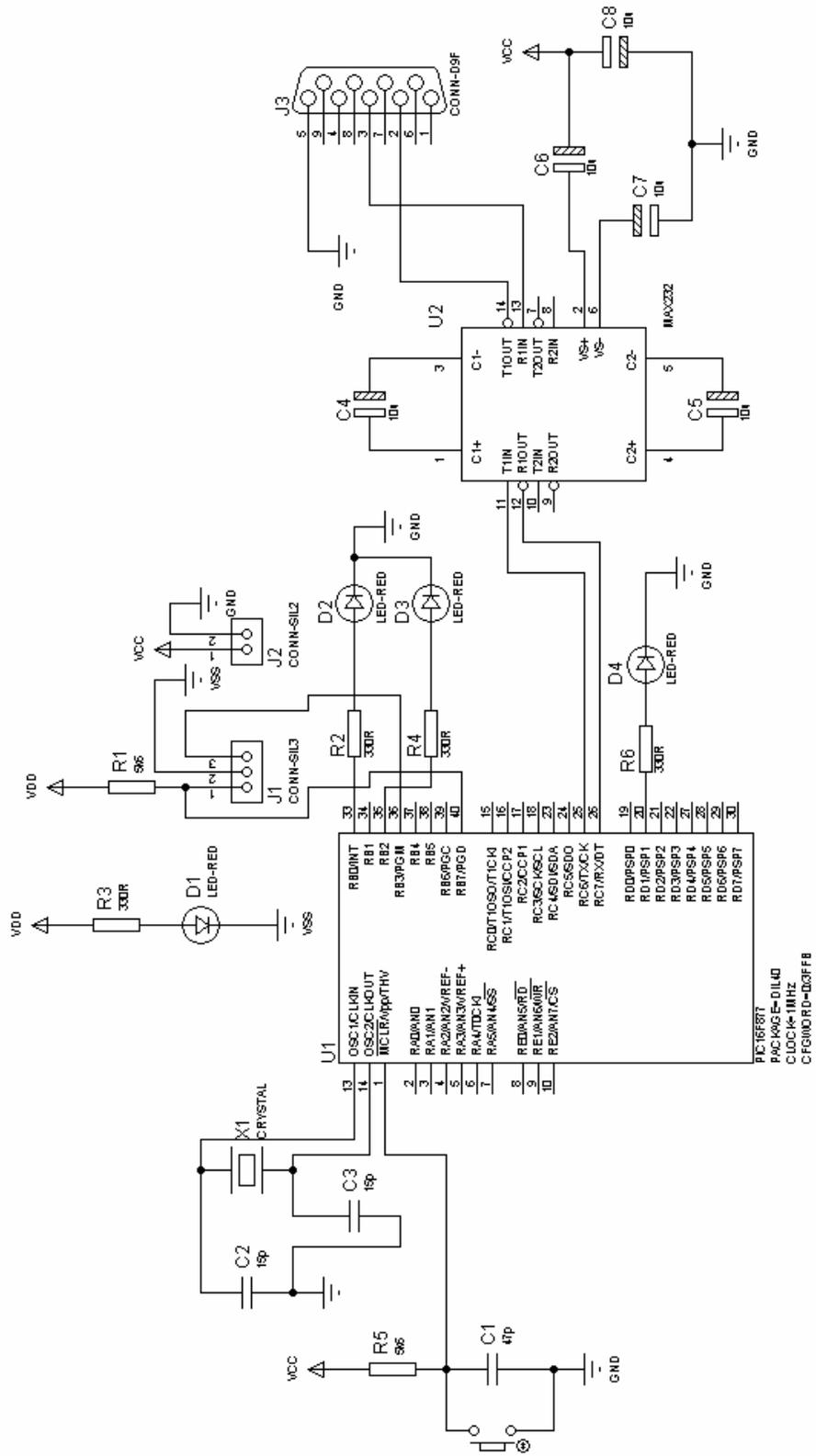


Figura 2.12. Circuito completo del Módulo de Control.

## **2.3. DISEÑO DEL MÓDULO RECEPTOR**

El módulo receptor es el encargado de extraer las señales X-10 de la red eléctrica y de realizar las acciones de control correspondientes. Para ello el módulo receptor utiliza un filtro pasa altos capaz de funcionar a 120V. También se necesita un demodulador que convierta las señales alternas de 120kHz en información digital que se pueda procesar mediante un microcontrolador. Para poder leer los comandos X-10 es necesario diseñar un detector de cruce por cero capaz de funcionar a 120V. Finalmente, se requiere de dos fuentes de voltaje, una para alimentar al microcontrolador y otra para alimentar los circuitos de demodulación.

### **2.3.1 SELECCIÓN DEL MICROCONTROLADOR**

Para seleccionar el microcontrolador adecuado para las tareas del módulo receptor se tomaron en cuenta los siguientes aspectos:

1. Los requerimientos de procesamiento aritmético y lógico de datos no son muy elevados así que basta con un procesador de 8 bits.
2. Se requieren al menos un timer debido a que se necesita cumplir las especificaciones de tiempo del protocolo X-10.
3. Debe estar en capacidad de atender interrupciones externas, preferentemente por cambio de estado para hacer que la rutina de sincronización con el cruce por cero sea más eficiente.
4. Es conveniente que el microcontrolador tenga incorporado un conversor análogo digital para adquirir la señal del demodulador.

En base a las consideraciones anteriores, el microcontrolador que más se ajusta a éstos requerimientos es el PIC 12F675 porque tiene un procesador de 8 bits; tiene dos timers, uno de 8 bits y otro de 16 bits; tiene interrupciones por cambio de estado y un conversor análogo digital.

Para la construcción del prototipo se utilizó el PIC 16F877A debido a que se lo puede conseguir fácilmente en el mercado local. Es fundamental para el proceso de diseño poder reemplazar rápidamente cualquier elemento dañado y el PIC 12F675 no se encuentra disponible en el mercado local. Otra ventaja de usar el microcontrolador 16F877A para la construcción del prototipo es que dispone de más puertos que el 12F675, los cuales sirven para colocar LEDs de visualización para facilitar la detección de fallas en el programa y agilizar el proceso de depuración del programa.

### 2.3.2 DISEÑO DEL FILTRO PASA ALTOS

Éste filtro es el encargado de separar la onda senoidal de 60Hz de la red eléctrica de la onda de 120kHz del código X-10. Debido a que se hace caro construir un filtro activo capaz de funcionar con voltajes de 170V pico, se decidió emplear un filtro pasivo. El filtro está compuesto por un capacitor y una resistencia (Red RC) calculados para detener la baja frecuencia (60Hz en este caso). El esquema del filtro se lo puede observar en la Figura 2.13.

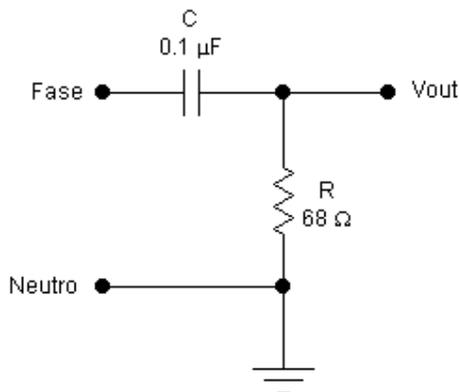


Figura 2.13. Filtro pasa altos pasivo.

La frecuencia de corte para este tipo de filtro pasivo esta dada por la ecuación 2.1

$$f_c = \frac{1}{2\pi\sqrt{RC}} \quad \text{Ecuación 2.1}$$

Se asumió un capacitor de 0,1 μF y de allí se calculó el valor de la resistencia.

$$R = \frac{1}{2\pi \cdot fc \cdot \sqrt{C}} \quad \text{Ecuación 2.2}$$

$$\frac{1}{2\pi \cdot 60 \cdot \sqrt{0,1 \cdot 10^{-6}}} = 70,36\Omega$$

El valor de resistencia más cercano que se puede conseguir es de 68Ω. Para comprobar que el filtro va a funcionar adecuadamente se realizó el calculo de la impedancia del capacitor a las frecuencias de interés: 60Hz y 120kHz

La impedancia de un capacitor se calcula con la Ecuación 2.3

$$X_c = \frac{1}{2\pi f \cdot C} \quad \text{Ecuación 2.3}$$

Para 60Hz la impedancia del capacitor de 0.1μF es:

$$X_{c_{60}} = \frac{1}{2\pi \cdot 60 \cdot 0,1 \cdot 10^{-6}} = 26,5k\Omega$$

Mientras que para una frecuencia de 120kHz la impedancia del mismo capacitor será:

$$X_{c_{120k}} = \frac{1}{2\pi \cdot 120 \cdot 10^3 \cdot 0,1 \cdot 10^{-6}} = 13,26\Omega$$

Como se puede observar en los cálculos anteriores, la impedancia  $X_{c_{60}}$  es 2000 veces mayor que  $X_{c_{120k}}$  lo que garantiza una atenuación mayor a 3dB (1000 veces). Para hacer las pruebas del filtro y poder comprobar su funcionamiento con un osciloscopio es preferible tener una forma de onda periódica. Por esta razón se realizó una modificación al programa del Módulo de Control para permitir que éste envíe un uno lógico X-10 cada vez que exista un cruce por cero de la línea de poder. De esta manera se puede obtener una onda X-10 periódica para poder

realizar el diseño del resto del Hardware. El programa utilizado para enviar un uno lógico periódico es “Envío código X10 para filtro.asm”. Éste programa envía un uno lógico durante un milisegundo cada vez que detecta un cruce por cero en la línea de poder. Las formas de onda obtenidas se encuentran en la Figura 2.14.

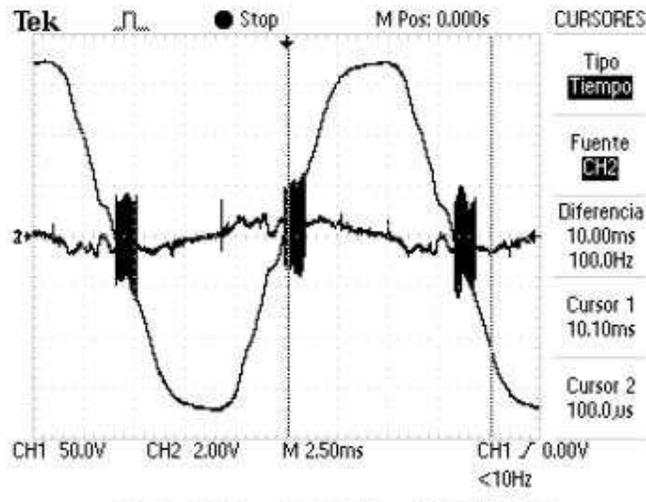


Figura 2.14. Señal de entrada y de salida del filtro RC

Como se puede observar la componente de 60Hz ha sido casi eliminada y su amplitud es mucho menor que la amplitud de la onda de 120kHz. El resto del acondicionamiento de la señal se la realizará con el circuito demodulador.

### 2.3.3 DISEÑO DEL DEMODULADOR

El demodulador es el encargado de convertir las señales X-10 de 120 kHz en una señal digital y acondicionarla para que esté en el rango de voltaje TTL (de cero a cinco voltios). Aunque podría parecer sencillo, fue bastante difícil transformar el pulso de 120kHz en un uno lógico que cumpla con los estándares eléctricos del Protocolo X-10, debido a que la amplitud de la señal X-10 variaba debido al ruido presente en la red eléctrica. Se realizaron varias pruebas y mediciones en la red pero no se logró ni siquiera establecer un patrón respecto al tiempo para estas variaciones. En algunos momentos la señal X-10 llegaba a un voltaje de pico a pico de 5V mientras que en otras ocasiones era de 1,2 V. De todos los circuitos

probados, solo uno funcionó apropiadamente para la segunda condición según los estándares X-10. Para el diseño del circuito demodulador se buscó la mayor simplicidad posible para reducir su tamaño y su costo de fabricación. Primero se trató de realizar la demodulación con elementos pasivos para evitar la construcción de fuentes de voltaje pero al no obtenerse resultados satisfactorios se tuvo que aumentar la complejidad de los circuitos. A continuación se presentan los circuitos que más éxito tuvieron.

### 2.3.3.1 Demodulador X-10 con regulador de voltaje 7805

Este circuito aprovecha la capacidad que tiene el regulador de voltaje 7805 para convertir señales de corriente alterna en corriente continua con un menor rizado. Para esta aplicación no era posible aplicar un filtro capacitivo ya que en las peores condiciones la señal era demasiado débil y desaparecía con todos los diferentes valores de capacitores con los que se realizó las pruebas (entre  $0.22\mu\text{F}$  y  $1000\mu\text{F}$ ). Esto se debe a que la señal es de alta frecuencia y tiene pequeña amplitud y potencia. Al utilizar el regulador de voltaje tomando como entrada el voltaje que cae sobre la resistencia de  $68\Omega$  del filtro pasivo, se logró eliminar gran parte de los semiciclos negativos de la forma de onda así como la componente de 60Hz. El circuito utilizado se muestra en la Figura 2.15.

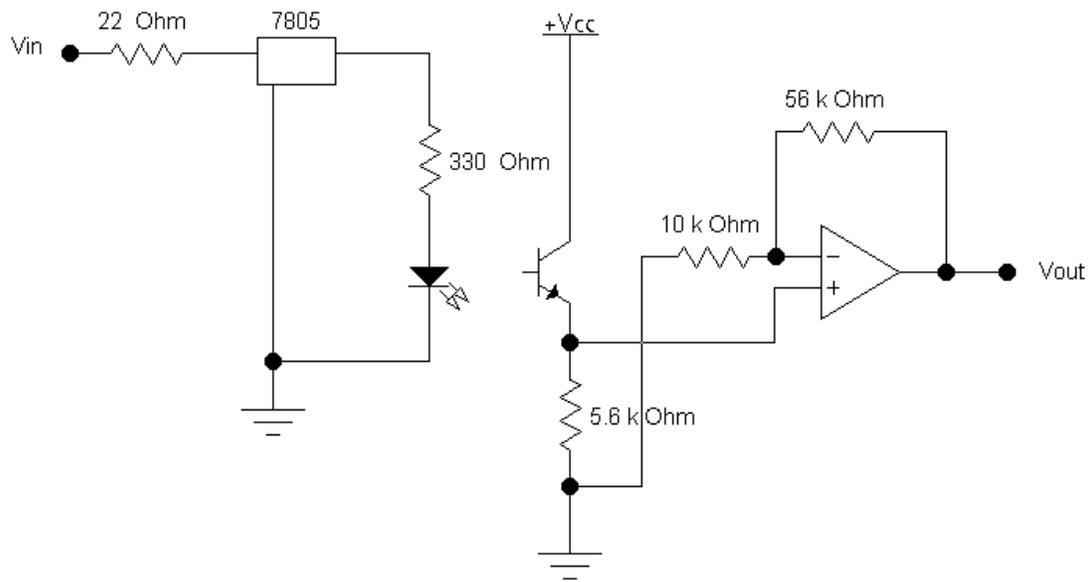


Figura 2.15. Demodulador X-10 con regulador de voltaje 7805

Las formas de onda obtenidas a la salida del regulador de voltaje se pueden observar en la Figura 2.16

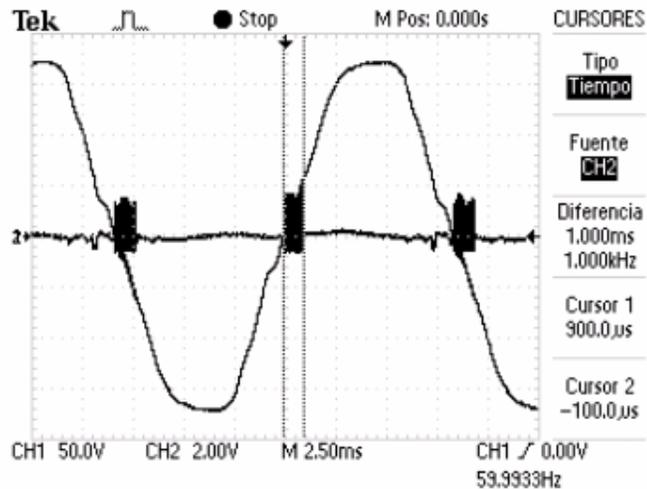


Figura 2.16. Formas de onda a la salida del regulador de voltaje.

Estas formas de onda luego serán llevadas hacia un optotransistor para brindar aislamiento óptico al microcontrolador. Adicionalmente se aprovecha la respuesta lenta de los optotransistores para convertir la señal de 120kHz en una señal de corriente continua con poco rizado. Esto es posible gracias a que el encendido de los optotransistores es muy rápido en comparación a su tiempo de apagado. Cuando se enciende el opto transistor llega rápidamente al pico de conducción pero cuando se apaga lo hace en un tiempo mayor que el tiempo que se tarda en llegar el siguiente pico de conducción. Cada pico de conducción ocurre cuando la señal de 120kHz acondicionada llega a su valor máximo. Al tratarse de una señal de 120kHz, cada pico se produce cada  $8,33 \mu\text{s}$  como se puede observar en la Figura 2.17. El tiempo que tarda en apagarse por completo el opto transistor bajo las condiciones del circuito es de alrededor de  $22 \mu\text{s}$  por lo que a la salida de éste se obtiene una señal con una componente DC de 3,1V durante cada uno lógico. El opto transistor seleccionado para esta aplicación es el 4N25 o NTE 3040.

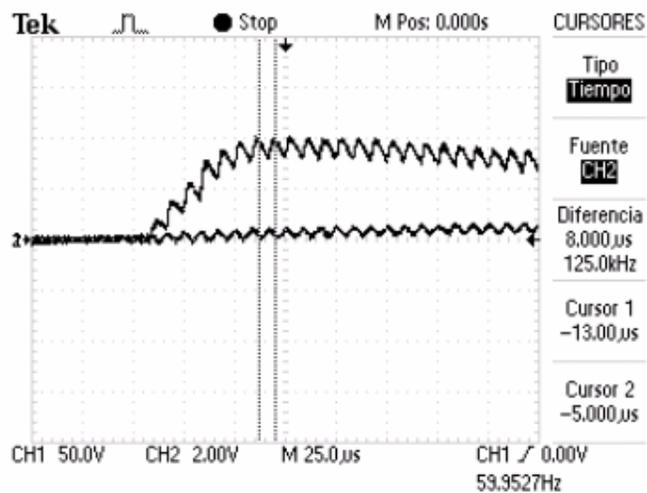


Figura 2.17. Forma de onda a la salida del opto transistor.

Esta propiedad del opto transistor 4N25 será aprovechada en todos los demás tipos de demoduladores para brindar aislamiento óptico y a la vez obtener un pulso de 5 voltios durante el 1 milisegundo que dura el tren de pulsos de 120kHz. La Figura 2.18 muestra como se ve un uno lógico completo.

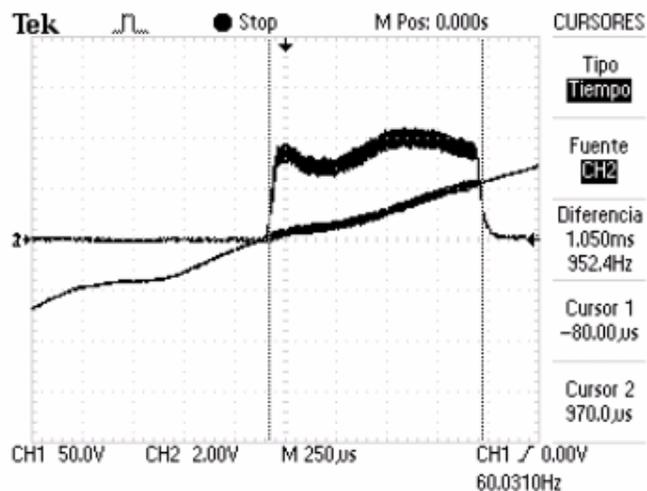


Figura 2.18. Uno lógico X-10 luego de optotransistor.

La Figura 2.15 muestra el uno lógico que viaja sobre el semiciclo positivo de la red. El problema para demodular los unos lógicos X-10 se presenta cuando se tiene que demodular un uno lógico que viaja sobre el semiciclo negativo de la red eléctrica. Éste uno lógico tiene una menor amplitud y cuando las condiciones son

favorables se obtienen formas de onda como las mostradas en la Figura 2.19. pero cuando el voltaje de pico a pico obtenido en el filtro pasivo cae hasta 1,2V se obtienen formas de onda como las de la Figura 2.20.

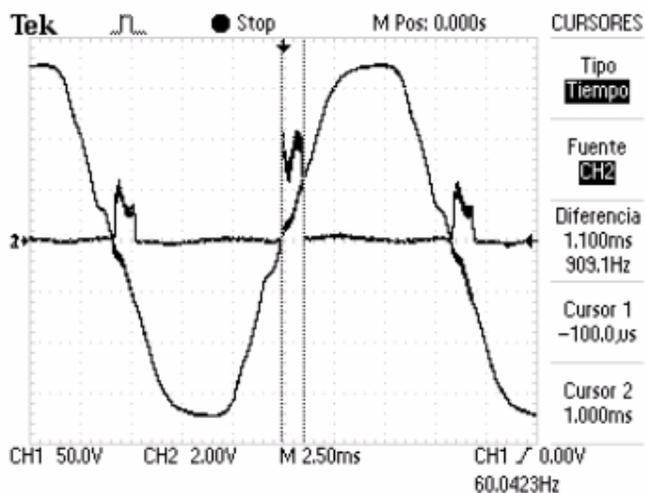


Figura 2.19. Mejor condición de voltaje de salida en el opto transistor.

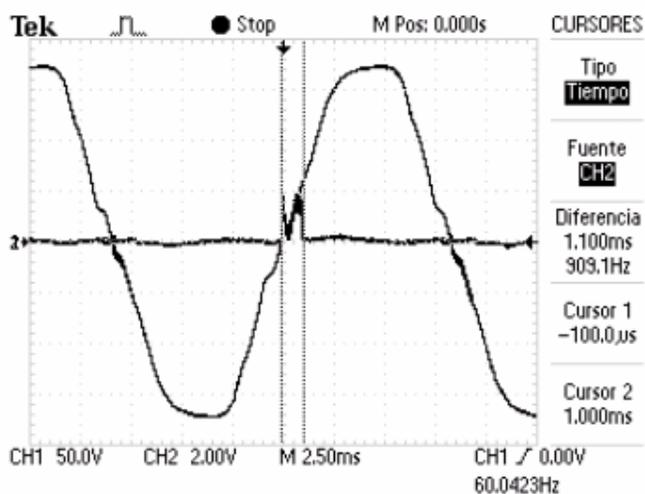


Figura 2.20. Peor condición de voltaje de salida en el opto transistor.

Cuando las formas de onda obtenidas son como en la Figura 2,19 no hay ningún problema con el acondicionamiento final. El amplificador operacional de ganancia 6,6 hace que la menor señal obtenida (0,8 V ) se amplifique hasta provocar la saturación del amplificador. El voltaje al que debería amplificarse la menor señal es de 4,5 V pero la saturación del operacional ocurre a los 4V. La saturación es

deseable ya que permite eliminar la mayor parte del rizado. La forma de onda obtenida luego del amplificador operacional se muestra en la Figura 2.21

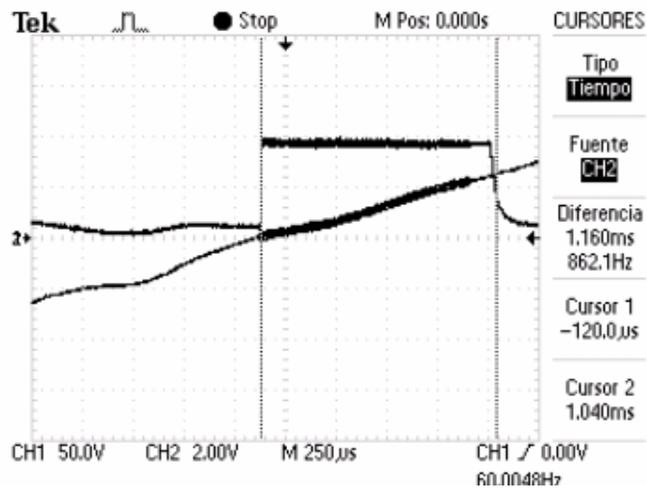


Figura 2.21. Salida del amplificador operacional.

La salida del amplificador operacional produce una onda cuadrada de un poco mas de 1,16 ms de duración mientras que el pulso X-10 dura 1ms. Debido a que el próximo uno lógico vendra luego de 8,33 ms no existe ningún problema a causa de esta prolongación del uno lógico que el circuito demodulador provoca.

El verdadero problema surge cuando se presentan las peores condiciones y se obtienen formas de onda como las de la Figura 2.20. En éste caso es imposible recuperar el uno lógico que viaja en el semiciclo negativo ya que desaparece por completo. Bajo estas condiciones el circuito demodulador no puede realizar su trabajo así que es necesario probar otra alternativa.

### 2.3.3.2 Demodulador X-10 con puente rectificador

Debido a que los voltajes obtenidos con el circuito del numeral anterior no eran estables durante todo el tiempo que dura el puso X-10, se decidió hacer una mejora. Se construyó un circuito rectificador de onda completa que toma el voltaje

de la resistencia del filtro pasivo y tiene como carga una resistencia 10 veces mayor a dicha resistencia, es decir,  $680\Omega$ . El circuito implementado se lo puede observar en la Figura 2.22.

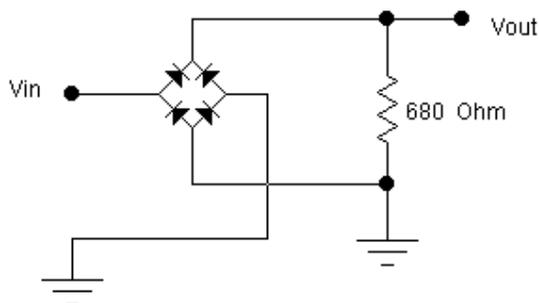


Figura 2.22. Demodulador X-10 con puente rectificador

El resultado de este circuito para buenas condiciones de las señales de voltaje X-10 fue muy satisfactorio como se muestra en la Figura 2.23. Se puede observar que tiene una respuesta mas estable tanto para el semiciclo positivo como para el negativo. Pero, para las peores condiciones, el voltaje del filtro se pierde por completo en el puente rectificador como se observa en la Figura 2.24 así que esta opción tampoco funciona apropiadamente.

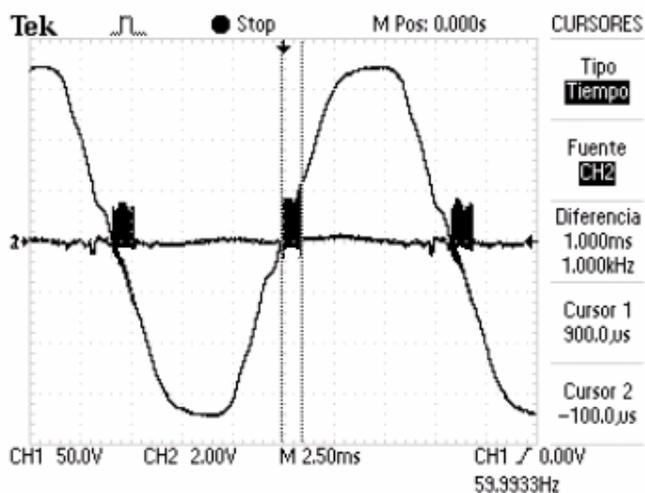


Figura 2.23. Mejor condición de voltaje de salida del puente de diodos

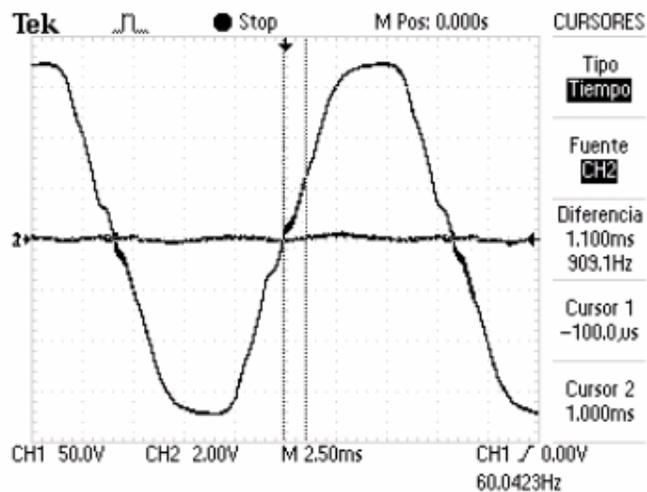


Figura 2.24. Peor condición de voltaje de salida del puente de diodos

### 2.3.3.3 Demodulador X-10 con transistor

Éste circuito utiliza un transistor en configuración de emisor-seguidor para adquirir la señal del filtro pasivo. Se conectó una resistencia diez veces mayor que la resistencia del filtro pasivo a la base del transistor y una resistencia de  $330\Omega$  en el emisor como se muestra en la Figura 2.25. Luego se conectó en serie el fotodiodo del opto transistor.

Experimentalmente se determinó que una corriente apropiada para el funcionamiento del fotodiodo que se está usando debe ser mayor a 10mA. La caída de voltaje en el fotodiodo es de 1,2V y el voltaje de saturación colector-emisor es de máximo 0.4V para la corriente de base que se está utilizando, y el voltaje de alimentación es de 5V. Esto significa que se debe dividir 3,4V para 10mA para obtener el valor de la resistencia adecuada. El resultado de ésta división es  $340\Omega$  así que se selecciona el valor de resistencia estándar más apropiado, en este caso  $330\Omega$ .

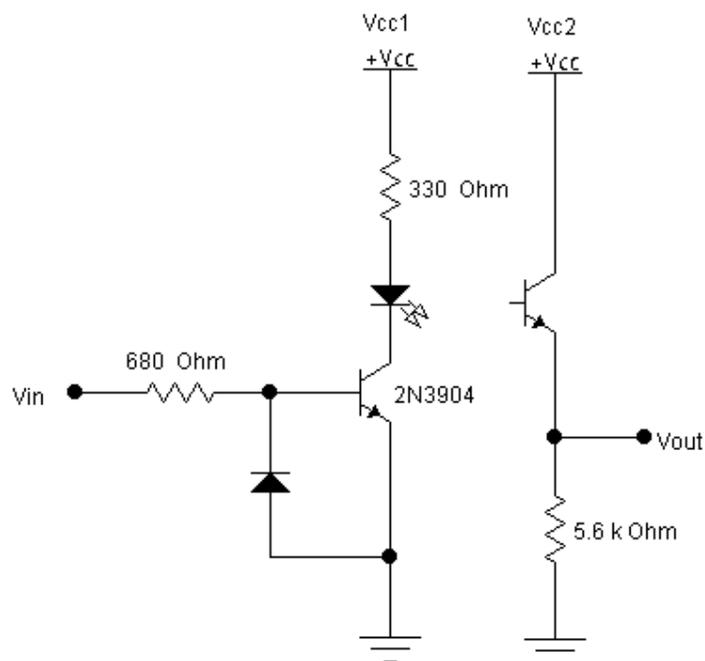


Figura 2.25. Demodulador X-10 con transistor

El diodo conectado a la base del transistor sirve para eliminar los semiciclos negativos. Las formas de onda obtenidas con este circuito se las puede observar en la Figura 2.26 para el caso de las mejores condiciones y en la Figura 2.27 cuando se presentan las peores condiciones.

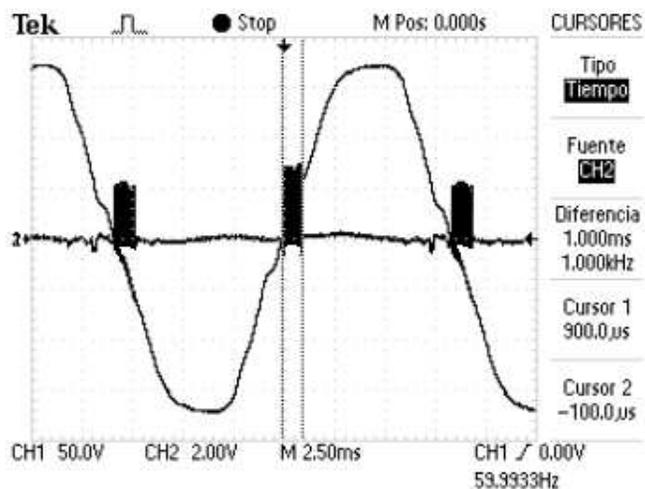


Figura 2.26. Mejor condición de voltaje de salida del transistor emisor-seguidor

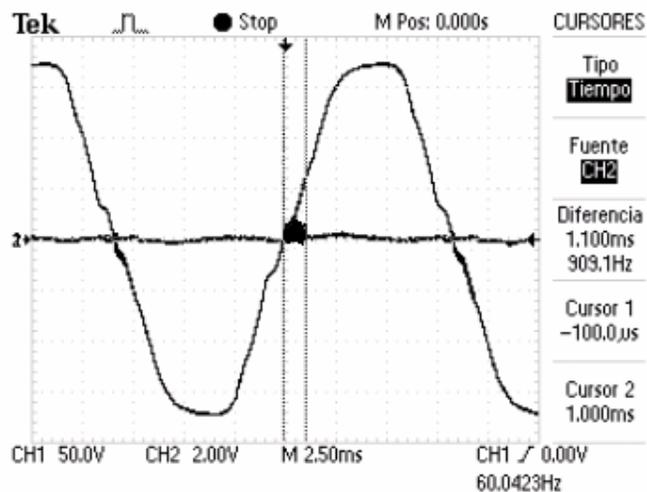


Figura 2.27. Peor condición de voltaje de salida del transistor emisor-seguidor

Como se puede observar en la Figura 2.26, cuando las señales X-10 son de buena calidad, se pueden adquirir tanto los unos lógicos X-10 que viajan en el semiciclo positivo y en el semiciclo negativo, pero cuando la señal X-10 se deteriora se pierde el uno lógico que viaja sobre el semiciclo negativo por lo que este tipo de demodulador tampoco es el apropiado. Se hicieron varias pruebas modificando los valores de las resistencias pero no se lograron mejoras significativas.

#### 2.3.3.4 Demodulador X-10 con opto transistor Darlington

Éste circuito utiliza una resistencia de  $330\ \Omega$  para tomar la señal del filtro pasivo y llevarla al fotodiodo del opto transistor tipo Darlington como se muestra en la Figura 2.28. Se eligió este valor de resistencia ya que es el máximo permitido para garantizar que el fotodiodo se encienda apropiadamente de acuerdo con la aplicación. Se eligió el máximo valor admitido porque con valores de resistencia cercanos a los  $68\ \Omega$  de la resistencia del filtro se afecta su funcionamiento. Se coloca un diodo en inverso paralelo al fotodiodo de aislamiento para eliminar los voltajes negativos. Las formas de onda que se obtuvieron para la mejor señal X-10 y para la peor señal se muestran en la Figura 2.29 y 2.30 respectivamente.

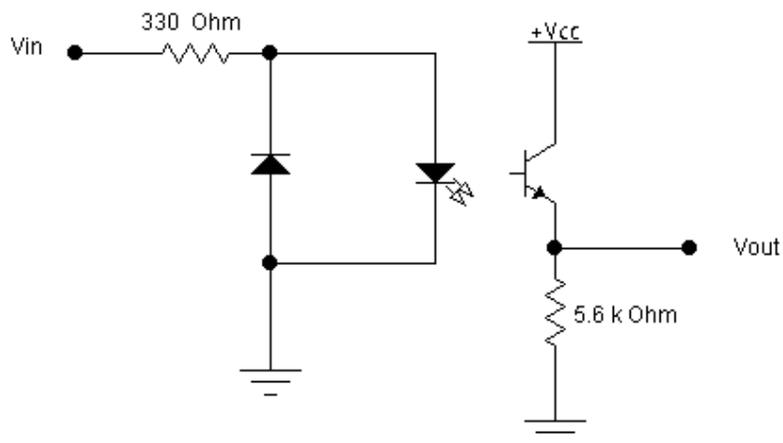


Figura 2.28. Demodulador X-10 con opto transistor Darlington.

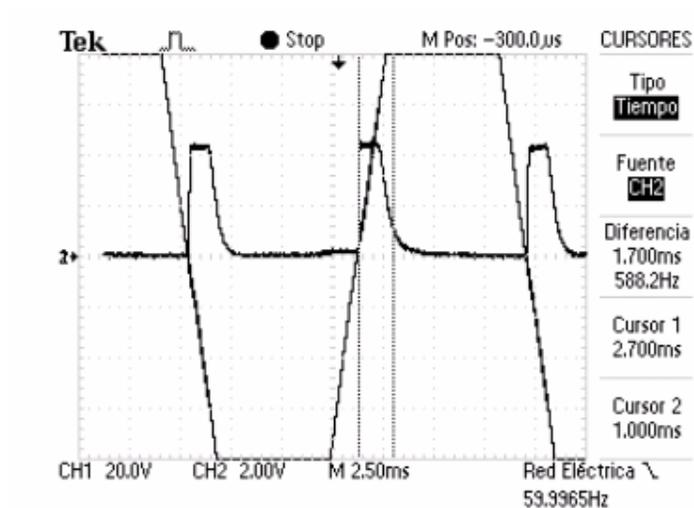


Figura 2.29. Salida del opto transistor con buena calidad de señal X-10

Como se puede observar en la Figura 2.29, el circuito propuesto entrega una señal de muy buena calidad a la salida del opto transistor cuando la calidad de la señal X-10 es buena y conserva parte de la señal X-10 que viaja en el semiciclo negativo cuando la señal X-10 esta deteriorada, como se puede observar en la Figura 2.30.

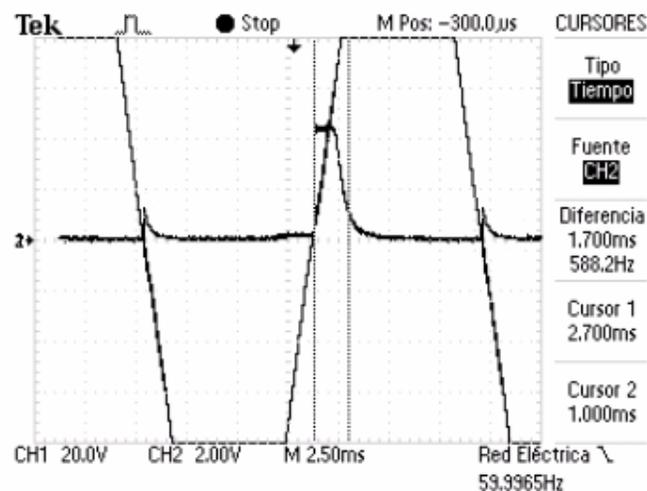


Figura 2.30. Salida del opto transistor con mala calidad de señal X-10

Lastimosamente, para poder utilizar la señal de la Figura 2.30 es necesario amplificarla al menos quince veces. Si se usa un circuito amplificador con esa ganancia se recuperará la señal X-10 para la peor condición pero en cambio se daña la señal X-10 que se obtiene en las mejores condiciones. Por esta razón se descartó al circuito demodulador con opto transistor Darlington.

### 2.3.3.5 Demodulador X-10 con seguidor de voltaje y opto transistor

Luego de que no se tuvo éxito con el diseño de un demodulador que no requiera alimentación de voltaje, y por consiguiente una fuente de voltaje adicional para mantener el aislamiento óptico, se decidió utilizar un amplificador operacional en configuración de seguidor de voltaje para desacoplar la impedancia del filtro y con un diodo conectado en la entrada no inversora del amplificador operacional, como se muestra en la Figura 2.31, para eliminar los voltajes negativos. Si no se coloca dicho diodo, las ondas X-10 se deforman y dejan de ser utilizables. Se eligió el circuito integrado LA6324N porque se lo puede alimentar con una sola fuente de voltaje, de entre 3V y 30V. Debido a que el diodo solo permite la conducción en un sentido se eliminaron los voltajes negativos.

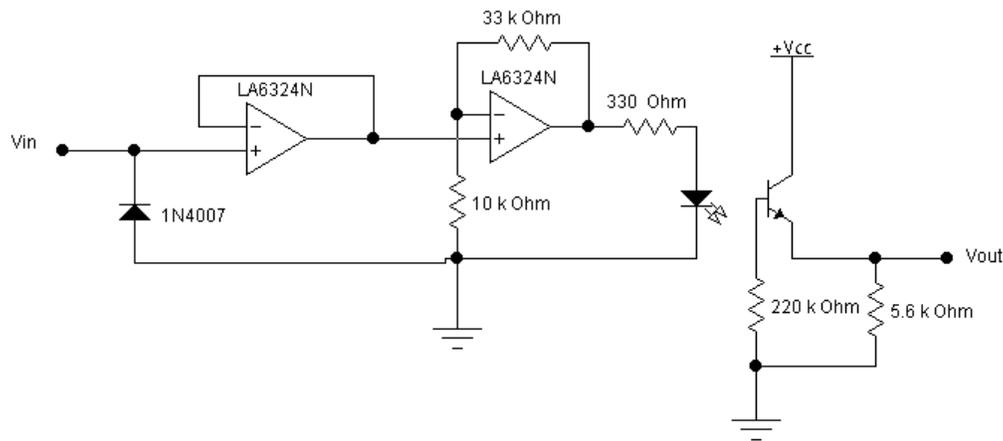


Figura 2.31. Demodulador con seguidor de voltaje y opto transistor.

Las formas de onda a la salida del seguidor de voltaje se muestran en la Figura 2.32

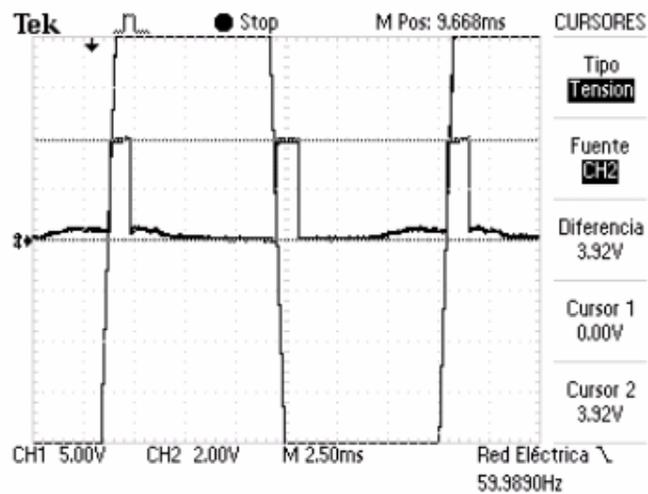


Figura 2.32 Formas de onda a la salida del seguidor de voltaje.

Las formas de onda a la salida del seguidor de voltaje son cuadradas y prácticamente se ha eliminado el ruido. Las formas de onda de 120kHz se transformaron en ondas cuadradas de manera similar a cuando se utilizó un opto transistor.

Para garantizar que la señal siempre sea utilizable por el microcontrolador es necesario amplificarla. Para este propósito se coloca un amplificador no inversor.

La ganancia de dicho amplificador está dada por la Ecuación 2.4

$$G = 1 + \frac{R_f}{R_1} \quad \text{Ecuación 2.4}$$

Se eligió un valor de  $R_1$  igual a  $10\text{k}\Omega$  así que despejando  $R_f$  para realizar su calculo se obtiene la Ecuación 2.5

$$R_f = (G - 1)R_1 \quad \text{Ecuación 2.5}$$

La ganancia adecuada para éste amplificador se la determinó experimentalmente. Se tomó como base una ganancia de 1 y se la fue aumentando hasta que la amplificación del ruido llegó a deformar la onda a tal punto que se perdía la información de la señal X-10. Dicha ganancia límite es de 7,4. Luego se obtuvo el promedio entre la ganancia mas baja y la ganancia mas alta permitidas para encontrar el punto de operación más seguro, tanto para las mejores como para las peores condiciones de la señal X-10. Se obtuvo el promedio entre las dos ganancias debido a que los amplificadores operacionales son circuitos lineales. Por estas razones se utilizó una ganancia de 4,2 para calcular la resistencia  $R_f$ .

Luego, utilizando la Ecuación 2.5 se determinó que el valor de  $R_f$  apropiado era de  $32\text{k}\Omega$ , al cual se lo aproximó a  $33\text{k}\Omega$  ya que es el valor más apropiado de resistencia estándar.

El voltaje de salida del amplificador no inversor presenta una alta cantidad de ruido luego de ser amplificado pero se garantiza que el voltaje pico de las ondas cuadradas correspondientes a los semiciclos positivos y negativos siempre sean de 4V sin importar la calidad de la señal X-10. La señal de salida en el opto transistor se la puede observar en la Figura 2.34

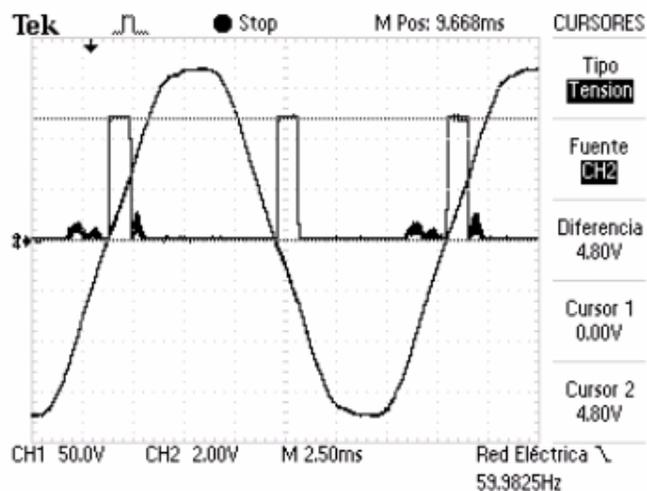


Figura 2.34. Forma de onda de salida en el opto transistor.

La forma de onda que se obtuvo a la salida del opto transistor casi no contiene ruido y el valor del voltaje obtenido para los pulsos X-10 demodulados es de 4,8V a la entrada del canal del microcontrolador.

La gran ventaja de este circuito es que como la adquisición del voltaje se la realiza directamente de la red eléctrica, la señal X-10 que viaja en el semiciclo negativo se la puede recuperar prácticamente con la misma intensidad con la que se recupera a la señal X-10 que viaja en el semiciclo positivo. Por esta razón las formas de onda a la salida del seguidor de voltaje son casi iguales para las mejores y para las peores condiciones de la señal X-10.

El problema con este circuito es que el amplificador operacional seleccionado no responde bien ante la frecuencia de 120kHz. Inicialmente es beneficioso ya que el seguidor de voltaje convierte las señales de entrada en pulsos cuadrados. El problema se presenta debido a los transitorios que se generan en el filtro pasivo cuando se lo conecta a la red eléctrica. Durante el transitorio se aplican 120 voltios a la entrada del amplificador operacional, lo cual termina dañando al circuito integrado. Para evitar esto sería necesario utilizar el primer operacional en configuración de amplificador inversor debido a que se incluye una resistencia en la entrada al circuito integrado, la cual sirve adicionalmente de limitadora de corriente. El problema al hacer esto es que la señal se atenúa y se observan los

misimos problemas que con todos los circuitos anteriores cuando la señal X-10 que viaja en la red se deteriora. Por estos motivos también se desechó este circuito.

### **2.3.3.6 Demodulador X-10 con filtro pasa banda y opto transistor**

Debido a que no fue posible recuperar la señal X-10 con circuitos sencillos, es necesario diseñar un demodulador un poco mas complejo tomando como base el circuito demodulador con seguidor de voltaje ya que es el que mejor resultados ha dado. También es necesario diseñar un filtro pasabanda que elimine el ruido blanco de la red eléctrica, el cual es la causa de que no funcionen adecuadamente ninguno de los circuitos demoduladores diseñados en los numerales anteriores.

Para el diseño de un filtro pasabanda es necesario utilizar un circuito operacional con una buena respuesta a frecuencias como la que utiliza el protocolo X-10. Uno de estos circuitos integrados es el TL084 ya su ancho de banda con ganancia unitaria es típicamente de 3MHz . La desventaja es que éste circuito requiere de dos fuentes de voltaje: una positiva y una negativa. Como no se logró obtener resultados compatibles con el protocolo X-10 con los demoduladores diseñados en los numerales 2.3.3.1 a 2.3.3.5, se decidió utilizar el circuito integrado TL084 para construir un filtro pasabanda que elimine el ruido de la red aunque para ello sea necesario utilizar una fuente adicional. La adquisición de la señal del filtro pasivo se la realizará a través de un amplificador inversor, para evitar que el circuito integrado se dañe durante el transitorio inicial. El circuito diseñado se lo presenta en la Figura 2.33.

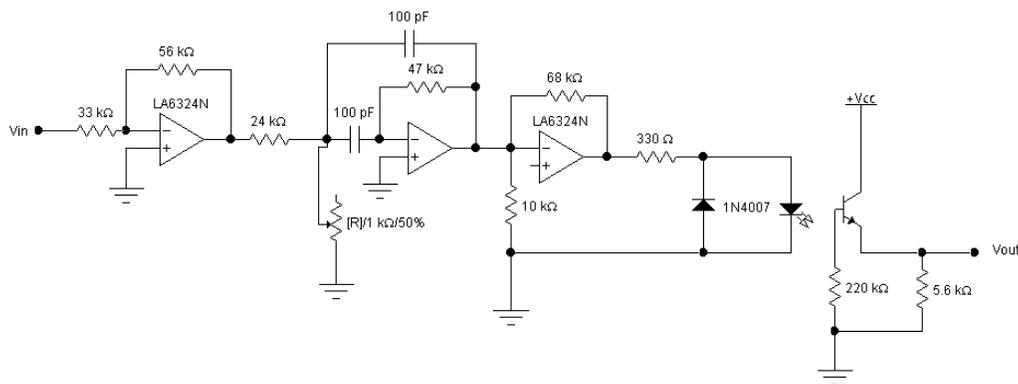


Figura 2.33. Demodulador con filtro pasabanda y opto transistor.

La primera etapa de amplificación permite mejorar la señal de entrada y por la configuración de amplificador inversor se provee al circuito integrado de un limitador de corriente durante el transitorio inicial. El valor de la resistencia de entrada se la calculó para limitar la corriente máxima a 5mA pico durante el transitorio. Se eligió el valor de 5mA ya que la capacidad de disipación máxima de este circuito integrado es de 680mW. Con una corriente de 5mA a 120V se tiene una potencia de 600mW. De esta manera se garantiza que el transitorio durante la carga inicial del capacitor no podrá dañar al circuito integrado. Para calcular el valor de la resistencia que garantice que la mayor corriente pico que pueda pasar por la resistencia sea de 5mA se aplica la Ecuación 2.6.

$$R = \frac{V}{I} \quad \text{Ecuación 2.6.}$$

Donde R representa la resistencia, V el voltaje e I la corriente. Utilizando el valor pico del voltaje de la red, el valor de la resistencia necesaria es:

$$R = \frac{170V}{5mA} = 34k\Omega$$

El valor de resistencia estándar mas apropiado es de 33kΩ. Adicionalmente se busca tener una pequeña ganancia en esta etapa. No se puede tener demasiada ganancia porque los amplificadores podrían entrar en saturación y evitar el

correcto funcionamiento del filtro. Como el voltaje pico de la señal que se obtiene del filtro pasivo es de 2,5V y voltaje de saturación del amplificador operacional es de 4,5V; la ganancia más apropiada se la obtiene dividiendo el voltaje de saturación para el voltaje pico de entrada. El resultado es 1,8. Como la ganancia en un amplificador operacional en configuración de amplificador inversor está dada por la Ecuación 2.7, el valor de la resistencia de alimentación  $R_f$  será el producto de la ganancia por la resistencia  $R$ .

$$G = \frac{R_f}{R} \quad \text{Ecuación 2.7}$$

Donde  $G$  es la ganancia,  $R_f$  es la resistencia de realimentación y  $R$  es la resistencia de entrada.

Realizando el producto de la ganancia por la resistencia de entrada se obtiene un valor de  $59,4\Omega$  para la resistencia de realimentación. Como lo deseable es que no se llegue a saturación es conveniente que la resistencia estándar seleccionada sea la de valor inmediato inferior. Por esta razón se eligió una resistencia de  $56k\Omega$ .

La segunda etapa contiene el filtro pasabanda. Debido a que se busca eliminar la mayor cantidad de ruidos y de frecuencias indeseables se eligió una configuración de filtro pasabanda de banda angosta. El diagrama de éste tipo de filtros se lo muestra en la Figura 2.34.

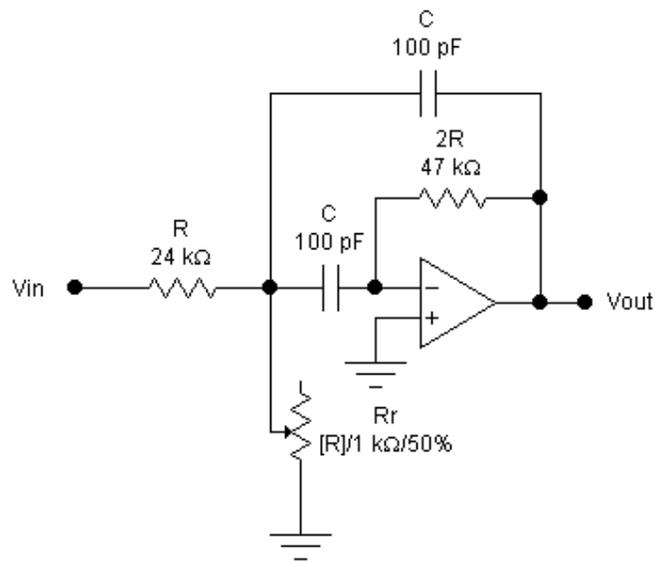


Figura 2.34. Filtro pasabanda de banda angosta.

Éste tipo de filtro permite seleccionar el ancho de banda deseado en base al factor de calidad  $Q$  y a la frecuencia resonante  $F_r$  según la Ecuación 2.6.

$$Q = \frac{F_r}{B} \quad \text{Ecuación 2.6}$$

Para determinar el ancho de banda del filtro es necesario determinar cuales son las frecuencias que más afectan al circuito. La frecuencia que más interfiere en el funcionamiento de la adquisición es la frecuencia de 60Hz de la red eléctrica. También se necesita eliminar el ruido blanco. Mediante algunas mediciones con el osciloscopio se determinó que la mayor frecuencia de ruido blanco es de aproximadamente 145 Mhz. Debido a que las frecuencias identificadas se encuentran bastante lejos de la frecuencia resonante, se decidió utilizar un ancho de banda de 60kHz. Cuando se utiliza éste ancho de banda para el diseño del filtro, se obtienen valores de resistencias y capacitancias fáciles de encontrar en el mercado local y que además permiten el adecuado funcionamiento del amplificador operacional.

Para calcular el factor de calidad para el ancho de banda seleccionado se utiliza la ecuación 2.6 de la siguiente manera:

$$Q = \frac{120\text{kHz}}{60\text{kHz}} = 2$$

Otra ecuación necesaria para el diseño del filtro es la Ecuación 2.7.

$$B = \frac{0,1591}{RC} \quad \text{Ecuación 2.7.}$$

Se asume el valor del capacitor  $C = 100\text{nF}$ , y utilizando la Ecuación 2.7 se puede calcular el valor de la resistencia  $R$  de la siguiente manera:

$$R = \frac{0,1591}{B \cdot C} = \frac{0,1591}{60 \cdot 10^3 \cdot 100 \cdot 10^{-9}} = 26,5\text{k}\Omega$$

El valor de resistencia estándar más apropiado es de  $24\text{k}\Omega$  para a resistencia  $R$ . En consecuencia, la resistencia  $2R$  debe tener un valor lo más cercano posible a  $48\text{k}\Omega$ . El valor más apropiado que se puede encontrar entre los valores estándar es de  $47\text{k}\Omega$ . El último parámetro que falta por calcular es la resistencia  $R_r$ , para lo cual se utiliza la Ecuación 2.8.

$$R_r = \frac{R}{2Q^2 - 1} \quad \text{Ecuación 2.8.}$$

Reemplazando los valores calculados se puede obtener:

$$R_r = \frac{24000}{2 \cdot 2^2 - 1} = 3428\Omega$$

Para obtener la resistencia que mejor seleccione una frecuencia resonante de 120kHz se utiliza un potenciómetro de precisión de 10k $\Omega$ . La Figura 2.35 verifica el funcionamiento del filtro y de la primera etapa de amplificación.

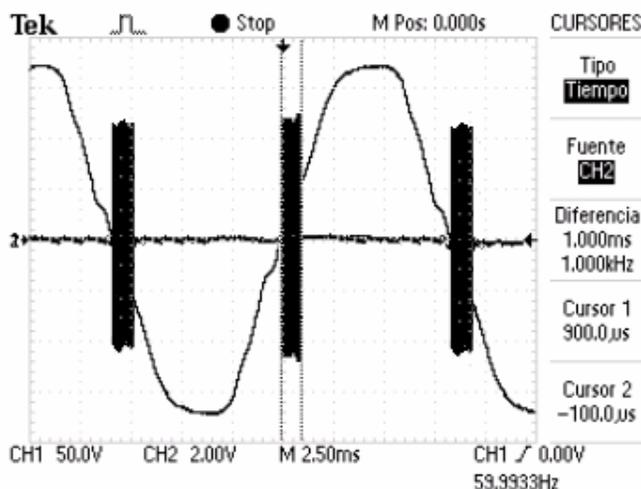


Figura 2.35. Salida del filtro pasabanda de banda angosta.

Como se puede observar en la Figura 2.35, la respuesta del filtro pasabanda de banda angosta junto con la primera etapa de amplificación es muy buena. El ruido tanto de alta como de baja frecuencia ha sido eliminado y la señal X-10 tiene una buena amplitud y calidad. Posteriormente se trató de reducir el ancho de banda del filtro pero también se atenúa la señal X-10 sin obtener resultados significativos. Por esta razón se conservó el filtro diseñado porque es el que mejor resultados produce para la señal X-10.

Para garantizar que la señal siempre sea legible por el microprocesador se amplificará la señal de salida del filtro pasabanda mediante una tercera etapa, utilizando un amplificador operacional en configuración de amplificador no inversor. Ésta última etapa de amplificación es necesaria debido a que la carga que se necesita manejar es un fotodiodo en inverso paralelo con un diodo de señal. Como la corriente que debe pasar por el fotodiodo debe ser mayor a 10mA para llegar a la saturación del fotodiodo (determinado experimentalmente), la carga provoca que el voltaje de salida del amplificador operacional caiga. Al amplificar la señal de salida del filtro pasabanda se obtiene un mayor voltaje y por

lo tanto más corriente a través del fotodiodo, con lo que se garantiza que se va a llegar a la saturación del fototransistor.

La ganancia adecuada para éste amplificador se la determinó experimentalmente ya que no se encontraron las ecuaciones para calcular compensaciones de voltaje para la carga conectada. Se tomó como base una ganancia de 1 y se la fue aumentando hasta que el voltaje de salida dejó de aumentar significativamente, lo cual significa que se llegó al límite de corriente que puede entregar el amplificador operacional. Cuando se obtiene la corriente máxima que puede entregar el amplificador operacional TL084 no se corre el riesgo de dañar el circuito integrado porque está diseñado para soportar cortocircuitos en sus terminales de salida y con la corriente obtenida no se supera el límite de disipación de energía de 680mW. La ganancia necesaria es de 6,9 y la resistencia  $R_f$  es de 68k $\Omega$  para una resistencia  $R$  de 10k $\Omega$ . La salida de voltaje del amplificador no inversor con carga (fotodiodo) se la encuentra en la Figura 2.36

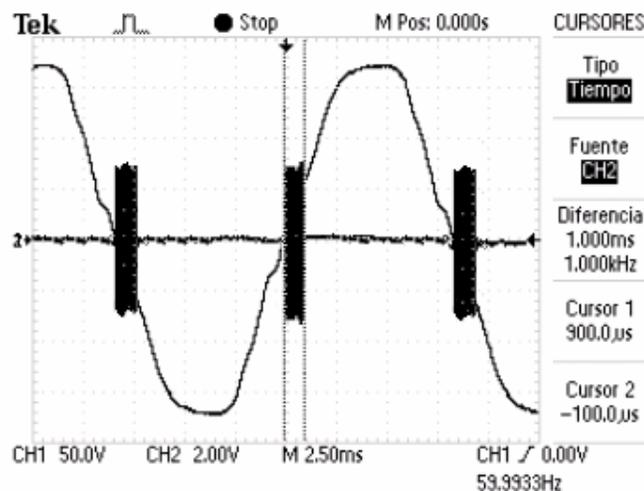


Figura 2.36. Salida del amplificador no inversor

El voltaje de salida del amplificador no inversor presenta una baja cantidad de ruido, prácticamente nula considerando que dicho ruido se ha amplificado casi 7 veces. La corriente que se logró obtener para la carga del fotodiodo con una resistencia limitadora de corriente de 330 $\Omega$  es de 10,9 mA; la cual es suficiente

para llevar al optotransistor a la saturación. La señal de salida en el optotransistor se la puede observar en la Figura 2.37

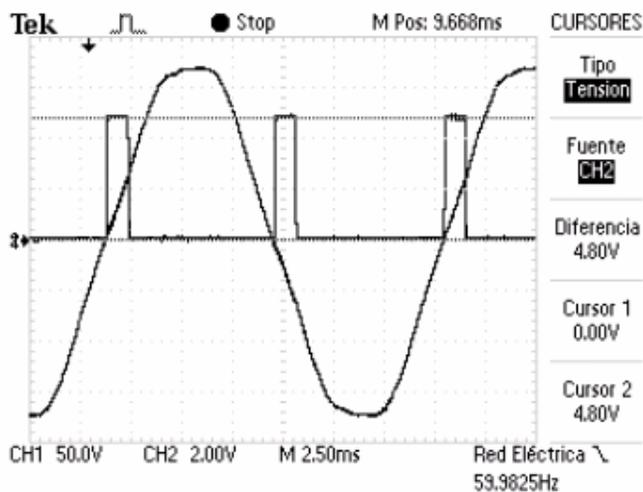


Figura 2.37. Forma de onda de salida en el opto transistor.

La forma de onda que se obtuvo a la salida del opto transistor practicamente no contiene ruido y el valor del voltaje obtenido para los pulsos X-10 demodulados es de 4,8V. El ruido presente en las formas de onda puede ser ignorado fácilmente si la adquisición de datos se la realiza mediante el conversor análogo digital del PIC. Simplemente se establece el valor de uno lógico sobre el nivel de voltaje del ruido. Para asegurar que la calidad del voltaje de la señal X-10 que viaja en el semiciclo negativo sea buena se tomó una forma de onda más detallada, la cual se muestra en la Figura 2.38.

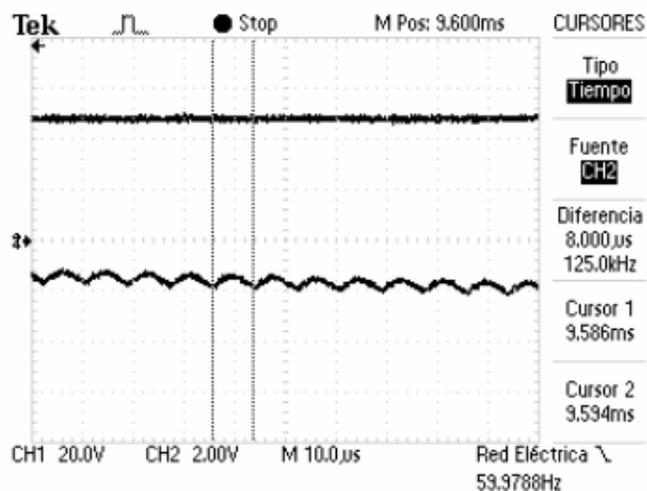


Figura 2.38. Acercamiento la señal demodulada del semiciclo negativo.

En la Figura 2.38 se observa la forma de onda X-10 demodulada en la parte superior y la forma de onda X-10 que viaja sobre la corriente alterna de la red en la parte inferior. Como se puede observar, el demodulador construido ha logrado eliminar el ruido de la señal demodulada para los pulsos correspondientes a un uno lógico. Las últimas comprobaciones que se deben realizar a las formas de onda X-10 demoduladas tienen que ver con sus tiempos de duración establecidos por el Protocolo X-10. Primeramente, un pulso uno lógico X-10 debe estar retrasado un tiempo máximo de  $50\mu\text{s}$  respecto al cruce por cero. La medición de éste tiempo se presenta en la Figura 2.39.

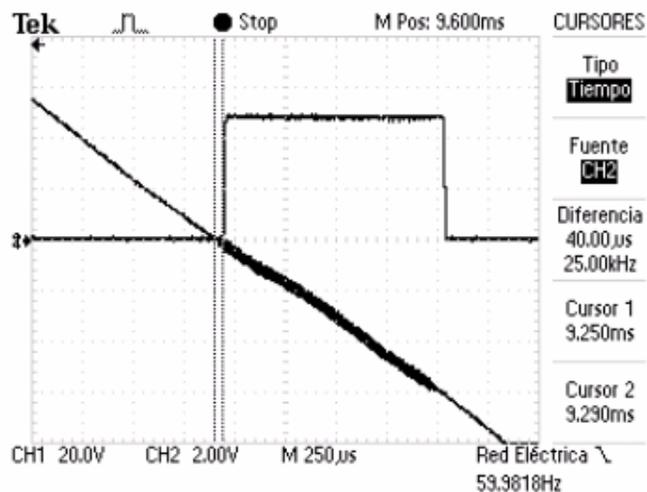


Figura 2.39. Retraso de la señal demodulada respecto al cruce por cero.

El tiempo medido en la Figura 2.39 fue de  $40\mu\text{s}$ , tiempo que se encuentra dentro del rango permitido. La segunda comprobación de tiempos de los pulsos uno lógico tienen que ver con su tiempo en alto: Debe durar al menos  $1\text{ms}$ . La medición del tiempo de duración del uno lógico que viaja en el semiciclo positivo y del uno lógico que viaja en el semiciclo negativo se muestra en la Figura 2.40 y 2.41 respectivamente.

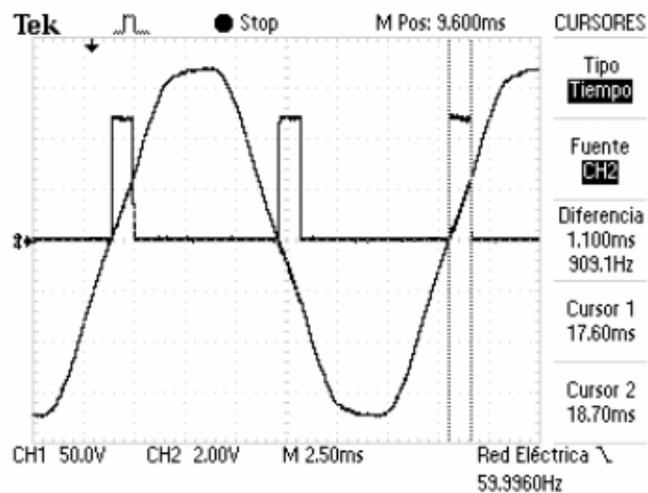


Figura 2.40. Duración del uno lógico que viaja en el semiciclo positivo

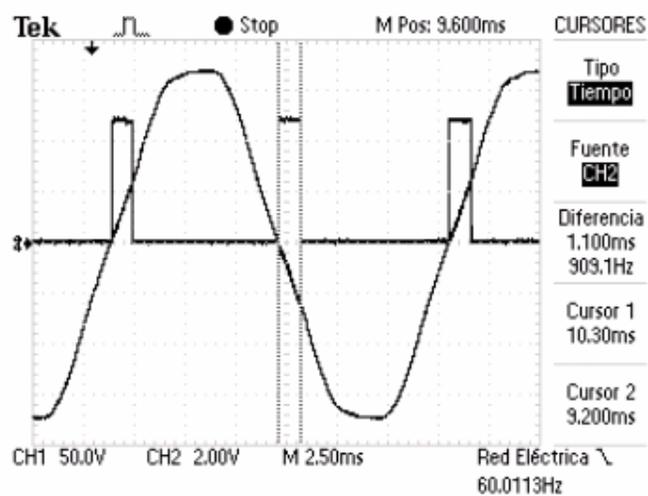


Figura 2.41. Duración del uno lógico que viaja en el semiciclo negativo

Los resultados de las dos mediciones fueron de 1,1ms para ambos casos. Éste resultado coincide con el tiempo en alto que entrega el módulo comercial PSC05 por lo que se puede afirmar que se logró igualar el rendimiento de dicho módulo pero utilizando dispositivos electrónicos que se encuentran fácilmente en el mercado ecuatoriano.

Como conclusión de todas las mediciones realizadas se puede afirmar que el demodulador construido cumple con todas las especificaciones eléctricas del Protocolo X-10, lo cual brindará total compatibilidad con los módulos comerciales a nivel de capa física.

#### **2.3.4 DISEÑO DEL DETECTOR DE CRUCE POR CERO**

Para el detector de cruce por cero de una señal AC con voltajes de 120V se decidió emplear un circuito similar al que contiene el módulo PSC04 debido a su simplicidad y confiabilidad. Como se puede observar en la Figura 2.42, el circuito utiliza un transistor en configuración de seguidor-emisor para adquirir la forma de onda de 120 V de la red. Para limitar la corriente proveniente de la red eléctrica se utiliza una resistencia de 240k $\Omega$  conectada a la base del transistor, para tener una corriente máxima de 0,7 mA. Debido a que en la configuración de seguidor emisor la ganancia de voltaje es aproximadamente 1, prácticamente no influye la resistencia del colector en la ganancia. La única condición es que ésta resistencia sea de un valor mucho mayor que la impedancia  $r_e$  del transistor, la cual está dada típicamente en unidades de ohmio. Por este motivo, la selección de ésta resistencia se la hace en relación a la corriente que el fotodiodo del circuito integrado 4N25 requiere para encenderse completamente. Se conectó una resistencia de 330 $\Omega$  debido a las razones expuestas en el numeral 2.3.3.3. También se conectó una resistencia de 220k $\Omega$  en la base del opto transistor para darle más estabilidad al detector de cruce por cero. El diodo conectado a la base del transistor sirve para eliminar los semiciclos negativos de la señal de la red. La resistencia que se conecta al emisor del opto transistor sirve solamente para Pull-down así que su valor será de 5,6k $\Omega$ . El canal del microcontrolador que permitirá la interrupción por cambio de estado será conectada a dicha resistencia.

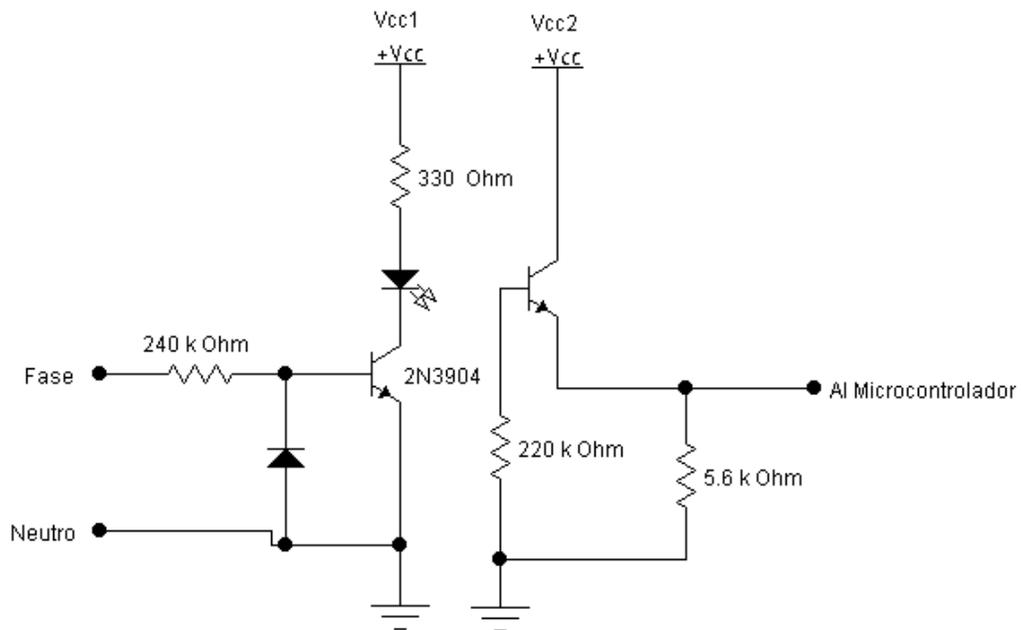


Figura 2.42. Detector de cruce por cero

Las formas de onda obtenidas de este circuito se pueden observar en la Figuras 2.43 y Figura 2.44

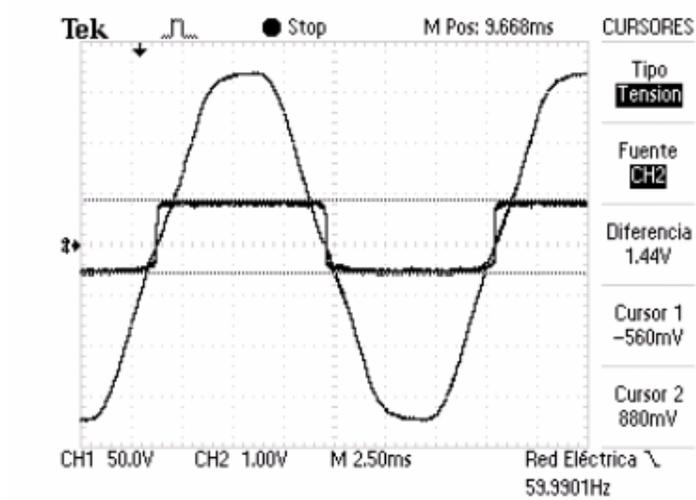


Figura 2.43. Entrada del transistor del detector de cruce por cero

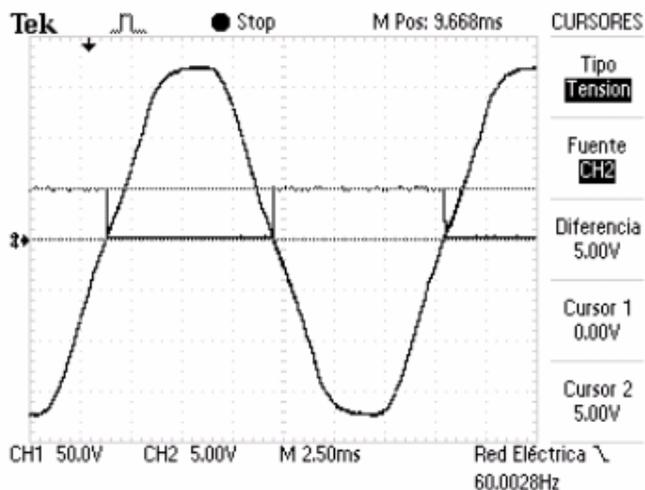


Figura 2.44. Salida del Detector de cruce por cero

Se puede observar que la forma de onda de salida del detector de cruce por cero es una onda cuadrada cuyos cambios de estado coinciden con los cruces por cero de la corriente alterna de la red, lo cual comprueba que el circuito diseñado cumple apropiadamente con su tarea.

### 2.3.5 DISEÑO DE LAS FUENTES DE ENERGÍA

En el caso del módulo receptor no se necesita ningún respaldo de energía ya que si el módulo no esta conectado a la red, no podrá recibir ningún comando X-10. Es necesario construir dos fuentes de energía debido a que se esta utilizando el circuito TL084 para realizar la demodulación y éste circuito requiere una fuente de voltaje positiva y una negativa. Por este motivo se requiere de una fuente para el circuito del microcontrolador y otra negativa para la alimentación del amplificador operacional.

Cada una de las fuentes de energía para el módulo receptor debe ser capaz de entregar un voltaje constante, con el menor rizado posible. Adicionalmente, es necesario que tengan la capacidad de entregar la corriente necesaria para el funcionamiento de todos los circuitos del Módulo receptor. La familia de microcontroladores seleccionada requiere de un voltaje de entre 2,0 a 5,5V para

funcionar apropiadamente. Por este motivo la fuente de voltaje positivo debe ser de 5V. En consecuencia, la fuente de voltaje negativo debe ser de  $-5V$  para que el amplificador operacional seleccionado funcione apropiadamente.

### 2.3.5.1 Diseño de la fuente de energía de 5V

El circuito elegido como fuente de alimentación es un rectificador de onda completa que tiene a su entrada un voltaje alterno de 12V. Se eligió un puente rectificador de onda completa ya que permite obtener una mejor calidad de voltaje para la alimentación del microcontrolador. El voltaje de 12V se lo obtiene de una de las bobinas de un transformador reductor de dos bobinas independientes que convierte los 120V de la red eléctrica en 12V. La otra bobina será utilizada para la fuente de voltaje negativo.

Luego de que la corriente alterna es rectificada se utiliza un filtro capacitivo para tener una corriente continua mucho más pura. Finalmente, para garantizar que el voltaje de alimentación sea siempre de 5V se utiliza un circuito integrado regulador de voltaje NTE960, o su equivalente genérico el CI 7805. Éste circuito integrado requiere como entrada un voltaje de 7V a 20V y entrega hasta 1A de corriente. El circuito de una de las fuentes se encuentra en la Figura 2.45

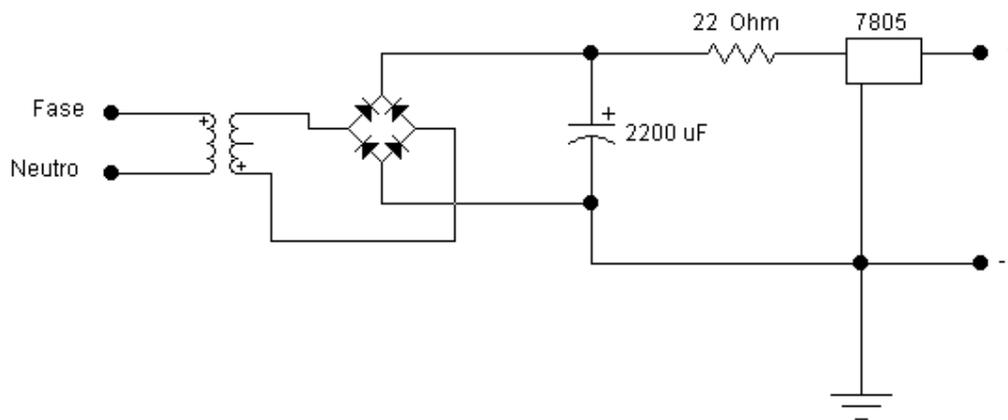


Figura 2.45. Fuente de energía de 5V.

### 2.3.5.2 Diseño de la fuente de energía de -5V

El circuito elegido como fuente de alimentación es un rectificador de media onda que tiene a su entrada un voltaje alterno de 12V. Se eligió un rectificador de media onda ya que ésta fuente solo tiene que alimentar al circuito integrado TL084. El voltaje de 12V se lo obtiene del otro secundario del transformador de dos bobinas independientes con relación de transformación 10:1. Luego de que la corriente alterna es rectificada se utiliza un filtro capacitivo para tener corriente continua más pura. De igual manera que para la fuente de voltaje positivo, para garantizar que el voltaje de alimentación sea siempre de 5V se utiliza un circuito integrado regulador de 7805. Como el transformador utilizado tiene bobinas independientes se cortocircuita la salida positiva de ésta fuente con la salida negativa de la fuente de voltaje positivo para tener los -5V necesarios. El circuito de una de las fuentes se encuentra en la Figura 2.46

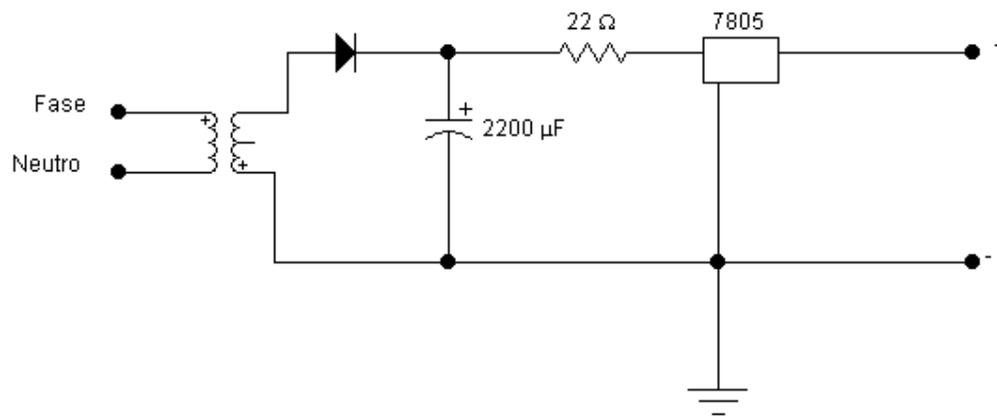


Figura 2.45. Fuente de energía de -5V.

### 2.3.6 DISEÑO DE LOS ACTUADORES

Los actuadores son los encargados de ejercer las acciones de control sobre la iluminación o los electrodomésticos. Para el presente proyecto se implementan tanto actuadores para iluminación como actuadores de potencia.

### 2.3.6.1 Diseño de los actuadores de iluminación

Éste tipo de actuador debe ser capaz de manejar cualquier tipo de elemento de iluminación cuya corriente sea menor que 1 amperio. Se pensó en 1 amperio debido a que los elementos de iluminación más comunes (bombillas incandescentes de 100W) consumen alrededor de 0,83 amperios y todos los demás elementos de iluminación para el hogar consumen corrientes menores. El circuito diseñado para este propósito se lo puede observar en la Figura 2.46

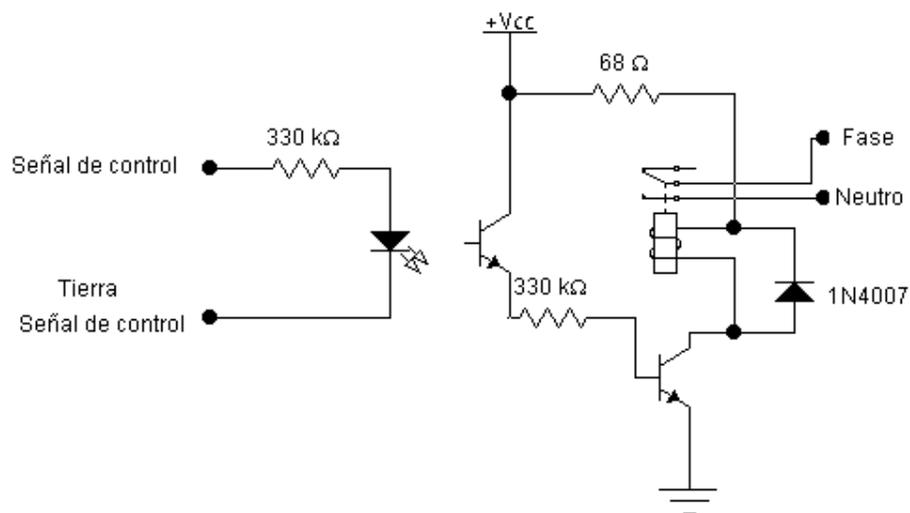


Figura 2.46. Circuito del actuador de Iluminación.

Los actuadores de iluminación están en capacidad de manejar corrientes de hasta 1 amperio a un voltaje de 120V, por esta razón se lo puede utilizar para controlar bombillas incandescentes de hasta 100W con total seguridad así como también puede manejar luminarias ahorradoras de energía. Para aislar la bobina del relé del circuito de control se utilizó un optotransistor 4N25. Para limitar la corriente de activación para el fotodiodo se utilizó una resistencia de 330Ω por las razones expuestas en los numerales anteriores. Se debió utilizar un transistor NPN 2N3904 para amplificar corriente debido a que al fototransistor no puede activar a la bobina del relé por si solo. Para la activación de la bobina del relé se utilizan 15 voltios de corriente continua, los cuales se toman desde el capacitor de la fuente positiva de voltaje. La resistencia de 68Ω es necesaria para reducir el voltaje que

cae sobre la bobina del relé a 12V y de ésta manera no sobrepasar su valor nominal de voltaje. El calculo de ésta resistencia se la hizo aplicando un divisor de voltaje. La resistencia de la bobina del relé es de  $270\Omega$  (Valor medido) y se necesita que caigan sobre ella 12 V. Aplicando un divisor de voltaje se obtiene una resistencia de  $67\Omega$ . El valor de resistencia estándar más apropiado es, por lo tanto,  $68\Omega$

El diodo en inverso paralelo a la bobina del relé sirve para descargar la energía acumulada en forma de campo magnético.

Luego de construido el circuito se lo probó con una bombilla incandescente de 100W y con una lámpara ahorradora de energía de 25W y en ambos casos funcionó apropiadamente. Las pruebas completas de funcionamiento se las realizará en el capítulo de pruebas.

### 2.3.6.2 Diseño de los actuadores de Potencia

Éste tipo de actuador debe ser capaz de manejar cualquier tipo de electrodoméstico cuya corriente sea menor que 20 amperios. La capacidad de corriente está dada por el relé que se utilice. El circuito diseñado para este propósito se lo puede observar en la Figura 2.47

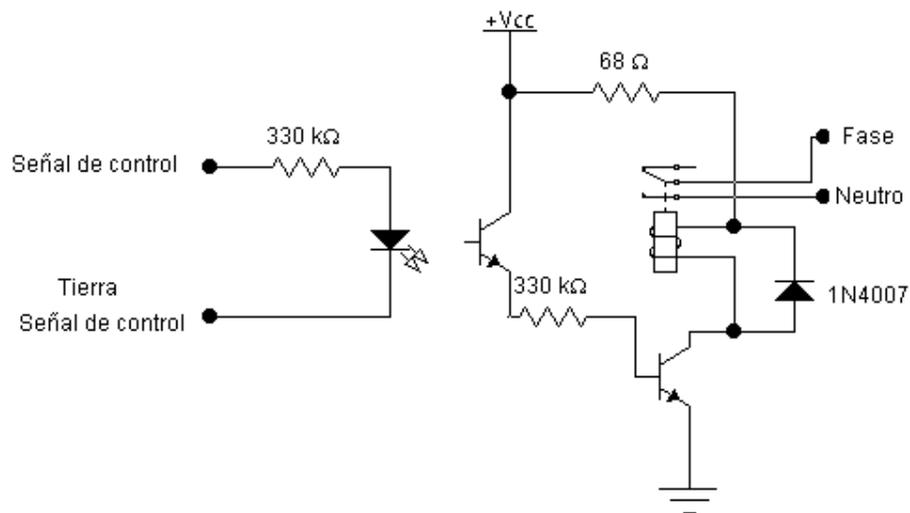


Figura 2.47. Circuito del actuador de potencia.

Como se puede observar en la figura 2.47, el circuito del actuador de potencia es idéntico al circuito del actuador de iluminación a excepción del relé utilizado. La capacidad de conducción del relé utilizado para corriente alterna es: 10A a 125V o 7A a 240V. El optotransistor sirve para aislar eléctricamente a la bobina del relé del circuito de control. El transistor tipo NPN permite amplificar corriente y actúa exclusivamente como interruptor (entre corte y saturación). Debido a que las características de las bobinas son casi idénticas se usa el mismo valor de resistencia limitadora para la bobina del relé. El diodo permite descargar la corriente almacenada en la bobina sin dañar ningún elemento del circuito.

Luego de terminar el diseño de los circuitos del Módulo de Control y de los Módulos receptores es posible diseñar el Software para los módulos antes mencionados y para la computadora. El proceso de diseño del software para el Módulo de Control, para los Módulos receptores y para la computadora (Interfaz del usuario) se desarrolla en el siguiente capítulo.

# **CAPÍTULO 3**

## **DISEÑO DEL SOFTWARE**

## **CAPÍTULO 3: DISEÑO DEL SOFTWARE**

El software necesario para este proyecto debe permitir la correcta comunicación por un lado entre el Módulo de Control diseñado, el módulo comercial PSC04 y una PC. Por otro lado, debe permitir la comunicación dentro de la red X-10 entre el Módulo de Control y los Módulos actuadores diseñados. Para ello, el primer paso necesario es elaborar un protocolo de comunicación entre la PC y el Módulo de Control. Para la comunicación entre el Módulo de Control y el módulo PSC04 es necesario seguir las reglas que establece el fabricante de dicho módulo. Finalmente, para la comunicación entre el Módulo de Control y los Módulos actuadores es necesario aplicar el protocolo X-10 Pro para tener compatibilidad con cualquier módulo X-10 que se encuentre instalado o que se pueda encontrar en el mercado. Luego de estas consideraciones generales se procede a diseñar los tres programas: Software para el Módulo de Control, software para la computadora y el software para los Módulos actuadores.

Para el diseño del software del Módulo de Control fue necesario desarrollar al mismo tiempo algunos subVI para poder verificar el funcionamiento de los códigos programados para el microcontrolador. Estos subVI son posteriormente usados para desarrollar el software para la computadora.

### **3.1 SOFTWARE PARA EL MÓDULO DE CONTROL**

El Módulo de Control es el encargado de manejar al módulo PSC04 para que éste último envíe los códigos X-10 a través de la red eléctrica. También es el encargado de comunicarse con la computadora para recibir los comandos del usuario y ejecutarlos como comandos X-10, ya sea en tiempo real, o para almacenarlos en memoria no volátil y ejecutarlos a la fecha y hora especificadas por el usuario.

Es necesario que el programa sea modular para facilitar cualquier actualización o mejora, y para poder garantizar la eficacia y eficiencia de cada uno de los

algoritmos utilizados. Los módulos en los que se dividió al programa son los siguientes:

1. Transmisión y recepción de datos entre la PC y el Microcontrolador
2. Sincronización con la red eléctrica y codificación de comandos recibidos a código X-10.
3. Reloj en tiempo real.
4. Escritura en memoria no volátil.
5. Método de almacenamiento, algoritmo de lectura y procedimientos de ejecución de comandos guardados en memoria EEPROM.
6. Algoritmo de igualación del reloj

El diagrama de flujo del programa principal del Módulo de Control se lo presenta en la Figura 1.1 y el diagrama de flujo para las interrupciones en el Módulo de Control se lo muestra en la Figura 2.2.

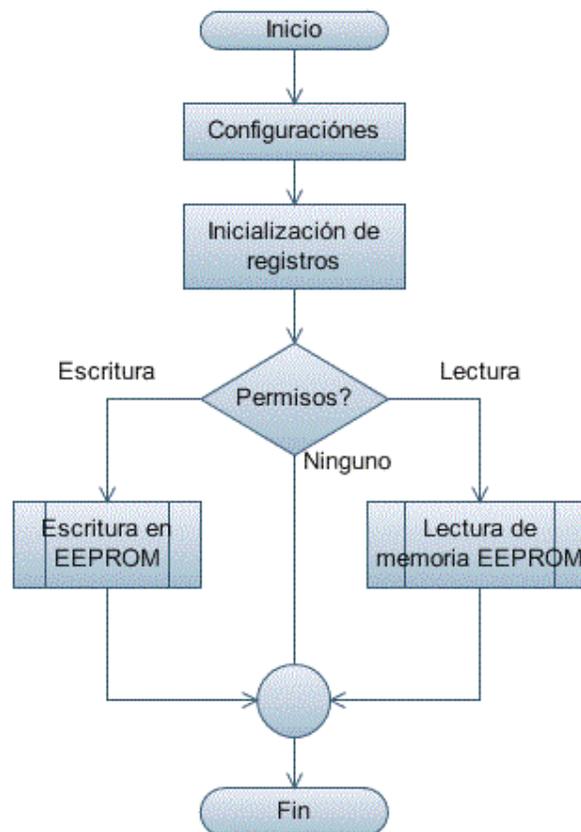


Figura 1.1. Diagrama de flujo del Programa del Módulo de Control.

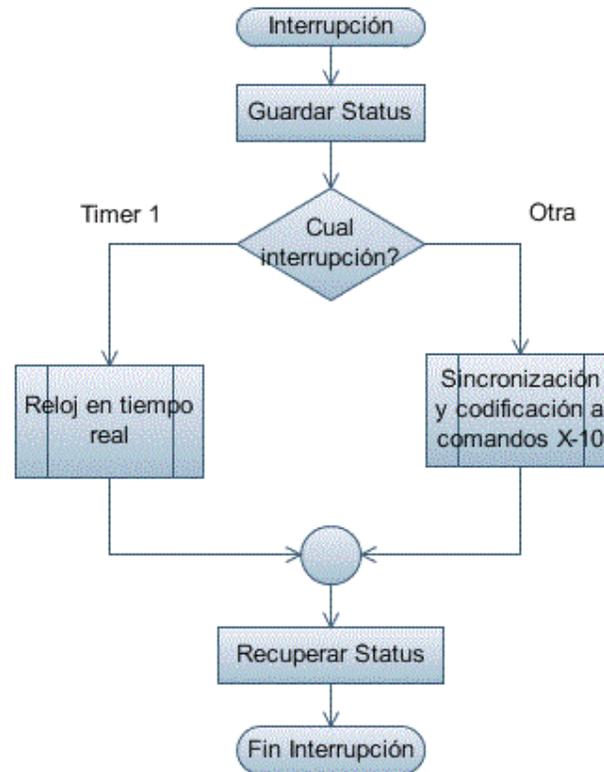


Figura 1.2. Diagrama de flujo de las Interrupciones del Módulo de Control.

### 3.1.1 TRANSMISIÓN Y RECEPCIÓN DE DATOS ENTRE LA PC Y EL MICROCONTROLADOR

Este módulo se lo programó primero ya que es imprescindible para el desarrollo del programa y para garantizar la correcta comunicación entre el Módulo de Control y la PC. El primer paso para garantizar la correcta comunicación fue la creación de un programa que reciba un dato y lo devuelva. Esto permitió comprobar el correcto funcionamiento del hardware. Simultáneamente se desarrolló un programa para que la PC envíe un dato por el puerto serial, luego lea un dato desde puerto serial y presente el valor recibido en pantalla. El programa de la PC que se utilizó fue “comunicación serial.vi”, mientras que el programa en assembler para el microcontrolador es “Comunicación serial.asm”.

El código de “comunicación serial.vi” permite configurar el puerto serial de la computadora y esperar a que el usuario presione el botón para enviar un dato de 8 bits. El dato que se envía es el que se encuentre en el control “Enviado”. Luego

el programa espera 1 segundo y procede a leer el dato que se encuentre en el buffer de recepción del mismo puerto serial usado para la transmisión. Éste dato es presentado en el recuadro “Recibido”. El panel de control de éste VI se lo puede observar en la Figura 3.3.



Figura 3.3 Panel de control del subVI “Comunicación Serial.VI”

El código de “Comunicación serial.asm” permite configurar el microcontrolador para recibir datos seriales de manera continua, utilizando el puerto serial en modo de recepción continua y utilizando la interrupción de recepción serial. Luego de recibido el dato, éste es inmediatamente retransmitido hacia la computadora. Luego de terminada la transmisión, el microcontrolador retorna a estado de espera para estar listo para un nuevo dato.

Luego de desarrollados ambos códigos se realizaron 20 pruebas de comunicación a la velocidad de 9600 bps cuyos resultados se presentan en la Tabla 3.1.

Tabla 3.1 Pruebas de comunicación serial

Dato enviado por la PC	Dato recibido por la PC
0	0
2	2
4	4
8	8
16	16
32	32
64	64
128	128
255	255
1	1
7	7
99	99
77	77
27	27
98	98
254	254
31	31
133	133
12	12
46	46

Como se puede observar, las pruebas de comunicación serial fueron exitosas en un 100%. De esta manera se demuestra que el hardware para la comunicación serial funciona apropiadamente; es decir, el circuito del Max232, y las conexiones con el microcontrolador, así como el cable construido para la comunicación con el puerto serial de la computadora.

También se comprobó que la recepción y transmisión de datos por parte del microcontrolador fue correctamente programada, así como también se verificó la adecuada configuración del puerto serial de la computadora.

En un experimento posterior se trató de incrementar el baud-rate, pero la transmisión perdió confiabilidad; por lo cual se decidió hacer la comunicación a 9600 bps. Esta falla en la transmisión se le atribuye al cable para transmisión serial utilizado ya que el microcontrolador PIC16F877A teóricamente está en

capacidad de transmitir / recibir hasta 1250 kbps, mientras que el MAX232 teóricamente puede transmitir / recibir hasta 120 kbps.

Luego de garantizar la correcta comunicación entre la computadora y el microcontrolador, el siguiente paso fue determinar la manera de enviar los datos de dirección de casa y dirección de dispositivo, así como del código de función desde la computadora hacia el microcontrolador. Para ahorrar memoria no volátil en el microcontrolador se decidió comprimir la trama X-10 completa de 50 bits a 16 bits, evitando enviar el encabezado X-10 (el cual es siempre el mismo), así como los bits complementados de las direcciones de casa, dispositivo y Código Clave. También se omitieron los bits de parada entre el primer byte X-10 y el segundo. De esta manera se consiguió que solo se requieran 2 bytes de 8 bits para enviar un comando X-10 completo desde la PC hasta el microcontrolador. Posteriormente el microcontrolador reconstruirá la trama X-10 para transmitir los comandos X-10 a través de la red. El primer byte a enviar se divide en dos partes: los cuatro primeros bits corresponden a las cuatro cifras mas significativas del Código de función (el bit menos significativo siempre será 1 cuando corresponda al Código de función.) y los cuatro bits menos significativos corresponden al Código de casa. El segundo bit mas significativo también fue dividido en 2 partes: Los tres bits mas significativos corresponden a la acción que debe ejecutar el microcontrolador con los bytes recibidos, mientras que los 5 bits menos significativos corresponden al código de dirección. Los tres primeros bits del segundo byte corresponden a varios comandos propios del presente proyecto, los cuales se describen a continuación:

- |       |     |   |                                       |
|-------|-----|---|---------------------------------------|
| Bit 7 | = 1 | → | Escribir comando en memoria EEPROM    |
|       | = 0 | → | Comando a ejecutarse en tiempo real   |
| Bit 6 | = 1 | → | Comando para igualar el reloj interno |
|       | = 0 | → | No se va a igualar el reloj interno   |
| Bit 5 | = 1 | → | Es una instrucción DIM o Bright       |
|       | = 0 | → | No es una instrucción DIM o Bright    |

Luego de establecido el contenido de los 2 bytes que se recibirán en cada transmisión se debe establecer el modo de transmisión. Debido a que la cantidad de datos que se deben transmitir no es muy alta y que cada comando X-10 tarda 0.833 segundos en enviarse a través de la red eléctrica, se decidió utilizar un método mas seguro que el bit de paridad para verificar que la comunicación funcione apropiadamente. El método elegido es enviar cada dato dos veces y comparar si los dos datos recibidos son idénticos. Éste tipo de comprobación vuelve a la transmisión más lenta pero garantiza que la comunicación sea muy fiable. Si los dos datos coinciden, el microcontrolador enviará un código de recepción exitosa. En caso contrario el microcontrolador enviará un código de error y esperará el reenvío del último dato. Éste método de transmisión será aplicado con todos los bytes que se deban enviar. Para aplicar este método de transmisión se desarrolló un subVI llamado “Envío 1 dato.VI” y un programa para el microcontrolador llamado “Recibo 1 dato.asm”. El panel de control del “Envío 1 dato.VI” se muestra en la Figura 3.4



Figura 3.4 Panel de control del subVI Envío 1 dato.VI

El código de “Envío 1 dato.VI” es básicamente igual al de “comunicación serial.vi” con la diferencia de que “Envío 1 dato.VI” envía el dato del recuadro de control

“Enviado” dos veces y luego espera la respuesta del microcontrolador. El diagrama de flujo de “Recibo 1 dato .VI” se lo puede observar en la Figura 3.5.

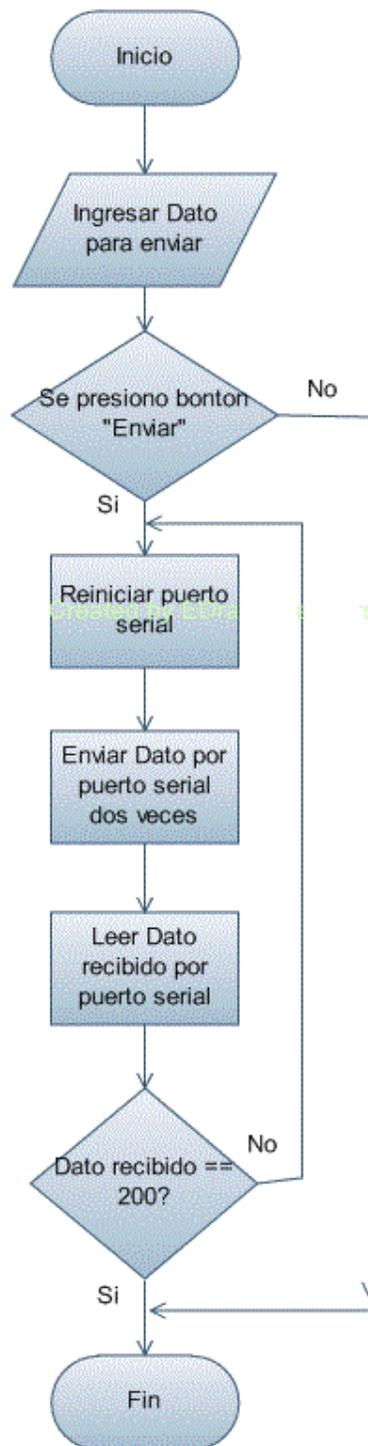


Figura 3.5. Diagrama de flujo de “enviar 1 dato.VI”

Para el caso del microcontrolador, el código de “Recibo 1 dato.asm” es muy similar al código de “comunicación serial.asm” con la diferencia de que “Recibo 1 dato.asm” recibe dos datos y los compara. Si los dos datos son iguales envía el número 200 como respuesta a la computadora. Si los dos bytes son diferentes envía un 30 como respuesta para la computadora.

Luego de enviar la respuesta a la computadora sale de la interrupción y espera otros dos bytes. El diagrama de flujo de “Recibo 1 dato.asm” se lo pude observar en la Figura 3.6.

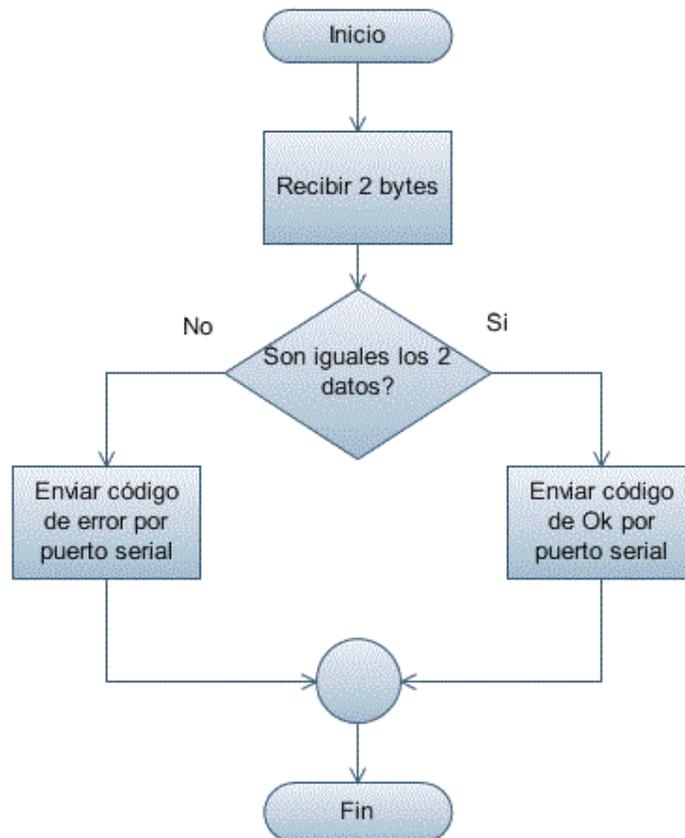


Figura 3.6. Diagrama de flujo de “Recibo 1 dato.asm”

Para probar que el sistema de transmisión esté funcionando correctamente y sin errores se hicieron 20 pruebas de envío de datos. Los resultados se pueden observar en la Tabla 3.2

Tabla 3.2 Pruebas de Recibo 1 dato

Dato enviado por la PC	Dato recibido por la PC
13	13
24	24
36	36
49	49
58	58
67	67
89	89
92	92
109	109
118	118
123	123
136	136
147	147
160	160
171	171
185	185
191	191
208	208
211	211
224	224

Como se puede observar en la Tabla 3.2, el envío del dato por duplicado sigue teniendo una efectividad del 100%. Con estos resultados se puede afirmar que el sistema de transmisión es muy fiable y se puede confiar en su buen funcionamiento, en conjunto con los módulos de software restantes que lo utilizarán para cumplir sus tareas.

### 3.1.2 SINCRONIZACIÓN CON LA RED ELÉCTRICA Y CODIFICACIÓN DE COMANDOS RECIBIDOS A CÓDIGO X-10

Debido a la gran importancia de los tiempos de transmisión y recepción en el protocolo X-10, es necesaria una buena sincronización de la transmisión del código X-10 con los cruces por cero de la línea de poder. Como ya se vio anteriormente, en el capítulo de hardware, el módulo PSC04 detecta por hardware los cruces por cero de la línea de poder, y el microcontrolador detecta el cruce a través de uno de los canales del puerto B que permiten interrupciones por cambio de estado. La onda que entrega el módulo PSC04 es una onda cuadrada, cuyo tiempo en alto coincide con el semiciclo positivo y su tiempo en bajo coincide con el semiciclo negativo. Como la señal se la adquiere utilizando una interrupción por cambio de estado, se producirá una interrupción en el momento en que ocurra un cruce por cero, ya sea por el cambio del semiciclo negativo al positivo (flanco ascendente de la señal del PSC04) o por el cambio del semiciclo positivo al negativo (flanco descendente de la señal del PSC04). Es necesario detectar ambos cruces por cero debido a que el protocolo X-10 exige que se transmita un bit durante el comienzo de un semiciclo y su complemento durante el siguiente semiciclo (a excepción del código de inicio). La detección de cruce por cero del microcontrolador solo debe estar activa mientras se realiza una transmisión de código X-10 hacia la red eléctrica, para no distraer al microcontrolador de otras operaciones que también utilizan interrupciones.

La rutina del microcontrolador de interrupción por cambio de estado, necesaria para realizar la transmisión del código X-10, se la desarrolló inicialmente en "envío código completo.asm". Éste programa está fundamentalmente diseñado para facilitar la depuración del programa para codificación a X-10 de un comando enviado desde la computadora. Se completó esta rutina en "Envío código X10.asm".

El código de "envío código completo.asm" incluye la configuración de la interrupción por cambio de estado para el canal del puerto B y las modificaciones necesarias en la rutina de interrupciones, para que durante el pooling de las

fuentes de interrupción se pueda detectar si la fuente de interrupción fue un cambio de estado en el puerto B o no.

Es necesario que se verifique primero esta fuente de interrupción debido a que el retardo máximo permitido por el protocolo X-10 entre el cruce por cero de la línea de poder y el inicio de la transmisión de un bit es de 50  $\mu$ s. Luego de que se determina que la interrupción fue causada por un cambio de estado en el puerto B, se procede a determinar que bit de la trama X-10 es que se va a transmitir, mediante el uso de un contador (puntero), el cual se encera al inicio del programa y luego de que toda la trama X-10 ha sido transmitida.

El siguiente paso es realizar un conteo del milisegundo que dura el dato X-10, luego del cual se llevará a cero lógico la salida del canal encargado de la transmisión de comandos X-10, para cumplir con las especificaciones de tiempo en alto del uno lógico. Para el caso del cero lógico esto último no importa ya que cero lógico es el estado por defecto. Para ello se desarrolló el código de “Envío código X10.asm”, cuyo diagrama de flujo se encuentra en la Figura 3.7.

El programa de “Envío código X10.asm” añade el código necesario para la configuración del timer 2, para el conteo del milisegundo que debe durar cada bit X-10. Ésta tarea será realizada por el timer 2 debido a que éste timer no requiere recargar ningún valor en sus registros.

Como se trata de una transmisión de un código X-10 completo, también es necesario incluir las rutinas necesarias para crear la pausa de 3 ciclos completos de la línea de poder entre los dos bytes X-10 que conforman la mayoría de comandos X-10.

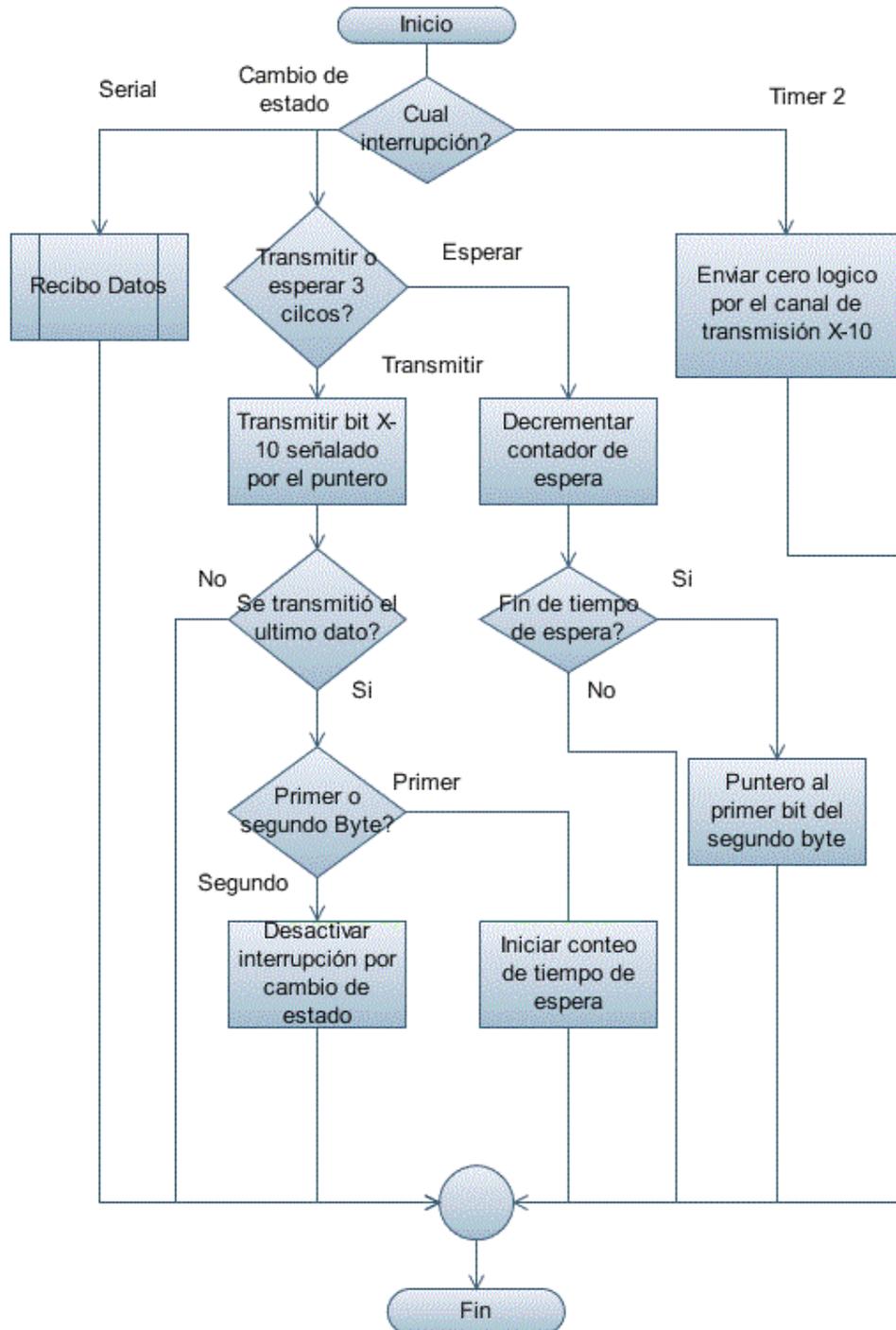


Figura 3.7. Diagrama de flujo de la sincronización con la red eléctrica y codificación de comandos X-10

La función “Recibo Datos” se explica a continuación en lenguaje estructurado.

**Recibo Datos**

Si            dato recibido es el Primer Byte  
                  Preparar recepción del Segundo Byte  
                  Guardar Primer Byte

Caso contrario    dato recibido es el Segundo Byte  
                  Preparar recepción del Primer Byte  
                  Guardar Segundo Byte  
                  Borrar Flag de Cambio de estado  
                  Habilitar interrupción por cambio de estado

**Fin Tarea**

Luego de que el código se logró depurar por completo mediante el uso del simulador, se procedió a realizar pruebas con el hardware para verificar el funcionamiento del programa.

Las pruebas realizadas consistieron en transmitir códigos para encender y apagar los dispositivos comerciales. Los dispositivos comerciales que se usaron para las pruebas se adquirieron en [www.X-10.com](http://www.X-10.com) y fueron los siguientes: 1 "SocketRocket", 1 "Lamp Module" y 1 "Appliance Module"

El módulo "SocketRocket" es el módulo básico para control de iluminación. Solo permite comando On/Off sobre bombillas incandescentes. Éste módulo está diseñado para uso en el interior solamente. Para las pruebas, este dispositivo fue configurado con la dirección "A1". Se lo puede observar en la Figura 3.8.



Figura 3.8. Módulo "SocketRocket", X-10.com

El módulo "LampModule" está diseñado para el control de iluminación de lámparas de bombillas incandescentes. Éste módulo permite comandos de on, off, dim y bright. Éste módulo está diseñado para uso en el interior solamente. Para

las pruebas, este dispositivo fue configurado con la dirección “A2”. Se lo puede observar en la Figura 3.9.



Figura 3.9. Módulo “Lamp Module”, X-10.com

El módulo “Appliance Module” permite controlar cargas resistivas con corrientes máximas de 15 amperios. También puede manejar motores de hasta 1/3 HP y televisiones de hasta 400W. Este módulo permite solamente el control on/off del aparato. Éste módulo está diseñado para uso en el interior solamente. Para las pruebas, este dispositivo fue configurado con la dirección “A3”. Se lo puede observar en la Figura 3.10.



Figura 3.10. Módulo “Appliance Module”, X-10.com

Para probar el funcionamiento de las rutinas de sincronización con la red eléctrica y de la codificación de comandos a Código X-10 se utiliza el subVI “Envío 1 dato.VI” junto con el software desarrollado para el Módulo de Control en el presente numeral. Para ello fue necesario calcular los números que se debían enviar desde la computadora tanto en el primer como en el segundo byte. Se debe tener en cuenta que como los 4 bits más significativos del primer Byte corresponderán al Código de Función, es necesario obtener el equivalente en decimal del código binario correspondiente a los 4 bits más significativos del Código de Función y multiplicarlo por 16 (rotar 4 posiciones a la izquierda). Éstas

operaciones aritméticas son necesarias para acomodar 2 tipos de códigos X-10 de 4 bits dentro de un solo byte de 8 bits. Los números decimales que resultan de esta operación se muestran en la Tabla 3.3.

Tabla 3.3 Número decimales correspondientes a los Códigos de función.

Código X-10	Equivalente decimal x 16	
Encender	2	32
Apagar	3	48
Encender todas las luces	1	16
Apagar todas las luces	6	96
Apagar todos los dispositivos	0	0
Dim	4	64
Bright	5	80

Los 4 bits menos significativos del primer byte corresponden al código de casa. Para las pruebas se configuró a los módulos comerciales con el Código de Casa "A". El número decimal correspondiente al Código de Casa "A" es 6. Los demás números decimales correspondientes a los otros Códigos de Casa se muestran en la Tabla 3.4.

Tabla 3.4 Número decimales equivalentes al Código de Casa X-10

Código de Casa	
Código X-10	Equivalente decimal
A	6
B	14
C	2
D	10
E	1
F	9
G	5
H	13
I	7
J	15
K	3
L	11
M	0
N	8
O	4
P	12

Por lo tanto, el número que corresponde al primer byte que se tiene que enviar al Módulo de Control se lo obtiene de la suma del equivalente decimal del Código de Función (ya desplazado 4 posiciones a la izquierda), con el equivalente del Código de Casa. Con ésta operación aritmética se construye el arreglo que permite comprimir la trama X-10 de 22 bits a solamente 8 bits para la comunicación entre la PC y el Módulo de Control (y posteriormente permitirá optimizar la cantidad de memoria no volátil utilizada para guardar instrucciones X-10). Por ejemplo, para enviar el comando “encender” un dispositivo con Código de Casa “A”, se deberá enviar el número:  $32 + 6 = 38$ . Para enviar el comando “apagar” ese mismo dispositivo se deberá enviar:  $48 + 6 = 54$ .

Para determinar el número decimal que se tiene que enviar en el segundo byte, es necesario encontrar el valor decimal equivalente del Código de Dirección del dispositivo. Según el orden establecido anteriormente, el código de dirección ocupará los 5 bits menos significativos del segundo Byte. Estos equivalentes se pueden encontrar en la Tabla 3.5.

Tabla 3.5 Equivalente decimal del Código de Dirección X-10.

Código de Dirección	
Código X-10	Equivalente decimal
1	12
2	28
3	4
4	20
5	2
6	18
7	10
8	26
9	14
10	30
11	6
12	22
13	0
14	16
15	8
16	24

Los 3 bits mas significativos del segundo Byte corresponden al comando que debe ejecutar el microprocesador. El equivalente decimal de estos comandos debe ser multiplicado por 32 (rotar 5 espacios a la izquierda). Éstos equivalentes decimales se listan en la Tabla 3.6.

Tabla 3.6 Equivalente decimal de los Comandos del microprocesador.

Comando de Microprocesador	Equivalente decimal	x 32
Ejecutar comando X-10	0	0
Dim o Bright	1	32
Igualar Reloj	2	64
Escribir comando en EEPROM	4	128

De manera similar que para el primer byte, es necesario sumar el equivalente decimal del Código de Dirección con el equivalente decimal de los comandos del microprocesador, para obtener arreglo de 8 bits que se debe enviar como segundo Byte desde la computadora.

Luego de determinar los números decimales que se deben enviar desde la computadora mediante “Envío 1 dato.VI”, se procedió a realizar varias pruebas de funcionamiento de Envío código X-10.asm”, las cuales se resumen en la Tabla 3.7.

Tabla 3.7 Resumen de pruebas para el código de “Envío código X-10.asm”

Cód.Casa	Cód. Función	1 <sup>er</sup> Byte	Comando micro.	Cód. Dir.	2 <sup>do</sup> Byte	Resultado
A	Encender	38	Ejecutar comando X-10	1	12	Disp. A1 encendido
A	Apagar	54	Ejecutar comando X-11	1	12	Disp. A1 apagado
A	Encender	38	Ejecutar comando X-10	2	28	Disp. A2 encendido
A	Apagar	54	Ejecutar comando X-11	2	28	Disp. A2 apagado
A	Encender	38	Ejecutar comando X-10	3	4	Disp. A3 encendido
A	Apagar	54	Ejecutar comando X-11	3	4	Disp. A3 apagado
A	Encender Luces	22	Ejecutar comando X-11	N/A	0	Disp. A1 encendido
						Disp. A2 encendido
A	Apagar luces	102	Ejecutar comando X-11	N/A	0	Disp. A1 apagado
A	Apagar todo	6	Ejecutar comando X-12	N/A	0	Disp. A1 apagado
						Disp. A2 apagado
						Disp. A3 apagado

Las pruebas descritas en la Tabla 3.7 se repitieron 5 veces y se obtuvo siempre el mismo resultado. De esta manera se comprobó que la codificación a protocolo X-10 de los comandos enviados desde la computadora, así como la transmisión del código X-10 se realizó correctamente por parte del Módulo de Control.

Luego de haber verificado que todo el programa funciona bien, se debe dar el último paso para terminar con la codificación de comandos X-10: Permitir la codificación de instrucciones espaciales, es decir Dim y Bright. La diferencia entre la instrucción Dim (o Bright), y todas las demás es que luego de los dos Bytes que normalmente se transmiten, es necesario reenviar el Código de Función de Dim tantas veces como pasos de luminosidad se desee atenuar la luz. De acuerdo con el protocolo, para el reenvío de éste código no se debe dejar la pausa de tres ciclos de la línea de poder entre los Bytes que sigan al segundo Byte X-10. Cuando la instrucción a ejecutar sea Dim (o Bright) se requiere recibir un tercer byte desde la computadora para saber cuantas veces se debe reenviar el Código de Función de Dim (o Bright). El programa desarrollado para este propósito es "Envío código X-10\_dim\_z.asm". Para la creación del código de "Envío código X-10\_dim\_z.asm" fue necesario modificar en "Envío código X-10.asm" la parte correspondiente al código de la función "Recibir Datos". Su nuevo contenido se detalla a continuación en lenguaje estructurado.

#### Recibo Datos

```

Si      dato recibido es Tercer o Cuarto Byte
      Si      dato recibido es Tercer Byte
            Preparar recepción del Primer Byte
            Guardar Tercer byte en Contador Dim
            Borrar Flag de Cambio de estado
            Habilitar interrupción por cambio de estado

Caso contrario dato recibido es el Primer o Segundo Byte
      Si      dato recibido es Primer Byte
            Preparar recepción del Segundo Byte
            Guardar Primer Byte
      Caso contrario dato recibido es Segundo Byte
            Preparar recepción del Primer Byte
            Guardar Segundo Byte
            Borrar Flag de Cambio de estado

```

Habilitar interrupción por cambio de estado

Si        dato recibido corresponde a comando Dim o Bright  
              Preparar recepción del Tercer Byte  
              Preparar transmisión continua para Dim o Bright

**Fin Tarea**

Si se trata de una instrucción Dim (o Bright) se debe habilitar la recepción de un tercer byte, el cual debe contener el número de repeticiones de la instrucción Dim (o Bright). Luego de que se reciba el tercer byte, el programa realiza la transmisión de los dos primeros bytes normalmente, pero cuando transmita el último bit del segundo byte X-10, en lugar de terminar la transmisión, el programa repite el envío del segundo byte X-10 por un número de veces igual al recibido en el tercer byte. Es decir que el tercer byte recibido desde la computadora se lo utiliza como un contador para la ejecución del comando Dim.

La última modificación hecha en el programa permite que luego de que se transmita el segundo byte X-10 se desactive la pausa de tres ciclos de la línea de poder.

Una vez realizadas estas modificaciones y de haber depurado el programa, se precedió a realizar pruebas para comprobar su correcto funcionamiento. Éstas pruebas se las resumen en la Tabla 3.8 para la instrucción Dim y en la Tabla 3.9 para la instrucción Bright. Las pruebas fueron hechas solamente con el "Lamp Module" ya que es el único de los dispositivos comerciales adquiridos que puede ejecutar comandos Dim o Bright.

Para realizar las pruebas del comando Dim es necesario ejecutar una instrucción de encendido del dispositivo A2, tal como se describió en la Tabla 3.7, antes de cada prueba del comando Dim. Los resultados de las pruebas se los resume en la Tabla 3.8.

Tabla 3.8. Resultados de las pruebas con el comando Dim

Cód.C.	Cód. F.	1 <sup>er</sup> Byte	Com. micro.	Cód. D.	2 <sup>do</sup> Byte	3 <sup>er</sup> Byte	Resultado
A	Dim	70	Ejec. X-10	2	60	2	Disp. A2 Atenuado
A	Dim	70	Ejec. X-10	2	60	4	Disp. A2 Atenuado
A	Dim	70	Ejec. X-11	2	60	6	Disp. A2 Atenuado
A	Dim	70	Ejec. X-12	2	60	8	Disp. A2 Atenuado
A	Dim	70	Ejec. X-13	2	60	10	Disp. A2 Atenuado
A	Dim	70	Ejec. X-14	2	60	12	Disp. A2 Atenuado
A	Dim	70	Ejec. X-15	2	60	14	Disp. A2 Atenuado
A	Dim	70	Ejec. X-16	2	60	16	Disp. A2 Atenuado
A	Dim	70	Ejec. X-17	2	60	18	Disp. A2 Atenuado
A	Dim	70	Ejec. X-18	2	60	20	Disp. A2 Apagado

Previo a la realización de cada prueba con el comando Bright es necesario, en primer lugar, ejecutar una instrucción de encendido del dispositivo A2 tal como se describió en la Tabla 3.7 y posteriormente se requiere ejecutar un comando Dim (con un número 20 como tercer Byte para apagar completamente el dispositivo y así poder apreciar el efecto del comando Bright). Luego de estas dos acciones se puede enviar un comando Bright. Estas dos acciones son necesarias debido a que el “Lamp Module” fue diseñado para ejecutar instrucciones de Bright solamente después de que se haya ejecutado un comando Dim. Los resultados de las pruebas se los resume en la Tabla 3.9.

Tabla 3.9. Resultados de las pruebas del comando Bright

Cód.C.	Cód. F.	1 <sup>er</sup> Byte	Com. micro.	Cód. D.	2 <sup>do</sup> Byte	3 <sup>er</sup> Byte	Resultado
A	Bright	94	Ejec. X-10	2	60	2	Disp. A2 Iluminado
A	Bright	94	Ejec. X-10	2	60	4	Disp. A2 Iluminado
A	Bright	94	Ejec. X-11	2	60	6	Disp. A2 Iluminado
A	Bright	94	Ejec. X-12	2	60	8	Disp. A2 Iluminado
A	Bright	94	Ejec. X-13	2	60	10	Disp. A2 Iluminado
A	Bright	94	Ejec. X-14	2	60	12	Disp. A2 Iluminado
A	Bright	94	Ejec. X-15	2	60	14	Disp. A2 Iluminado
A	Bright	94	Ejec. X-16	2	60	16	Disp. A2 Iluminado
A	Bright	94	Ejec. X-17	2	60	18	Disp. A2 Iluminado
A	Bright	94	Ejec. X-18	2	60	20	Disp. A2 Encendido

Lastimosamente, no se contó con un luxómetro para realizar estas pruebas. Por éste motivo no se pudo cuantificar el nivel de luminosidad obtenido con cada incremento de dos pasos. Solamente se puede apreciar el incremento de luminosidad para cada prueba. De estos experimentos se pudo determinar que el número de pasos para un comando Dim o Bright es de 20. También se pudo observar que la variación de la luminosidad no es directamente proporcional al número de pasos, lo que sugiere que el “Lamp Module” está diseñado para atenuar o incrementar el voltaje (o el ángulo de disparo de los tiristores) proporcionalmente al número de pasos pero no así la luminosidad, debido a que la luminosidad de una bombilla incandescente no es directamente proporcional al voltaje aplicado a sus terminales.

### **3.1.3 RELOJ EN TIEMPO REAL.**

El reloj en tiempo real es necesario para que el Módulo de Control funcione de manera autónoma. Luego de que los comandos sean escritos en la memoria EEPROM es necesario leer la fecha y hora de ejecución prevista para cada comando y compararla con la hora que marca el reloj interno. Cuando se de la coincidencia entre la fecha y hora de un comando grabado en la memoria EEPROM y la fecha y hora del reloj interno del Módulo de Control, se procederá a ejecutar el comando grabado en memoria. Debido a que los comandos solo se podrán grabar con una resolución de segundos (menores resoluciones no son necesarias para aplicaciones domésticas), la comparación entre fecha y hora de los comandos y la fecha y hora actual se realiza cada segundo.

Con el objetivo de implementar el reloj, primero se realizó un programa para garantizar que todas las transiciones de segundos, minutos, horas días, meses y años (incluidos los bisiestos) se realicen correctamente. El programa desarrollado para este propósito es “Sim Reloj.asm”. El reloj está diseñado para llevar la cuenta del tiempo con una resolución de hasta 5 décimas de segundo. Los registros necesarios para el funcionamiento del reloj son: Año; Año\_aux, utilizado para diferenciar los años bisiestos; mes; día; hora; minuto; segundo y D\_semana, utilizado para registrar el día de la semana correspondiente. El reloj fue calibrado

para tener como fecha inicial las cero horas del primero de enero del 2005, día sábado de un año no bisiesto. La importancia de conocer el día de la semana vinculado a una fecha específica radica en el hecho de que existirán tareas que solo se van a ejecutar de lunes a viernes, y no son necesarias el fin de semana. Por ejemplo, el fin de semana no es necesario madrugar para ir al trabajo, así que una rutina que permita encender una cafetera a las 7 de la mañana tendrá que ser ejecutada más tarde en los fines de semana, dependiendo de los hábitos de cada persona. El diagrama de flujo de la rutina de interrupción que permite actualizar los registros del reloj cada 0,5 s se lo presenta en la Figura 3.11.

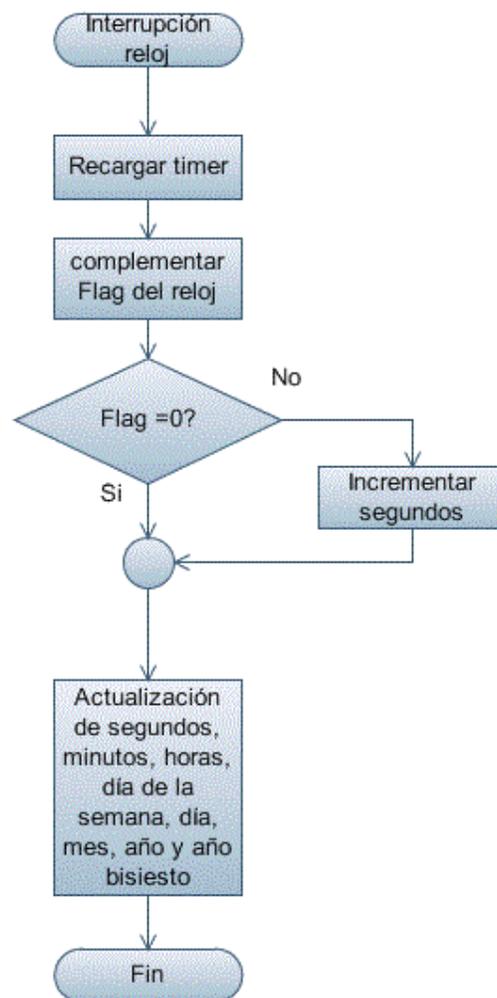


Figura3.11. Diagrama de flujo Reloj

Luego de que se depuró por completo el código de “Sim Reloj.asm” se procedió a integrar este código al programa del Módulo de Control, “Envio codigo X-10\_dim\_z.asm” para obtener el código “Envio codigo X-10\_dim\_ck\_b.asm” . Para el conteo del tiempo se utilizó el timer 1 configurándolo para realizar el conteo de 500ms.

Cada vez que se desborda el timer entra a una rutina de interrupción, la cual actualiza los registros de tiempo. Para verificar el funcionamiento del reloj se utiliza un LED de visualización que parpadea cada segundo. Éste LED permite verificar que los cálculos y la calibración del timer estén bien hechos, y además servirá durante el resto del proceso de programación como indicador de que el microcontrolador está funcionando correctamente.

#### **3.1.4 ESCRITURA EN MEMORIA NO VOLÁTIL.**

Almacenar datos en memoria no volátil es indispensable para permitir que el Módulo de Control pueda actuar de manera autónoma, para que pueda realizar tareas rutinarias sin la necesidad de la intervención humana. Ésta rutina debe permitir escribir cualquier dato en la siguiente posición no usada de memoria. También debe ser capaz de permitir sobrescribir datos antiguos para poder borrar comandos que ya no son necesarios.

Para cumplir con estas tareas se emplean dos bytes para escribir desde la computadora hacia el Módulo de Control: 1 byte de comando y un byte de datos. Así, el byte de datos es escrito en memoria de la manera en que el byte de comando lo indique. La rutina desarrollada para hacer este trabajo es “Envio codigo X10\_dim\_wr\_b.asm” y los Comandos de Microprocesador correspondientes a la escritura en la memoria EEPROM se resumen en la Tabla 3.11.

Tabla 3.10. Comandos de Microprocesador utilizados en el modo de escritura en memoria EEPROM

Comando de Microprocesador	Equivalente decimal	x 16
Ejecutar comando X-10	0	0
Escribir desde inicio (Escritura EEPROM)	1	16
Fin de programación (Escritura EEPROM)	2	32
Igualar Reloj	4	64
Escribir comando en EEPROM	8	128

El comando “Escribir desde el inicio” y el comando “Fin de programación” son los dos comandos nuevos que se introducen para el modo de escritura en memoria EEPROM.

El comando “Escribir desde el inicio” sirve para reescribir desde el inicio la memoria EEPROM mientras que el comando “Fin de programación” se lo utiliza para detener el proceso de programación, volver al modo normal de recepción de comandos y permitir la lectura de la memoria EEPROM.

Es importante detener el proceso de lectura mientras se escriban nuevos comandos, porque la lectura de un comando incompleto puede provocar un malfuncionamiento del programa.

El último dato enviado debe ser el número 255, el cual corresponde al byte de fin de archivo (EOF). El diagrama de flujo del algoritmo para la escritura en memoria EEPROM se lo puede observar en la Figura 3.12.

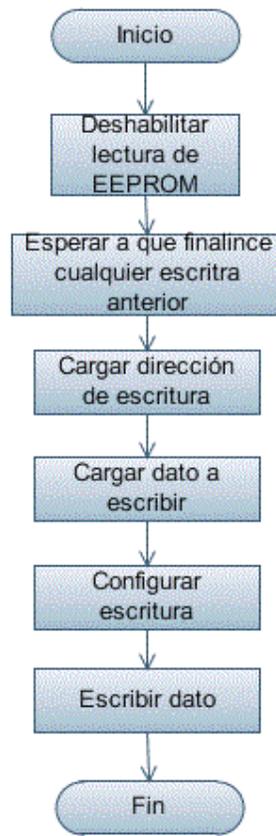


Figura 3.12 Diagrama de flujo del algoritmo de escritura en EEPROM

Para permitir que la rutina de interrupción para comunicación serial identifique que el dato enviado no es un comando de ejecución en tiempo real, sino de que se trata de un comando para escribir en memoria EEPROM, es necesario hacer modificaciones en la recepción del segundo Byte. Estas modificaciones se explican a continuación en lenguaje estructurado.

#### Recibo Datos

Si	dato recibido es Tercer o Cuarto Byte
Si	dato recibido es Tercer Byte
	Preparar recepción del Primer Byte
	Guardar Tercer byte en Contador Dim
	Borrar Flag de Cambio de estado
	Habilitar interrupción por cambio de estado
Caso contrario	dato recibido es el Primer o Segundo Byte
Si	dato recibido es Primer Byte
	Preparar recepción del Segundo Byte

```

                                Guardar Primer Byte
Caso contrario  dato recibido es Segundo Byte
                                Preparar recepción del Primer Byte
                                Guardar Segundo Byte

                                Si      dato recibido contiene Flag de escritura EEPROM
                                Ir a rutina PROGRAMAR

                                Borrar Flag de Cambio de estado
                                Habilitar interrupción por cambio de estado

                                Si      dato recibido corresponde a comando Dim o Bright
                                Preparar recepción del Tercer Byte
                                Preparar transmisión continua para Dim o Bright

```

**Fin Tarea****PROGRAMAR**

```

Si      Flag para igualación de reloj está activa
        Ir a rutina de Igualación de Reloj12
Preparar dato recibido para Escritura EEPROM
Setear Flags para Escritura EEPROM
Si      Flag de comando Fin de escritura está activo
        Activar Flag de Fin de escritura EEPROM

```

**Fin Tarea**

El programa “Envio codigo X10\_dim\_wr\_b.asm” permite discriminar si el comando que se envió desde la computadora es un comando para ejecutarse en tiempo real, o si se tratan de datos para la programación de comandos que se ejecutarán posteriormente (comando de escritura en memoria EEPROM). El primer byte enviado desde la computadora es el byte de datos, el cual contiene la información que será escrita en la memoria EEPROM del Módulo de Control. El segundo byte corresponde a un comando que le indicará al Módulo de Control en que posición de la memoria EEPROM debe escribirse el byte de datos. Gracias a éste algoritmo la escritura de cada byte es completamente independiente, lo cual permite una programación mucho mas fácil desde la computadora y además facilita las futuras evoluciones del programa de la computadora ya que no será necesario realizar cambios en el software del microprocesador.

---

<sup>12</sup> La rutina de Igualación de reloj se la dejó vacía porque se la desarrollará en el numeral 3.1.6, solamente se dejó preparado el programa para recibirla

Para verificar que el programa de escritura esté funcionando correctamente se realizaron varias pruebas. Utilizando el programa “Envío 1 dato.VI” se envió un número cualquiera seguido por el Byte de comando correspondiente a la escritura. Se escribieron varios datos y luego se los leyó de la memoria EEPROM del microcontrolador mediante un programador. Los resultados de estas pruebas se los resume en la Tabla 3.11.

Tabla 3.11. Pruebas de escritura en memoria EEPROM

1er Byte	Com. micro.	2do Byte	Memoria EEPROM	Equivalente
1	Wr / inicio	144	01	1
2	Write	128	02	2
4	Write	128	04	4
8	Write	128	08	8
16	Write	128	10	16
32	Write	128	20	32
64	Write	128	40	64
128	Write	128	80	128
255	Write	128	FF	255
0	Write	128	00	0

Como se puede observar en la Tabla 3.11, los datos escritos en la memoria EEPROM del microcontrolador fueron los mismos que se enviaron desde la computadora. De esta manera se comprueba que el algoritmo de escritura implementado en el PIC esta funcionando apropiadamente. Para verificar que el comando de sobre escritura esté funcionando correctamente se realizó una segunda prueba, cuyos resultados se los resume en la Tabla 3.12.

Este segundo grupo de pruebas se lo realizó sobrescribiendo las cuatro primeras localidades de memoria. El primer comando fue una sobre escritura de la primera posición de memoria, el segundo comando fue idéntico al primero y los tres siguientes comandos fueron de sobre escritura para la segunda, tercera y cuarta posición de memoria.

Tabla 3.12. Pruebas de sobre escritura en memoria EEPROM.

1er Byte	Com. micro.	2do Byte	Memoria EEPROM	Equivalente
0	Wr / inicio	144	00	0
0	Wr / inicio	144	00	0
0	Write	128	00	0
0	Write	128	00	0
0	Write	128	10	16
			20	32
			40	64
			80	128
			FF	255
			00	0

Como se puede apreciar en la Tabla 3.12, se consiguió completamente el objetivo de esta prueba ya que las 4 primeras localidades de memoria se encuentra vacías, mientras que las restantes 10 conservan los valores programados durante la primera practica de escritura en memoria EEPROM.

Con estas dos pruebas se verificó que el algoritmo de escritura trabaja sin ningún error.

### **3.1.5 ALMACENAMIENTO, LECTURA Y EJECUCIÓN DE COMANDOS GUARDADOS EN MEMORIA EEPROM.**

Luego de haber depurado el código para escribir cualquier dato en memoria EEPROM, es necesario crear un protocolo de escritura/lectura de la memoria para organizar la información almacenada. Este algoritmo de almacenamiento de datos permitirá que el microcontrolador pueda decodificar apropiadamente los datos almacenados en su memoria EEPROM, lo cual permitirá que el microcontrolador pueda actuar autónomamente de manera eficiente.

El algoritmo de lectura permitirá que el microcontrolador busque la hora y fecha de cada comando almacenado en su memoria y, de ser tiempo de ejecutarlo, debe permitir que el microcontrolador decodifique apropiadamente el comando y

lo transmita a través de la red eléctrica mediante procedimientos eficientes que no interfieran con las otras funciones del microcontrolador.

### 3.1.5.1 Almacenamiento de datos en memoria EEPROM

Existen dos tipos de comandos que serán almacenados: Comandos repetitivos y comandos no repetitivos. Los comandos repetitivos son comandos que se ejecutarán periódicamente; es decir, cada semana o cada día a una hora determinada. Este tipo de comandos sirven principalmente para ahorro de energía o para automatizar tareas diarias que siempre se ejecutan a la misma hora. Por otro lado, los comandos no repetitivos son comandos que se ejecutarán una sola vez. Gracias a este tipo de comandos se pueden automatizar eventos que salen de la rutina, brindando de esta manera el apoyo de la automatización en la mayor cantidad posible de tareas dentro del hogar.

Para agilizar la lectura de la memoria EEPROM, y poder discriminar entre los dos tipos de comandos, todos los comandos poseerán: Un encabezado, un segundo grupo de bytes que contengan la información temporal (Registros de tiempo) y un tercer grupo de bytes en los cuales se guardará la información correspondiente al comando X-10 que se tiene que ejecutar (Registros de comandos). La Tabla 3.13 muestra la distribución de bytes de los comandos repetitivos y no repetitivos.

Tabla 3.13. Distribución de bytes para el almacenamiento en memoria EEPROM

Número de byte	Com. Repetitivo	Com. No Repetitivo
1	Encabezado	Encabezado
2	Mes	D_Semana
3	Día	Hora
4	Hora	Minuto
5	Minuto	Segundo
6	Segundo	Comando
7	Comando	Dirección
8	Dirección	Contador Dim
9	Contador Dim	

El espacio que ocupe cada comando en la memoria EEPROM depende del tipo de comandos del que se trate. Se decidió implementar ésta manera de almacenar comandos en memoria no volátil para aprovechar al máximo la memoria disponible.

El encabezado contendrá información sobre el tipo de comando que se está procesando, así como del número de bytes que dicho comando ocupa en la memoria. De esta manera, al leer un comando se puede saber que tipo de información se va a encontrar. También permitirá saltar al siguiente comando si se determina que todavía no es el momento de ejecutar el comando que el microcontrolador está leyendo. Así, se puede ahorrar tiempo y leer la memoria de una manera más rápida. Se puede apreciar que estos saltos son convenientes, si se considera el hecho de que solo se va a ejecutar un comando durante cada ciclo de lectura. Gracias a este modo de lectura se puede reducir el tiempo hasta el 25% del tiempo que se tardaría si se leen todos los datos. Tomado en consideración lo expuesto, la distribución de los bits del encabezado de cada comando almacenado en la memoria EEPROM es la siguiente:

Bit 7 = 1 → Comando repetitivo  
 = 0 → Comando no repetitivo

Bit 6 = 0 → debe ser 0 para que sea un encabezado válido (el número 255 en decimal o FF en hexadecimal es el bit de Fin de archivo o EOF)

Bit 5-0 → equivalente en binario a la extensión en bytes del comando

Por ejemplo, el encabezado para un comando repetitivo cuya extensión es de 8 bytes tendrá la siguiente forma:

Bit	7	6	5	4	3	2	1	0
	1	0	0	0	1	0	0	0

Luego del encabezado viene el grupo de bytes que permiten identificar el tiempo adecuado para ejecutar cada comando (Registros de tiempo). Ésta información depende del tipo de comando del que se trate. Para el caso de los comandos repetitivos se utilizan 4 bytes para guardar la información temporal: día de la semana, hora, minuto y segundo. En el caso de comandos no repetitivos se utilizan 5 bytes: mes, día, hora, minuto y segundo. Para el caso de los comandos repetitivos, el registro D\_semana (día de la semana) posee opciones adicionales:

Bit 7 = 1 → Todos los días  
 = 0 → Desactivado

Bit 6 = 1 → Modo repetición por días hábiles / no hábiles  
 = 0 → Modo repetición por días

Si bit 6 = 1

Bit 5 = 1 → Ejecutar fin de semana  
 = 0 → Ejecutar entre semana

Bits 2-0 no importan

Si bit 6 = 0, bit 5 no importa

Bits 2-0:

Bit			Equivalencia
2	1	0	
0	0	0	Domingo
0	0	1	Lunes
0	1	0	Martes
0	1	1	Miércoles
1	0	0	Jueves
1	0	1	Viernes
1	1	0	Sábado

El último grupo de bytes (Registros de comandos) se refieren al comando X-10 que se requiere ejecutar y esta conformado por dos bytes para el caso de todos los comandos X-10, a excepción de los comandos Dim y Bright que requieren tres bytes. Éste tercer grupo de Bytes esta comprendido por el Byte de "Comando" seguido por el byte de "Dirección" como se puede apreciar en la Tabla 3.13. El

tercer byte solo se envía para comandos Dim o Bright y corresponde a un “Contador Dim” que indica el número de pasos Dim o Bright que se deben ejecutar.

Este grupo de bytes tienen un formato similar a los bytes que se reciben desde la computadora cuando se envían comandos que van a ser ejecutados en tiempo real por el microcontrolador. El byte “Comando” contiene el Código de dirección en sus 5 bits menos significativos y la flag de comando Dim/Bright en el quinto bit. El byte “Dirección” contiene la dirección de casa en los 4 bits menos significativos y el código de función en los 4 bits más significativos. La similitud entre los datos que se reciben de la PC, cuando un comando debe ser ejecutado en tiempo real, y los comandos almacenados en la memoria EEPROM es necesaria, para que se pueda reconstruir la trama X-10 utilizando el mismo algoritmo y de esta manera poder utilizar el mismo módulo de transmisión de comandos X-10.

### **3.1.5.2 Algoritmo de lectura y procedimientos de ejecución de comandos almacenados en memoria EEPROM**

El algoritmo de lectura es el encargado de buscar un dato en la memoria EEPROM y compararlo con el registro correspondiente del reloj. En caso de que se produzca una coincidencia, el algoritmo de lectura procede a realizar la comparación del siguiente registro hasta llegar a la comparación de los segundos. Si luego de realizar la comparación de los segundos se llega a una coincidencia el algoritmo debe permitir la ejecución del comando X-10 almacenado en memoria. El algoritmo de almacenamiento definido anteriormente requiere que el primer byte de cada comando almacenado sea un encabezado que identifique el tipo de comando y la extensión de dicho comando. Por este motivo, el algoritmo de lectura debe comenzar la lectura buscando un encabezado en la primera localidad de memoria EEPROM. El diagrama de flujo del algoritmo de lectura de la memoria EEPROM se encuentra en la Figura 3.13.

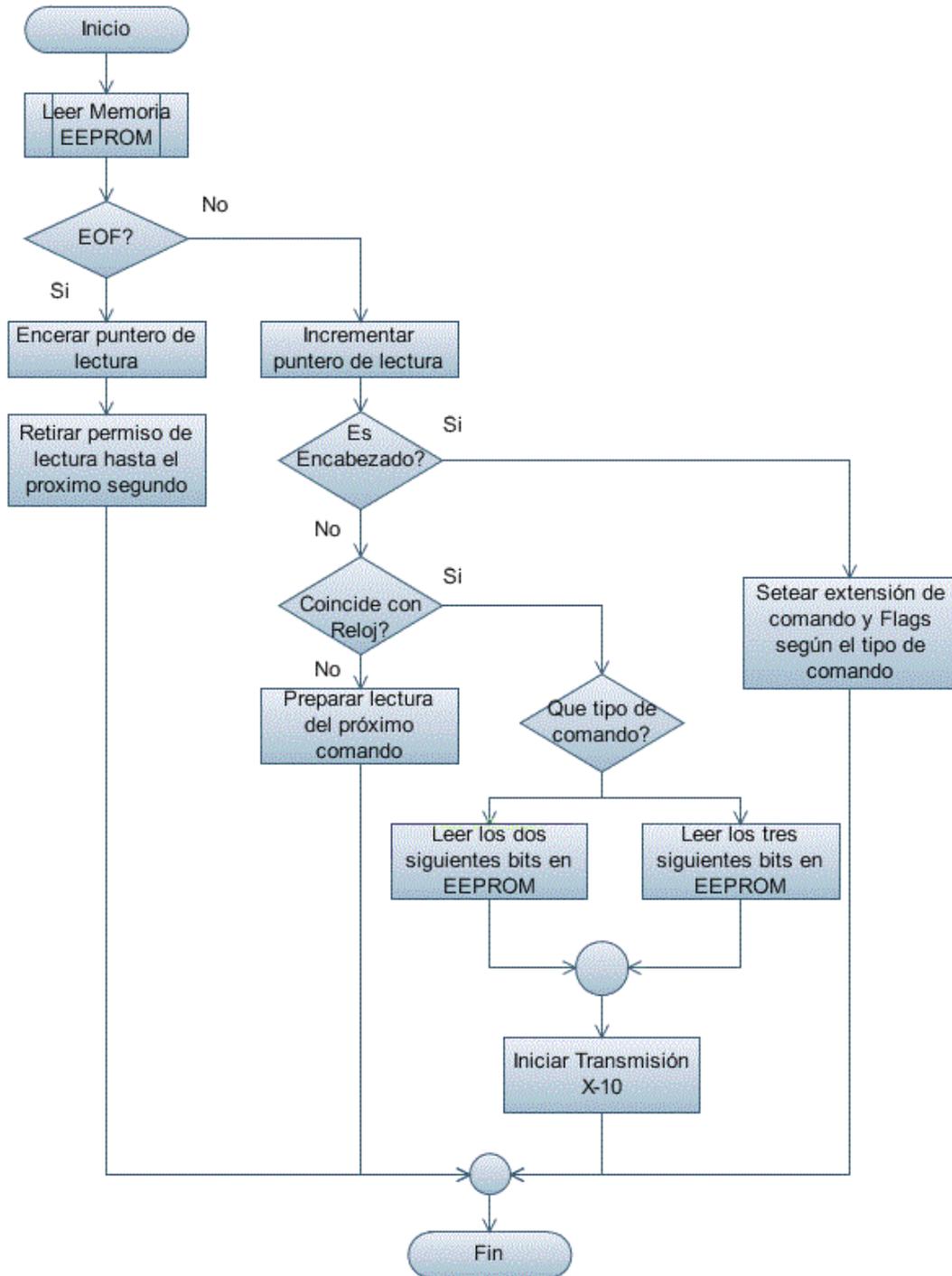


Figura 3.13. Algoritmo de lectura de memoria EEPROM.

La rutina para leer memoria EEPROM se la explica a continuación en lenguaje estructurado

**Leer memoria EEPROM**

- Seleccionar Banco 2
- Seleccionar Dirección de memoria para lectura
- Seleccionar Banco 3
- Iniciar lectura
- Seleccionar Banco 2
- Guardar dato

**Fin Tarea**

Para verificar la validez del encabezado, el programa revisa si el valor obtenido de la lectura de la memoria EEPROM es distinto de FF hexadecimal. Como en un encabezado válido el sexto bit es cero, cualquier encabezado válido será diferente de FF hexadecimal. Cuando la búsqueda de un encabezado encuentra el número FF hexadecimal significa que se ha llegado al final del archivo (fin de la memoria EEPROM programada) y por lo tanto se procede a encerrar el puntero de lectura de la memoria EEPROM, para comenzar una nueva lectura en cuanto ocurra un cambio de segundos del reloj interno.

En el caso de identificar un encabezado válido, el siguiente paso es identificar el tipo de comando del que se trata y configurar el sistema de lectura de acuerdo al tipo de comando que se va a leer y se procede a apuntar a la siguiente localidad de memoria EEPROM.

Luego de identificado el tipo de comando se procede a hacer la primera comparación entre el tiempo almacenado en memoria y su correspondiente del reloj. Si los registros no coinciden, el puntero de lectura de memoria EEPROM saltará a la localidad de memoria señalada para el encabezado del siguiente comando y se repetirá el procedimiento de lectura.

En el caso de que exista una coincidencia se incrementará el puntero de lectura y se procederá a realizar la siguiente comparación. En el momento en que se llegue a una coincidencia de los registros de segundos, el programa autorizará la ejecución de los comandos X-10 según sea conveniente para el tipo de comando que se está leyendo en ese momento.

Para la ejecución del comando X-10 se leen los registros correspondientes y se los copia a los registros que forman parte del módulo de transmisión de comandos X-10. Para este último proceso es necesario discriminar si se trata de un comando Dim/Bright, o de un comando normal, para saber si el comando ocupará dos o tres bytes. Luego se alza la bandera que activa la sincronización de la transmisión de comandos X-10 con la red eléctrica y se inicia la transmisión del comando. Una vez terminada la transmisión se vuelve a la lectura de la memoria EEPROM, repitiéndose todo el proceso anterior.

### **3.1.6 ALGORITMO DE IGUALACIÓN DEL RELOJ**

El último algoritmo a implementar es el algoritmo de igualación del reloj. Para ello se insertó en el código de programación de memoria EEPROM una rutina que permite discriminar si se trata de un comando de escritura en memoria no volátil o si es un comando para igualar el reloj. Si se determina que el comando recibido es para igualar el reloj, el programa determina los registros de tiempo que se van a igualar con los datos recibidos y copia la información actualizada en dichos registros. El primer byte recibido lo copia tal como fue recibido, pero el segundo byte requiere que se aplique una máscara para eliminar los comandos correspondientes a la igualación del reloj. El diagrama de flujo que describe el funcionamiento del algoritmo de igualación de reloj se lo muestra en la Figura 3.14.

La selección del grupo de registros que se requiere igualar se utilizó un contador y una tabla. Luego de identificado el grupo de registros que se van a igualar se copia los datos recibidos desde la PC en los registros correspondientes del reloj. Para ello se copia el primer byte recibido en el registro del reloj correspondiente al año, y el segundo byte recibido se lo copia en el registro correspondiente a la cuenta de años bisiestos. Luego se incrementa el puntero que permite igualar todos los registros del reloj y se espera los siguientes dos bytes.

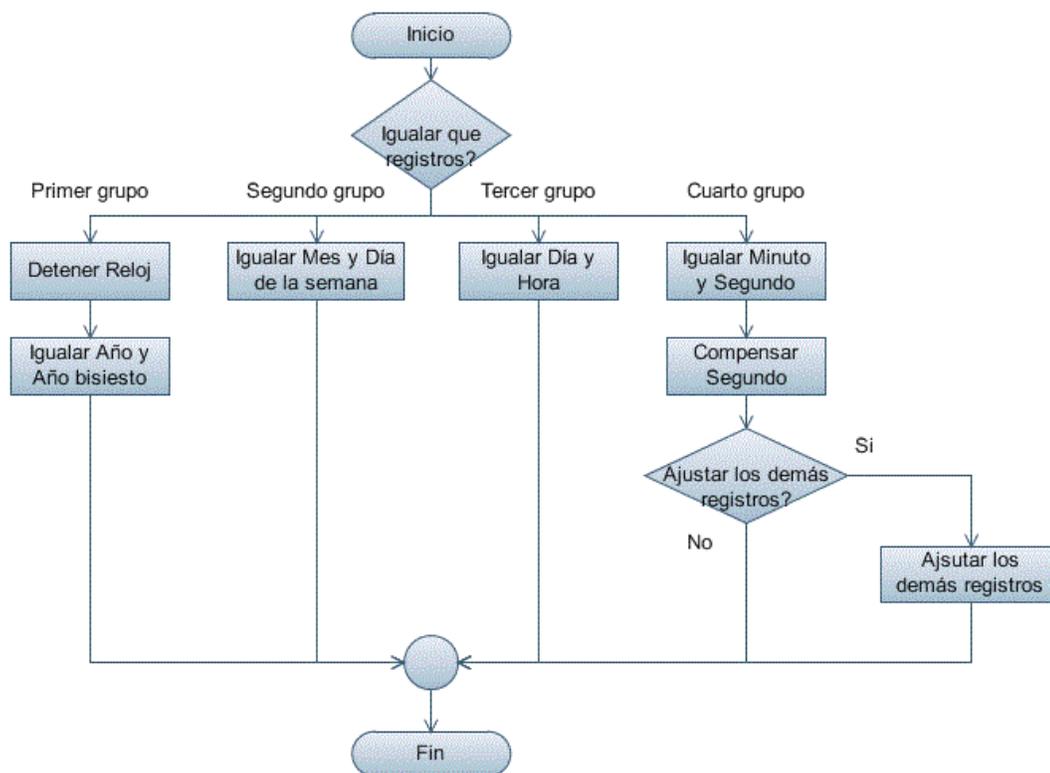


Figura 3.14. Algoritmo de igualación de reloj

El primer byte del siguiente grupo recibido se copiará al registro de meses, y el segundo byte se copiará al registro de día de la semana. El primer Byte del tercer grupo será copiado en el registro del día, y el segundo byte se copiará en el registro de hora. Finalmente, luego de recibir el cuarto grupo de bytes se copiará el primer byte al registro de minutos, y el segundo byte al registro de segundos. Para compensar el retardo introducido por la comunicación serial es necesario sumar 2 segundos al registro de segundos. Luego de esto, en caso de que sea necesario, se procede a ajustar los demás registros de la misma manera en que se actualizan los registros cuando un cambio de segundos ocurre en el reloj en tiempo real, y finalmente se sale de la rutina.

Éste código se encuentra desarrollado en el archivo “envio codigo X-10\_dim\_rd1com\_h.asm”. Luego de que se termino la depuración e integración de todos los módulos se obtuvo el código de “Tesis JAS\_final.asm”, el cual permite el funcionamiento del Módulo de Control con todas las funciones programadas.

### 3.1.7 PRUEBAS DE FUNCIONAMIENTO DEL MÓDULO DE CONTROL

Luego de que se depuró por completo éste código y que se logró la integración con todos los demás módulos, se procedió a realizar varias pruebas para verificar que el programa del Módulo de Control funcione correctamente. Las pruebas consistieron en escribir byte por byte varios comandos en la memoria EEPROM utilizando un programador de microcontroladores. Luego se utilizó el programa “enviar 1 dato.VI” para igualar el reloj del microcontrolador. Las pruebas se las realizó tanto con comandos repetitivos como con comandos no repetitivos.

#### Prueba 1: Comandos no repetitivos

En esta prueba se busca primeramente comprobar el funcionamiento del algoritmo de lectura solamente. En una segunda parte de la prueba se buscará verificar el funcionamiento de la rutina para igualar el reloj interno del Módulo de Control y verificar que la lectura y ejecución de comandos repetitivos se esté realizando correctamente. El primer paso para esta prueba es programar la memoria EEPROM del microcontrolador con dos comandos no repetitivos. Estos comandos se presentan en la Tabla 2.14.

Tabla 3.14. Comandos programados en memoria EEPROM

	Byte	Dato decimal
Comando 1	Encabezado	07
	Mes	01
	Día	01
	Hora	00
	Minuto	00
	Segundo	05
	Comando	12
	Dirección	38
Comando 2	Encabezado	07
	Mes	01
	Día	01
	Hora	00
	Minuto	00
	Segundo	10
	Comando	12
	Dirección	54

El primer comando programado ordena que se encienda el dispositivo A1, el primero de enero a las 00h00 con 5 segundos. El segundo comando ordena que se apague el dispositivo A1, el primero de enero a las 00h00 con 10 segundos.

Luego de programados los comandos se enciende el Módulo de Control para observar los resultados. No es necesario igualar el reloj ya que la hora por defecto con la que se enciende el reloj son las 00h00, 0 segundos del primero de enero del 2005 (día sábado).

El primer resultado de prueba 1 fue que luego de 5 segundos de encendido el Módulo de Control se enciende el dispositivo con la dirección A1 y 5 segundos más tarde el mismo dispositivo se apaga automáticamente. Por lo tanto se puede afirmar que la prueba se realizó satisfactoriamente.

Para verificar el resultado se iguala el reloj del Módulo de Control a través del programa "Envío 1 dato.VI" a las 00h00, 0 segundos del primero de enero. De igual manera que en caso anterior, luego de 5 segundos de igualada la hora del Módulo de Control se enciende el dispositivo con la dirección A1 y 5 segundos más tarde el mismo dispositivo se apaga automáticamente.

La conclusión obtenida de la prueba 1 es que el algoritmo de lectura para comandos no repetitivos trabaja apropiadamente, y que el algoritmo para igualar la hora del reloj del Módulo de Control también funciona correctamente.

Prueba 2: Comandos repetitivos.

En esta prueba se busca comprobar que el algoritmo de lectura esté funcionando correctamente cuando sea necesario leer comandos repetitivos.

De igual manera que para la prueba anterior, lo primero que se debe hacer es programar los comando repetitivos en la memoria EEPROM del microcontrolador. Estos comandos se presentan en la Tabla 2.15.

Tabla 3.15. Comandos programados en memoria EEPROM

	Byte	Dato decimal
Comando 1	Encabezado	134
	D_semana	64
	Hora	22
	Minuto	13
	Segundo	00
	Comando	12
	Dirección	38
Comando 2	Encabezado	134
	D_semana	96
	Hora	12
	Minuto	15
	Segundo	00
	Comando	12
	Dirección	38

El primer comando programado ordena encender el dispositivo A1 a las 22h13, 0 segundos de todos los días entre semana (lunes, martes, miércoles, jueves, y viernes).

El segundo comando programado ordena apagar el dispositivo A1 a las 12h15, 0 segundos de todos los días del fin de semana (sábado y domingo).

Luego de programados estos comando se enciende el Módulo de Control y se iguala la hora del reloj interno utilizando el mismo procedimiento utilizado para la prueba 1. La hora que se va a enviar al Módulo de Control son las 22h12, 50 segundos del 3 de enero del 2005 (lunes).

El primer resultado de la prueba 2 fue que el Módulo de Control encendió el módulo A1 10 segundos después de que se igualó el reloj interno.

Para verificar el correcto funcionamiento del Módulo de Control cuando se programan comandos repetitivos, se igualó el reloj interno del Módulo de Control enviando una nueva hora: 12h14, 45 segundos, del 1 de enero del 2005 (sábado).

El resultado fue que el dispositivo A1 se apagó 15 segundos después de que la rutina para igualar la hora se ejecutó.

La conclusión obtenida de la prueba 2 es que el algoritmo de lectura funciona adecuadamente con comandos repetitivos, tanto para fines de semana como para días de trabajo. Adicionalmente se verificó el funcionamiento de la rutina para igualar el reloj interno del Módulo de Control.

Prueba 3: Secuencia de eventos y prueba de comandos Dim y Bright.

Esta es la prueba final para verificar que todas las funciones de la lectura de memoria EEPROM estén funcionando correctamente.

Para la realización esta prueba se ejecutara una secuencia de 4 comandos: Encender, Dim, Bright y Apagar.

Los comandos que se tiene que programar en la memoria EEPROM del microcontrolador se presentan en la Tabla 3.16.

El primer comando programado ordena encender el dispositivo A2 a las 12h15, 10 segundos de cualquier fin de semana. El segundo comando ordena reducir la intensidad luminosa (comando Dim) del dispositivo A2 hasta el 50% a las 12h15, 11 segundos de cualquier fin de semana. El tercer comando incrementa la luminosidad del dispositivo A2 hasta el 100% a las 12h15, 20 segundos de cualquier fin de semana. Finalmente, el cuarto comando apaga el dispositivo A2 a las 12h15, 25 segundos de cualquier fin de semana.

Para la ejecución de la prueba se iguala el reloj del Módulo de Control a las 12h15, 0 segundos del 1 de enero de 2005 (sábado).

Tabla 3.16. Comandos programados en memoria EEPROM

	Byte	Dato decimal
Comando 1	Encabezado	134
	D_semana	96
	Hora	12
	Minuto	15
	Segundo	10
	Comando	28
	Dirección	38
Comando 2	Encabezado	135
	D_semana	96
	Hora	12
	Minuto	15
	Segundo	12
	Comando	60
	Dirección	70
Comando 3	Cont_Dim	10
	Encabezado	135
	D_semana	96
	Hora	12
	Minuto	15
	Segundo	20
	Comando	60
	Dirección	86
Comando 4	Cont_Dim	10
	Encabezado	134
	D_semana	96
	Hora	12
	Minuto	15
	Segundo	25
	Comando	28
	Dirección	54

El resultado de esta prueba fue que a los 10 segundos de haberse igualado el reloj del Módulo de Control se encendió el módulo A2 y casi de inmediato se redujo su intensidad luminosa a la mitad. A las 12h15 con 20 segundos se volvió a incrementar la intensidad luminosa hasta llegar al máximo y 5 segundos mas tarde se apagó.

Para verificar los resultados se volvió a igualar el reloj del Módulo de Control y se observaron los mismos resultados.

La conclusión de esta prueba fue que el algoritmo de lectura está funcionando correctamente para secuencias pequeñas.

Para verificar que todos los componentes diseñados hasta el momento no presenten conflictos se repitieron las pruebas de los numerales 3.1.2; 3.1.4 y la prueba 3 del numeral 3.1.5.2 de manera consecutiva. Debido a que los resultados fueron los mismos que se obtuvieron cuando se realizó las pruebas por primera vez, en cada unos de los numerales citados, se puede concluir que el funcionamiento de la programación del Módulo de Control es satisfactoria. Para hacer una comprobación exhaustiva de todos los componentes de la programación del Módulo de Control se realizarán pruebas adicionales, las cuales serán ejecutadas luego de que todos los componentes del proyecto esten completamente operativos. Estas pruebas y resultados serán expuestos en el capítulo 4 de éste proyecto de titulación.

### **3.2 SOFTWARE PARA LA COMPUTADORA**

El software desarrollado para la computadora es la interfaz de usuario con el Módulo de Control. En consecuencia, es necesario que éste programa posea la capacidad de brindar todas las herramientas necesarias para que el usuario pueda manejar el Módulo de Control de una manera rápida y sencilla. Esta interfaz debe permitir que el usuario pueda controlar fácilmente todos los dispositivos instalados en cada uno de los diferentes ambientes del hogar. Debe permitir una fácil configuración de los nuevos componentes que se instalen en el sistema. Debe ser capaz de igualar el reloj del Módulo de Control de manera automática. Debe permitir una programación sencilla del Módulo de Control. Debe permitir el acceso a través de Internet a todas las funciones del Módulo de Control.

Para que el proceso de programación sea más rápido y flexible se desarrollarán componentes básicos de carácter genérico (Sub VIs) que se convertirán en las herramientas que la interfaz de usuario utilizará para realizar sus tareas. La herramienta básica para la transmisión de datos ya fue desarrollada durante el proceso de elaboración del software para el Módulo de Control. El código de "Envío 1 dato.VI" es el VI básico que permite la comunicación con el Módulo de Control bajo los parámetros establecidos en el Numeral 3.1.5.

El programa para la PC se lo realizó en LabView y está dividido por ambientes. Los ambientes están separados en distintas pestañas y cada ambiente tiene asignado un código de casa. Aparte de las pestañas correspondientes a cada ambiente, hay tres pestañas más que sirven para programar comandos en memoria no volátil. Las dos primeras pestañas de programación sirven para programar cualquier comando, mientras que la última pestaña sirve para ejecutar las rutinas de ahorro de energía.

La conexión al panel de control a través de Internet utiliza el administrador para conexión remota al panel de control o en inglés "Remote Panel Connection

Manager” . El diagrama de flujo para el Panel de control principal se lo muestra en la Figura 3.15.

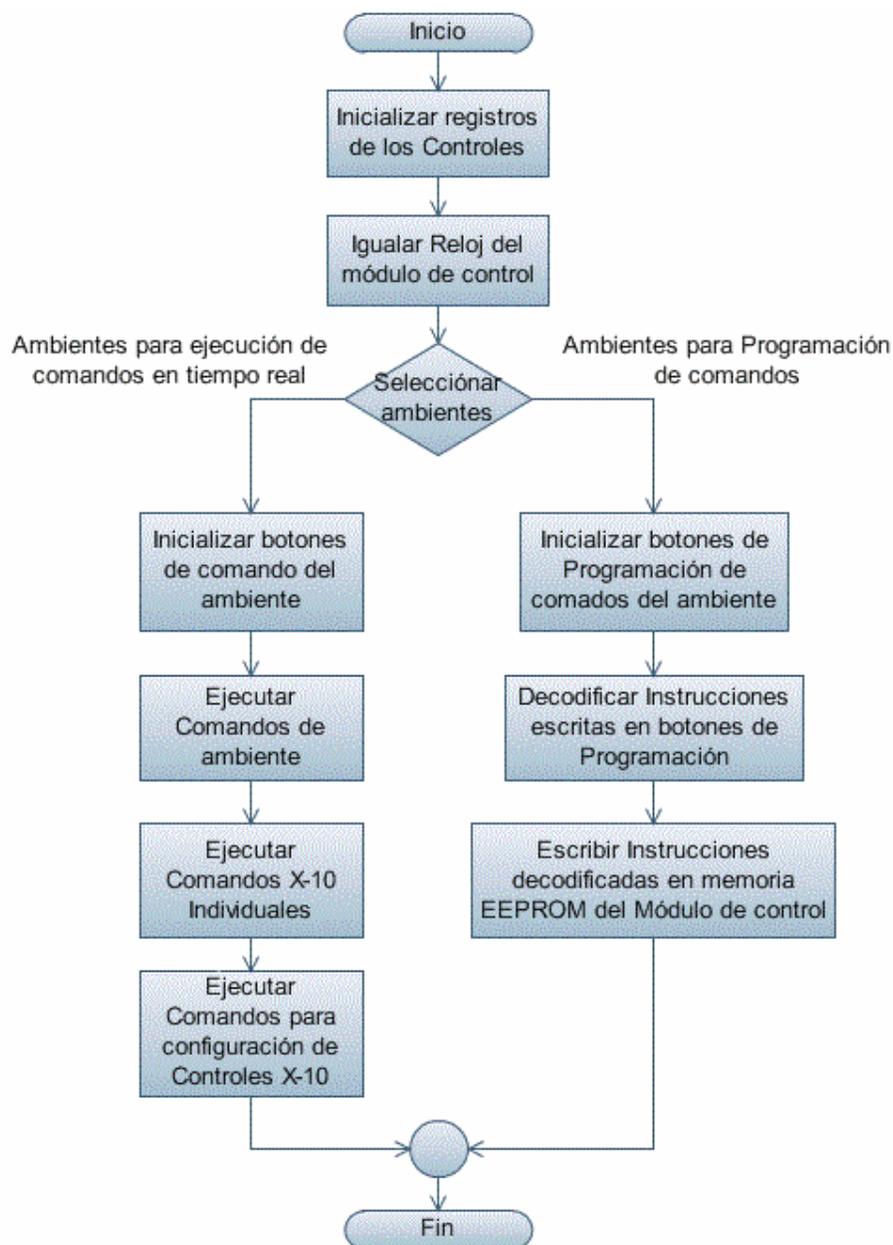


Figura 3.15. Diagrama de flujo de la Interfaz del usuario

### 3.2.1 SUBVI'S PARA ENVÍO DE COMANDOS X-10

Se pensó en crear un subVI para cada tipo de comando que se fuera a enviar al Módulo de Control, para facilitar la construcción del Panel de control principal. De esta manera se tendrá el menor número de elementos en el diagrama de dicho

panel, lo cual permitirá optimizar recursos como espacio ocupado en disco y tiempo de programación. Adicionalmente, la programación que se encuentre en el diagrama del panel de control será más fácil de entender para que en el futuro se puedan actualizar las funciones del presente proyecto.

### **3.2.1.1 Comandos de ambiente**

Los comandos de ambiente son los encargados de encender o apagar todos los dispositivos que tengan el mismo Código de casa; es decir, todos los dispositivos que se encuentren en un mismo ambiente. Estos comandos son: “Encender Todas las Luces”, “Apagar Todas las Luces” y “Apagar Todo”.

#### *3.2.1.1.1 Comando Encender Todas las Luces*

Éste comando permite encender todos los actuadores X-10 diseñados para manejar iluminación y cuya dirección de casa sea la misma dirección de casa que la enviada en el comando. El diagrama de flujo del comando “Encender Todas las Luces” se muestra en la Figura 3.16.

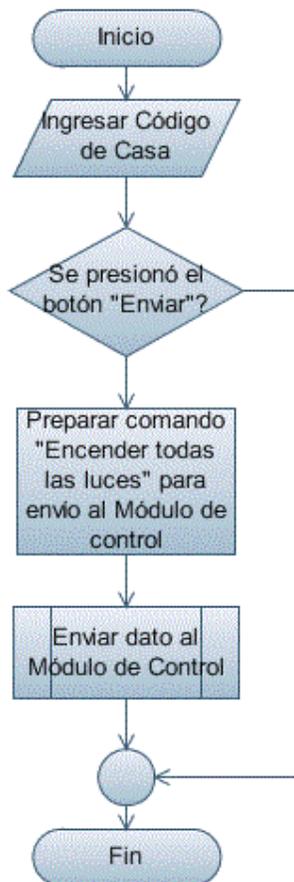


Figura 3.16. Diagrama de flujo del comando “Encender Todas las Luces”

El algoritmo para preparar el comando “Encender Todas las Luces” para ser enviado al Módulo de Control se lo explica a continuación en lenguaje estructurado.

**Ingresar Código de Casa**

Leer dato ingresado por el usuario

**Fin Tarea**

**Preparar comando “Encender Todas las Luces”**

Ejecutar subVI “Codificador Alfa.VI”

Sumar número 16 al resultado de Codificador Alfa

**Fin Tarea**

**Enviar dato al Módulo de Control**

Ejecutar subVI “Envío 1 dato.VI”

**Fin Tarea**

Éste comando junto con los comandos “Apagar todas las luces” y “Apagar todo” afectan a todos los dispositivos que tengan la misma dirección de casa. El programa diseñado es “Encender Todas las Luces.VI” y es una modificación del subVI genérico “Envío 1 dato.VI”. Debido a que éste comando se aplica sobre todos los dispositivos que tengan la misma dirección de casa, no es necesario incluir el código de dirección.

Para la construcción del diagrama de este subVI fue necesario crear un subVI que permita convertir un carácter alfabético en una dirección de casa X-10. Esta tarea la realiza “Codificador alfa.VI”. Éste programa toma como entrada un dato de tipo String (cadena de caracteres) y lo convierte a su equivalente X-10 decimal según la Tabla 3.17.

Tabla 3.17. Equivalencia entre Código de Casa alfabético, binario y decimal.

Casa	H1	H2	H4	H8	Equivalente decimal
A	0	1	1	0	6
B	1	1	1	0	14
C	0	0	1	0	2
D	1	0	1	0	10
E	0	0	0	1	1
F	1	0	0	1	9
G	0	1	0	1	5
H	1	1	0	1	13
I	0	1	1	1	7
J	1	1	1	1	15
K	0	0	1	1	3
L	1	0	1	1	11
M	0	0	0	0	0
N	1	0	0	0	8
O	0	1	0	0	4
P	1	1	0	0	12

El resultado de la conversión realizada por “Codificador alfa.VI” es un número decimal que ya puede ser usado directamente en cualquier operación matemática necesaria para calcular el dato que se va a enviar al microprocesador.

El equivalente decimal del código de función correspondiente al comando “Encender Todas las Luces” es el número 1. Éste número debe ser multiplicado por 16 para cumplir con lo establecido en 3.1.2. En la Tabla 3.3 se pueden observar las equivalencias de los comandos X-10 más importantes.

Para tener el dato que se debe enviar al microcontrolador es necesario sumar el resultado de “Codificador alfa.VI” con el número 16, correspondiente al código de función “Encender Todas las Luces”. Éste dato se lo envía al microcontrolador mediante el mismo procedimiento empleado por “Envío 1 dato.VI” para que sea ejecutado en tiempo real. El panel de control de “Encender Todas las Luces.VI” se muestra en la Figura 3.17.



Figura 3.17. Panel de control del SubVI “Encender Todas las Luces.VI”

#### 3.2.1.1.2 Comando Apagar Todas las Luces

Éste comando permite apagar todos los actuadores X-10 diseñados para manejar iluminación y cuya dirección de casa sea la misma dirección de casa que la enviada en el comando, afectando a todos los dispositivos que tengan la misma dirección de casa. El diagrama de flujo del algoritmo que permite enviar el comando “Apagar Todas las Luces” al Módulo de Control se lo puede encontrar en la Figura 3.18.

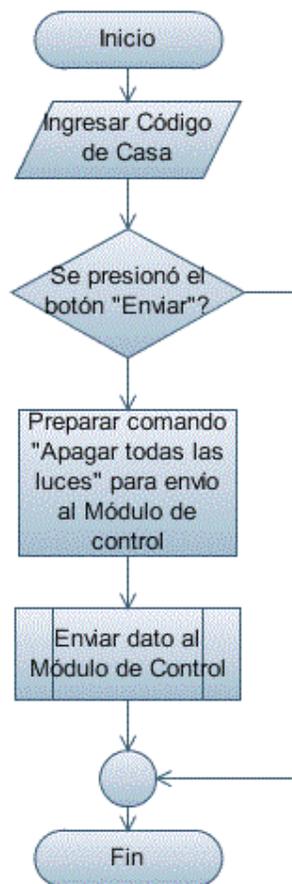


Figura 3.18 Diagrama de flujo del comando “Apagar Todas las Luces”

El programa para preparar el comando “Apagar Todas las Luces” para ser enviado al Módulo de Control se lo explica en seguida en lenguaje estructurado.

**Ingresar Código de Casa**

Leer dato ingresado por el usuario

**Fin Tarea**

**Preparar comando “Apagar Todas las Luces”**

Ejecutar subVI “Codificador Alfa.VI”

Sumar número 96 al resultado de Codificador Alfa

**Fin Tarea**

**Enviar dato al Módulo de Control**

Ejecutar subVI “Envío 1 dato.VI”

**Fin Tarea**

El programa diseñado es “Apagar Todas las Luces.VI”. Del mismo modo que “Encender Todas las Luces.VI”, éste SubVI es una modificación del subVI genérico “Envío 1 dato.VI”. Como este comando solo necesita una Dirección de Casa X-10 en formato decimal, se hace uso del “Codificador alfa.VI” diseñado en el numeral anterior. De igual manera que en el comando “Encender Todas las Luces”, se le suma el equivalente decimal del comando “Apagar Todas las Luces” el cual se puede encontrar en la Tabla 3.3. Luego de realizada dicha suma ya se encuentra listo el dato que debe ser enviado al microcontrolador para la ejecución del comando en tiempo real. El panel de control del subVI “Apagar Todas las Luces” se puede observar en la Figura 3.19

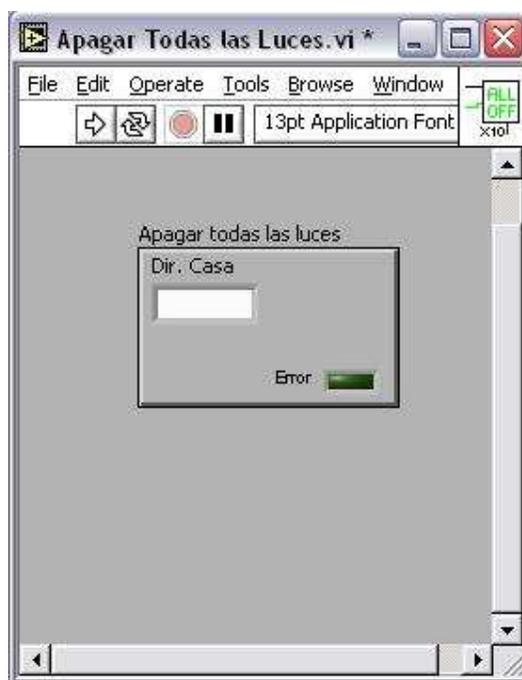


Figura 3.19. Panel de control del SubVI “Apagar Todas las Luces.VI

#### 3.2.1.1.3 Comando Apagar Todo

Éste comando permite apagar todos los actuadores X-10 que se encuentren instalados dentro de un ambiente y cuya dirección de casa sea la misma dirección de casa que la enviada en el comando. Éste comando se ejecuta tanto en los

actuadores de iluminación como en los actuadores de potencia, sin importar si el actuador permite instrucciones Dim o Bright. El diagrama de flujo del algoritmo para enviar el comando “Apagar Todo” al Módulo de Control se lo presenta en la Figura 3.20.

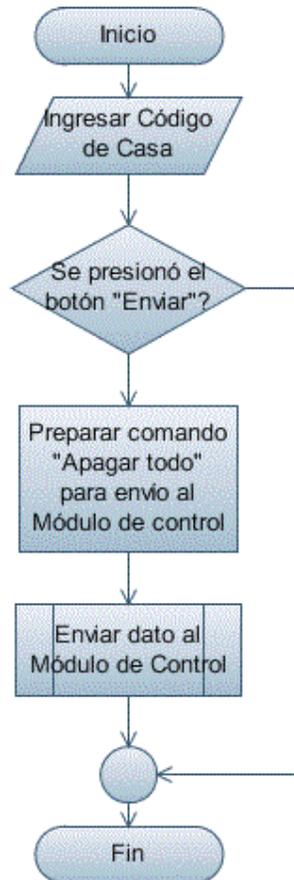


Figura 3.20 Diagrama de flujo del comando “Apagar Todo”

La manera para prepara el comando “Apagar Todo” para ser enviado al Módulo de Control se describe a continuación el lenguaje estructurado.

**Ingresar Código de Casa**

Leer dato ingresado por el usuario

**Fin Tarea**

**Preparar comando “Apagar Todo”**

Ejecutar subVI “Codificador Alfa.VI”

Sumar número 0 al resultado de Codificador Alfa

**Fin Tarea****Enviar dato al Módulo de Control**

Ejecutar subVI "Envío 1 dato.VI"

**Fin Tarea**

El programa diseñado para realizar esta tarea es "Apagar Todo.VI" y al igual que los dos anteriores subV, fue diseñado tomando como base "enviar 1 dato.VI" y también utilizando "Codificador alfa.VI".

Según la Tabla 3.3, el equivalente decimal para el comando "Apagar Todo" es el número 0. Una vez realizada la suma se envía éste dato al microcontrolador para que éste ejecute el comando "Apagar Todo" en tiempo real. El panel de control de "Apagar Todo.VI" se muestra en la Figura 3.21.

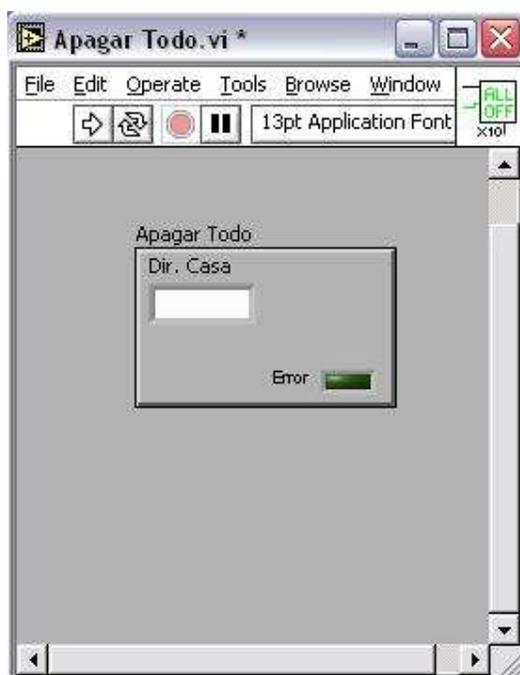


Figura 3.21. Panel de control del SubVI "Apagar Todo.VI"

### 3.2.1.2 Comandos X-10 individuales

Los comandos X-10 individuales son aquellos que solo se ejecutan sobre un dispositivo a la vez. Estos comandos requieren de un Código de Casa y de un Código de dirección.

Los comandos de éste tipo que se necesita implementar para cumplir con los objetivos del presente proyecto son: “Encender”, “Apagar”, “Dim” y “Bright”.

Adicionalmente a los comandos, también se requiere construir subVi's que permitan reducir al máximo el número de controles que se van a presentar en el panel de control. Esto no solo facilita y agiliza el proceso de programación sino que también facilita la comprensión del programa.

Los subVi's encargados de estas tareas son: botón general y generador de pulsos. El “botón general” permite agrupar todos los comandos implementados en este numeral mientras que el “generador de pulsos” permite reducir el número de controles necesarios en la interfaz del usuario. Gracias a éstos dos subVI es posible diseñar una interfaz de usuario con una apariencia similar a los controles de iluminación tradicionales (interruptores mecánicos) y de ésta manera facilitar el manejo del sistema por parte del usuario.

#### *3.2.1.2.1 Comando Encender*

Éste comando permite encender el actuador X-10 cuyo Código de Casa y Código de dirección sea el mismo que el enviado en dicho comando. Éste comando se ejecuta tanto en los actuadores de iluminación como en los actuadores de potencia, sin importar si el actuador permite instrucciones Dim o Bright.

El algoritmo necesario para implementar el comando “Encender” se muestra en la Figura 3.22.

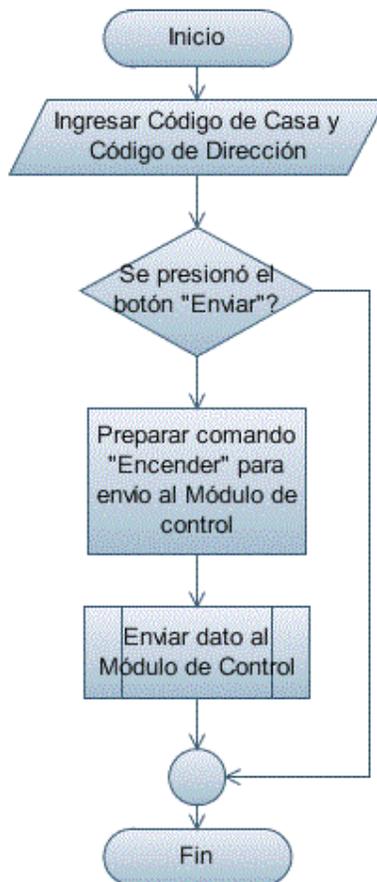


Figura 3.22. Diagrama de flujo del comando “Encender”

El método empleado para preparar el comando “Encender” para ser enviado al Módulo de Control se lo explica con lenguaje estructurado en las siguientes líneas.

**Ingresar Código de Casa**

Leer dato ingresado por el usuario

**Fin Tarea**

**Preparar comando “Encender”**

Ejecutar subVI “Codificador Alfa.VI”

Sumar número 32 al resultado de Codificador Alfa

Seleccionar resultado de la suma como Primer dato a enviar

Ejecutar subVI “Codificador Num.VI”

Seleccionar resultado de Codificador Num como segundo dato a enviar

**Fin Tarea**

**Enviar dato al Módulo de Control**

Ejecutar subVI "Envío 1 dato.VI"

**Fin Tarea**

El programa diseñado para enviar este comando al microprocesador es "Encender Luces.VI". Éste programa utiliza como base para su construcción el programa genérico "enviar 1 dato.VI" y además hace uso del "Codificador alfa.VI" ya diseñado en los numerales anteriores y del "Codificador Num.VI". Éste ultimo subVI funciona de manera similar al "Codificador alfa.VI" pero convierte las direcciones de casa X-10 en su equivalente decimal según la Tabla 3.18

Tabla 3.18. Equivalencia entre Código de Dirección alfabético, binario y decimal.

Dirección	D1	D2	D4	D8	Equivalente decimal
1	0	1	1	0	6
2	1	1	1	0	14
3	0	0	1	0	2
4	1	0	1	0	10
5	0	0	0	1	1
6	1	0	0	1	9
7	0	1	0	1	5
8	1	1	0	1	13
9	0	1	1	1	7
10	1	1	1	1	15
11	0	0	1	1	3
12	1	0	1	1	11
13	0	0	0	0	0
14	1	0	0	0	8
15	0	1	0	0	4
16	1	1	0	0	12

El programa "Encender Luces.VI" tiene que enviar dos datos en lugar del un solo dato que requerían los comandos de ambiente. Para lograr este propósito se debe añadir una estructura de secuencia que permita enviar en primer lugar el byte que contiene el Código de Casa y el código de función (Primer dato) para luego enviar el segundo byte con el código de Dirección y los Comandos de Microprocesador (Segundo dato) de la manera que se estableció en el numeral 3.1.2. El

equivalente decimal del código de función X-10 para el comando “Encender” que se debe sumar al equivalente del Código de Casa es el número 32 como se puede observar en la Tabla 3.3. Debido a que el equivalente decimal del Comando de Microprocesador para ejecutar un comando en tiempo real es cero según la Tabla 3.6, no es necesario sumar nada al equivalente decimal del código de dirección.

El panel de control de “Encender Luces.VI” se lo puede observar en la Figura 3.23.



Figura 3.23. Panel de control del SubVI “Encender Luces.VI”

#### 3.2.1.2.2 Comando Apagar

Éste comando permite apagar el actuador X-10 cuyo Código de Casa y Código dirección de dispositivo sea el mismo que la enviado en dicho comando. Éste comando se ejecuta tanto en los actuadores de iluminación como en los actuadores de potencia, sin importar si el actuador permite instrucciones Dim o

Bright. El diagrama de flujo que explica la manera de enviar éste comando al Módulo de Control se puede observar en la Figura 3.24.

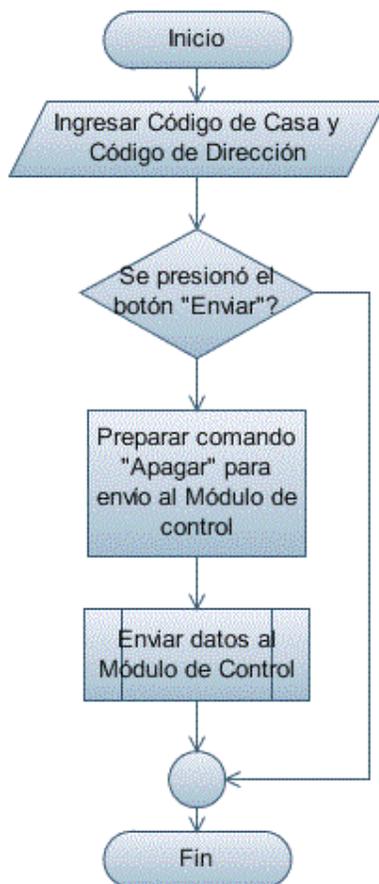


Figura 3.24. Diagrama de flujo del comando "Apagar"

El Algoritmo empleado para preparar el comando "Apagar" para ser enviado al Módulo de Control se lo describe a continuación con lenguaje estructurado.

**Ingresar Código de Casa y Código de Dirección**

Leer datos ingresados por el usuario

**Fin Tarea**

**Preparar comando "Apagar"**

Ejecutar subVI "Codificador Alfa.VI"

Sumar número 48 al resultado de Codificador Alfa

Seleccionar resultado de la suma como Primer dato a enviar

Ejecutar subVI "Codificador Num.VI"

Seleccionar resultado de Codificador Num como segundo dato a enviar

#### Fin Tarea

#### Enviar dato al Módulo de Control

Ejecutar subVI "Envío 1 dato.VI"

#### Fin Tarea

El programa diseñado para enviar este comando al microprocesador es "Apagar Luces.VI". Éste programa utiliza como base para su construcción el programa genérico "enviar 1 dato.VI" y los subVI "Codificador alfa.VI" y "Codificador Num.VI" diseñados anteriormente. De manera similar que para el diseño de "Encender Luces.VI", es necesario añadir una estructura de secuencia ya que para este comando también es necesario enviar un código de dirección. Dicha estructura de secuencia permite enviar como primer byte el código de dirección y el código de función correspondiente al comando "Apagar" (Primer dato), para luego enviar como segundo byte el código de dirección junto con los Comandos de Microprocesador (Segundo dato).

El equivalente decimal del comando "Apagar" es el número 48 y se lo debe sumar al equivalente decimal del código de casa. No se tiene que sumar nada al código de dirección ya que el equivalente decimal del Comando de Microprocesador necesario para transmitir códigos X-10 en tiempo real es 0.

El panel de control de "Apagar Luces.VI" se muestra en la Figura 3.25



Figura 3.25. Panel de control del SubVI “Apagar Luces.VI”

### 3.2.1.2.3 Comandos *Dim* y *Bright*

Éstos comandos permiten atenuar o incrementar el voltaje entregado por el actuador X-10 a la carga conectada a dicho actuador. Para que se ejecute el comando *Dim* es necesario que el dispositivo que se quiere controlar se encuentre previamente encendido y que su dirección de casa y dirección de dispositivo sea la misma que la enviada en el comando X-10.

Para que el comando *Bright* pueda ser ejecutado es necesario que el dispositivo que se requiere controlar esté previamente encendido y que el voltaje aplicado a este dispositivo haya sido atenuado (ejecución previa de un comando *DIM*).

Éstos dos comandos se ejecutan solamente en actuadores que permiten instrucciones *Dim* o *Bright*, generalmente sobre dispositivos para iluminación, aunque su implementación también es posible sobre dispositivos de potencia.

El programa diseñado para enviar este comando al microprocesador es “*Dim subVI.VI*”. Éste programa utiliza como base para su construcción un subVI llamado “*Dim-bright Luces.VI*”, el mismo que a su vez se basa en el programa genérico “*enviar 1 dato.VI*”.

El Programa “*Dim-bright Luces.VI*” debe enviar tres bytes hacia el Módulo de Control. Los dos primeros Bytes son enviados de igual manera que para los comandos “*Encender*” o “*Apagar*” y corresponden al Código de casa, Código de dirección, Código de función y Comandos para el microprocesador.

El tercer byte corresponde al Contador *Dim*. El diagrama de flujo que explica el funcionamiento del algoritmo para enviar los comandos *Dim* o *Bright* se muestra en la Figura 3.26.

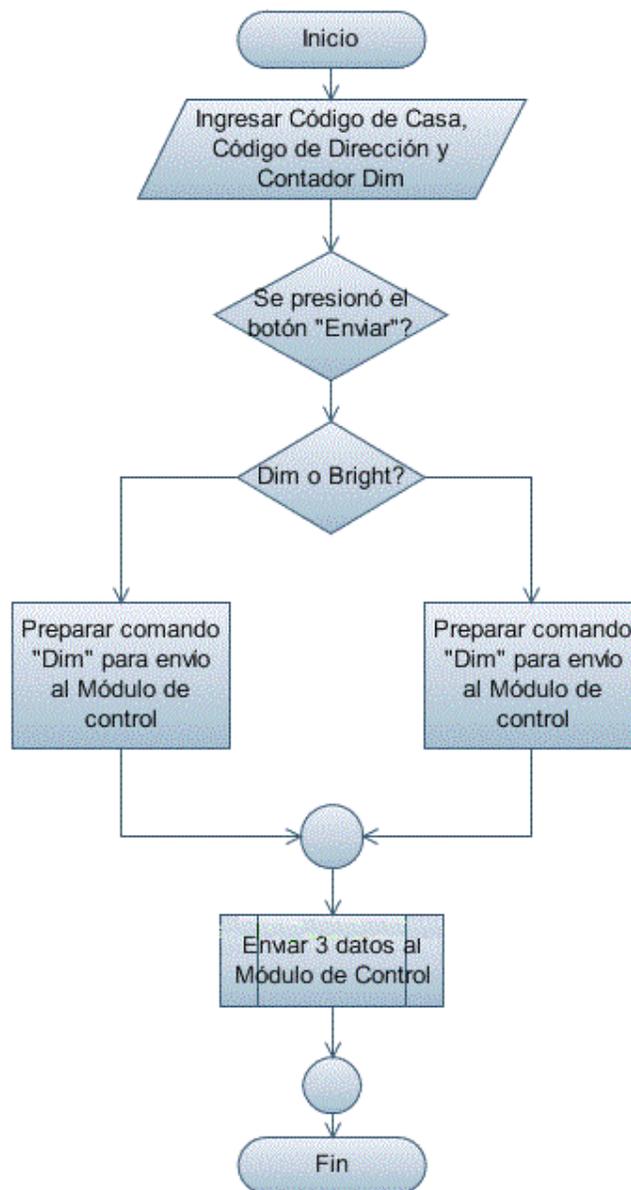


Figura 3.26. Diagrama de flujo de los comandos Dim y Bright

Los pasos necesarios para preparar el comando “Dim” para ser enviado al Módulo de Control se los describe a continuación con lenguaje estructurado.

**Ingresar Código de Casa y Código de Dirección y Contador Dim**

Leer datos ingresados por el usuario

**Fin Tarea**

**Preparar comando “Dim”**

Ejecutar subVI "Codificador Alfa.VI"

Sumar número 64 al resultado de Codificador Alfa

Seleccionar resultado de la suma como Primer dato a enviar

Ejecutar subVI "Codificador Num.VI"

Sumar número 32 al resultado de Codificador Num

Seleccionar resultado de la suma como Segundo dato a enviar

Dividir valor de Contador Num para 5

Seleccionar resultado de la división como Tercer dato a enviar

#### **Fin Tarea**

#### **Preparar comando "Bright"**

Ejecutar subVI "Codificador Alfa.VI"

Sumar número 80 al resultado de Codificador Alfa

Seleccionar resultado de la suma como Primer dato a enviar

Ejecutar subVI "Codificador Num.VI"

Sumar número 32 al resultado de Codificador Num

Seleccionar resultado de la suma como Segundo dato a enviar

Dividir valor de Contador Num para 5

Seleccionar resultado de la división como Tercer dato a enviar

#### **Fin Tarea**

#### **Enviar 3 datos al Módulo de Control**

Ejecutar subVI "Envío 1 dato.VI"

#### **Fin Tarea**

En el primer byte se envía el Código de Casa y el Código de Función. Para convertir el Código de Casa a su equivalente decimal se usa el "Codificador alfa.VI". El código de función se lo obtiene de un selector operado por un comando booleano que se utiliza para seleccionar entre el comando Dim o Bright. El equivalente decimal según la Tabla 3.3 para el comando Dim es 64 mientras que para el comando Bright es 80. Finalmente se suman los dos equivalentes decimales obtenidos para enviar el primer Byte.

El segundo byte está formado por el código de dirección más el Comando de Microprocesador correspondiente. Para obtener el decimal equivalente del código de dirección se utiliza el "Codificador Num.VI". El Comando de Microprocesador

correspondiente a los comandos Dim o Bright es el número 32 según la Tabla 3.6. Luego de sumar el equivalente decimal del código de dirección con el Comando de Microprocesador el segundo byte esta listo para ser transmitido.

Finalmente, el último byte que debe ser transmitido es el que sirve para determinar el número de pasos Dim o Bright que debe enviar el Módulo de Control en el comando X-10. Para este control se eligió un control vertical que indicará el nivel de iluminación seleccionado de 0 a 100. El valor entregado por este control se lo divide para 5, para ponerlo en la escala apropiada, ya que el número máximo de pasos Dim/bright es de 20. El resultado de esta división es el número que se va a transmitir como tercer Byte.

Luego de enviados estos tres bytes, el Módulo de Control ya tiene los datos necesarios para ejecutar el comando de Dim o Bright en tiempo real. El panel de control de “Dim-bright Luces.VI” se muestra en la Figura 3.27.

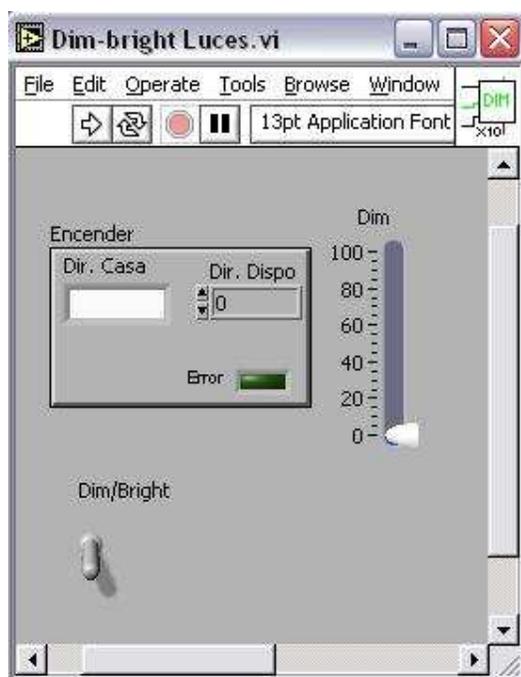


Figura 3.27 Panel de control del SubVI “Dim-bright.VI”

Para diseñar el programa “Dim subVI.VI” se utiliza el último subVI diseñado para enviar los datos necesarios al Módulo de Control. Debido a que se piensa utilizar

un comando numérico vertical como el que se muestra en la Figura 3.28, el programa “Dim subVI.VI” debe estar en condición de determinar si corresponde enviar un comando Dim o Bright hacia el subVI “Dim-bright Luces.VI”.



Figura 3.28. Control vertical que comanda las instrucciones Dim/Bright

Éste programa también debe determinar el valor del Contador Dim en función del valor anterior y del valor actual de dicho control vertical, el cual representa el nivel de luminosidad deseado. El diagrama de flujo de “Dim subVI.VI” se lo puede observar en la Figura 3.29.

Para poder determinar que tipo de comando enviar, “Dim subVI.VI” utiliza un array para guardar el estado anterior del control. Para ejecutar este subVI es necesario que el usuario elija el nivel de luminosidad que desea y de un clic en el botón “OK”. Si el estado anterior y el estado actual del control vertical son iguales el programa no ejecuta ninguna acción. Si ha habido algún cambio, el programa compara el estado anterior y el actual para ver cual es mayor.

Si el estado actual es menor que el estado anterior, el programa actualiza el registro del estado anterior con el nuevo valor del control vertical y ordena que “Dim-bright Luces.VI” envíe un comando Dim al Módulo de Control.

Si el estado actual es mayor que el estado anterior el programa actualiza de igual manera el registro del estado anterior con el nuevo valor del control vertical y ordena que “Dim-bright Luces.VI” envíe un comando Bright al Módulo de Control.

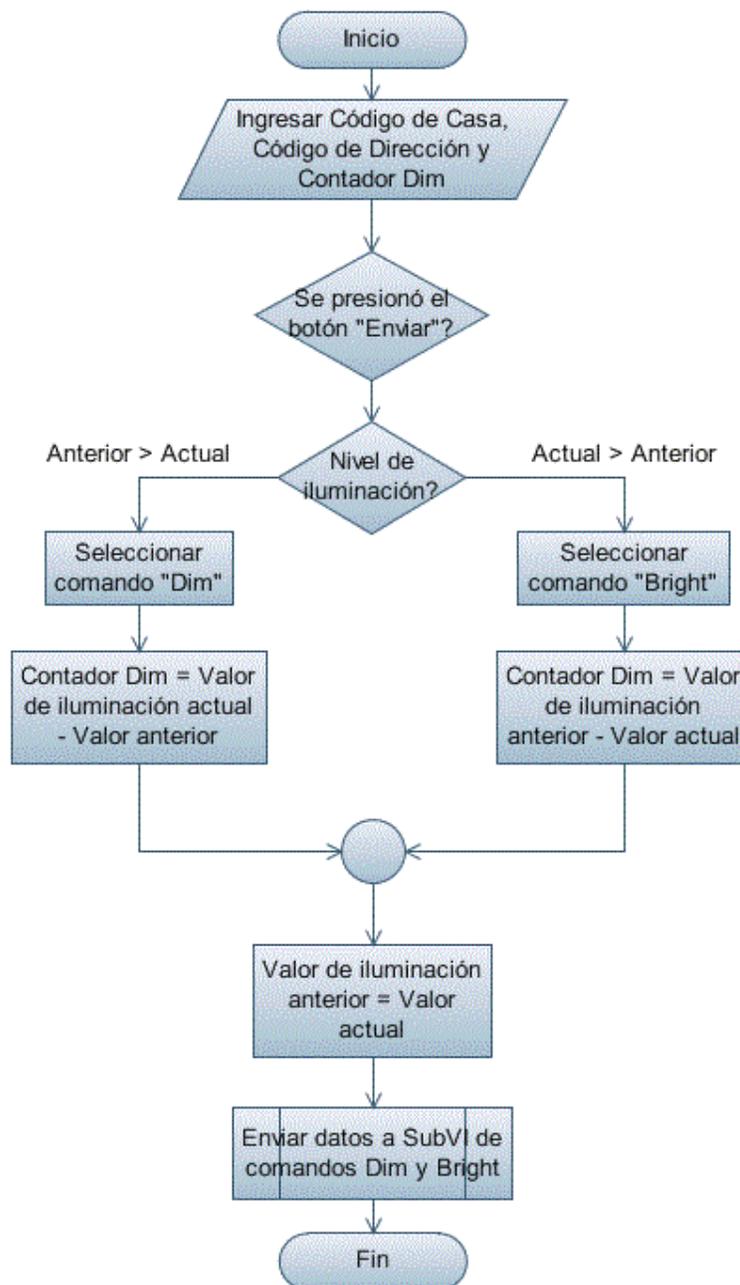


Figura 3.29. Diagrama de flujo de "Dim subVI.VI"

Finalmente se realiza la diferencia entre el número mayor y el menor de la comparación entre el registro del estado anterior y el actual. Este número se envía como el número de pasos Dim/bright que se deben ejecutar.

Ya sea que se envíe un comando Dim o un Bright, el Módulo de Control envía el comando X-10 en tiempo real. El panel frontal de “Dim subVI.VI” se lo puede encontrar en la Figura 3.30.

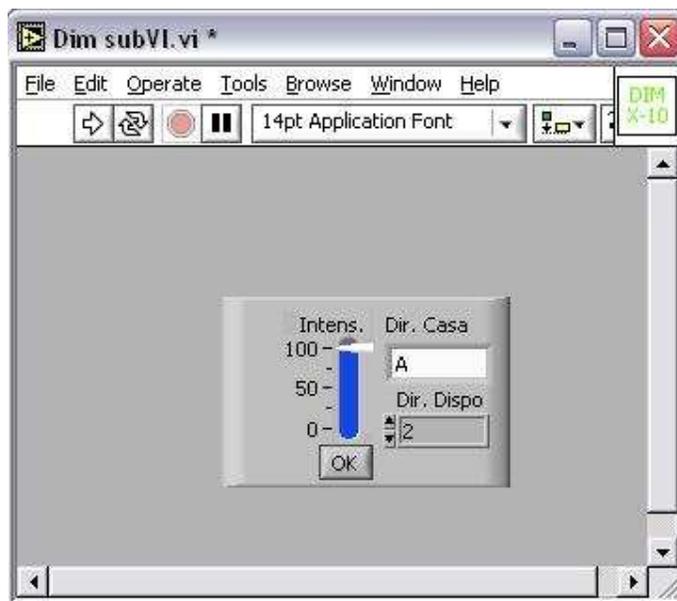


Figura 3.30. Panel de control del subVI “Dim subVI.VI”

#### 3.2.1.2.4 Botón genérico

Luego de que ya se encuentran desarrollados los Vi necesarios para enviar los comandos X-10 al Módulo de Control, se realizó el diseño de un botón genérico de control para agrupar a los comandos “Encender”, “Apagar”, “Dim” y Bright”, y de esta manera reducir al máximo el número de terminales que se deban conectar en el diagrama de la interfaz del usuario.

El subVI “Botón Gen b1.VI” es el encargado de unificar los subVI que envían al Módulo de Control los comandos antes mencionados. Para hacer más amigable la interfaz del usuario, se decidió utilizar un comando “Menu Ring” en el Panel de Control de la Interfaz de usuario así que es necesario sumar un número 1 al código de dirección que se reciba desde la Interfaz del usuario para permitir su funcionamiento con los subVI’s antes diseñados. También se requieren de tres comando booleanos que serán los encargados de iniciar la transmisión de los

comandos hacia el microcontrolador. El Código de Casa y el nivel de intensidad no requieren ningún cambio. Luego se procede a conectar los Códigos de Casa, Códigos de Dirección, control de intensidad y comandos booleanos a los terminales correspondientes de los subVI: Encender Luces.VI, Apagar Luces.VI y Dim subVI.VI. El panel de control de “Boton Gen b1.VI” se muestra en la Figura 3.31.

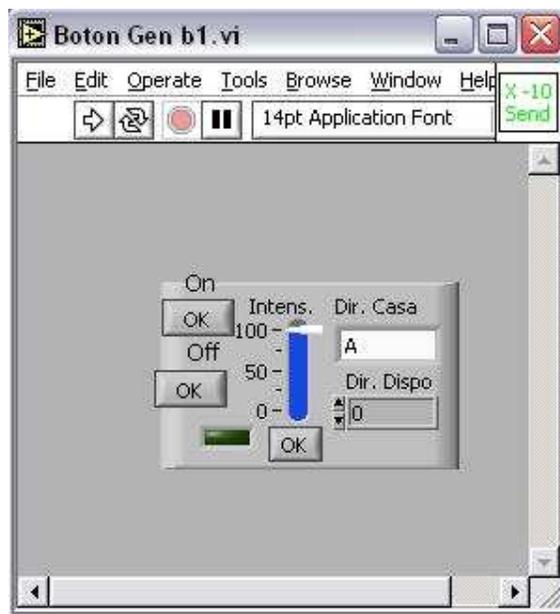


Figura 3.31. Panel de control del subVI “Boton Gen b1.VI”

#### 3.2.1.2.5 Generador de pulsos

Debido a que el subVI “botón general” requiere de dos controles booleanos para funcionar, es necesario crear un subVI que permita reducir éste número de controles a un solo interruptor. El diagrama de flujo para la creación de “generador de pulsos” se lo muestra en la Figura 3.32.

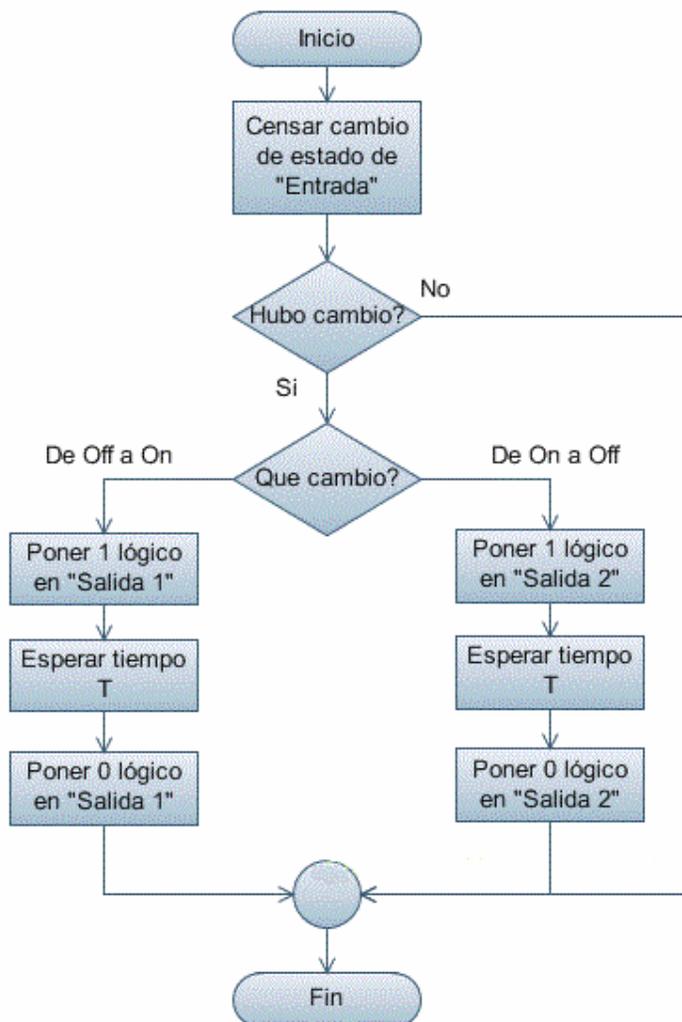


Figura 3.32. Diagrama de flujo de “Generador de pulsos”

Éste subVI transforma un cambio de estado de su control booleano en dos señales de salida. Si el cambio de estado es con flanco positivo (de off a on) envía un pulso a su indicador “Pulso On” y ninguna señal a su indicador “Pulso Off”. Si el cambio de estado es con flanco negativo (de on a off) el pulso lo envía al indicador “Pulso Off”.

El subVI “Generador pulsos b1.VI” evita que una señal “On” en la Interfaz de Usuario envíe continuamente comandos X-10 innecesarios, saturando de señales X-10 el medio físico; es decir, el cableado del hogar.

Cuando se requiera encender un dispositivo X.10, el usuario dará clic sobre el interruptor que aparece en la Interfaz de Usuario, y este subVI transforma este cambio de estado en un pulso que se transmitirá al subVI encargado de enviar comandos X-10 al Módulo de Control, para que dicho comando se envíe una sola vez.

Para el diseño de “Generador pulsos b1.VI” se utilizó un array de controles booleanos que servirá para registrar el estado anterior del control booleano “Entrada”. Mediante el uso de operadores lógicos se determina si hubo un flanco positivo o negativo y se elige entre una de las dos estructuras de caso.

Si es un flanco positivo se elige la estructura de caso que envía un 1 lógico al indicador “Pulso On”, actualiza el estado del array de controles booleanos e introduce un retardo para permitir que la señal sea leída por el subVI que se encargará de enviar el comando X-10 al microcontrolador.

Si se trata de un flanco negativo se elige la estructura que envía un 1 lógico al indicador “Pulso Off”, cambia el valor registrado en el array e introduce un retardo igual que en el caso anterior.

Luego de que se termina la ejecución de “Generador pulsos b1.VI” ambos indicadores vuelven a su estado predeterminado, en este caso “Off” (0 lógico).

Es necesario duplicar este subVI para cada Interruptor que se use en la Interfaz, ya que se necesita guardar el estado de cada uno de dichos interruptores. El panel de control de “Generador pulsos b1.VI” se lo puede encontrar en la Figura 3.33.

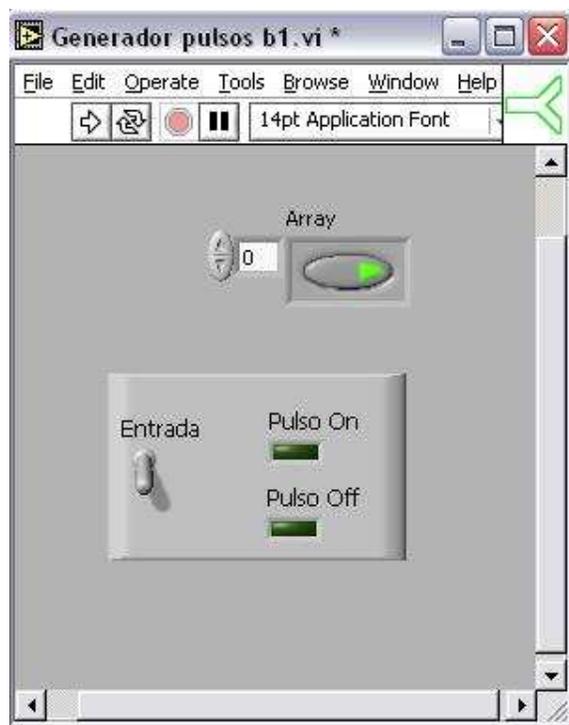


Figura 3.33. Panel de control del subVI “Generador pulsos b1.VI”

### 3.2.1.3 Comando para Igualar el reloj del Módulo de Control

Éste comando permite igualar tanto la hora como la fecha del reloj interno implementado en el microcontrolador. Este comando debe permitir adquirir la hora y fecha de la computadora y enviarla con el formato que utiliza el reloj del Módulo de Control. Para aprovechar la rutina de envío de datos ya implementada, los registros a enviar se agrupan de dos en dos. El diagrama de flujo del comando para igualar el reloj del Módulo de Control se lo observa en la Figura 3.31.

El subVI encargado de realizar esta tarea es “Igualar reloj 2.VI”. Éste subVI permite obtener la hora actual según la configuración actual de la computadora en la que corra la aplicación. Para ello primero utiliza la función “Get date/ time in seconds”, la cual cuenta los segundos transcurridos desde las 12 am GMT del 1 de Enero de 1904 y lo utiliza como dato para la función “Seconds to date/ time” para convertir estos segundos en un cluster de 9 enteros de 32 bits que especifican: segundo, minuto, hora, día del mes, mes, año, día de la semana, día

del año, y si se utiliza el horario estándar o el horario de verano. Éste cluster se lo convierte en un array y mediante la función “Index Array” se extrae el dato que se requiera para su posterior procesamiento.

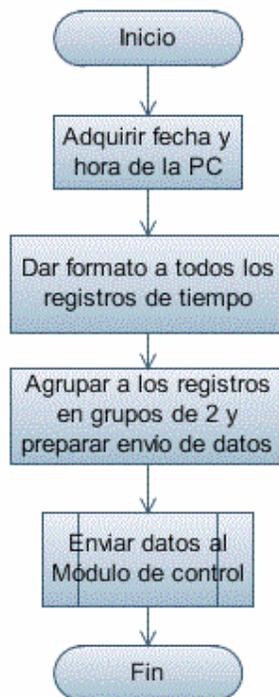


Figura 3.31. Diagrama de flujo del comando Igualar reloj.

Para poner el dato del año entregado por la computadora en el mismo formato que utiliza el Módulo de Control es necesario tomar el elemento 5 del array y restarle 2000 ya que la cuenta del reloj interno del microcontrolador comienza en el año 2000, mientras que el año 2000 en el reloj del Módulo de Control se lo representa con el número 0.

Para obtener el Año\_aux (conteo de años bisiestos) es necesario obtener un número comprendido entre 0 y 3, de acuerdo al programa del microcontrolador. Para ello es necesario obtener el residuo de la división para 4 de la variable Año y multiplicarlo por 4 para obtener un número entero comprendido entre 0 y 3.

Para obtener el día de la semana se toma el elemento número 6 del array y se le resta 1 para que coincida con el formato utilizado por el microcontrolador.

El resto de datos del array se los utiliza con el mismo formato que entrega la función “Seconds to date/ time” .

Para el proceso de escritura, éste subVI se utiliza un control booleano para iniciar el envío de datos y el subVI “Envío codigo X-10.VI”. El panel de control de control de “Envío codigo X-10.VI” se lo puede apreciar en la Figura 3.32

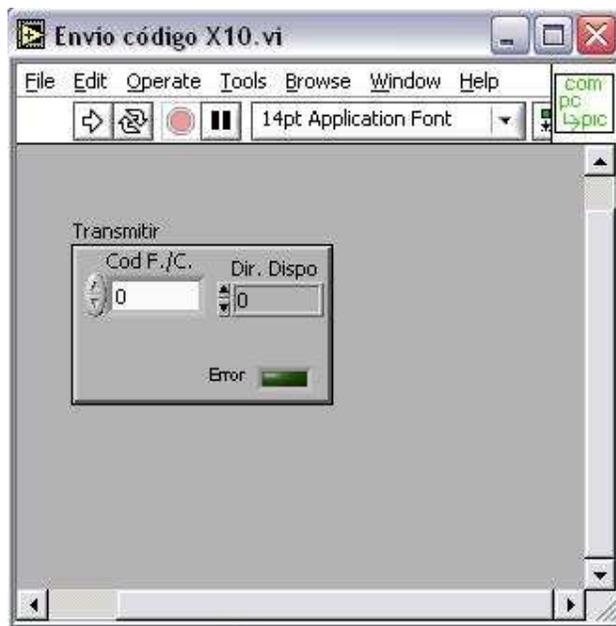


Figura 3.32. Panel de control del subVI “Envío codigo X-10.VI”

Una vez concedido el permiso de envío de datos, el programa envía los datos en grupos de dos mediante el uso de una estructura de secuencia y de “Envío codigo X-10”. El envío de datos debe respetar el orden antes establecido en la función para igualar el reloj del Módulo de Control. Por este motivo primero envía el dato del Año y el Año\_aux, luego envía el mes y el Día de la semana, luego el día y la hora, y finalmente el dato sobre minuto y segundo.

En cada caso es necesario que al segundo elemento de cada grupo, Año\_aux, D\_semana, hora y segundo, se le sume el número 192, el cual es el Comando de Microprocesador que permite igualar el reloj interno del Módulo de Control. El número 192 es la suma de 128 (comando que permite el acceso a las rutinas de escritura en memoria EEPROM) y el número 64 (comando que envía los datos a

la subrutina para igualar el reloj interno). Estos comandos se los puede consultar en la Tabla 3.6. Luego del envío de los 8 datos, el microcontrolador actualiza los registros correspondientes. El panel de control de “Igualar reloj 2.VI” se puede observar en la Figura 3.33.

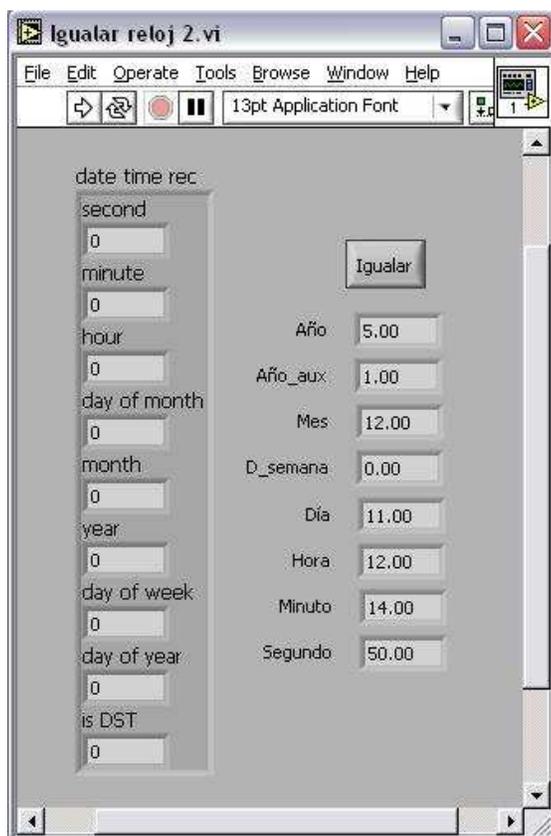


Figura 3.33. Panel de control del subVI “Igualar reloj 2.VI”

#### 3.2.1.4 Comando para escribir en la memoria EEPROM

Este comando permite la escritura de datos en la memoria EEPROM del microcontrolador. Requiere como entradas los datos que se necesitan enviar en formato de Array y el número de datos a enviar. El programa envía los datos uno a uno junto con el comando adecuado para escribir en la memoria EEPROM del Módulo de Control. El diagrama de flujo de este programa se lo presenta en la Figura 3.34

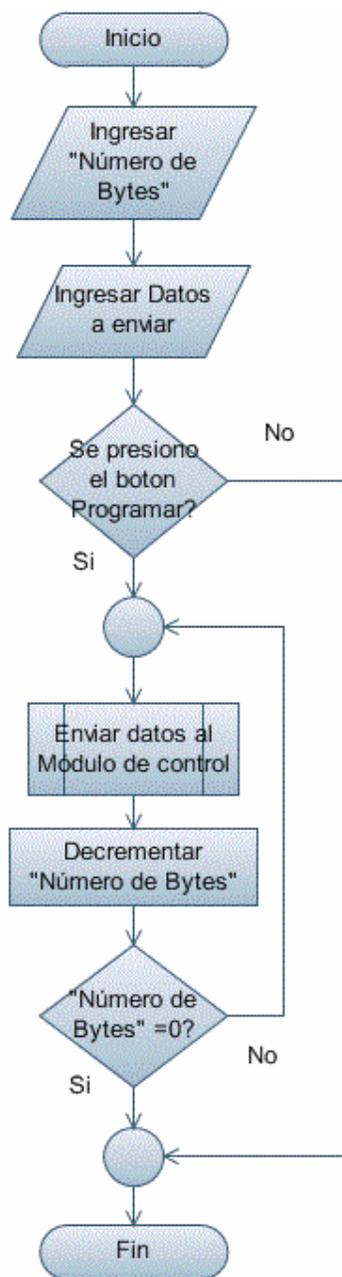


Figura 3.34. Diagrama de flujo del comando para escribir en la memoria EEPROM

El programa diseñado para este propósito es "Programar\_e.VI". Este programa utiliza un array para guardar la información que tiene que ser enviada, un control numérico que especifica cuantos bytes del array van a ser enviados, un control booleano que sirve para iniciar la escritura en la memoria EEPROM, otro control booleano que habilita o deshabilita el despliegue de un mensaje de programación exitosa, y un último control booleano que indica si la escritura debe realizarse

desde el inicio de la memoria EEPROM del microcontrolador o si se debe continuar desde la última posición de memoria escrita.

Cuando se da la orden de iniciar la escritura, el programa envía los datos del array uno a uno mediante el uso de una estructura “For loop” que se ejecuta un número N de veces igual al valor del control numérico. El programa utiliza el contador de iteraciones “i” como índice de la función “Index array” para extraer el número que debe ser enviado en la iteración actual. Para enviar dicho dato el programa utiliza el subVI “Envío código X-10.VI”. El primer dato enviado es el que se extrae del array, mientras que el segundo dato corresponde al Comando de Microprocesador adecuado (o a la suma de Comandos de Microprocesador requeridos).

Estos comandos se los puede observar en la Tabla 3.10. Cuando se llega al último byte a enviar se suma el número 32, correspondiente al Comando de Microprocesador de Fin de programación. Si se completa el proceso sin problemas, se muestra un mensaje que indica que la programación fue exitosa (si el control booleano correspondiente esta activado). El panel de control de “Programar\_e.VI” se lo puede observar en la Figura 3.35.

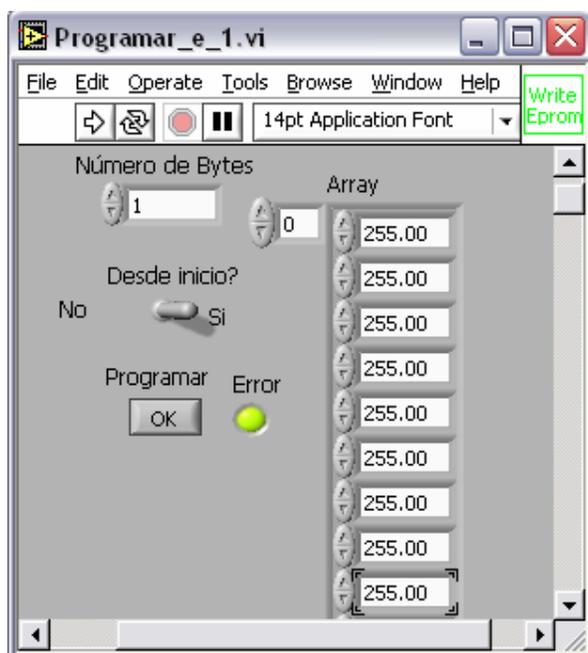


Figura 3.35. Panel de control del subVI “Programar\_e.VI”

Para permitir que la programación de comandos X-10 sea sencilla para el usuario, se implementó un subVI encargado de codificar las instrucciones ingresadas por el usuario, en el formato establecido para el almacenamiento de datos en la memoria EEPROM del microcontrolador. El diagrama de flujo de este “Codificador para Programación” se lo presenta en la Figura 3.36.

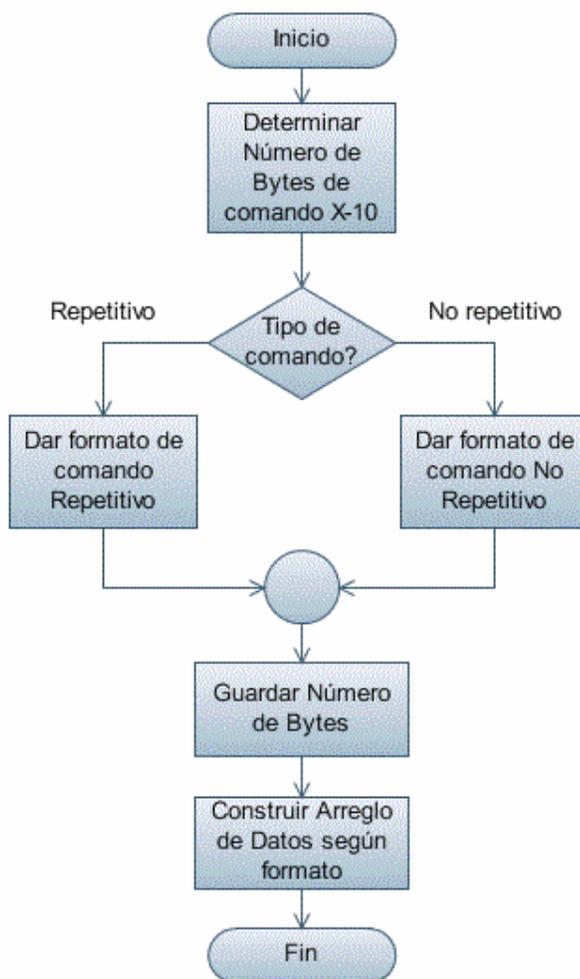


Figura 3.36. Diagrama de flujo de “Codificador de Programación”

El subVI diseñado para cumplir con ésta es “Codif Programar.VI”. Este programa permite determinar el número de bytes que el comando X-10 programado ocupará en la memoria de Módulo de Control. El valor de la extensión del comando es

necesaria para realizar el envío de los datos a través del “Comando para escribir en la memoria EEPROM”.

Según el tipo de comando seleccionado, el programa da formato de comando Repetitivo o No repetitivo a los datos ingresados por el usuario.

Para dar formato de comando Repetitivo, el programa construye un Array con los datos de: Encabezado para un comando Repetitivo, Día de la semana en que se ejecutará el comando, hora, minuto, segundo, Comando de procesador sumado al Código de dirección, Código de función sumado al Código de Casa y el Contador Dim. Este formato ya fue establecido y explicado en los Numerales 3.1.4 y 3.1.5. Estos ocho datos son enviados a una función “Index Array” para enviar solo los datos que sean necesarios, según el número de bytes establecidos anteriormente.

Para el formato de un comando No Repetitivo, el programa construye un Array con los datos de: Encabezado para un comando No Repetitivo, mes, día, hora, minuto, segundo, Comando de procesador sumado al Código de dirección, Código de función sumado al Código de Casa y el Contador Dim. Este formato ya fue establecido y explicado en los Numerales 3.1.4 y 3.1.5. Estos nueve datos son enviados a una función “Index Array”, para enviar solo los datos que sean necesarios según el número de bytes establecidos anteriormente.

### **3.2.2 CONSTRUCCIÓN DEL PANEL DE CONTROL DE LA INTERFAZ DE USUARIO**

La interfaz de usuario es la que permite controlar todos los dispositivos de la Red X-10. Ésta interfaz debe ser de fácil uso y permitir que el usuario pueda controlar electrodomésticos e iluminación, ya sea en tiempo real o mediante la programación de secuencias de comandos que serán ejecutados posteriormente.

La programación de dichos comandos debe ser de fácil comprensión y posibilitar la implementación de rutinas que permitan el ahorro de energía.

Para facilitar la comprensión de los diagramas del VI del panel de control, es preferible reducir al máximo el número de elementos que vayan directamente sobre dicho diagrama; es decir, se va a tratar de usar todos los subVI genéricos que sean necesarios para reducir al mínimo los elementos que aparezcan en el diagrama. El diagrama de flujo para la Interfaz del usuario ya se lo presentó en la Figura 3.15.

El Panel de Control dará la bienvenida al usuario e inmediatamente le preguntará si desea recibir ayuda para la configuración de los dispositivos X-10 instalados o si desea ingresar directamente a comandar su red X-10.

Gracias a esta interfaz, el usuario puede encender o apagar sus dispositivos X-10 y atenuar o incrementar el voltaje aplicado a ellos, si los actuadores instalados tienen ésta capacidad. Puede añadir o retirar dispositivos de la red X-10 con facilidad y podrá ejercer control de todos los ambientes de su hogar en tiempo real. También puede programar secuencias de comandos para realizar tareas rutinarias o de ahorro de energía.

El panel de control de la interfaz de usuario permitirá el control de cada uno de los 16 ambientes posibles asociados con los respectivos 16 códigos de casa, mediante pestañas. Otras tres pestañas permitirán la programación de los comandos que posteriormente será grabados en la memoria EEPROM del Módulo de Control. El panel de control del modo de ejecución se puede observar en la Figura 3.37

En cada uno de los ambientes se podrá tener hasta 16 Botones Genéricos para comandar los 16 dispositivos que pueden tener el mismo código de casa. Cada uno de estos Botones Genéricos tendrá un código de dirección predeterminado (de 1 a 16) según la posición que ocupe dentro del panel de control. Estos botones genéricos se pueden configurar según el tipo de actuador X-10 que esté asociado a su dirección de casa y su dirección de dispositivo.

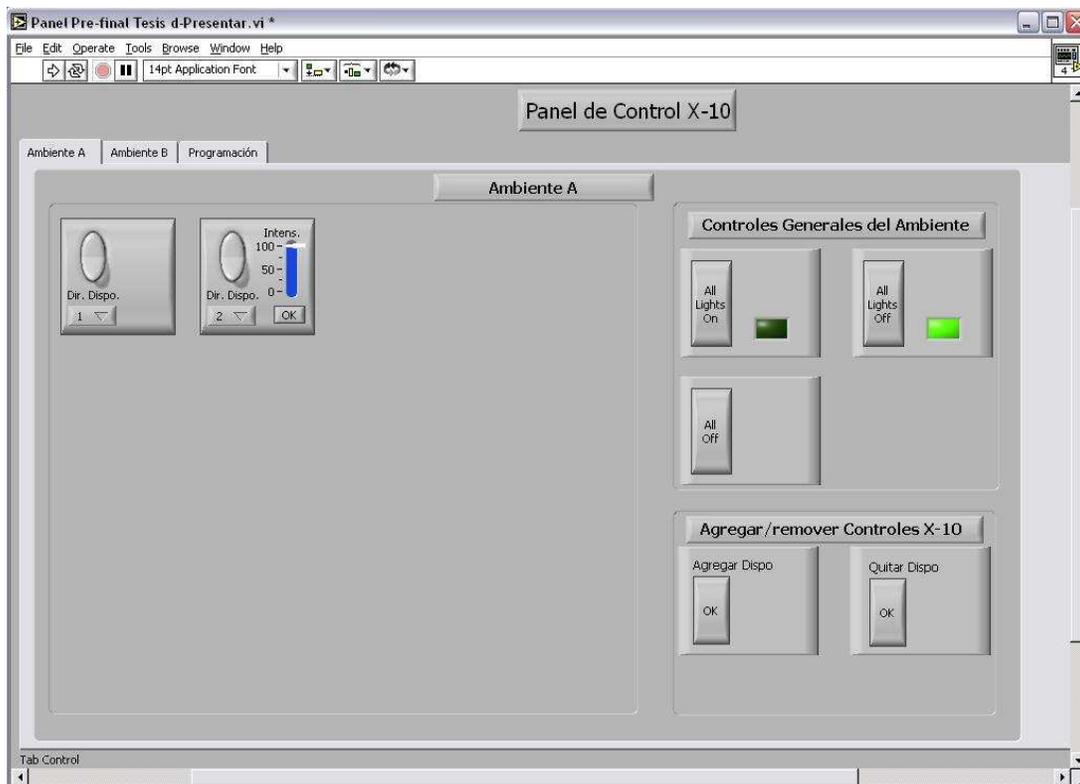


Figura 3.37 Panel de control del modo de ejecución

El botón que se encuentra en la parte superior izquierda de la Figura 3.18 estará asociado a un actuador que solamente admite comandos X-10 de tipo On / Off ya que tiene como controles un interruptor que permite enviar comandos X-10 de Encender y Apagar, y un Menu ring para modificar el código de dirección. El Código de Casa al cual se encuentra asociado este botón es la "A", ya que se encuentra en la pestaña correspondiente al Ambiente A.

El botón que se encuentra a la derecha del anterior estará asociado a un actuador que admita comandos de tipo On / Off y además Dim / Bright, ya que tiene como controles un interruptor para encender y apagar el actuador, un Menu ring que permite seleccionar el código de dirección, un control de tipo vertical pointer slide para seleccionar la intensidad a la cual se quiere que el actuador funcione y un botón de tipo OK para enviar comando para obtener el nivel de iluminación que se seleccionó con el control anterior.

En ambos casos el control correspondiente al interruptor se encuentra conectado al subVI “Generador pulsos b1.VI”, y éste a su vez está conectado a “Boton Gen b1.VI”. Los demás controles están conectados a los terminales correspondientes del subVI “Boton Gen b1.VI”.

En la parte derecha de este panel de control se han colocado dos grupos de botones. Los botones que corresponden a los controles Generales del Ambiente (Figura 3.37) permiten efectuar acciones de control sobre todos los dispositivos que se encuentren dentro del ambiente seleccionado. El botón All Lights On permite encender todas los actuadores que se hayan definido como actuadores para iluminación. Dicho botón está conectado a una estructura de caso que permite la ejecución por una sola vez de “Encender Todas las Luces.VI” y además cambia el estado del indicador a encendido.

El botón All Lights Off permite apagar todos los dispositivos del ambiente que se hayan definido como actuadores para iluminación. Dicho botón esta conectado a una estructura de caso que permite la ejecución por una sola vez de “Apagar Todas las Luces.VI” y además cambia el estado del indicador a apagado.

Finalmente el botón All Off permite apagar cualquier actuador que se encuentre dentro del ambiente seleccionado. Dicho botón esta conectado a una estructura de caso que permite la ejecución por una sola vez de “Apagar Todo.VI” y además cambia el estado del indicador a apagado.

El segundo grupo de botones que se encuentran en la parte derecha del panel de control son los que se puede observar en la sección Agregar / Remover controles X-10 de la Figura 3.37. Estos controles permiten modificar la configuración de cada ambiente. El momento que se da clic en el botón “Agregar dispositivo” el programa despliega un Tabla de dialogo que realiza una pregunta al usuario para determinar si se trata de un actuador de tipo On / Off o de tipo Dim / Bright. Cuando el actuador que se va a añadir es de tipo On / Off, se ocultan los controles “Vertical pointer slide” y el botón “OK”; mientras que cuando se trata de un actuador que admite comandos Dim / Bright, se muestran dichos controles.

La configuración de cada ambiente se la guarda en un Array llamado "Estado" de 16 datos numéricos. Si el dato correspondiente al primer control es 0 significa que dicho control está desactivado. Si es un número 1 significa que se trata de un control tipo On / Off y por lo tanto se deben ocultar los controles "Vertical pointer slide" y su botón de "OK". Si es un número 2 significa que el actuador soporta comandos Dim / Bright y por lo tanto se muestran todos los controles. Para mostrar u ocultar los 16 botones genéricos se utilizan 16 estructuras de caso que son comandadas por funciones "Index array", las cuales permiten extraer el dato del array que corresponde al botón asociado a la estructura de caso. Dependiendo del número que se extraiga del array se asignan valores a las propiedades de cada uno de los controles.

Los botones "Agregar dispo" y "Quitar dispo" permiten incrementar o decrementar el valor de un control que sirve de índice (puntero) para modificar el array. Luego se despliega un cuadro de texto que permite discriminar si el nuevo actuador soporta comandos Dim / Bright o no. Finalmente, con los dos datos recogidos en los pasos anteriores se procede a modificar el array "Estado".

### 3.3 SOFTWARE PARA LOS MÓDULOS RECEPTORES

El software para los módulos receptores es el encargado de recibir los comandos X-10 demodulados por hardware y comparar si la dirección recibida coincide con la suya. En caso de que esto último se cumpla, el programa debe interpretar el comando X-10 y ejecutarlo.

Para poder leer los comandos X-10, es necesario que la lectura se la haga sincronizándola con los cruces por cero de la corriente alterna de la red. Luego de que las lecturas de los bits estén bien sincronizadas, el programa debe buscar continuamente el encabezado de un comando X-10, y cuando lo encuentre debe recibir toda la trama y verificar que no tenga errores. Si hay algún error tiene que desechar la trama y si no hay errores debe verificar que el Código de Casa y de dirección recibidos le correspondan. Si no le corresponden, desechará la información recibida y volverá a buscar un encabezado X-10. Si le corresponde los códigos de casa y de dirección recibidos, el programa debe interpretar el comando X-10 y ejecutarlo en tiempo real. Los dispositivos receptores no almacenan ningún tipo de información. El diagrama de flujo del programa principal para los módulos receptores se encuentra en la Figura 3.38 y el diagrama de flujo de las interrupciones utilizadas para dicho programa se presentan en la Figura 3.39.



Figura 3.38. Diagrama de flujo del Programa para los Actuadores

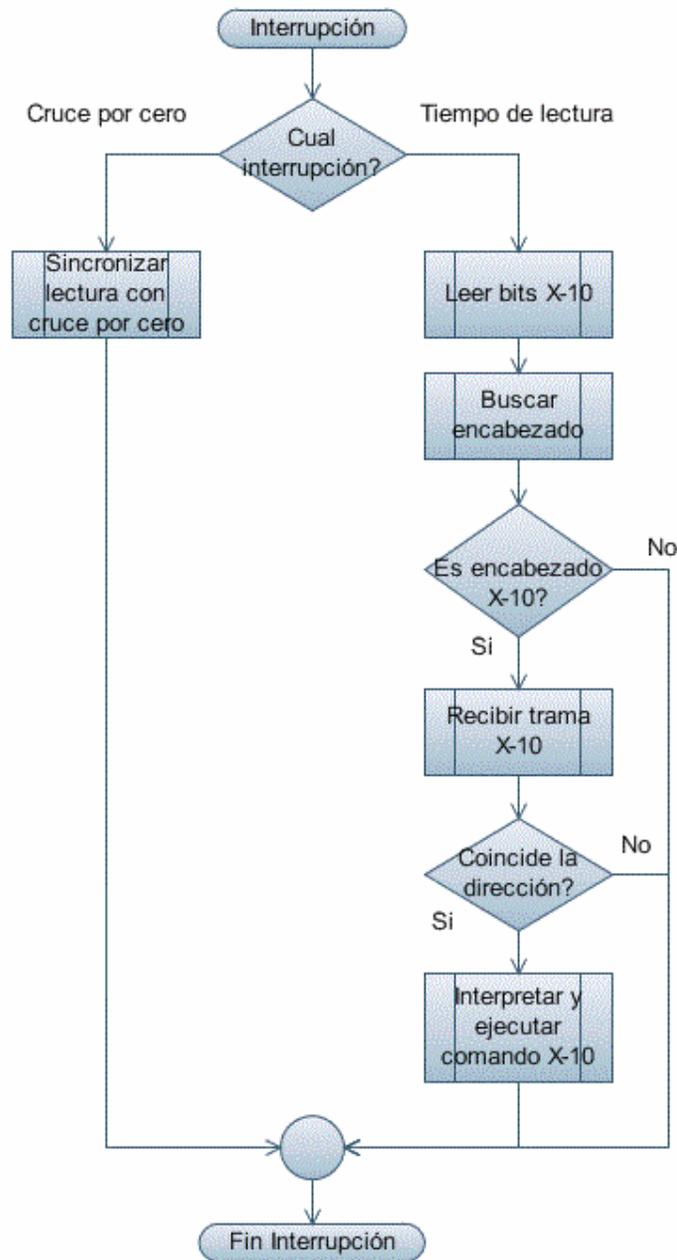


Figura 3.39. Diagrama de flujo de las interrupciones para el programa de los actuadores.

Para cumplir con las tareas descritas anteriormente se diseñó un programa llamado “Lector X-10 an\_B.asm”.

### 3.3.1 DISEÑO DEL PROGRAMA PARA LOS RECEPTORES

El software para los receptores fue diseñado de manera modular para facilitar futuras mejoras. Para realizar estas tareas se utilizan dos registros de control: Control y Control2. Los módulos en los que se subdivide el software son:

1. Lector de Bits X-10
2. Identificador de encabezados
3. Receptor de tramas X-10
4. Interprete de comandos X-10

#### 3.3.1.1 Lector de Bits X-10

Es necesario que la lectura de cada bit X-10 esté sincronizada con los cruces por cero de la línea de poder. Para lograr dicha sincronización, es necesario adquirir la señal de cruce por cero a través de una interrupción por cambio de estado. Luego de que la interrupción por cambio de estado identifique un cruce por cero se debe realizar su lectura dentro del 1ms que establece el protocolo X-10. El diagrama de flujo para la ejecución de este función se lo encuentra en la Figura 3.40.

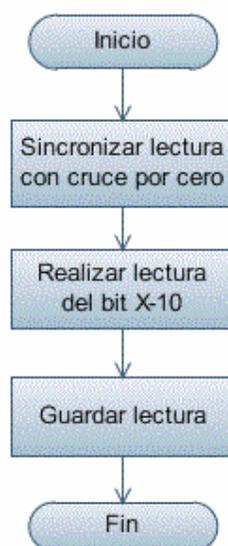


Figura 3.40. Diagrama de flujo para leer bits

Cada vez que un cambio de estado ocurra, el programa arranca el timer 1 para realizar una cuenta de 500  $\mu$ s. Ésta cuenta es necesaria para realizar la lectura del bit X-10 luego de que la señal proveniente del Filtro X-10 se haya estabilizado, por este motivo se decidió tomar la medición justo en la mitad del tiempo de duración de cada bit.

Luego de finalizado el conteo de los 500 $\mu$ s, se realiza la adquisición del bit X-10. Para ello se utiliza el conversor análogo / digital a través del canal 0. Después de que la conversión se ha terminado se compara éste resultado con el número decimal 40, el cual equivale a 0,8 V cuando se toma como referencia la alimentación de 5V. Un voltaje de entrada superior a 0,8 se considera un uno lógico, mientras que un voltaje inferior se considera un cero lógico.

El límite de voltaje configurable para determinar si se trata de un uno lógico o un cero lógico permite que señales X-10 en mal estado puedan ser reconocidas y a la vez para evita que algún ruido en la red sea leído erróneamente como uno lógico.

El valor de 0,8V se lo determinó experimentalmente luego de varias mediciones sobre el hardware. El resultado de la comparación se guarda en el bit7 del registro "Control" para ser utilizado en los siguientes módulos.

### **3.3.1.2 Identificador de encabezados**

Luego de que el bit X-10 fue leído, el identificador de encabezados crea un registro de desplazamiento de 8 bits con cada uno de los bit X-10 recibido.

Luego de cada actualización de dicho registro, el programa lee los cuatro bits menos significativos del registro en busca de un código de inicio para una trama X-10. El diagrama de flujo para la identificación de encabezados se lo presenta en la Figura 3.41.



Figura 3.41. Diagrama de flujo para la identificación de encabezados X-10

Para actualizar el registro de desplazamiento con el último bit recibido, el programa toma el último bit guardado por la función “leer bits X-10”, lo copia en bit de carry (registro STATUS) y rota el registro de desplazamiento, para que el contenido del carry ocupe el bit menos significativo de dicho registro. Luego se comparan los 4 bits menos significativos del registro de desplazamiento con el número binario “1110” correspondiente al encabezado X-10. Si los bytes son iguales, el programa escribe un uno lógico en el bit 0 del registro Control para establecer que ya se recibió el encabezado X-10 y que los siguientes 18 bits corresponden al resto de la trama X-10 (preparar la recepción de trama X-10). A continuación, el programa sale de la rutina de interrupción para esperar el siguiente bit X-10. Si no se produce la coincidencia entre los bits recibidos y el encabezado X-10, el programa sale de la rutina de interrupción y sigue esperando la llegada de un encabezado X-10.

### 3.3.1.3 Receptor de tramas X-10

Luego de que se identificó un encabezado X-10, es necesario recibir los 18 bits del resto de la trama X-10 y guardarlos para chequear la integridad de los datos recibidos. El diagrama de flujo del receptor de tramas X-10 se lo presenta en la Figura 3.42.

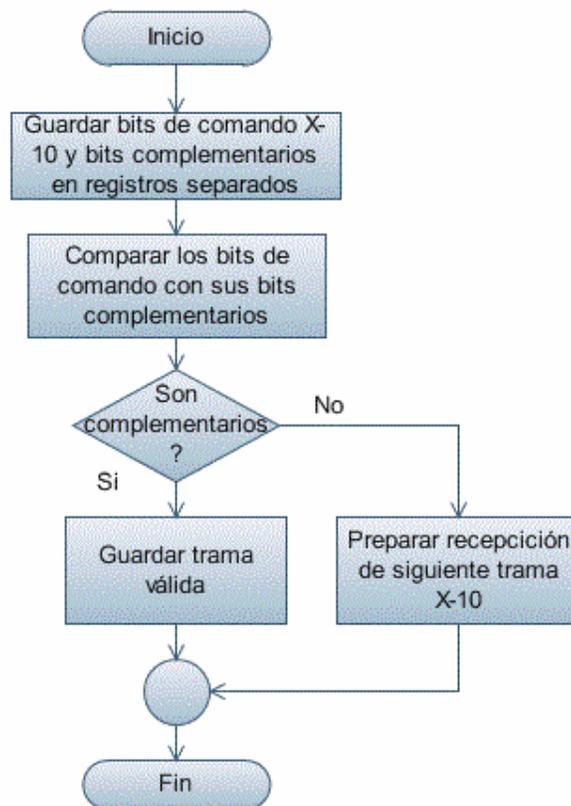


Figura 3.42. Diagrama de flujo del receptor de tramas X-10

Luego de que se ha recibido un encabezado X-10, es necesario separar los 9 bits correspondientes al comando X-10 de sus 9 bits complementarios. Para ello, los 16 bits que lleguen a continuación son distribuidos en dos registros de desplazamiento: Recept y Recept\_compl.

Los bits correspondientes al comando X-10 son almacenados en el registro Recept y los bits complementarios se envían al registro Recept\_compl. Como luego de cada bit de comando X-10 se debe enviar su bit complementario, es necesario que los bits se distribuyan alternadamente entre los dos registros. Para

realizar la tarea de conmutación se emplea el bit 1 del registro Control. Cuando dicho bit es un cero lógico, el programa envía el bit recibido al registro Recept. Cuando el bit 1 del registro Control es un uno lógico, el programa envía éste bit hacia el registro Recept\_compl.. El noveno bit del comando X-10 se envía a la posición 6 del registro Control y su complemento se envía a la posición 5 del mismo registro. Cuando el último bit de la trama es recibido (el bit complementario del noveno bit de comando X-10), el programa ejecuta la rutina de verificación de errores.

La rutina de verificación de errores utiliza una operación Or exclusiva para garantizar que los bits del comando X-10 que fueron recibidos y sus bits complementarios, en realidad son complementarios y así garantizar que no hubo ningún error en la transmisión. Para comprobar la integridad de los dos últimos bits recibidos se utiliza una rutina de comparaciones. Luego de garantizada la integridad de la trama X-10, éstos datos pueden ser usados por el interprete de comandos X-10.

#### **3.3.1.4 Interprete de comandos X-10**

Una vez verificada la validez de la trama X-10 se procede a identificar el tipo de comando X-10 enviado en la trama. Puede tratarse de un Código de Casa y de dirección o de un Código de Casa y de función.

La diferencia entre estos dos tipos de tramas X-10 es que el último bit de la trama es un cero lógico para el código de dirección y un uno lógico para el código de función.

Luego de identificar el tipo de comando que se recibió, es necesario interpretar dicho comando y ejecutar las acciones de control que correspondan. El diagrama de flujo para esta operación se lo encuentra en la Figura 3.43.

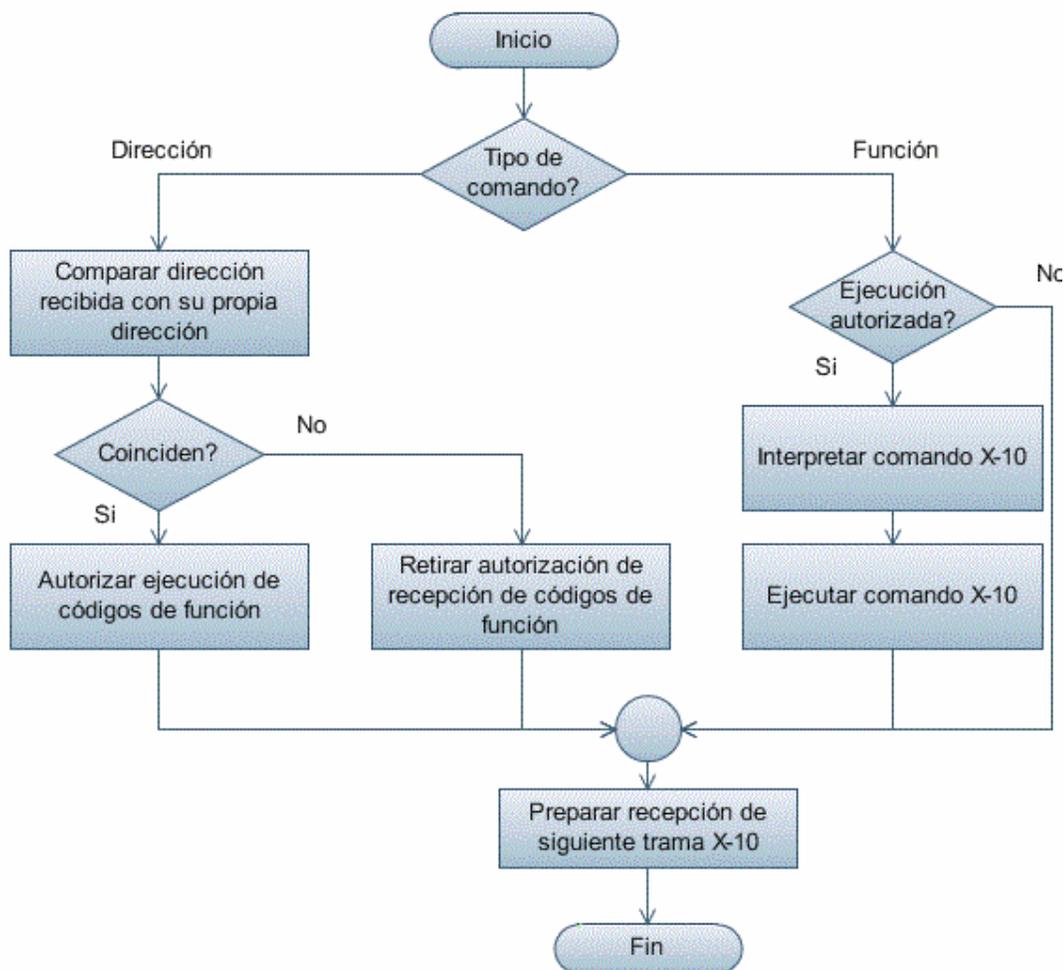


Figura 3.43. Diagrama de flujo del Interprete de comandos X-10

Para determinar el tipo de comando recibido, el interprete de comandos revisa el noveno bit recibido. En el caso de que sea un cero lógico el programa selecciona la rutina para comparar los códigos de casa y los códigos de dirección. En cambio, si dicho bit es un uno lógico, el programa selecciona la rutina del Interprete de comandos X-10 para decodificar el código de función recibido.

La rutina para comparar si la dirección enviada en la trama X-10 corresponde al dispositivo que la recibe es muy sencilla. Simplemente se compara la dirección asignada al dispositivo con la parte de la trama X-10 recibida por el registro Recept.. Si los dos bytes son idénticos, se pone un uno lógico en el bit 0 del registro Control2 para indicar que la siguiente trama X-10 contiene comandos

dirigidos a éste dispositivo que deben ser ejecutados (Autorización de códigos de función).

Por otro lado, si los dos bytes comparados son diferentes, se pone un cero lógico en el bit 0 del registro Control2, para indicar que los próximos códigos de función que sean enviados estarán dirigidos a otro dispositivo. Luego el programa se prepara para recibir la siguiente trama X-10 y sale de la rutina de interrupción.

Si el programa selecciona la rutina del interprete de comandos X-10, la primera acción a tomarse es verificar que el código de casa, incluido en la trama, corresponda a éste dispositivo. Para este propósito se toman los 4 bits más significativos del registro que guarda la dirección asignada a dicho dispositivo y se los compara con los 4 bit más significativos que se recibieron en el registro Recept. Si el resultado de la comparación indica que se trata de códigos de casa distintos, el programa desecha la trama ya que corresponde a otro dispositivo y se prepara para recibir un nueva trama X-10. Pero si el resultado de la comparación indica que los dos códigos de casa son iguales, entonces se procede a llamar a la tabla encargada de decodificar el comando X-10. Para ello se extraen los 4 bits menos significativos del dato contenido en el registro Recept y se los carga en el acumulador. Inmediatamente se llama a la tabla que según el comando recibido, selecciona la subrutina apropiada.

Cuando el comando recibido es un "All units Off" simplemente envía un cero lógico a su actuador para apagarlo.

Si el comando recibido es un "All lights on", se verifica si éste dispositivo esta definido como un actuador para iluminación . Si el bit 1 del registro Control2 es un uno lógico se trata de un actuador para iluminación y el programa envía un uno lógico, a su actuador para encender el foco alimentado por dicho actuador.

En caso de encontrarse un cero lógico en la posición 1 del registro Control2, éste dispositivo no es un actuador de iluminación y no se realiza ninguna acción de control.

Cuando el comando recibido es "On" el programa verifica si el bit0 del registro Control2 es un uno lógico. Si es así, significa que la trama X-10 anterior autorizó a éste actuador para recibir el presente comando y por tal motivo el programa envía un uno lógico a su actuador para encenderlo.

En el caso en que el comando recibido sea "Off" el programa verifica si el bit0 del registro Control2 es un uno lógico. Si es así, significa que la trama X-10 anterior autorizó a éste actuador para recibir el presente comando y por tal motivo el programa envía un cero lógico a su actuador para apagarlo.

Finalmente, si el comando recibido es "All lights off" se verifica si el dispositivo está definido como actuador para iluminación de la misma manera que se lo hizo para el comando "All lights on". Si se verifica que se trata de un dispositivo para iluminación, el programa envía un uno lógico a su actuador para encenderlo. Si la comprobación determinó que no se trata de un actuador para iluminación, entonces no se ejecuta ninguna acción.

Luego de ejecutado el comando apropiado, el programa se prepara para recibir una nueva trama X-10 y sale de la rutina de interrupción para esperar el siguiente cruce por cero y reiniciar todo el proceso de recepción. Una vez integrado éste último módulo al programa para los módulos receptores, se ha finalizado con el proceso de desarrollo del software.

Finalmente, es necesario verificar el funcionamiento en conjunto del hardware y del software diseñados. En el siguiente capítulo se harán pruebas de funcionamiento y compatibilidad con algunos módulos comerciales. También se realizarán pruebas para verificar el correcto funcionamiento del sistema, programando y observando la ejecución de secuencias de comandos que permitan facilitar las tareas cotidianas de un hogar.

# **CAPÍTULO 4**

## **PRUEBAS Y RESULTADOS**

## **CAPÍTULO 4: PRUEBAS Y RESULTADOS**

Luego de que se comprobó el correcto funcionamiento del Hardware y del Software es necesario realizar pruebas de funcionamiento de todo el sistema para comprobar que la red X-10 funcione correctamente. También es necesario realizar pruebas de compatibilidad con algunos dispositivos X-10 comerciales para garantizar que se logró la compatibilidad con los equipos análogos ya existentes en el mercado.

La organización de este capítulo está enfocada a comprobar que los objetivos planteados para este proyecto se hayan cumplido a cabalidad.

Primeramente se comprobará que los dispositivos diseñados funcionen adecuadamente en varios entornos y con diferentes longitudes de cable.

Posteriormente se verificará la compatibilidad entre los dispositivos diseñados y los dispositivos que se pueden encontrar comercialmente.

Luego se contrastará que todas las secuencias programadas en el Módulo de Control se cumplan en los tiempos programados. Se buscará comprobar que no existan errores en la ejecución de los comandos grabados en la memoria EEPROM del microcontrolador y que los errores en los tiempos de ejecución sean de máximo un segundo, tiempo que corresponde a la resolución del reloj interno.

Finalmente se documentará la efectividad de las rutinas para ahorro de energía y se realizará un estudio comparativo para determinar el ahorro obtenido.

### **4.1 PRUEBAS DE FUNCIONAMIENTO GENERAL**

Estas pruebas buscan comprobar que todo el sistema esté funcionando apropiadamente y que la red X-10 trabaje de una manera eficaz. Las pruebas realizadas en este literal consistieron en enviar todos los comandos

implementados a varias distancias con veinte repeticiones por comando. Las distancias de prueba fueron: menos de un metro, cinco metros, diez metros y veinte metros. Las pruebas se repitieron para cada tipo de actuador.

Debido a que no todos los comandos X-10 pueden ser ejecutados en todos los diferentes tipos de actuadores existentes, se dividió a los actuadores en 3 grupos básicos: Dispositivos de iluminación tipo On / Off, dispositivos de iluminación atenuables y dispositivos de potencia. Es importante aclarar que toda distancia mencionada en el presente subcapítulo es aproximada, ya que se utilizó el cableado y los tomacorrientes ya instalados en los diferentes sitios de prueba.

#### 4.1.1 DISPOSITIVOS DE ILUMINACIÓN TIPO ON / OFF

Estos dispositivos están diseñados para manejar bombillas incandescentes. Se pueden usar para manejar cargas resistivas de baja potencia hasta 100W con una corriente máxima de 1A. Las instrucciones que estos dispositivos pueden ejecutar son: On, Off, All lights on, All lights off y All off.

Los resultados de las pruebas para los dispositivos de iluminación tipo On / Off realizadas utilizando el cableado instalado de un hogar se encuentran resumidos en la Tabla 4.1 para distancias menores a un metro, en la Tabla 4.2 para una distancia de cinco metros, en la Tabla 4.3 para una distancia de diez metros y en la Tabla 4.4 para una distancia de veinte metros.

Tabla 4.1. Pruebas para dispositivos de iluminación tipo On / off – menos de 1m

<b>Comando</b>	<b>Total de paquetes enviados</b>	<b>Recibidos</b>	<b>Perdidos</b>	<b>Porcentaje de éxito</b>
On	20	20	0	100.00%
Off	20	20	0	100.00%
All lights on	20	20	0	100.00%
All lights off	20	20	0	100.00%
All off	20	20	0	100.00%

Tabla 4.2. Pruebas para dispositivos de iluminación tipo On / off – 2m

<b>Comando</b>	<b>Total de paquetes enviados</b>	<b>Recibidos</b>	<b>Perdidos</b>	<b>Porcentaje de éxito</b>
On	20	20	0	100.00%
Off	20	20	0	100.00%
All lights on	20	20	0	100.00%
All lights off	20	20	0	100.00%
All off	20	20	0	100.00%

Tabla 4.3. Pruebas para dispositivos de iluminación tipo On / off – 10m

<b>Comando</b>	<b>Total de paquetes enviados</b>	<b>Recibidos</b>	<b>Perdidos</b>	<b>Porcentaje de éxito</b>
On	20	20	0	100.00%
Off	20	20	0	100.00%
All lights on	20	20	0	100.00%
All lights off	20	20	0	100.00%
All off	20	20	0	100.00%

Tabla 4.4. Pruebas para dispositivos de iluminación tipo On / off – 20m

<b>Comando</b>	<b>Total de paquetes enviados</b>	<b>Recibidos</b>	<b>Perdidos</b>	<b>Porcentaje de éxito</b>
On	20	20	0	100.00%
Off	20	20	0	100.00%
All lights on	20	20	0	100.00%
All lights off	20	20	0	100.00%
All off	20	20	0	100.00%

Como se puede observar en la Tabla 4.1, Tabla 4.2, Tabla 4.3 y Tabla 4.4 la comunicación utilizando el protocolo X-10 es muy confiable para distancias de hasta 20 metros cuando se esta comandando dispositivos de iluminación tipo On /off dentro de un hogar. Para verificar estos resultados en otros entornos se repitieron las pruebas usando el cableado de otros dos hogares y adicionalmente se realizaron pruebas en el Laboratorio de Instrumentación y en el Laboratorio de Instrumentación Industrial de la Escuela Politécnica Nacional. La distancia elegida para dichas pruebas fue de 20 metros ya que es la mayor distancia a la

cual se busca garantizar la conectividad. Los resultados de las pruebas realizadas en los cinco entornos diferentes se resumen en la Tabla 4.5.

Como los resultados fueron satisfactorios para todos los entornos no fue necesario repetir las pruebas para las distancias de 1, 5 y 10 metros. Estas pruebas verifican que la comunicación con los dispositivos de iluminación tipo On /Off tiene una muy alta confiabilidad para distancias de hasta 20 metros.

Tabla 4.5. Resumen de pruebas para dispositivos de iluminación tipo On / Off – 20m

<b>Comando</b>	<b>Total de paquetes enviados</b>	<b>Recibidos</b>	<b>Perdidos</b>	<b>Porcentaje de éxito</b>
On	100	100	0	100.00%
Off	100	100	0	100.00%
All lights on	100	100	0	100.00%
All lights off	100	100	0	100.00%
All off	100	100	0	100.00%

#### **4.1.2 DISPOSITIVOS DE ILUMINACIÓN ATENUABLES**

Este grupo de dispositivos está comprendido por los actuadores capaces de reducir el voltaje que entregan a la carga. Estos dispositivos están diseñados para soportar cargas de iluminación equivalentes a las bombillas incandescentes de 100W y, al igual que los dispositivos de la sección 4.1, pueden manejar también cargas resistivas de hasta 1A. Los comandos que pueden ejecutar este tipo de dispositivos son: On, Off, All lights on, All lights off, All off, Dim y Bright.

Los resultados de las primeras pruebas para los dispositivos de iluminación atenuables realizadas utilizando el cableado instalado de un hogar se registraron en la Tabla 4.6, para distancias menores a un metro, en la Tabla 4.7 para una distancia de cinco metros, en la Tabla 4.8 para una distancia de diez metros y en la Tabla 4.9 para una distancia de veinte metros.

Tabla 4.6. Pruebas para dispositivos de iluminación atenuables – menos de 1m

<b>Comando</b>	<b>Total de paquetes enviados</b>	<b>Recibidos</b>	<b>Perdidos</b>	<b>Porcentaje de éxito</b>
On	20	20	0	100.00%
Off	20	20	0	100.00%
All lights on	20	20	0	100.00%
All lights off	20	20	0	100.00%
All off	20	20	0	100.00%
Dim	400	400	0	100.00%
Bright	400	400	0	100.00%

Tabla 4.7. Pruebas para dispositivos de iluminación atenuables – 5m

<b>Comando</b>	<b>Total de paquetes enviados</b>	<b>Recibidos</b>	<b>Perdidos</b>	<b>Porcentaje de éxito</b>
On	20	20	0	100.00%
Off	20	20	0	100.00%
All lights on	20	20	0	100.00%
All lights off	20	20	0	100.00%
All off	20	20	0	100.00%
Dim	400	400	0	100.00%
Bright	400	400	0	100.00%

Tabla 4.8. Pruebas para dispositivos de iluminación atenuables – 10m

<b>Comando</b>	<b>Total de paquetes enviados</b>	<b>Recibidos</b>	<b>Perdidos</b>	<b>Porcentaje de éxito</b>
On	20	20	0	100.00%
Off	20	20	0	100.00%
All lights on	20	20	0	100.00%
All lights off	20	20	0	100.00%
All off	20	20	0	100.00%
Dim	400	400	0	100.00%
Bright	400	400	0	100.00%

Tabla 4.9. Pruebas para dispositivos de iluminación atenuables – 20m

<b>Comando</b>	<b>Total de paquetes enviados</b>	<b>Recibidos</b>	<b>Perdidos</b>	<b>Porcentaje de éxito</b>
On	20	20	0	100.00%
Off	20	20	0	100.00%
All lights on	20	20	0	100.00%
All lights off	20	20	0	100.00%
All off	20	20	0	100.00%
Dim	400	400	0	100.00%
Bright	400	400	0	100.00%

Como se puede observar en la Tabla 4.6, Tabla 4.7, Tabla 4.8 y Tabla 4.9 la comunicación utilizando el protocolo X-10 es muy confiable para distancias de hasta 20 metros, cuando se está comandando dispositivos de iluminación atenuables dentro de un hogar. Para verificar estos resultados en otros entornos se repitieron las pruebas usando el cableado de los mismos dos hogares utilizados en el numeral anterior y adicionalmente se realizaron pruebas en el Laboratorio de Instrumentación y en el Laboratorio de Instrumentación Industrial. La distancia elegida para dichas pruebas fue de igualmente 20 metros. Los resultados de las pruebas realizadas en los cinco entornos diferentes se resumen en la Tabla 4.10

. Tabla 4.10. Resumen de pruebas para dispositivos de iluminación atenuables –  
20m

<b>Comando</b>	<b>Total de paquetes enviados</b>	<b>Recibidos</b>	<b>Perdidos</b>	<b>Porcentaje de éxito</b>
On	100	100	0	100.00%
Off	100	100	0	100.00%
All lights on	100	100	0	100.00%
All lights off	100	100	0	100.00%
All off	100	100	0	100.00%
Dim	2000	2000	0	100.00%
Bright	2000	2000	0	100.00%

Como los resultados fueron satisfactorios para todos los entornos no fue necesario repetir las pruebas para las distancias de 1, 5 y 10 metros. Estas

pruebas verifican que la comunicación con los dispositivos de iluminación atenuables tiene una muy alta confiabilidad para distancias de hasta 20 metros.

#### 4.1.3 DISPOSITIVOS DE POTENCIA

Estos dispositivos están diseñados para soportar cargas resistivas o inductivas de hasta 3A y 360VA. Los comandos que pueden ejecutar este tipo de dispositivos son: On, Off, y All off.

Los resultados de las primeras pruebas para los dispositivos de potencia realizadas utilizando el cableado instalado en un hogar se registraron en la Tabla 4.11, para distancias menores a un metro, en la Tabla 4.12 para una distancia de cinco metros, en la Tabla 4.13 para una distancia de diez metros y en la Tabla 4.14 para una distancia de veinte metros.

Tabla 4.11. Pruebas para dispositivos de potencia – menos de 1m

<b>Comando</b>	<b>Total de paquetes enviados</b>	<b>Recibidos</b>	<b>Perdidos</b>	<b>Porcentaje de éxito</b>
On	20	20	0	100.00%
Off	20	20	0	100.00%
All off	20	20	0	100.00%

Tabla 4.12. Pruebas para dispositivos de potencia – 5m

<b>Comando</b>	<b>Total de paquetes enviados</b>	<b>Recibidos</b>	<b>Perdidos</b>	<b>Porcentaje de éxito</b>
On	20	20	0	100.00%
Off	20	20	0	100.00%
All off	20	20	0	100.00%

Tabla 4.13. Pruebas para dispositivos de potencia – 10m

<b>Comando</b>	<b>Total de paquetes enviados</b>	<b>Recibidos</b>	<b>Perdidos</b>	<b>Porcentaje de éxito</b>
On	20	20	0	100.00%
Off	20	20	0	100.00%
All off	20	20	0	100.00%

Tabla 4.14. Pruebas para dispositivos de potencia – 20m

<b>Comando</b>	<b>Total de paquetes enviados</b>	<b>Recibidos</b>	<b>Perdidos</b>	<b>Porcentaje de éxito</b>
On	20	20	0	100.00%
Off	20	20	0	100.00%
All off	20	20	0	100.00%

Como se puede observar en la Tabla 4.11, Tabla 4.12, Tabla 4.13 y Tabla 4.14 la comunicación utilizando el protocolo X-10 es muy confiable para distancias de hasta 20 metros cuando se esta comandando dispositivos de potencia dentro de un hogar. Para confirmar estos resultados se repitieron las pruebas en los mismos dos hogares utilizados en 4.1.1 y 4.1.2 y adicionalmente se realizaron pruebas en el Laboratorio de Instrumentación y en el Laboratorio de Instrumentación Industrial. Se utilizó la misma distancia de 20 metros. Los resultados de las pruebas realizadas en los cinco entornos se resumen en la Tabla 4.15

Tabla 4.15. Resumen de pruebas para dispositivos de potencia – 20m

<b>Comando</b>	<b>Total de paquetes enviados</b>	<b>Recibidos</b>	<b>Perdidos</b>	<b>Porcentaje de éxito</b>
On	100	100	0	100.00%
Off	100	100	0	100.00%
All off	100	100	0	100.00%

Como los resultados fueron satisfactorios para todos los entornos no fue necesario repetir las pruebas para las distancias de 1, 5 y 10 metros. Estas pruebas verifican que la comunicación con los dispositivos de potencia tiene una muy alta confiabilidad para distancias de hasta 20 metros.

Con los resultados obtenidos en los numerales 4.1.1, 4.1.2 y 4.1.3 se puede afirmar que la confiabilidad de la conexión entre los dispositivos de la red X-10 es del 100% dentro de un rango de hasta 20m de cable, utilizando el cableado estándar de cualquier hogar.

No se realizaron pruebas en ambientes industriales u otros con alto ruido ya que el protocolo X-10 solo está diseñado y recomendado para hogares.

Las pruebas realizadas en los laboratorios de Instrumentación y de Instrumentación Industrial demuestran que los dispositivos diseñados son capaces de admitir cierto nivel de ruido e incluso se demostró que el Protocolo X-10 puede funcionar apropiadamente dentro de una sala de computo de hasta 16 computadoras sin ningún problema.

## **4.2 PRUEBAS DE COMPATIBILIDAD**

Con las pruebas de compatibilidad se quiere determinar el grado en el cual los dispositivos diseñados pueden trabajar en conjunto con los dispositivos construidos por otros fabricantes. Debido a que no se dispone de una gran variedad de marcas y modelos, y tomando en consideración que el protocolo X-10 estandariza tanto la parte física como lógica del protocolo, se consideró suficiente realizar pruebas de compatibilidad con los equipos adquiridos previamente.

Las pruebas de compatibilidad se dividirán en dos etapas: Envío de comandos desde el Módulo de Control diseñado hacia actuadores comerciales, y envío de comandos desde módulos comerciales hacia los actuadores diseñados.

### **4.2.1 ENVÍO DE COMANDOS DESDE EL MÓDULO DE CONTROL DISEÑADO HACIA ACTUADORES COMERCIALES.**

Para la realización de esta prueba se enviarán 20 paquetes desde el Módulo de Control diseñado hacia cada tipo de actuador comercial. La prueba se la realizará para una distancia de 20 metros de cableado dentro de la instalación eléctrica de

un hogar. Se enviarán todos los comandos admitidos por el dispositivo que se esté probando para garantizar total compatibilidad. La prueba se la realiza con una distancia de 20 metros para probar la comunicación a la mayor distancia considerada en este proyecto.

Los resultados de las pruebas para el módulo “SocketRocket” se presentan en la Tabla 4.16, los resultados de las pruebas para el módulo “Lamp Module” se resumen en la Tabla 4.17 y los resultados de las pruebas para el “Appliance Module” se encuentran en la Tabla 4.18.

Tabla 4.16. Pruebas con el módulo “SocketRocket”

<b>Comando</b>	<b>Total de paquetes enviados</b>	<b>Recibidos</b>	<b>Perdidos</b>	<b>Porcentaje de éxito</b>
On	20	20	0	100.00%
Off	20	20	0	100.00%
All lights on	20	20	0	100.00%
All lights off	20	20	0	100.00%
All off	20	20	0	100.00%

Tabla 4.17. Pruebas con el “Lamp Module”

<b>Comando</b>	<b>Total de paquetes enviados</b>	<b>Recibidos</b>	<b>Perdidos</b>	<b>Porcentaje de éxito</b>
On	20	20	0	100.00%
Off	20	20	0	100.00%
All lights on	20	20	0	100.00%
All lights off	20	0	20	0.00%
All off	20	20	0	100.00%
Dim	400	397	3	99.25%
Bright	400	398	2	99.50%

Tabla 4.18. Pruebas con el “Appliance Module”

<b>Comando</b>	<b>Total de paquetes enviados</b>	<b>Recibidos</b>	<b>Perdidos</b>	<b>Porcentaje de éxito</b>
On	20	20	0	100.00%
Off	20	20	0	100.00%
All off	20	20	0	100.00%

Se puede observar que en la Tabla 4.16 y en la Tabla 4.18 los resultados son completamente satisfactorios con un porcentaje de éxito del 100% en el envío y recepción de los datos, con lo cual se puede demostrar que la compatibilidad entre el Módulo de Control diseñado y los módulos “socketRocket” y “Appliance Module” es completa. En la Tabla 4.17 se observa que el 100% de los comandos On, Off, All lights on y All off se recibieron correctamente. Para el Caso del “Lamp Module” no se recibió ningún comando All lights off. Por este motivo se puede concluir que dicho módulo no admite comandos All lights off a pesar de que está diseñado para el funcionamiento exclusivo con elementos de iluminación. La última afirmación se confirma por el hecho de que en las pruebas de la Tabla 4.16, Tabla 4.6, Tabla 4.7, Tabla 4.8, Tabla 4.9 y Tabla 4.10 ya se verificó el correcto envío del comando All lights off. Finalmente, en la misma Tabla 4.17 se puede observar que para los comandos Dim y Brights, se pierden algunos paquetes. Si éstos resultados se comparan con los obtenidos en la Tabla 4.6, Tabla 4.7, Tabla 4.8, Tabla 4.9 y Tabla 4.10, se puede concluir que el problema no es el envío de los paquetes desde el Módulo de Control sino que el problema es propio del receptor análogo “Lamp Module”. Esto muestra que la eficacia de los módulos digitales es superior a la de los módulos analógicos para tramas que deben ser enviadas consecutivamente, como es el caso de los comandos Dim y Bright. También se debe tener cuidado al emplear este tipo de módulos para utilizar comandos como Extended Data y Extended Code ya que se podrían perder datos si no se introduce una pequeña pausa entre las tramas.

#### **4.2.2 ENVÍO DE COMANDOS DESDE MÓDULOS COMERCIALES HACIA LOS ACTUADORES DISEÑADOS.**

De igual manera que en el numeral 4.2.1, la realización de esta prueba consistirá en enviar 20 paquetes por comando desde un “Tranceiver” comercial hacia cada tipo de actuador diseñado. La prueba se la realizará para una distancia de 20 metros de cableado dentro de la instalación eléctrica de un hogar. El módulo comercial que se utilizará para esta prueba es el “Wireless Transceiver TM751”. Es un módulo capaz de recibir comandos a través de un control remoto y transformarlos en códigos X-10 que son introducidos en la red eléctrica. Debido a

que el control remoto solo tiene implementados los comandos On, Off, Dim y Bright solo se pueden realizar pruebas con éstos comandos. De todas maneras, las pruebas con los demás comandos no son necesarias ya que todos los comandos implementados ya fueron probados en el numeral 4.1 y lo que se trata de comprobar en esta prueba es la conectividad entre un módulo comercial emisor de comandos X-10 y los dispositivos actuadores diseñados. La prueba se la realiza con una distancia de 20 metros para probar la comunicación a la mayor distancia considerada en éste proyecto.

Los resultados de las pruebas para el Módulo tipo On / Off” se presentan en la Tabla 4.19, los resultados de las pruebas para el Módulo atenuable se resumen en la Tabla 4.20 y los resultados de las pruebas para el Módulo de potencia se encuentran en la Tabla 4.21.

Tabla 4.19. Pruebas con el Módulo tipo On / Off

<b>Comando</b>	<b>Total de paquetes enviados</b>	<b>Recibidos</b>	<b>Perdidos</b>	<b>Porcentaje de éxito</b>
On	20	20	0	100.00%
Off	20	20	0	100.00%

Tabla 4.20 Pruebas con el Módulo atenuable

<b>Comando</b>	<b>Total de paquetes enviados</b>	<b>Recibidos</b>	<b>Perdidos</b>	<b>Porcentaje de éxito</b>
On	20	20	0	100.00%
Off	20	20	0	100.00%
Dim	400	400	0	100.00%
Bright	400	400	0	100.00%

Tabla 4.21 Pruebas con el Módulo de potencia

<b>Comando</b>	<b>Total de paquetes enviados</b>	<b>Recibidos</b>	<b>Perdidos</b>	<b>Porcentaje de éxito</b>
On	20	20	0	100.00%
Off	20	20	0	100.00%

En la Tabla 4.19, Tabla 4.20 y Tabla 4.21 se puede observar que el porcentaje de éxito de los comandos recibidos es del 100% para todos los casos. Esto demuestra que los dispositivos diseñados son completamente compatibles con el módulo comercial “Wireless Transceiver TM751” y que se puede utilizar dicho módulo para aprovechar de mejor manera las funciones de los dispositivos X-10 diseñados, brindando mayor confort al usuario.

### **4.3 PRUEBAS DE FUNCIONAMIENTO DE LA PROGRAMACIÓN DE SECUENCIAS**

Estas pruebas tratarán de verificar que los algoritmos de escritura y lectura de comandos en memoria no volátil funcionen con eficacia y precisión. Las secuencias de prueba involucrarán a todos los tipos de dispositivos y se realizarán para todos los tipos de comandos definidos en el Numeral 3.1.5.1, en los cuales se especifica la manera de ejecutar los comandos programados.

Para realizar las pruebas de programación de secuencias se elaborarán rutinas prácticas y se verificará su eficacia y precisión. La primera secuencia de comandos se encargará de ejecutar una rutina para automatizar las tareas matutinas y permitirá hacer el desayuno de una manera mucho más sencilla, sin necesidad de preocuparse de los electrodomésticos. La otra secuencia de prueba servirá para simular la presencia de personas en el hogar para los días en que la familia decida irse de vacaciones.

#### **4.3.1 SECUENCIA 1: RUTINA PARA AUTOMATIZAR LAS TAREAS MATUTINAS Y REALIZAR EL DESAYUNO.**

Estas rutinas de automatización permitirán automatizar tareas que cualquier persona realiza todas las mañanas. Dichas rutinas estarán programadas por defecto pero pueden ser modificadas fácilmente por el usuario. Estas rutinas preprogramadas también servirán de ejemplo para permitir al usuario entender más fácilmente la manera de programar los comandos X-10. La rutina a diseñar está pensada para controlar un termostato, las luces de la cocina y el dormitorio,

una televisión en el dormitorio, una cafetera, una sandwichera y una licuadora. Para la elaboración de esta rutina se asume que el usuario se levanta a las 7:30, se baña hasta las 7:45, toma su desayuno a las 7:50 y sale de su hogar a las 8:00. Con estas consideraciones iniciales, la rutina de control hará las siguientes tareas:

- Encender termostato a las 6:30
- Encender luces de dormitorio a las 7:30:00
- Encender televisión de dormitorio a las 7:30:01
- Encender luces de la cocina a las 7:31
- Apagar luces de la cocina a las 7:33
- Encender cafetera a las 7:38
- Encender sandwichera a las 7:40
- Apagar dispositivos de ambiente cocina a las 7:45
- Apagar televisión de dormitorio a las 7:45:01
- Apagar termostato a las 7:47
- Encender luces de la cocina a 7:50
- Apagar todas las luces de ambiente cocina a 8:05

Para la ejecución de esta rutina se configuraran los dispositivos como se indica en la Tabla 4.22.

Tabla 4.22. Configuración de actuadores para ejecutar la Secuencia 1.

<b>Dispositivo controlado</b>	<b>Tipo de Actuador</b>	<b>Dirección X-10</b>
Luz de dormitorio	Iluminación atenuable	A1
Televisión dormitorio	De potencia	A2
Termostato	De potencia	A3
Luces de cocina	Iluminación tipo On / Off	B1
Cafetera	De potencia	B2
Sanduchera	De potencia	B3

Para facilitar la simulación y la toma de datos se reemplazó el termostato por un foco de 150W. La secuencia se la configuró para ejecutarse de lunes a viernes y los datos tomados se presentan en la Tabla 4.23.

Tabla 4.23. Pruebas de ejecución de la Secuencia 1 – Día 1

<b>Dispositivo controlado</b>	<b>Hora programada</b>	<b>Comando X-10</b>	<b>Hora de ejecución</b>	<b>Error en Segundos</b>
Termostato	6:30:00	On	6:30:00	0
Luz de dormitorio	7:30:00	On	7:30:00	0
Televisión Dormitorio	7:30:01	On	7:30:01	0
Luces de cocina	7:31:00	On	7:31:00	0
Luces de cocina	7:33:00	Off	7:33:00	0
Cafetera	7:38:00	On	7:38:00	0
Sanduchera	7:40:00	On	7:40:00	0
Luces de cocina	7:45:00	All Off	7:45:00	0
Cafetera	7:45:00	All Off	7:45:00	0
Sanduchera	7:45:00	All Off	7:45:00	0
Televisión Dormitorio	7:45:01	Off	7:45:01	0
Termostato	7:47:00	Off	7:47:00	0
Luces de cocina	7:50:00	On	7:50:00	0
Luces de cocina	8:05:00	All Off	8:05:00	0
Cafetera	8:05:00	All Off	8:05:00	0
Sanduchera	8:05:00	All Off	8:05:00	0

Las Secuencia 1 se la repitió durante 5 días para poder observar si existe algún error en la ejecución de los comandos. Los resultados de estas cinco pruebas se las resume en la Tabla 4.24.

A partir de los resultados obtenidos en la Tabla 4.24 se puede concluir que la precisión del reloj programado es muy buena considerando la aplicación. No hubo ningún error en los tiempos de ejecución de ningún comando de la Secuencia 1 durante una semana de operación, de lunes a viernes. En lo que se refiere a tiempos de ejecución se podrían tolerar errores de algunos segundos ya que es muy difícil que el usuario note la diferencia en este tipo de aplicaciones. Sin embargo, se pudo constatar que durante una semana no existió ningún error en la ejecución de los comandos ni en su tiempo de ejecución.

Tabla 4.24. Errores en ejecución de la Secuencia 1

Dispositivo controlado	Error en segundos				
	Día 1	Día 2	Día 3	Día 4	Día 5
Termostato	0	0	0	0	0
Luz de dormitorio	0	0	0	0	0
Televisión Dormitorio	0	0	0	0	0
Luces de cocina	0	0	0	0	0
Luces de cocina	0	0	0	0	0
Cafetera	0	0	0	0	0
Sanduchera	0	0	0	0	0
Luces de cocina	0	0	0	0	0
Cafetera	0	0	0	0	0
Sanduchera	0	0	0	0	0
Televisión Dormitorio	0	0	0	0	0
Termostato	0	0	0	0	0
Luces de cocina	0	0	0	0	0
Luces de cocina	0	0	0	0	0
Cafetera	0	0	0	0	0
Sanduchera	0	0	0	0	0

#### 4.3.2 SECUENCIA 2: SIMULACIÓN DE PRESENCIA EN EL HOGAR.

La simulación de presencia está pensada para confundir a los delincuentes que quieran robar el domicilio en el cual este instalado este sistema. La simulación de presencia se la puede realizar cuando la familia salga de vacaciones e incluso para cuando la casa quede abandonada durante la jornada laboral. De esta manera será difícil que los delincuentes logren determinar los horarios de la familia y de esta manera disuadirlos de intentar robar dicho hogar.

La rutina deberá encender luces y algunos artefactos que produzcan ruido por intervalos que serían tomados como “normales” por cualquier observador. La Secuencia 2 encenderá las luces del dormitorio, la cocina y la sala de la manera en que lo hacen habitualmente los habitantes de dicho hogar todas las mañanas. A medio día encenderá las luces del comedor y posteriormente la televisión del

comedor a la hora de un programa de televisión. A la noche encenderá las luces de la sala y del comedor. Luego apagará dichas luces y encenderá la televisión del dormitorio y la apagará a la hora en que normalmente duermen los ocupantes. Los comandos específicos a ejecutarse son:

Encender luces de dormitorio a las 7:30:00  
 Encender televisión de dormitorio a las 7:30:01  
 Encender luces de la cocina a las 7:31  
 Apagar todo en ambiente dormitorio a las 7:45:00  
 Apagar todas las luces de ambiente cocina a las 8:05  
 Encender luces de comedor a las 13:04  
 Encender televisión de comedor a las 13:07  
 Apagar todo en ambiente social a las 14:12  
 Encender todas las luces de ambiente social a las 18:33  
 Apagar todas las luces de ambiente social a las 19:07  
 Encender televisión de dormitorio a las 19:09  
 Apagar televisión de dormitorio a las 21:02

Para la ejecución de esta rutina se configuraran los dispositivos como se indica en la Tabla 4.25.

Tabla 4.25. Configuración de actuadores para ejecutar la Secuencia 2.

<b>Dispositivo controlado</b>	<b>Tipo de Actuador</b>	<b>Dirección X-10</b>
Luz de dormitorio	Iluminación atenuable	A1
Televisión dormitorio	De potencia	A2
Luces de cocina	Iluminación tipo On / Off	B1
Luz de comedor	Iluminación tipo On / Off	C1
Televisión comedor	De potencia	C2
Luz sala	Iluminación atenuable	C3

La secuencia se la configuró para ejecutarse todos los días y los datos tomados durante el primer día se los presenta en la Tabla 4.26

Tabla 4.26. Pruebas de ejecución de la Secuencia 2 – Día 1

<b>Dispositivo controlado</b>	<b>Hora programada</b>	<b>Comando X-10</b>	<b>Hora de ejecución</b>	<b>Error en Segundos</b>
Luz de dormitorio	7:30:00	On	7:30:00	0
Televisión Dormitorio	7:30:01	On	7:30:01	0
Luces de cocina	7:31:00	On	7:31:00	0
Ambiente dormitorio	7:45:00	All Off	7:45:00	0
Luces de cocina	8:05:00	All lights Off	8:05:00	0
Luces de comedor	13:04:00	On	13:04:00	0
Televisión comedor	13:07:00	On	13:07:00	0
Ambiente Social	14:12:00	All Off	14:12:00	0
Ambiente social	18:33:00	All lights On	18:33:00	0
Ambiente Social	19:07:00	All lights Off	19:07:00	0
Televisión Dormitorio	19:09:00	On	19:09:00	0
Televisión Dormitorio	21:02:00	Off	21:02:00	0

Las Secuencia 2 se la repitió durante 7 días para verificar el funcionamiento adecuado de todos los comandos empleados así como su precisión de ejecución. Los resultados de estas siete pruebas se las resume en la Tabla 4.27.

Tabla 4.27. Errores en la ejecución de la Secuencia 2.

<b>Dispositivo controlado</b>	<b>Error en segundos</b>						
	<b>Día 1</b>	<b>Día 2</b>	<b>Día 3</b>	<b>Día 4</b>	<b>Día 5</b>	<b>Día 6</b>	<b>Día 7</b>
Luz de dormitorio	0	0	0	0	0	0	0
Televisión Dormitorio	0	0	0	0	0	0	0
Luces de cocina	0	0	0	0	0	0	0
Ambiente dormitorio	0	0	0	0	0	0	0
Luces de cocina	0	0	0	0	0	0	0
Luces de comedor	0	0	0	0	0	0	0
Televisión comedor	0	0	0	0	0	0	0
Ambiente Social	0	0	0	0	0	0	0
Ambiente social	0	0	0	0	0	0	0
Ambiente Social	0	0	0	0	0	0	0
Televisión Dormitorio	0	0	0	0	0	0	0
Televisión Dormitorio	0	0	0	0	0	0	0

Luego de analizar los resultados obtenidos en la Tabla 4.27 se pueden confirmar las conclusiones obtenidas en el Numeral 4.3.1. A pesar de que se podrían tolerar errores de pocos segundos, se puede observar que durante una semana completa de ejecución de la rutina, no existe ningún error ni de ejecución ni de tiempo. Esto confirma que el algoritmo de igualamiento de reloj, la precisión del reloj interno y los algoritmos para escritura y lectura en memoria no volátil funcionan adecuadamente y son capaces de trabajar sin cometer ningún error durante el periodo de una semana. No se realizó ninguna prueba para determinar el tiempo en el cual el reloj se llega a desigular un segundo, pero si se iguala el reloj cada semana está garantizado que no existirá ningún error de temporización. Como el reloj se iguala cada vez que el programa de Interfaz se ejecuta, las actualizaciones se realizarán en un periodo de tiempo menor a una semana la mayor cantidad de veces.

#### **4.4 PRUEBAS DE FUNCIONAMIENTO DE LAS RUTINAS DE AHORRO DE ENERGÍA**

El principal objetivo de estas pruebas es constatar la eficacia de las rutinas de ahorro de energía. Las rutinas de prueba involucraran a todos los tipos de dispositivos y se realizarán utilizando comandos para apagar todos los dispositivos de un ambiente. También provee de comandos para usar cierto tipo de artefactos eléctricos que según estudios son las causas más comunes de desperdicio de energía.

##### **4.4.1 PRINCIPALES CAUSAS DEL DESPERDICIO DE ENERGÍA.**

Según fuentes como la BBC de Londres y la EST (Energy Saving Trust) las causas mas comunes de derroche de energía son: olvidar luces encendidas, dejar cargadores de celulares conectados indefinidamente y dejar aparatos electrónicos en Stand-by. El sistema X-10 diseñado puede ayudar a solucionar los tres tipos

de descuidos fácilmente. Debido a que en este proyecto solo se han sentado las bases para crear una red de domótica completa, no se incluyeron en los objetivos el diseño de sensores ni ningún otro tipo de adquisición de datos pero se sugiere su diseño e implementación para futuros proyectos.

La manera en que el sistema aquí diseñado puede ayudar a la reducción en el desperdicio de energía es mediante la introducción de comandos que se ejecutarán automáticamente para apagar todos los dispositivos electrónicos que se hayan dejado encendidos por accidente. Según las recomendaciones hechas por el artículo "UK 'tops energy wasters league2' "<sup>13</sup>, deberían apagarse todos los equipos electrónicos que no estén en uso a excepción de las computadoras que se usen varias veces al día. Por este motivo se recomienda a los usuarios del sistema de domótica diseñado que instalen actuadores X-10 en los artefactos electrónicos mencionados, pues son causa frecuente de desperdicio de energía, para permitir su desconexión de manera automática en caso de que se olvide desconectarlos. También se recomienda instalar actuadores X-10 para iluminación en los cuartos de los niños ya que ellos olvidan las luces encendidas con frecuencia. Finalmente, se aconseja instalar un Módulo de potencia en el termostato para permitir que el sistema de control lo encienda solo a las horas que sea necesario. Esto último permitiría un gran ahorro de energía sin que se cause ningún inconveniente para el usuario.

#### **4.4.2 DISEÑO Y PRUEBA DE LAS RUTINAS DE AHORRO DE ENERGÍA.**

Las rutinas para ahorro de energía se las diseñó tomando en cuenta los estudios mencionados en el numeral anterior. Debido a que los mayores derrochadores de energía son los niños, la rutina de ahorro de energía apagará todos los aparatos eléctricos de sus cuartos a la hora en que ellos se van a la escuela y también algunos minutos después de que se van a dormir. También se apagará el termostato durante la noche cuando se lo haya dejado encendido. Otro descuido frecuente es dejar encendidos los aparatos eléctricos del dormitorio cuando se

---

<sup>13</sup> <http://news.bbc.co.uk/1/hi/uk/6075794.stm>

sale para el trabajo, así que la rutina de ahorro de energía apagará todos los dispositivos que se queden encendidos luego de la hora a la cual el usuario sale para su trabajo. Durante la noche se apagarán todos los aparatos electrónicos conectados en la sala y comedor y las luces de la cocina. El usuario tiene total libertad de cambiar los horarios de apagado e incluso de aumentar comandos si es necesario. Los comandos programados son:

Apagar todo en dormitorio infantil a las 7:45

Apagar todo en dormitorio infantil a las 20:30

Apagar termostato a las 21:00

Apagar todo en dormitorio principal a las 8:00

Apagar todo en ambiente social a las 22:00

Apagar todas las luces de la cocina a las 21:10

Si el usuario planea tener una reunión durante la noche, lo único que tiene que hacer es desactivar el comando que apaga las luces en el ambiente social y volver a activarlo al día siguiente.

Las configuraciones necesarias para ejecutar la rutina de ahorro de energía se pueden observar en la Tabla 4.28. Ya que se utilizan comandos para apagar todas la luces de un ambiente o apagar todo dentro de un ambiente es necesario que todos los dispositivos X-10 dentro de un mismo ambiente estén configurados con el mismo código de casa.

Tabla 4.28. Ejemplo de configuración para la Rutina de ahorro de energía.

<b>Dispositivo controlado</b>	<b>Tipo de Actuador</b>	<b>Dirección X-10</b>
Luz de dormitorio	Iluminación atenuable	A1
Televisión dormitorio	De potencia	A2
Termostato	De potencia	A3
Luces de cocina	Iluminación tipo On / Off	B1
Cafetera	De potencia	B2
Sanduchera	De potencia	B3
Luz de comedor	Iluminación tipo On / Off	C1
Televisión comedor	De potencia	C2
Luz sala	Iluminación atenuable	C3

Los resultados de la prueba de la rutina de ahorro de energía se pueden observar en la Tabla 4.29. El ambiente social incluye a la sala y al comedor.

Tabla 4.29. Pruebas de la Rutina de ahorro de energía.

<b>Dispositivo controlado</b>	<b>Hora programada</b>	<b>Comando X-10</b>	<b>Hora de ejecución</b>	<b>Error en Segundos</b>
Ambiente dormitorio Niños	7:45:00	On	7:45:00	0
Ambiente dormitorio Principal	8:00:00	On	8:00:00	0
Ambiente dormitorio Niños	20:30:00	On	20:30:00	0
Termostato	21:00:00	All Off	21:00:00	0
Luces de cocina	21:10:00	All lights Off	21:10:00	0
Ambiente social	22:00:00	On	22:00:00	0

Luego de ejecutar las pruebas para la rutina de ahorro de energía se verificó que todos los comandos se ejecutaron apropiadamente y que no causaron interferencias con la rutina diaria de los ocupantes del hogar. La rutina de ahorro de energía se dejó instalada dentro del hogar de prueba por una semana, durante la cual no se presentó ningún problema. Se recomienda que cada usuario configure las rutinas de ahorro de energía de acuerdo a su rutina diaria.

## **CAPÍTULO 5**

### **CONCLUSIONES Y RECOMENDACIONES**

## **CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES**

El presente proyecto ha permitido conocer a fondo el funcionamiento del protocolo X-10 junto con sus beneficios y sus debilidades. En este capítulo se busca expresar los puntos más importantes en lo que se refiere a experiencia adquirida y sugerencias para implementar exitosamente un sistema de domótica X-10. También se busca plantear las mejoras que se podrían realizar mediante nuevos proyectos que involucren otras áreas de conocimiento, una vez que se han desarrollado las herramientas básicas para la implementación de sistemas de domótica sobre protocolo X-10.

### **5.1 CONCLUSIONES**

1. Los sistemas de domótica basados en protocolo X-10 ofrecen una gran compatibilidad entre fabricantes ya que se encuentran estandarizadas tanto las características eléctricas como lógicas del protocolo.
2. El protocolo X-10 es fácilmente escalable ya que contiene instrucciones tanto para enviar datos como para enviar comandos extendidos, los cuales pueden ser establecidos por cada OEM. De esta manera se comparten las instrucciones básicas que permiten la completa funcionalidad del protocolo, y a la vez se permite tener comandos mas avanzados, si la aplicación lo requiere.
3. La principal debilidad del protocolo es que cualquier persona puede comandar cualquier dispositivo X-10, sin importar quien sea el propietario o el fabricante, ya que no se realiza ninguna autenticación de usuario. Cualquier persona con un transmisor de códigos X-10 puede manejar cualquier red X-10 siempre y cuando logre instalar su dispositivo en algún tomacorriente o cable conectado a dicha red. Es necesario realizar mejoras al protocolo X-10 para brindar alguna seguridad al protocolo.

4. Las señales X-10 pueden ser afectadas por interferencias externas. Por este motivo es recomendable que cualquier equipo que tenga que leer señales X-10 posea un buen sistema de filtrado de señales para permitir que las tramas X-10 sean leídas correctamente y evitar que la capa de Enlace de datos (según el modelo OSI) deseche la trama y se pierdan datos o comandos.
5. El protocolo X-10 es ideal para realizar tareas de control de baja complejidad dentro de un hogar puesto que utiliza el cableado ya existente y no es necesario preocuparse por la interferencia electromagnética proveniente de muchos de los aparatos modernos que si interferirían con sistemas de domótica basados en Bluetooth. Dispositivos inalámbricos, que cada vez se encuentran con mas frecuencia en los hogares, tales como: accesorios Bluetooth para teléfonos celulares o para computadoras, teléfonos inalámbricos, access points trabajan en bandas similares a las que utilizan los sistemas de control Bluetooth causando interferencia y dificultades de configuración que no podrían ser superadas por muchas personas que no conocen sobre ese tipo de tecnología.
6. Es complicado realizar una red X-10 para tareas complicadas ya que la comunicación se la realiza a través de un bus compartido y en modo half duplex. Para tareas de control que requieran el envío de gran cantidad de datos se necesita un protocolo de mayor ancho de banda que el protocolo X-10, cuyo ancho de banda es de 60 bps.
7. En ésta aplicación en particular, los dispositivos X-10 digitales tienen varias ventajas frente a los dispositivos analógicos debido a que las tareas son realizadas mediante software. Los dispositivos digitales permiten mayor flexibilidad y dan la posibilidad de que se realicen mejoras significativas sin necesidad de realizar ningún cambio de hardware.
8. Para las tareas de lectura de comandos Dim o Bright los dispositivos digitales tienen un mejor desempeño que los dispositivos analógicos ya

que los primeros no necesitan los 3 ciclos de espera que requieren los dispositivos analógicos. Por esta razón los dispositivos analógicos pierden algunos comandos Dim o bright mientras que los digitales no pierden ninguno.

9. Del resultado de todas las pruebas se puede extraer como conclusión final que éste trabajo ha cumplido totalmente con los objetivos planteados.

## **5.2 RECOMENDACIONES**

1. Para la instalación de un sistema de domótica basado en el protocolo X-10 es conveniente instalar filtros en el cable de alimentación del hogar, para evitar que sistemas similares de los vecinos afecten el sistema propio.
2. Si el hogar cuenta con alimentación bifásica o trifásica es conveniente instalar puentes de señal para poder controlar dispositivos conectados a una fase distinta a la cual esta conectada el Módulo de Control.
3. Sería conveniente que se realicen proyectos complementarios al presente, para brindar funciones de comunicación digital inalámbrica para permitir el control remoto de las funcionalidades del proyecto, así como para permitir la instalación de sensores en sitios donde no existan tomacorrientes. Estas funciones permitirían realizar un control mas eficiente y darían mayor confort al usuario. Se recomienda que la implementación de dichos sistemas de comunicación se la realice por medios físicos poco utilizados en los hogares modernos
4. Para realizar sistemas de seguridad en base al protocolo X-10 Pro es necesario implementar funciones de autenticación al protocolo X-10 Pro, para evitar que cualquier usuario no autorizado pueda lograr acceder al sistema controlado. Por este motivo no se recomienda que el protocolo X-10 básico sea utilizado para comandar cerraduras eléctricas o sensores

destinados a brindar servicios de seguridad. Las modificaciones sugeridas tienen que ver con establecer códigos de autenticación enviados a través de la función para enviar comandos extendidos y deshabilitar los códigos estándar del protocolo X-10, para los dispositivos relacionados con los servicios de seguridad.

5. Se recomienda instalar Módulos de potencia en dispositivos de alto consumo de energía eléctrica como por ejemplo en termostatos. Al instalar un Módulo de potencia en el termostato se le permite al sistema de control encenderlo solo durante el tiempo necesario, permitiendo un gran ahorro de energía sin que se cause ningún inconveniente para el usuario.
6. El protocolo X-10 Pro no es apropiado para sistemas de control en la industria o donde se necesite garantizar que la información fue recibida correctamente, puesto que éste protocolo no permite verificar que el receptor recibió la información. La única garantía que brinda este protocolo es que se puede verificar con un alto nivel de seguridad que el comando recibido es correcto o no. Se podría implementar una mejora en el protocolo para permitir al controlador saber si el dato fue recibido, pero es necesario recordar que es un protocolo de transición en modo half duplex lo que reduce su ancho de banda.
7. El uso del protocolo X-10 Pro, tal como esta planteado, solo es recomendable para la domótica en hogares y en oficinas pequeñas donde el número de tareas de control que se requiere realizar es limitado. Es decir, cuando no se requiere enviar mas de una trama X-10 por segundo como máximo.

## REFERENCIAS BIBLIOGRÁFICAS

Schweber, William; Electronic Communication Systems. Segunda Edición. Prentice Hall. New Jersey, Ohio, Estados Unidos. 1996.

Coughlin, Robert; Driscoll, Frederick; Amplificadores Operacionales y Circuitos Integrados Lineales. Prentice Hall. Quinta edición. México. 1999.

NTE Electronics inc; Semiconductors. Décima edición. NTE Electronics Inc. Estados Unidos. 2002

<http://www.x10.com/technology1.htm>

<http://www.x10pro.com/pro/pdf/technote.pdf>

[http://standards.ieee.org/announcements/pr\\_p1675.html](http://standards.ieee.org/announcements/pr_p1675.html)

[http://energiaycomputacion.univalle.edu.co/edicion19/revista19\\_10a.phtml](http://energiaycomputacion.univalle.edu.co/edicion19/revista19_10a.phtml)

[http://wolfstone.halloweenhost.com/TechBase/x10int\\_X10Intro.html](http://wolfstone.halloweenhost.com/TechBase/x10int_X10Intro.html)

<http://www.smarthome.com/aboutx10.html>

<http://www.computerhope.com/help/serial.htm>

<http://www.learobotics.com/proyectos/cuadernos/ct1/ct1.html>

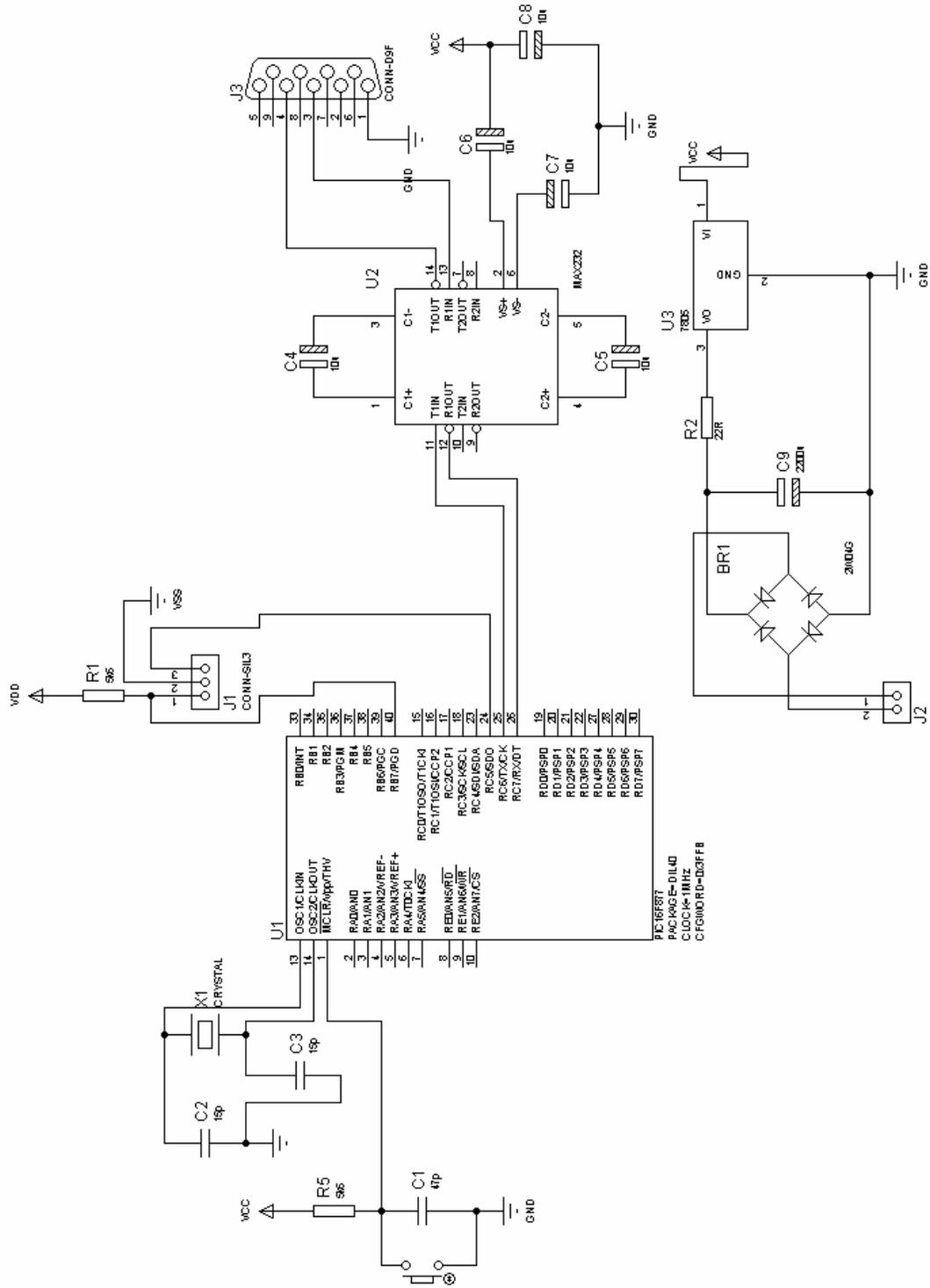
<http://www.est.org.uk/uploads/documents/housingbuildings/CE101.GPG171%20-%20Domestic%20energy%20efficiency%20primer.pdf>

<http://news.bbc.co.uk/1/hi/uk/6075794.stm>

# **ANEXOS**

## **ANEXO 1**

### **DIAGRAMA DEL CIRCUITO DEL MÓDULO DE CONTROL**



## **ANEXO 2**

### **DIAGRAMA DEL CIRCUITO DEL MÓDULO RECEPTOR**



## **ANEXO 3**

### **HOJA DE DATOS DEL CIRCUITO INTEGRADO MAX232**



**ELECTRONICS, INC.**  
**44 FALLAND STREET**  
**BLOOMFIELD, NJ 07003**  
**(973) 748-5089**  
<http://www.nteinc.com>

## NTE7142 Integrated Circuit +5V -Powered, Multichannel RS-232 Driver/Receiver

### Description:

The NTE7142 is a multichannel RS-232 driver/receiver in a 16-Lead DIP type package intended for all EIA/TIA-232E and V.28/V.24 communications interfaces, particularly applications where  $\pm 12V$  is not available.

### Features:

- Operate From Single +5V Power Supply
- Meet All EIA/TIA-232E and V.28 Specifications
- Multiple Drivers and Receivers
- 3-State Driver and Receiver Outputs

### Applications:

- Portable Computers
- Low-Power Modems
- Interface Translation
- Battery-Powered RS-232 Systems
- Multidrop RS-232 Networks

### Absolute Maximum Ratings: (Note 1)

Supply Voltage, $V_{CC}$ .....	-0.3V to +6V
Supply Voltage, $V_+$ .....	( $V_{CC} - 0.3V$ ) to +14V
Supply Voltage, $V_-$ .....	+0.3V to -14V
Input Voltage, $T_{IN}$ .....	-0.3V to ( $V_{CC} - 0.3V$ )
Input Voltage, $R_{IN}$ .....	$\pm 30V$
Output Voltage (Note 2), $T_{OUT}$ .....	$\pm 15V$
Output Voltage, $R_{OUT}$ .....	-0.3V to ( $V_{CC} + 0.3V$ )
Driver/Receiver Output Short Circuited to GND .....	Continuous
Continuous Power Dissipation ( $T_A = +70^\circ C$ ), $P_D$ .....	842mW
Dreate Above $+70^\circ C$ .....	10.53mW/ $^\circ C$
Operating Temperature Range, $T_{opr}$ .....	$0^\circ$ to $+70^\circ C$
Storage Temperature Range, $T_{stg}$ .....	$-65^\circ$ to $+160^\circ C$
Lead Temperature (During Soldering, 10sec), $T_L$ .....	$+300^\circ C$

Note 1. Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational section of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Note 2. Input voltage measured with  $T_{OUT}$  in high-impedance state,  $V_{CC} = 0V$ .

## **ANEXO 4**

### **HOJA DE DATOS DEL CIRCUITO INTEGRADO 4N25 (NTE 3040)**



**ELECTRONICS, INC.**  
**44 FARLAND STREET**  
**BLDGMFIELD, NJ 07003**  
**(973) 748-9000**  
<http://www.ntefno.com>

## NTE3040 Optoisolator NPN Transistor Output

### Description:

The NTE3040 is a gallium arsenide, infrared emitting diode in a 6-Lead DIP type package coupled with a silicon phototransistor.

### Applications:

- Power Supply Regulators
- Digital Logic Inputs
- Microprocessor Inputs

### Absolute Maximum Ratings: ( $T_A = +25^\circ\text{C}$ , unless otherwise specified)

#### Infrared Emitting Diode

Power Dissipation, $P_D$ .....	150mW
Derate above $25^\circ\text{C}$ ambient .....	2.0mW/ $^\circ\text{C}$
Forward Current, $I_C$	
Continuous .....	100mA
Peak (Pulse Width 1 $\mu\text{sec}$ , 300pps) .....	3A
Reverse Voltage, $V_R$ .....	6V

#### Phototransistor

Power Dissipation, $P_D$ .....	150mW
Derate above $25^\circ\text{C}$ ambient .....	2.0mW/ $^\circ\text{C}$
Collector-to-Emitter Voltage, $V_{CE0}$ .....	30V
Collector-to-Base Voltage, $V_{CBO}$ .....	70V
Emitter-to-Collector Voltage, $V_{ECO}$ .....	7V

#### Total Device

Power Dissipation, $P_D$ .....	250mW
Derate above $25^\circ\text{C}$ ambient .....	3.3mW/ $^\circ\text{C}$
Storage Temperature, $T_{stg}$ .....	$-55^\circ$ to $+150^\circ\text{C}$
Operating Temperature, $T_{opr}$ .....	$-55^\circ$ to $+100^\circ\text{C}$
Lead Soldering Temperature (10 seconds) .....	$+260^\circ\text{C}$

### Electrical Characteristics: ( $T_A = +25^\circ\text{C}$ , Note 1, unless otherwise specified)

Parameter	Symbol	Test Conditions	Min	Typ	Max	Unit
<b>Infrared Emitting Diode</b>						
Input Forward Voltage	$V_F$	$I_F = 10\text{mA}$	-	1.18	1.50	V
Reverse Leakage Current	$I_R$	$V_R = 6\text{V}$	-	0.001	10	$\mu\text{A}$

Note 1. Typical values at  $T_A = +25^\circ\text{C}$ .

## **ANEXO 5**

**HOJA DE DATOS DEL CIRCUITO INTEGRADO LM324N  
(NTE987)**



## NTE987/NTE987SM Integrated Circuit Quad, Low Power OP Amp

### Description:

The NTE987 and NTE987SM are low cost quad operational amplifiers with true differential inputs. These have several distinct advantages over standard operational amplifier types in single supply applications. The quad amplifier can operate at supply voltages as low as 3V or as high as 32V with quiescent currents about one fifth of those associated with the NTE941 (on a per amplifier basis). The common mode input range includes the negative supply, thereby eliminating the necessity for external biasing components in many applications. The output voltage range also includes the negative power supply voltage.

### Features:

- Available in 14-Lead DIP (NTE987) and Surface Mount, SOIC-14 (NTE987SM)
- Short Circuit Protected Outputs
- True Differential Input Stage
- Single Supply Operation: 3V to 32V
- Four Amplifiers per Package
- Internally Compensated
- Common Mode Range Extends to Negative Supply
- Industry Standard Pin-Outs

### Absolute Maximum Ratings: ( $T_A = +25^\circ\text{C}$ unless otherwise specified)

Power Supply Voltages	
Single Supply, $V_{CC}$ .....	32V
Split Supplies, $V_{CC}$ $V_{EE}$ .....	$\pm 16\text{V}$
Input Differential Voltage Range (Split Power Supplies), $V_{IDR}$ .....	$\pm 32\text{V}$
Input Common Mode Voltage range, $V_{ICR}$ .....	-0.3 to 32V
Input Forward Current ( $V_I < 0.3\text{V}$ , Note 1), $I_{IF}$ .....	50mA
Output Short Circuit Duration, $t_S$ .....	Continuous
Operating Junction Temperature, $T_J$ .....	$+150^\circ\text{C}$
Operating Ambient Temperature Range, $T_A$ .....	$0^\circ$ to $+70^\circ\text{C}$
Storage Temperature Range, $T_{stg}$ .....	$-55^\circ$ to $+125^\circ\text{C}$

Note 1. This input current will only exist when the voltage is negative at any of the input leads. Normal output states will reestablish when the input voltage returns to a voltage greater than -0.3V.

## **ANEXO 6**

**HOJA DE DATOS DEL CIRCUITO INTEGRADO TL084  
(NTE859)**



## NTE859/NTE859SM Integrated Circuit Quad, Low Noise, JFET Input Operational Amplifier

### Description:

The NTE859 (14-Lead DIP) and NTE859SM (SOIC-14 Surface Mount) JFET-input operational amplifiers are low noise amplifiers with low noise input bias, offset currents, and fast slew rate. The low harmonic distortion and low noise make these devices ideally suited as amplifiers for high-fidelity and audio preamplifier applications. Each amplifier features JFET-inputs (for high input impedance) coupled with bipolar output stages all integrated on a single monolithic chip.

### Features:

- Low Power Consumption
- Wide Common-Mode and Differential Voltage Ranges
- Low Input Bias and Offset Currents
- Output Short-Circuit Protection
- Low Total Harmonic Distortion: 0.003% Typ
- Low Noise:  $V_n = 18nV\sqrt{Hz}$  Typ
- High Input Impedance: JFET-Input Stage
- Internal Frequency Compensation
- Latch-Up Free Operation
- High Slew Rate: 13V/ $\mu$ s Typ

### Absolute Maximum Ratings: ( $T_A = 0$ to $+70^\circ\text{C}$ unless otherwise specified)

Supply Voltage (Note 1), $V_{CC}(+)$ .....	18V
Supply Voltage (Note1), $V_{CC}(-)$ .....	-18V
Differential Input Voltage (Note 2), $V_{ID}$ .....	$\pm 30\text{V}$
Input Voltage Range (Note 1, Note 3), $V_{IDR}$ .....	$\pm 15\text{V}$
Duration of Output Short Circuit (Note 4), $t_S$ .....	Unlimited
Power Dissipation ( $T_A = +25^\circ\text{C}$ ), $P_D$ .....	680mW
Derate Above $25^\circ\text{C}$ .....	10mW/ $^\circ\text{C}$
Operating Ambient Temperature Range, $T_A$ .....	$0^\circ$ to $+70^\circ\text{C}$
Storage Temperature Range, $T_{sig}$ .....	$-65^\circ$ to $+150^\circ\text{C}$
Lead Temperature (During Soldering, 1/16" from Case for 10sec), $T_L$ .....	$+260^\circ\text{C}$

Note 1. All voltage values, except differential voltages, are with respect to the midpoint between  $V_{CC}(+)$  and  $V_{CC}(-)$ .

Note 2. Differential voltages are at the non-inverting input pin with respect to the inverting pin.

Note 3. The magnitude of the input voltage must never exceed the magnitude of the supply voltage or 15V, whichever is less.

Note 4. The output may be shorted to GND or to either supply. Temperature and/or supply voltages must be limited to ensure that the dissipation rating is not exceeded.