



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE INGENIERÍA

CONTROL DE DISPOSITIVOS POR X-10

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO MECATRÓNICO

P R E S E N T A N:

CERVANTES REYES DAVID

PELAGIO HERNÁNDEZ GUILLERMO



DIRECTOR DE TESIS:

M.I ULISES MARTÍN PEÑUELAS RIVAS

MÉXICO D.F. AGOSTO 2013



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Agradecimientos.

Quiero agradecer a mi familia quienes en todo momento me han brindado su apoyo y sus consejos para poder seguir adelante.

A todos los demás familiares y amigos que han estado siempre ahí, apoyándome y alentándome, a todos ellos también les agradezco mucho.

A todos los profesores, nuestro asesor de tesis y los sinodales por dirigirnos y por haber empleado parte de su tiempo para darnos una formación.

A la Universidad Nacional Autónoma de México y a la Facultad de Ingeniería por darme la oportunidad de formarme como profesional.

Guillermo Pelagio Hernandez.

A mis padres, quienes siempre han estado a mi lado apoyándome en todos los aspectos de mi vida y han sabido guiarme.

A la facultad de ingeniería, por brindarme los conocimientos que he adquirido.

A nuestro asesor de tesis, por dirigirnos y por haber empleado parte de su tiempo para revisar nuestro trabajo.

A los sinodales, por el tiempo dedicado en la revisión de esta tesis.

*Finalmente, a todos mis profesores, quienes pusieron su empeño en mi educación profesional.
Gracias*

David Cervantes Reyes.

Resumen

En el presente trabajo se presenta el desarrollo de un sistema de automatización para el hogar utilizando el protocolo X-10, mediante el cual se envían las señales de control a través del 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.

El sistema planteado está dividido en dos módulos, el módulo emisor y los módulos receptores. El módulo emisor es controlado a través de una interfaz para PC y para un dispositivo móvil estas permiten tener el control de cada uno de los elementos de la red X-10, ya sea de manera alámbrica por conexión USB o inalámbrica a través de bluetooth. La interfaz de la PC está diseñada de tal forma que el usuario pueda cambiar el nombre del dispositivo así como la imagen que lo identifica. Los módulos receptores están diseñados para filtrar y demodular las señales X-10 que detecten viajando por la red eléctrica para luego determinar si son los destinatarios del paquete y toman las acciones de control apropiadas.

Para lograr exponer lo anterior, al inicio de este documento se mencionan los antecedentes de la domótica, así como los diferentes tipos de arquitecturas, medios de transmisión y protocolos que se encuentran actualmente en uso; todo esto para finalmente abordar en el protocolo elegido, exponiendo aspectos importantes del mismo, tales como el formato de la información enviada, tiempos y tolerancias de envío, cantidad de dispositivos a controlar, etc. A continuación se expone el desarrollo y la implementación de los elementos que componen al sistema domótico en cuestión, explicando su estructura y funcionamiento.

Como resultado del sistema propuesto, al final de este documento se muestran los resultados obtenidos a partir de las pruebas hechas con los módulos antes mencionados.

Finalmente se presentan diversas alternativas, con el fin de mejorar los módulos, hacer más confiable y robusto el sistema, así como las conclusiones del trabajo y algunas recomendaciones para implementar sistemas de este tipo.

Índice

Introducción	1
Objetivo	1
Capítulo 1: Domótica	2
1.1. Antecedentes	2
1.2. Definición y características	3
1.3. Tipos de arquitectura	4
1.3.1. Arquitectura centralizada	4
1.3.2. Arquitectura centralizada modular	5
1.3.3. Arquitectura descentralizada	5
1.3.4. Arquitectura distribuida	6
1.4. Medios de transmisión	6
1.4.1. Red eléctrica convencional	7
1.4.2. Cable coaxial	7
1.4.3. Cable de par trenzado	8
1.4.4. Cable de fibra óptica	8
1.4.5. Bluetooth	9
1.4.6. Wi-Fi	10
1.4.7. ZigBee	11
1.4.8. Infrarrojos	12
1.5. Protocolos de comunicación	13
1.5.1. Protocolo de control EIB	14
1.5.2. Protocolo de control EHS	15
1.5.3. Protocolo de control LonWorks	16
1.5.4. Protocolo de control CEBus	17
1.5.5. Protocolo de control BatiBUS	18
1.5.6. Protocolo X-10	18
1.6. Sistema X-10	19
1.6.1. Fundamentos del protocolo	19
1.6.2. Trama del protocolo	24
1.6.3. Ventajas y desventajas	27
Capítulo 2: Propuesta de diseño	30
Capítulo 3: Diseño del hardware	32
3.1. Módulo emisor	32
3.1.1. Elección del microcontrolador para el Módulo emisor	32
3.1.2. Funcionamiento	33
3.1.2.1. Comunicación a través del USB	33
3.1.2.2. Comunicación a través del Bluetooth	43
3.1.3. Fuente de alimentación	46
3.1.4. Detector de cruce por cero	47
3.1.5. Generador de 120 [kHz]	50
3.2. Módulo receptor	55
3.2.1. Detector de pulsos	56
3.2.2. Detector de cruce por cero	60
3.2.3. Procesamiento de las señales	60
3.2.4. Fuente de alimentación	61
3.2.5. Control de la iluminación	61

Capítulo 4: Diseño del software.....	66
4.1. Aplicación de escritorio	66
4.1.1. Elección del entorno de desarrollo	66
4.1.2. Estructura del programa	66
4.1.3. Interfaz gráfica.....	68
4.1.4. Transferencia de datos entre la aplicación y el módulo emisor.....	72
4.2. Aplicación móvil	72
4.2.1. Elección del entorno de desarrollo.....	73
4.2.2. Estructura del programa	73
4.2.3. Interfaz gráfica.....	75
4.2.4. Transferencia de datos entre la aplicación y el módulo emisor.....	79
Capítulo 5. Pruebas y resultados	80
5.1. Pruebas del módulo emisor	80
5.1.1. Comunicación mediante cable USB.....	82
5.1.2. Comunicación mediante Bluetooth	88
5.2. Pruebas del módulo receptor	92
Capítulo 6: Conclusiones y recomendaciones	107
6.1. Conclusiones	107
6.2. Recomendaciones	108
Capítulo 7. Trabajo a futuro	109
Anexo A: Esquemáticos	110
Anexo B: Circuitos impresos	113
Referencias.....	114

Introducción

Gracias a los avances tecnológicos generados a lo largo de la historia, los seres humanos han logrado mejorar su nivel de vida, en la que ya no se busca cumplir únicamente las necesidades básicas de supervivencia, también se busca lograr un nivel de vida con confort y control de los alrededores, así como; conocer, manipular y programar el ambiente en el que una persona se desenvuelve.

Ante estas nuevas necesidades, la tecnología se está involucrando en todos los aspectos de la vida de los seres humanos, creando tecnologías o nuevas disciplinas que cumplan con las nuevas necesidades. Así es como surge la domótica, un término que bien se puede definir, como un conjunto de sistemas capaces de automatizar una vivienda y que pretende introducir el control automático a los hogares; para brindar mayor confort y facilitar las tareas cotidianas de la vida.

Debido a lo anterior, los diversos fabricantes de productos electrónicos relacionados con la domótica, llevan años desarrollando productos que permitan administrar algunos de los aparatos existentes dentro de un hogar, pero a todas estas funciones se han sumado nuevas instalaciones que pueden llegar a ser muy costosas y complejas, debido a que para la transmisión de información entre los dispositivos que conforman la red domótica, se necesita de un cableado o de una transmisión inalámbrica.

Sin embargo, no todos los sistemas domóticos tienen los problemas anteriores, 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 otras tecnologías 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 hay que realizar grandes cambios para su instalación.

Objetivo

El objetivo de este trabajo consiste en el desarrollo e implementación de los elementos necesarios que nos permitan controlar dispositivos conectados a la red eléctrica del hogar, haciéndolo a través del protocolo X-10.

Capítulo 1. Domótica

1.1. Antecedentes

La domótica se conoce y se aplica desde los años 80, teniendo su aparición en Francia con el término *Domotique*; su desarrollo más importante ha tenido lugar en países como EE.UU. y Japón. En los últimos años a esta corriente se han unido países europeos (especialmente los nórdicos), impulsados por las nuevas tecnologías en comunicaciones.

En el año de 1987, apareció un sistema llamado *X-10*, desarrollado por *Pico Electronics Ltd.* una empresa instalada en Escocia; más tarde, a principios de los ochentas en Japón, se empezó a prestar un mayor interés en la introducción de la domótica para hacer posible la renovación del parque electrónico, y con ello conseguir una reducción en el gasto energético, ayudando a llevar la crisis de dicho sector.

Por otro lado, Estados Unidos lanzó un proyecto en 1984 llamado *Smart House*; dicho proyecto pretendía unir en un solo cable, todos los tipos de cable utilizados en una vivienda. En este mismo año también surge el sistema *CEBus*, (*Consumer Electronics Bus*) lanzado por la empresa *Electronics Industry Association*.

Más tarde en el año de 1985, en Europa y bajo la iniciativa del programa *EUREKA*, se lanza un proyecto llamado *Integrated Home Systems*, el cual consistió en reunir a las siete empresas de electrónica más importantes del mercado, con el objetivo de poner a punto una red doméstica con normas de utilización comunes.

A partir de los proyectos mencionados anteriormente, surge en Japón la normativa de bus doméstico, denominada *HBS (Home Bus System)* y presentada en el año de 1987; poco después; en el año de 1989, en Europa se crea el proyecto *Home Systems* dentro del programa *ESPRIT* con el objetivo de continuar el trabajo empezado bajo el programa *EUREKA*, y como resultado de esto surge el sistema *EHS (European Home Systems)* generado por un grupo cerrado de empresas, lo que provoca a inicios de los noventa la creación de otro sistema llamado *Batibus*; dicho sistema empezó a tener un gran éxito y fue implementado en múltiples puntos de Europa, sin embargo, más tarde aparece *EIB (European Installation Bus)* para hacerle competencia.

Para el año de 1992 la empresa Americana *Echelon Corporation*, lanza su sistema *LonWorks*, logrando acaparar principalmente el sector industrial; en respuesta a esto en 1997 en Europa se lanza el sistema *KONNEX*, cuyo origen se encuentra en la unión de los sistemas europeos: *EHS*, *EIB* y *Batibus*.

En la actualidad existen otros protocolos, sin embargo los de mayor desarrollo son los que se han mencionado, siendo el principal el protocolo X-10. Este protocolo, se encuentra 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, por ejemplo: *G.E.*, *Radio Shack (Plug-N-Power)*, *Sears*, *Stanley (Light Maker)*, *IBM (Home Director)*, *Zenith*, *X10 Pro*, *X10 Powerhouse*, *Active Home*, *Smart Home*, entre otras.

1.2. Definición y características

El término domótica viene de la unión de las palabras *Domus* (que significa *casa* en latín) y *tica* (de *automática*, palabra en griego, 'que funciona por sí sola'). Se entiende por domótica al conjunto de sistemas capaces de automatizar una vivienda mediante la incorporación de tecnología como electrónica e informática y, que permite gestionar de forma energéticamente eficiente, segura y confortable para el usuario, los distintos aparatos e instalaciones domésticas tradicionales que conforman una vivienda, como pueden ser la calefacción, la iluminación, climatización, así como el control de persianas, puertas, ventanas, cerraduras, riego, electrodomésticos, suministro de agua, suministro de gas, suministro de electricidad, etc. [1]

En un sistema domótico se pueden encontrar elementos de los siguientes tipos:

- **Controladores:** Dispositivos encargados de gestionar toda la información sobre los otros elementos del sistema. También incluye las interfaces necesarias para interactuar con el usuario u otra aplicación.
- **Actuadores:** Dispositivos capaces de recibir una orden del controlador y realizar la acción pertinente. Pueden ser por ejemplo, el motor de una puerta o el interruptor de luz.
- **Sensores:** Su función es la de obtener información del entorno, esta información se enviará al controlador para que este realice las acciones necesarias. Pueden ser sensores de presencia o temperatura o interruptores.

Puede darse el caso, que un mismo dispositivo sea un actuador y sensor e incluso controlador, esto lo convertiría en un elemento domótico autónomo del resto de dispositivos existentes en la vivienda.

Por otro lado, la domótica busca aportar servicios de gestión energética, seguridad, bienestar y comunicación; los cuales pueden estar integrados por medio de redes interiores y/o exteriores de comunicación, ya sean inalámbricas y/o cableadas, y cuyo control puede llegar a darse desde dentro y/o fuera del hogar.

Gracias a estos sistemas, la domótica ejerce acción principalmente en cuatro aspectos:

- Se consigue una mayor eficiencia energética: Esto debido a que administra y regula el consumo de energía haciendo uso de termostatos, sensores o relojes programadores.
- Se consigue una mayor comodidad del usuario: Debido a que se logra automatizar algunos servicios domésticos, por lo que no es necesario que el usuario este pendiente de estas actividades; entre los servicios que se encuentran están la calefacción, iluminación, refrigeración, el control de persianas, ventanas, o de riego.
- Un alto nivel de seguridad: En un sistema domótico se tiene la característica de poder encargarse de tareas especializadas para proteger la vivienda de intrusos o, poder dar

aviso de cuando hay accidentes o problemas, además de poder simular la presencia dentro de la vivienda.

- Un amplio nivel de comunicación: El usuario puede estar al tanto de todo lo que está sucediendo en la casa.



Fig. 1.1 Aspectos que integra la domótica [2]

1.3. Tipos de arquitectura

La arquitectura de un sistema domótico establece la ubicación de los diferentes elementos y módulos de control del sistema. Un sistema puede tener los siguientes cuatro tipos de arquitectura:

1.3.1. Arquitectura centralizada

En un sistema con arquitectura centralizada, los elementos que forman parte del sistema para el monitoreo y control de la vivienda, como sensores, luces, actuadores etc., se conectan por medio del bus hacia donde esté instalado el sistema central de control, el cual puede ser una computadora o algún otro dispositivo. El sistema central representa el corazón del sistema, por lo cual, si este falla, todo lo demás deja de funcionar, además de que los elementos que integran al sistema no tienen comunicación entre sí.

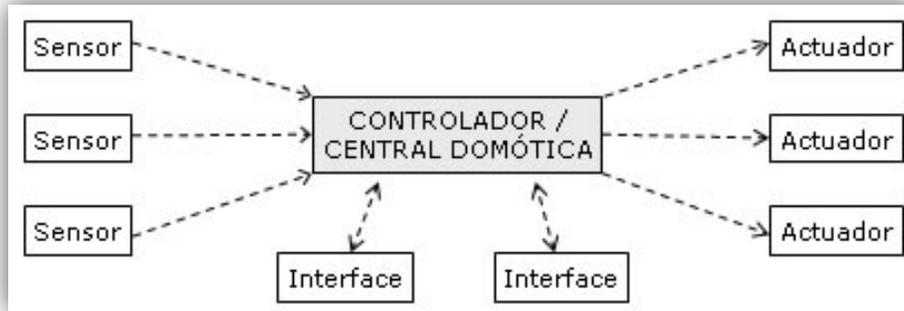


Fig. 1.2. Esquema de un sistema de arquitectura centralizada [3]

1.3.2. Arquitectura centralizada modular

Este tipo de arquitectura funciona de la misma manera que los de tipo centralizado, con la diferencia de que este tipo, permite la ampliación del sistema, ya que se pueden añadir módulos con diferentes funciones.

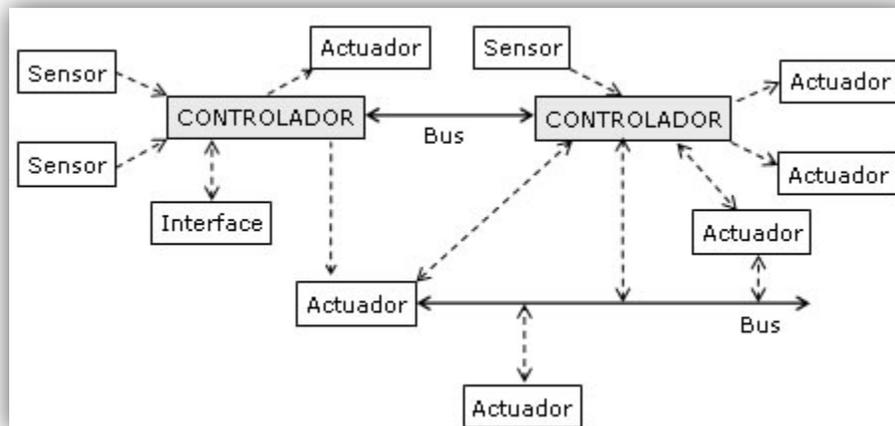


Fig. 1.3. Esquema de un sistema de arquitectura centralizada modular [3]

1.3.3. Arquitectura descentralizada

En este tipo de arquitectura, todos los elementos que conforman una red, pueden ser productores o consumidores de información, y a la vez, pueden procesar esta información o la proveniente de otros elementos. En este tipo de arquitectura, es necesario un protocolo de comunicaciones para que los elementos produzcan una acción coordinada.

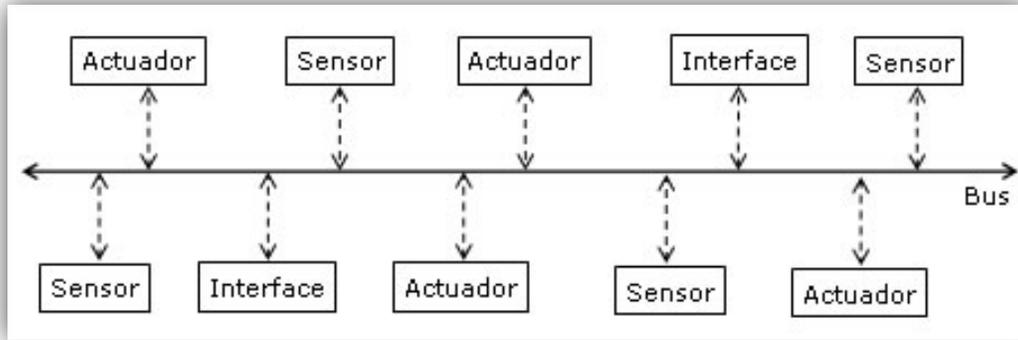


Fig. 1.4. Esquema de un sistema de arquitectura descentralizada [3]

1.3.4. Arquitectura distribuida

En este tipo de arquitectura se disponen de varios pequeños dispositivos, capaces de adquirir y procesar la información de múltiples sensores y transmitirlos al resto de dispositivos distribuidos por la vivienda.

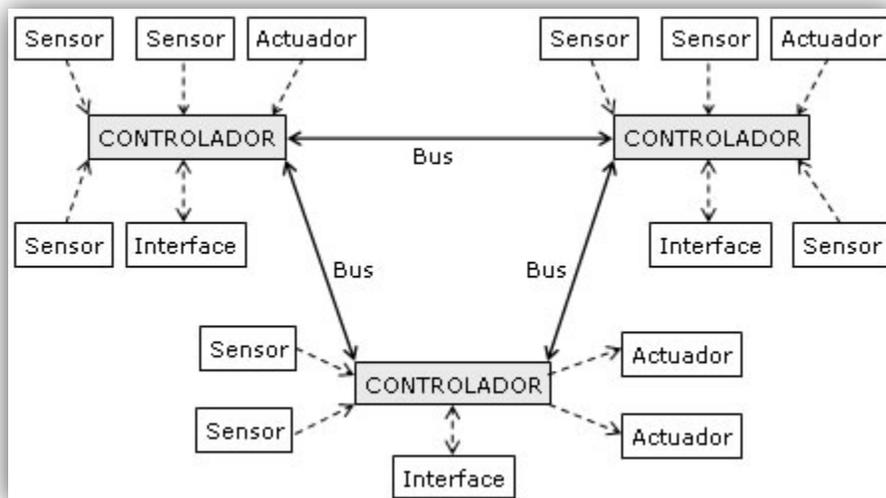


Fig. 1.5. Esquema de un sistema de arquitectura distribuida [3]

1.4. Medios de transmisión

En todo sistema domótico de cualquier tipo de arquitectura, los diferentes elementos de control deben intercambiar información unos con otros a través de un soporte físico (par trenzado, línea de potencia, red eléctrica, radio, infrarrojos, etc.), así bien como medio de transmisión se entenderá al soporte físico sobre el cual son transportados los datos de comunicaciones; a continuación listaremos los más importantes tipos de medios de transmisión aplicados a la domótica:

1.4.1. Red eléctrica convencional

Si bien no es el medio más adecuado para la transmisión de datos [4], si es una alternativa a tener en cuenta para las comunicaciones domésticas, dado el bajo costo que implica su uso, ya que se trata de una instalación existente de fácil adaptación a los elementos a controlar, por lo que es nulo el costo de la instalación.

Para aquellos casos en los que las necesidades del sistema no impongan requerimientos muy exigentes en cuanto a la velocidad de transmisión, la línea de distribución de energía eléctrica puede ser suficiente como soporte de dicha transmisión. Los rangos de frecuencia generalmente usados en este tipo de bus está entre 3 [kHz] a 149 [kHz].

1.4.2. Cable coaxial

El cable coaxial fue el primer cable empleado, aparte de cable eléctrico convencional, en la transmisión de información entre dispositivos y existen diferentes tipos según su uso y utilización.

- Señales de teledifusión que provienen de las antenas.
- Señales procedentes de las redes de TV por cable.
- Señales de control y datos a media y baja velocidad.

El cable coaxial tenía una gran utilidad en sus inicios por su propiedad idónea de transmisión de voz, audio y vídeo. Los factores a tener en cuenta a la hora de elegir un cable coaxial son su ancho de banda, su resistencia o impedancia característica, su capacidad y su velocidad de propagación. El ancho de banda del cable coaxial está entre los 500 [MHz], esto hace que el cable coaxial sea ideal para transmisión de televisión por cable por múltiples canales. La resistencia o la impedancia característica dependen del grosor del conductor central o malla; si varía éste, también varía la impedancia característica.

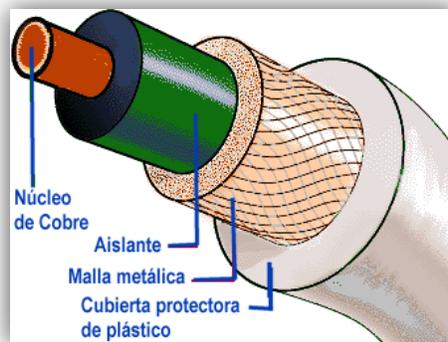


Fig. 1.6. Imagen de un cable coaxial y sus partes [5]

1.4.3. Cable de par trenzado

El cable de par trenzado es de los más antiguos en el mercado y en algunos tipos de aplicaciones es el más común. Consiste en dos alambres de cobre o a veces de aluminio, aislados con un grosor de 1 [mm²] aproximado. Los alambres se trenzan con el propósito de reducir la interferencia eléctrica de pares similares cercanos. Los pares trenzados se agrupan bajo una cubierta común de PVC (Poli cloruro de Vinilo) en cables multipares de pares trenzados (de 2, 4, 8, hasta 300 pares).

A pesar de que las propiedades de transmisión de cables de par trenzado son inferiores, y en especial por su sensibilidad ante perturbaciones externas a las del cable coaxial, su gran adopción se debe al costo, su flexibilidad y facilidad de instalación, así como las mejoras tecnológicas constantes introducidas en enlaces de mayor velocidad, longitud, etc.

El par trenzado, hasta ahora ha sido mejor aceptado por su costo, accesibilidad y fácil instalación. Sus dos alambres de cobre torcidos, aislados con plástico PVC han demostrado un buen desempeño en las aplicaciones de hoy, sin embargo, a altas velocidades puede resultar vulnerable a las interferencias electromagnéticas del medio ambiente. [4]

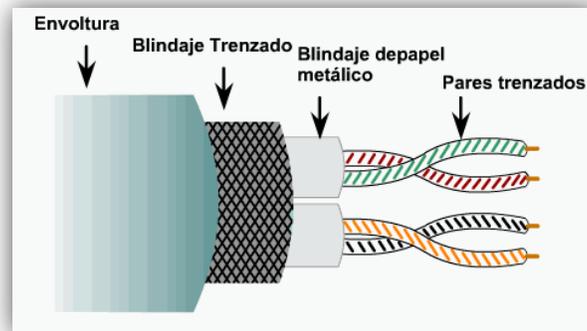


Fig. 1.7. Imagen de un cable trenzado y sus partes [6]

1.4.4. Cable de fibra óptica

En la última década la fibra óptica [4] ha pasado a ser una de las tecnologías más avanzadas que se utilizan como medio de transmisión. Los logros con este material han sido más que satisfactorios, desde lograr una mayor velocidad y disminuir casi en su totalidad ruidos e interferencias, hasta multiplicar las formas de envío en comunicaciones y recepción por vía telefónica.

La fibra óptica está compuesta por filamentos de vidrio de alta pureza muy compactos, el grosor de una fibra es como la de un cabello humano aproximadamente. Fabricadas a alta temperatura con base en silicio, su proceso de elaboración es controlado por medio de computadoras, para permitir que el índice de refracción de su núcleo, que es la guía de la onda luminosa, sea uniforme y evite las desviaciones.

Como características de la fibra podemos destacar que son compactas, ligeras, con bajas pérdidas de señal, amplia capacidad de transmisión y un alto grado de fiabilidad, ya que son inmunes a las

interferencias electromagnéticas de radio-frecuencia. Las fibras ópticas no conducen señales eléctricas, conducen rayos luminosos, por lo tanto son ideales para incorporarse en cables sin ningún componente conductor y pueden usarse en condiciones peligrosas de alta tensión.

En comparación con el sistema convencional de cables de cobre, donde la atenuación de sus señales es de tal magnitud que requieren de repetidores cada dos kilómetros para regenerar la transmisión, en el sistema de fibra óptica se pueden instalar tramos de hasta 70 [km] sin que haya necesidad de recurrir a repetidores, lo que también hace más económico y de fácil mantenimiento este material.

Con un cable de seis fibras se puede transportar la señal de más de cinco mil canales o líneas principales, mientras que se requiere de 10.000 pares de cable de cobre convencional para brindar servicio a ese mismo número de usuarios, con la desventaja que este último medio ocupa un gran espacio en los canales y requiere de grandes volúmenes de material, lo que también eleva los costos. Originalmente, la fibra óptica fue propuesta como medio de transmisión debido a su enorme ancho de banda; sin embargo, con el tiempo se ha introducido en un amplio rango de aplicaciones además de la telefonía, tales como la automatización industrial, la computación, los sistemas de televisión por cable y transmisión de información de imágenes astronómicas de alta resolución, entre otros.

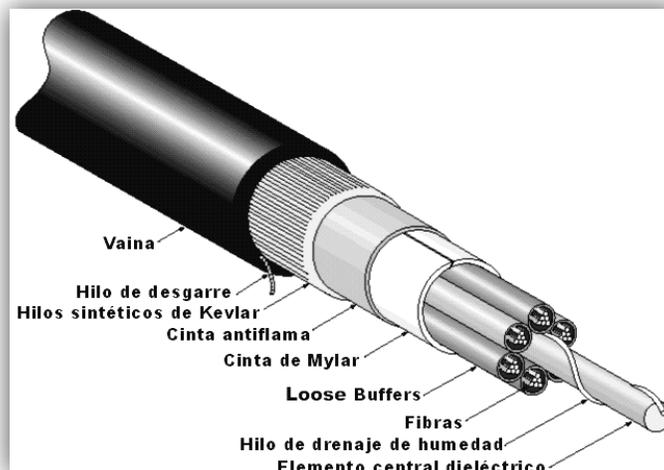


Fig. 1.8. Imagen del cable de fibra óptica y sus partes [7]

1.4.5. Bluetooth

El Bluetooth es una tecnología inalámbrica para el intercambio de datos en distancias cortas (<100 metros) para varios dispositivos fijos y móviles. Creado por el proveedor de telecomunicaciones de Ericsson en 1994, fue concebido originalmente como una alternativa inalámbrica a la comunicación RS-232 (Recommended Standard 232) ya que mejora las prestaciones de este; debido a que evita el uso de cables, aumenta la velocidad binaria y aporta movilidad dentro de un rango de hasta 100 metros (esta cambia dependiendo de la versión y/o país).



1.9. Imagen de varios dispositivos con comunicación Bluetooth [8]

El rango de frecuencias en que se mueve Bluetooth (2.402 [GHz] a 2.480 [GHz]) está dentro de una banda libre que se puede usar para aplicaciones ICM (Industrial, Científica y Médica) que no necesitan licencia.

Bluetooth es una opción interesante para intercambio de datos entre teléfonos móviles, agendas, pasarelas residenciales, centrales de seguridad/domótica, ordenadores, webcams, equipos o reproductores MP3, mandos a distancia universales, etc.

1.4.6. Wi-Fi

La tecnología Wi-Fi (*Wireless Fidelity*) es un mecanismo de conexión inalámbrica para dispositivos electrónicos y es el instrumento ideal para crear redes de área local en las viviendas, cuando es imposible instalar nuevos cables o se necesita movilidad total dentro de estos entornos. Más en detalle, permite navegar por Internet con un portátil o una tableta electrónica desde cualquier punto de una casa, aportando la difusión necesaria en muchas aplicaciones diarias de la vivienda. [9]

Las ventajas que se pueden notar para este nivel físico en el caso de un edificio, es la ausencia de cableado para crear una red de datos, dentro de las limitaciones del sistema. Como inconvenientes aportaremos dos fundamentales, una es la seguridad y otro es el coste de los dispositivos que incorporaran Wi-Fi.



1.10. Imagen de varios elementos con interconexión Wi-Fi [10]

1.4.7. ZigBee

ZigBee [4] es una alianza sin ánimo de lucro, de 25 empresas (la mayoría de ellas fabricantes de semiconductores), con el objetivo de favorecer el desarrollo e implantación de una tecnología inalámbrica de bajo coste. Destacan empresas como *Invensys*, *Mitsubishi*, *Philips* y *Motorola*, que trabajan para crear un sistema estándar de comunicaciones, vía radio y bidireccional, para usarlo dentro de dispositivos de domótica, control industrial, periféricos de PC y sensores médicos.

Esta tecnología inalámbrica es de baja velocidad y bajo consumo, con velocidades comprendidas entre 20 [kb/s] y 250 [kb/s] y rangos de 10 [m] a 75 [m]. Una red *ZigBee* puede estar formada por hasta 255 nodos los cuales tienen la mayor parte del tiempo un receptor *ZigBee* que no se encuentra activo mientras no reciba o envíe instrucciones, esto con el objeto de consumir menos que otras tecnologías inalámbricas. El objetivo es que un sensor equipado con un receptor *ZigBee* pueda ser alimentado con dos pilas AA durante al menos 6 meses y hasta 2 años. Como comparativa, la tecnología Bluetooth es capaz de llegar a 1 [MB/s] en distancias de hasta 10 [m] operando en la misma banda de 2.4 [GHz], sólo puede tener 8 nodos por celda y está diseñado para mantener sesiones de voz de forma continua.



1.11. Logo de ZigBee [11]

1.4.8. Infrarrojos

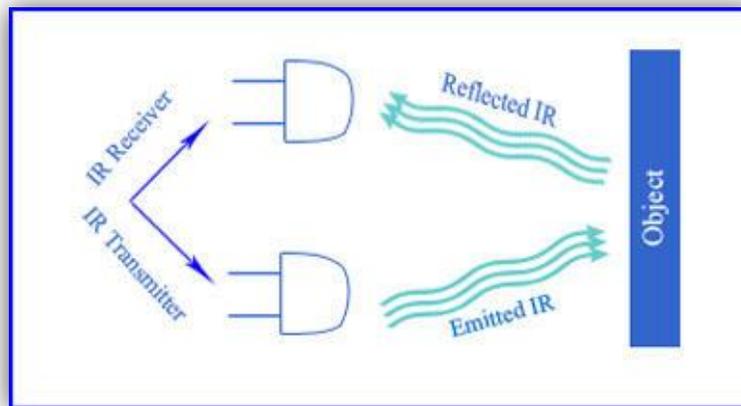
Los infrarrojos son utilizados habitualmente en mandos a distancia para pequeños dispositivos en los edificios, estos están cayendo en desuso debido al abaratamiento de las tecnologías inalámbricas por radiofrecuencia expuestas con anterioridad.

La comunicación mediante infrarrojos se realiza entre un diodo emisor, que emite una luz en la banda de Infrarrojo sobre la que se superpone una señal, convenientemente modulada con la información de control, y un fotodiodo receptor, cuya misión consiste en extraer de la señal recibida, dicha información.

Al tratarse de un medio de transmisión óptico, es inmune a las radiaciones electromagnéticas producidas por los equipos domésticos o por los demás medios de transmisión (coaxial, par trenzado, corrientes portadoras, etc.). Sin embargo, habrá que tomar precauciones en los siguientes casos:

Las interferencias electromagnéticas sólo afectarán a los extremos del medio IR, es decir, a todos aquellos dispositivos opto electrónicos (diodo emisor y fotodiodo receptor).

Es necesario tener en cuenta otras posibles fuentes de infrarrojo. Hoy en día existen diferentes dispositivos de iluminación que emiten cierta radiación IR.



1.12. Funcionamiento de los rayos infrarrojos [12]

Para finalizar con este apartado; las ventajas e inconvenientes de los sistemas de transmisión mostrados anteriormente se resumen en el comparativo de la siguiente tabla:

Tabla 1. Ventajas y desventajas de los diferentes medios de transmisión [13]

Sistema	Ventajas	Desventajas
Red eléctrica preexistente	Bajo Costo. Fácil Instalación	No inmune a ruido e interferencias electromagnéticas.
Cable de par trenzado	Bajo Costo. Fácil Instalación.	Escasa transmisión de información.

Cable coaxial	Bajo Costo. Fácil Instalación.	No Inmune a ruido e interferencias electromagnéticas.
Fibra óptica	Alta transmisión Alta velocidad. Permite grandes distancias. Seguridad en la transmisión de datos. Inmunidad frente a Interferencias electromagnéticas y de radiofrecuencias.	Adaptación de todos los dispositivos para recibir información óptica. Costo alto de cables y conexiones
Infrarrojos	Aplicable a cualquier inmueble.	Cada uno de los dispositivos debe tener un emisor/receptor de infrarrojos. Distancias cortas.
Radiofrecuencia (Wi-fi, Bluetooth, ZigBee)	Fácil interpretación de las comunicaciones. Aplicable en cualquier inmueble.	Todos los dispositivos deben tener un emisor/receptor de radio. Alta sensibilidad a las interferencias.

1.5. Protocolos de comunicación

Una vez establecido el soporte físico y la velocidad de comunicaciones, un sistema domótico se caracteriza por el protocolo de comunicaciones que utiliza; dicho “protocolo”, no es otra cosa que el idioma o formato de los mensajes utilizados por los diferentes elementos de control del sistema, con el objetivo de entenderse unos con otros y además para poder intercambiar información de una manera coherente.

De acuerdo con su estandarización los protocolos pueden ser:

- **Estándar:** Son aquellos que ya existen y cuyas empresas utilizan para desarrollar productos compatibles con estos, algunos ejemplos son el *X-10*, el *EHS*, el *EIB* y el *Batibus*.
- **Propietarios:** Son propios de una empresa, desarrollados por ellas y que fabrican sus propios productos que son capaces de comunicarse entre sí.

A continuación, se presenta un cuadro con los protocolos más importantes que están contribuyendo de mayor forma a la industria de la domótica. Algunos de estos protocolos, que tienen o tendrán un papel clave en el desarrollo de la domótica, en seguida los describiremos con mayor detalle.

Tabla 2. Protocolos existentes y algunas de sus características [14]

Protocolo	Medio de transmisión	Velocidad de transmisión	Ventajas	Desventajas
X-10	Línea de corriente de baja tensión	60 [bps]	Fácil integración y bajo costo	No inmune a ruido
EIB	Par trenzado	9600 [bps]	Posibilidad de usar mayor número medios de transmisión	Aplica solo en Europa
EHS	Todos	9600 [bps]	Bastante maduro y accesible en cuanto precio	Aplica solo en Europa
LonWorks	Todos	7800 [bps]	Gran fiabilidad y robustez	Sistemas muy caros y dependientes solo de algunas compañías.
Batibus	Todos	4800 [bps]	Gran fiabilidad, además de admitir múltiples medios físicos.	Sistemas muy caros, baja velocidad de transmisión.
CEBus	Todos	7500 [bps]	Gran versatilidad y admite múltiples medios físicos	Solo aplica en EE.UU.

1.5.1. Protocolo de Control EIB

EIB (European Installation Bus) es un protocolo domótico desarrollado bajo los auspicios de la Unión Europea con el objetivo de contrarrestar las importaciones de productos similares que se estaban produciendo desde el mercado japonés y el norteamericano, donde estas tecnologías se fueron desarrollando antes que en Europa.

El objetivo es crear un estándar europeo, con el suficiente número de fabricantes, instaladores y usuarios, que permita comunicarse a todos los dispositivos de una instalación eléctrica como: contadores, equipos de climatización, de custodia y seguridad, de gestión energética y los electrodomésticos.

El EIB es un sistema descentralizado en el que cada dispositivo puede ejercer una serie de funciones de forma autónoma o relacionada con otros dispositivos. Al ser un sistema descentralizado, si un elemento del sistema falla, éste puede seguir funcionando aunque sea parcialmente. Esto es así porque todos los dispositivos que se conectan al bus de comunicación de datos tienen su propio microprocesador y electrónica de acceso al medio.

En resumen las principales características del protocolo EIB son:

- Sistema descentralizado.
- Soporta varios medios de transmisión como:
 - Par trenzado a 24 [V] y 9600 [bps].

- Corrientes portadoras a 230 [V_{AC}]/50 [Hz] y 1200/2400 [bps].
 - Radiofrecuencia.
 - Infrarrojo.
- El número máximo de componentes a controlar es 57375.



1.13. Logo del protocolo de comunicación EIB [15]

1.5.2. Protocolo de control EHS

El estándar *EHS (European Home System)* ha sido otro de los intentos de la industria europea (año 1984), con el fin de crear una tecnología que permitiera la implantación de la domótica en el mercado residencial de forma masiva; el resultado fue la especificación del *EHS* en el año 1992. Está basada en una topología de niveles, en donde se especifican los niveles: físico, de enlace de datos, de red y de aplicación.

Desde su inicio, han estado involucrados los fabricantes europeos más importantes de electrodomésticos, las empresas eléctricas, las operadoras de telecomunicaciones y los fabricantes de equipamiento eléctrico. La idea fue crear un protocolo abierto que permitiera cubrir las necesidades de interconexión de los productos de todos estos fabricantes y proveedores de servicios. Tal y como fue pensado, el objetivo de la *EHS* es cubrir las necesidades de automatización de la mayoría de las viviendas europeas cuyos propietarios que no se pueden permitir el lujo de usar sistemas más potentes pero también más caros, (como *LonWorks*, *EIB* o *Batibus*) debido a la mano de obra especializada que exige su instalación.

Una de las principales ventajas de *EHS* es que los dispositivos se pueden configurar automáticamente haciendo más sencilla la ampliación de las instalaciones, además de la compatibilidad total entre dispositivos *EHS*.

En resumen, las principales características del protocolo EIB son:

- Arquitectura distribuida.
- Los medios de transmisión son:
 - Par trenzado a 24 [V] y 48 [kbps].
 - Corrientes portadoras a 230 [V_{AC}]/50 [Hz] y 2400 [bps].
 - Infrarrojos a 1200 [bps].

- Radiofrecuencia a 1100 [bps].
- El sistema es capaz de controlar a $10E^{12}$ elementos.



1.14. Logo del protocolo de comunicación EHS [16]

1.5.3. Protocolo de control LonWorks

Este protocolo de comunicación fue lanzado en 1992 por la compañía americana *Echelon Corp.*, es aplicado a sistemas del tipo distribuido y su medio de transmisión es a través del llamado cableado dedicado, aunque puede utilizar una gran variedad de medios de transmisión. Este protocolo es característico por su fiabilidad, ya que es demasiado robusto; sin embargo su implementación resulta muy costosa y es por ello que se usa principalmente en edificios de oficinas, hoteles e industria.

El éxito que ha tenido *LonWorks* en instalaciones profesionales, en las que importa mucho más la fiabilidad y robustez que el precio, se debe a que desde su origen ofrece una solución con arquitectura descentralizada, extremo a extremo, que permite distribuir la inteligencia entre los sensores y los actuadores instalados en la vivienda y, que cubre desde el nivel físico al nivel de aplicación de la mayoría de los proyectos de redes de control.

En resumen las principales características del protocolo LonWorks son:

- Arquitectura distribuida.
- Los medios de transmisión son:
 - Par trenzado a 24 [V] y 78 [kbps].
 - Corrientes portadoras a 5400 [bps] ó 3600 [bps].



1.15. Logo del protocolo de comunicación LonWorks [17]

1.5.4. Protocolo de control CEBus

Este protocolo surgió en el año de 1984 cuando diversos miembros de una empresa norteamericana llamada *Electronics Industry Association* se propusieron unificar los protocolos de señalización infrarroja para el control remoto de electrodomésticos; el protocolo se extendió a todo el ámbito de la domótica hasta el año de 1992, en donde especificaron y desarrollaron un estándar llamado *CEBus* (*Consumer Electronic Bus*).

El medio de transmisión del protocolo puede ser a través de: red eléctrica, cable trenzado, cable coaxial, infrarrojos, radio frecuencia, fibra óptica y bus de audio-vídeo. Los dispositivos pertenecientes a un sistema *CEBus* se comunican entre sí mediante mensajes, cuyo formato es independiente del medio de transmisión empleado; cada mensaje contiene la dirección de destino del receptor sin ninguna referencia acerca del medio físico en el que está situado el receptor o el transmisor. Al ser así, *CEBus* genera una red uniforme de comunicación a nivel lógico, lo que permite la implementación de más dispositivos que cuenten con una interfaz adecuada. Una ventaja es que los dispositivos además de recibir mensajes individuales, pueden recibir tramas con destino a todos los dispositivos. Los dispositivos se pueden agrupar (*group address*) y de esta forma, es posible mandar un único mensaje a varios dispositivos al mismo tiempo.

Como parte de la especificación *CEBus* se ha definido un lenguaje común para el diseño y especificación de la funcionalidad de un nodo, a este lenguaje lo han llamado *CAL* (*Common Application Language*) y es un lenguaje orientado a objetos.



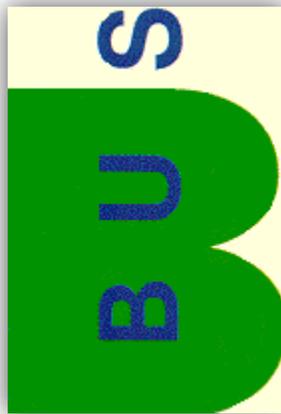
1.16. Logo del protocolo de comunicación CEBus [18]

1.5.5. Protocolo de control BatiBUS

Este protocolo fue desarrollado por una compañía Francesa llamada *Merlin Gerin*, se aplica en sistemas centralizados, utilizando como medio de transmisión el cable de par trenzado, aunque en algunas ocasiones también se hace mediante cable telefónico o eléctrico, este protocolo es capaz de trabajar a una velocidad de 4800 [bps] y es capaz de administrar hasta 500 puntos de control.

Este protocolo es de fácil instalación, bajo costo y capacidad de evolución, ya que se pueden añadir funciones conforme las necesidades lo exijan; aunque se puede ver como desventaja el hecho de que la transmisión sea mediante cable ya que limita la instalación a recintos nuevos.

En una red, todos los sistemas BatiBUS escuchan la información transmitida por cualquier otro y la procesan, pero sólo aquellos que hayan sido programados para ello, filtrarán la trama y la subirán a la aplicación empotrada en cada dispositivo. A nivel de acceso, este protocolo usa una técnica similar a Ethernet pero con resolución positiva de las colisiones, es decir si dos dispositivos intentan acceder al mismo tiempo al bus, ambos detectan que se está produciendo una colisión, pero sólo el que tiene más prioridad continúa transmitiendo, mientras que el otro deja de poner señal en el bus.



1.17. Logo del protocolo de comunicación BatiBUS [19]

1.5.6. Protocolo X-10

El X-10 es un protocolo estándar con transmisión por la línea eléctrica, este estándar fue introducido por primera vez en 1978 por *Sears Home Control System* y *Radio Shack Plug'n Power System*. Desde entonces, X-10 ha sido desarrollado y se han elaborado versiones para el sistema de control domótico por varias empresas. Actualmente, se puede encontrar en Europa con tres grandes familias de productos basadas en X-10, teóricamente compatibles entre sí, estas son: *Netzbus*, *Timac* y *Home Systems*.

Debido su estancia por más de 20 años en el mercado y a la tecnología empleada los productos X-10 tienen un precio muy competitivo, de manera que es líder en el mercado norteamericano

residencial y de pequeñas empresas (realizadas por los usuarios finales o electricistas sin conocimientos de automatización).

Se puede afirmar que el X-10 es ahora mismo la tecnología más accesible para realizar una instalación domótica no muy compleja, en el siguiente apartado explicaremos más a detalle su funcionamiento y los elementos necesarios para poder llevar a cabo la comunicación entre dispositivos.

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.



1.18. Logo del protocolo de comunicación X-10 [20]

1.6. Sistema X-10

X-10 fue desarrollado en 1975 por *Pico Electronics Ltd.*, en Escocia, con el fin de permitir el control remoto de los dispositivos domésticos, fue la primer tecnología domótica en aparecer, además de seguir siendo la más ampliamente disponible.

Para permitir el control de los dispositivos domésticos X-10, se utiliza la línea eléctrica (220 [V_{AC}] ó 110 [V_{AC}]) para transmitir señales de control entre equipos de automatización del hogar; a continuación se explica detalladamente cómo se realiza esto.

1.6.1. Fundamentos del protocolo

La comunicación entre dispositivos X-10 se hace a través de la red eléctrica, inyectando y leyendo señales contenidas en ésta. Las señales son transmitidas con ráfagas de frecuencia de 120 [kHz], que representan la información digital. La transmisión de señales se sincroniza con el punto de cruce en cero de la corriente eléctrica (de 50 [Hz] ó 60 [Hz]). Se intenta transmitir lo más cerca del punto cero como sea posible, pero se acepta una variación de 200 [µs] desde el punto cero.

La información que se transmite mediante la red eléctrica se representa mediante bits (0,1), ya que estos no se pueden mandar de manera digital mediante la línea eléctrica, se usa un formato establecido; de esta manera, para representar el uno binario se inyecta una ráfaga de 120 [kHz] de

1 milisegundo de duración, está ráfaga debe inyectarse en el punto cero de cruce de un medio ciclo de la corriente eléctrica. Para compatibilidad con sistemas trifásicos el 1 binario se transmite 3 veces, como 3 ráfagas de 120 [kHz] para coincidir con el punto cero de las tres fases de la corriente.

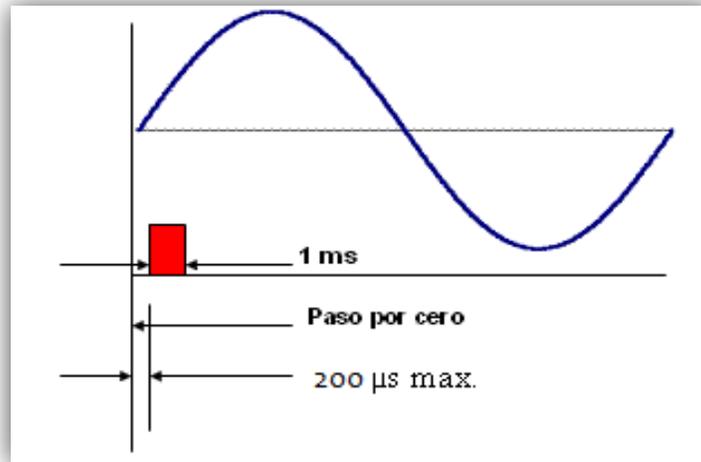


Fig. 1.19. Tiempo de sincronización de la señal en X-10 [21]

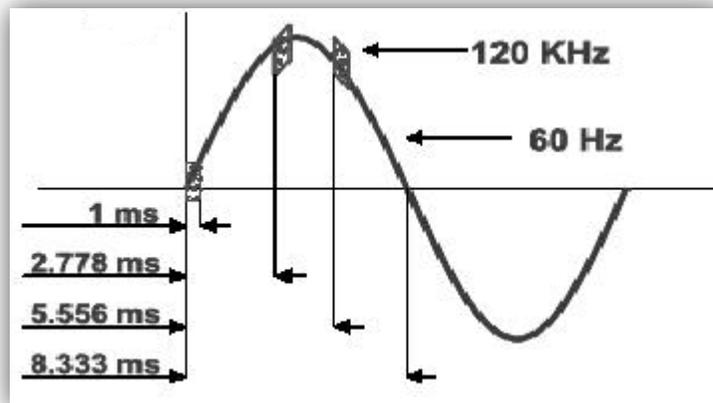


Fig. 1.20. Tiempos de sincronización en una señal trifásica [22]

Un mensaje completo en X-10 está compuesto por el código de comienzo (1110), seguido por el código de la letra de la casa y por un código de control. El código de control puede ser o una dirección de unidad o un código de comandos, lo cual se verá reflejado en un bit de sufijo.

En la tabla 3, 4 y 5 se muestran las diferentes direcciones de casa, de dispositivo y los comandos válidos, así como los bits que conforman cada una de estas direcciones.

Tabla 3. Diferentes direcciones de casa existentes [23]

Dirección de casa	Código de casa en bits			
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

Tabla 4. Códigos de control para unidades [23]

Dirección de Unidad	Código en bits				Sufijo
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 5. Códigos de control para comandos [23]

Comandos	Código en bits				Sufijo
Apagar todas las unidades	0	0	0	0	1
Encender todas las luces	0	0	0	1	1
Encender dispositivo	0	0	1	0	1
Apagar dispositivo	0	0	1	1	1
Atenuar intensidad	0	1	0	0	1
Aumentar intensidad	0	1	0	1	1
Apagar todas las luces	0	1	1	0	1
Código extendido	0	1	1	1	1
Petición de saludo	1	0	0	0	1
Aceptación de saludo	1	0	0	1	1
Atenuación preestablecida	1	0	1	X	1
Datos extendidos	1	1	0	0	1
Estado=ON	1	1	0	1	1
Estado=OFF	1	1	1	0	1
Petición de estado	1	1	1	1	1

Los comandos soportados por X-10 son explicados de manera breve a continuación:

- Apagar todas las luces: Código encargado de apagar todas las luces.
- Encender todas las luces: Código encargado de encender todas las luces.
- Encender/Apagar: Código encargado de encender/apagar el dispositivo.
- Atenuar/Aumentar intensidad: Código encargado de variar la intensidad luminosa.
- Código extendido: Extensión de código que puede ser utilizado tanto para control como para información.
- Petición de saludo: Código que sirve para buscar otros dispositivos capaces de enviar códigos X-10 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 "Petición de saludo" y tenga el mismo código de casa, además le permitirá saber si existe otro dispositivo con la misma dirección. La petición de saludo se responde con un comando de "Aceptación de Saludo".

- Aceptación de saludo: Código para aceptación de saludo, este se da cuando se recibió el código de "Petición de saludo" y tiene el mismo código de casa que el dispositivo que lo envió.
- Atenuación preestablecida: Código encargado de fijar el nivel de iluminación a un valor constante.
- Datos extendidos: Código encargado de enviar señales analógicas, generalmente 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 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.
- Estado = On: Código encargado de marcar el estado encendido.
- Estado = Off: Código encargado de marcar el estado apagado.
- Petición de estado: Código encargado para dispositivos con capacidad de transmisión bidireccional y sirven para monitorear el funcionamiento de los módulos X-10.

Cuando transmitimos cualquier código de las tablas antes expuestas, dos pasos por cero son usados para transmitir cada bit como una pareja de bits complementarios.

El uno binario es representado como un pulso en el primer semiciclo y una ausencia del pulso en el segundo semiciclo; en el cero binario el pulso se encuentra en la segunda parte del ciclo y la ausencia de pulso está en la primera parte del mismo.

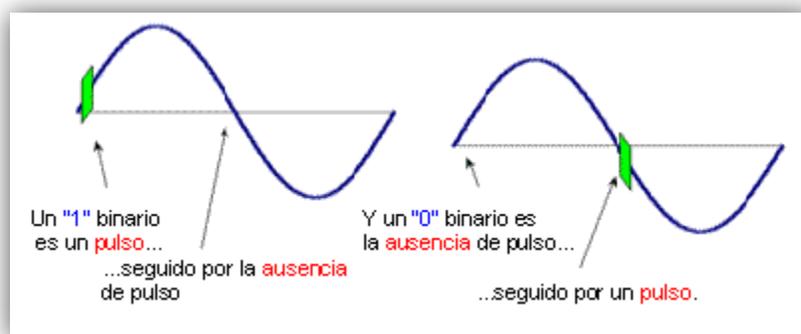


Fig. 1.21. Formato de los bits que se mandan por la red [23]

El código de comienzo (1110) es el único que no se envía de forma complementaria.

1.6.2. Trama del protocolo

Como se mencionó anteriormente, un mensaje completo en X-10 está compuesto por el código de comienzo (1110), el código de casa y por un código de control, que a la vez puede ser o una dirección de unidad o un código de comandos.

La trama siempre empieza con el mismo código de inicio "1110", dicho código es el único que no se envía de manera complementaria, por lo que se necesitan 4 semiciclos (2 ciclos) para transmitir esta parte de la trama.

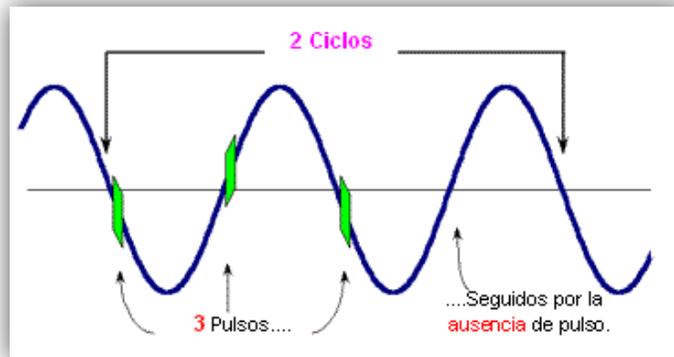


Fig. 1.22. Formato del código de inicio [23]

Inmediatamente después del código de inicio se transmite la dirección de casa, al ser 4 bits y transmitirse de manera complementaria se necesitan 4 ciclos.

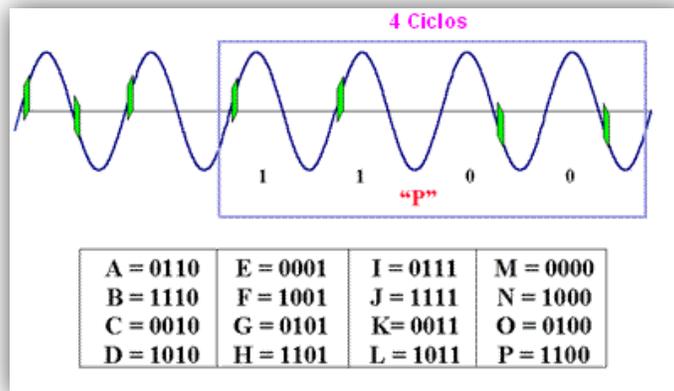


Fig. 1.23. Formato del código de casa [23]

Después de enviar el código de la casa, enviamos la dirección de unidad, aquí el sufijo siempre será cero; para esta parte se necesitan 5 ciclos.

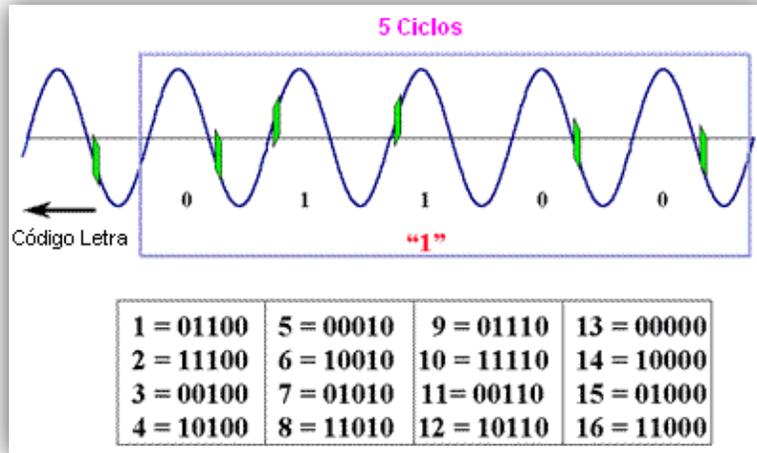


Fig. 1.24. Formato de la dirección de dispositivo [23]

Un bloque completo de datos o paquete de información se compone de código de inicio, código de la casa, código de control y sufijo. Para darle una mayor fiabilidad al sistema, se transmite dos veces cada uno de estos bloques de información.

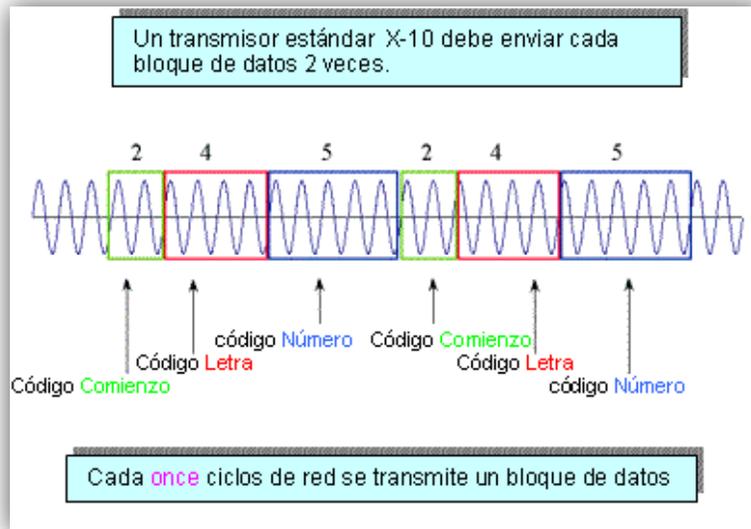


Fig. 1.25. Bloque de datos y su repetición [23]

Después de estos 2 bloques de información que contienen el código de dirección, le siguen 2 bloques de información pero que contienen el código de comando. Cada par de bloques de información deben estar separados por 6 pasos por cero.



Fig. 1.26. Ciclos de espera entre transmisiones [23]

Después de esta pausa, se debe enviar el código de comienzo, seguido del código de la letra y el código de control, finalmente irá el sufijo, teniendo que ser en este caso igual a 1 para que el código de control sea interpretado como un comando y no como una dirección por el receptor.

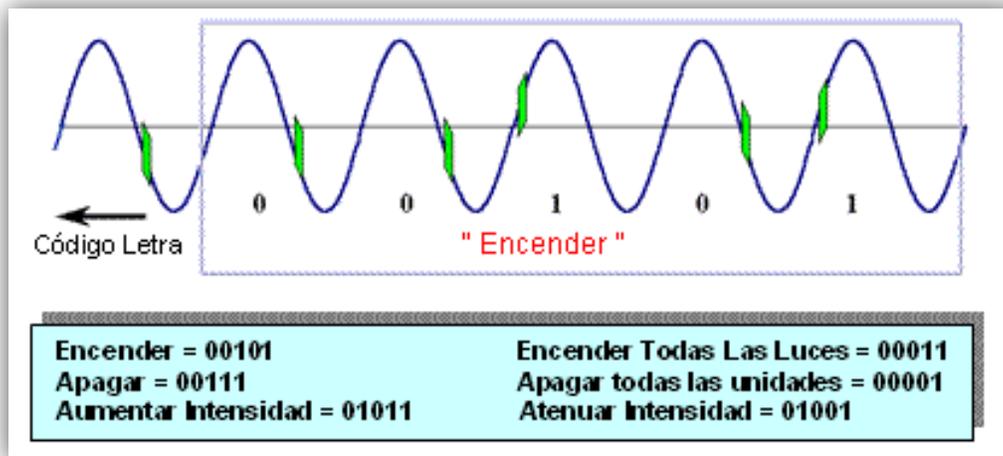


Fig. 1.27. Formato del código de comando [23]

Este paquete de datos, pero ahora conteniendo al comando, será transmitido 2 veces. Para transmitir la trama completa de X-10 se necesitan 47 ciclos.

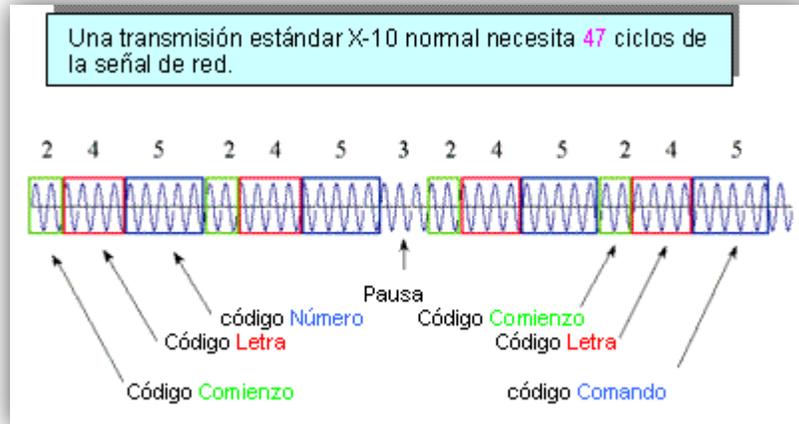


Fig. 1.28. Transmisión total en X-10 [23]

A una frecuencia de 60 [Hz] ello supone un tiempo igual a 0.7833 segundos en transmitir una orden completa.

1.6.3. Ventajas y desventajas

Ventajas:

- De instalación sencilla: Esto debido a que no se necesitan instalar más cables sobre en el lugar, todo se hace mediante la red eléctrica preexistente.
- Facilidad de uso: Ya que las comunicaciones se establecen a través de la línea eléctrica; lo que hace que si tenemos una toma de red eléctrica, tenemos una toma de red domótica.
- Configurable: El usuario puede configurar los dispositivos según sus necesidades.
- Ampliable: Compatibilidad con productos de la misma gama.

Desventajas:

- El ruido de la línea eléctrica: La transmisión de señales de pulsos a alta frecuencia a través de la red eléctrica puede verse afectada por interferencias. Las fuentes típicas que producen interferencias son aparatos eléctricos como televisores, videograbadoras, equipos de sonido, computadoras, monitores, transformadores e incluso los cables preparados con filtros tienen la tendencia de depositar ruido eléctrico sobre los cables de la red. Muchos de los nuevos aparatos electrónicos que se utilizan para uso doméstico utilizan circuitos para evitar los ruidos eléctricos, pero cuando esto ocurre a veces generan ruido en la red eléctrica. Cuando el ruido se encuentra sobre la red eléctrica puede ocasionar atenuación o bloqueo de las señales transmitidas o recibidas en los dispositivos X-10.

- Colisiones: El protocolo X-10 estándar, vigente desde 1978, es vulnerable a las colisiones, ya que se puede dar el caso en el que 2 dispositivos emisores estén enviando información al mismo tiempo generando el choque de información en la línea eléctrica y, generando nuevos bits de información que al ser recibidos por los dispositivos receptores ejecutarían otra instrucción
- Tiempo de respuesta largo: El tiempo de respuesta de los módulos X-10 es largo, y si hay mensajes extendidos, o hay colisiones entre dispositivos o varios dispositivos intentando acceder simultáneamente al mismo canal, tendremos tiempos de respuesta que pueden llegar a los muchos segundos. Esto se soluciona haciendo zonas independientes dentro de la casa, filtrando señales X-10 entre ellas, y teniendo más de un interfaz con X-10.
- Cambio de un código a otro código por el ruido eléctrico: Según aumentamos los dispositivos X-10 (y no son necesarios demasiados dispositivos para que comience este fenómeno), la distancia hamming entre códigos de identificación válidos de dos dispositivos reales disminuye. En la práctica, la probabilidad de que el ruido haga saltar de un código válido de un dispositivo a otro código válido de otro dispositivo aumenta. La solución es similar al problema de los retrasos, pero mientras que un problema puntual de atrasos es asumible, un problema de estas características puede ser peligroso. Por ello se han desarrollado sistemas domóticos más robustos como el protocolo EIB.

Capítulo 2. Propuesta de diseño

La propuesta de este trabajo es desarrollar un sistema de automatización para el hogar, que permita el control de encendido y apagado de aparatos eléctricos, así como el encendido, apagado y control del nivel de luminosidad de bombillas incandescentes, todo ello haciendo uso del protocolo X-10.

Para ello se ha pensado en el siguiente diseño: un módulo emisor conformado por un microcontrolador, un detector de cruce por cero y un generador de 120 [kHz] que inyectará señales a esta frecuencia a través de la red eléctrica, manteniendo la codificación del protocolo X-10. Cada elemento que se desee controlar deberá de tener su propio módulo receptor, compuesto por un microcontrolador, un filtro a 120 [kHz], algún circuito encargado de preparar las señales antes de ser pasadas al microcontrolador y un detector de cruce por cero, que extraerá las señales enviadas por el módulo de control; posteriormente realizará la decodificación y la comprobación de dirección, de tal manera que si la dirección recibida coincide con la suya, decodificará y ejecutará el comando X-10.

El módulo emisor deberá de permitir comunicación mediante USB con una computadora y de manera inalámbrica mediante bluetooth con un dispositivo móvil, los cuales servirán de interfaces entre el usuario y el módulo emisor; para ejecutar la secuencia determinada por el usuario.

El programa de la computadora y la aplicación móvil estarán diseñados para ser de fácil manejo para el usuario, con una organización donde sea sencillo encontrar el dispositivo que se quiera controlar y dar el comando deseado.

El microcontrolador que se va usar será elegido de acuerdo a su capacidad de memoria de acceso aleatorio (RAM), memoria de programa (ROM), frecuencia de operación, periféricos y costos del mismo.

El diagrama de bloques propuesto se muestra a continuación:

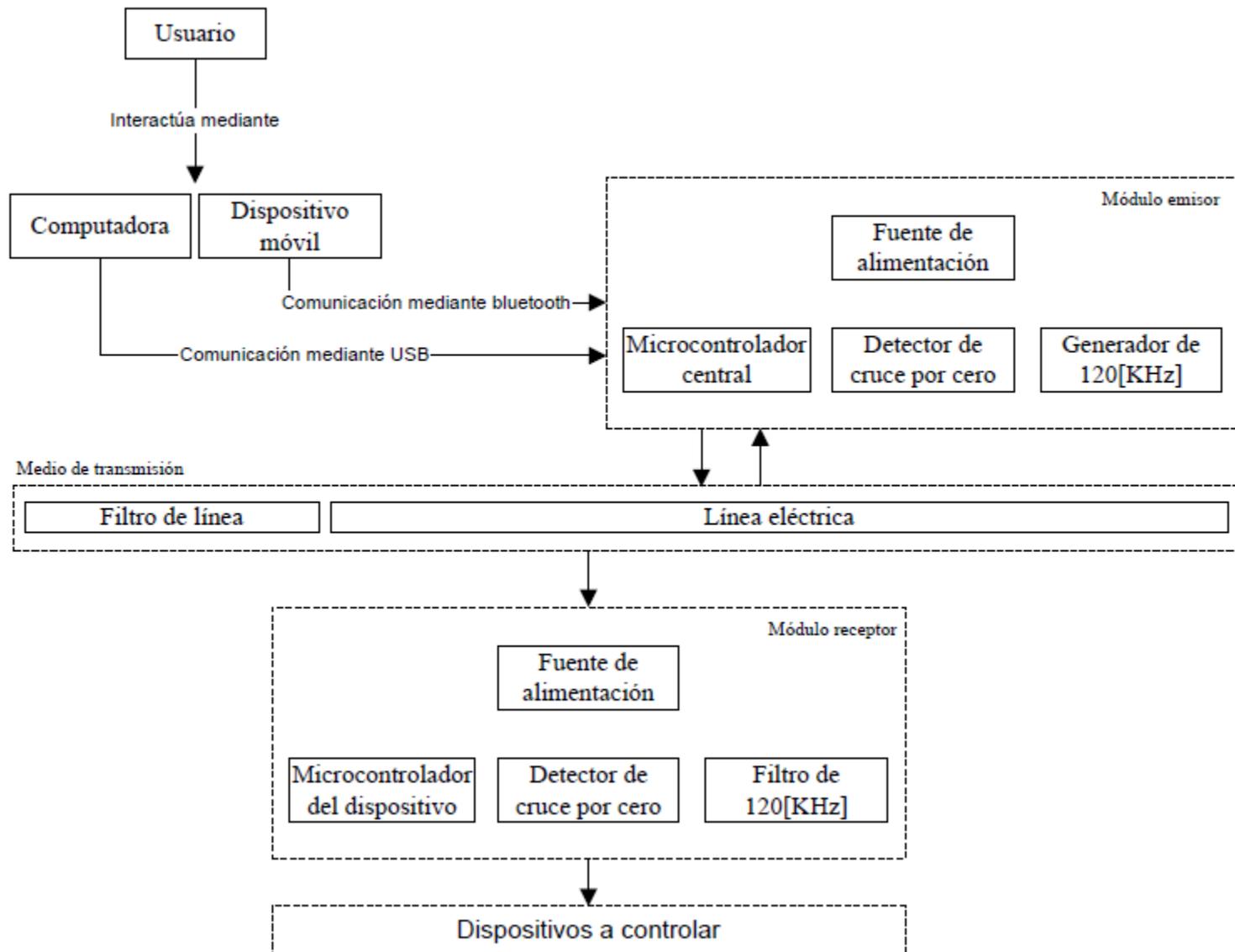


Fig. 2.1. Diagrama de bloques propuesto

Capítulo 3. Diseño del hardware

En este capítulo se muestra como se llevó a cabo el diseño de los elementos físicos, necesarios para realizar el control de dispositivos mediante el protocolo X-10.

3.1. Módulo emisor

Como se mostró en el capítulo anterior, el módulo emisor planteado se deberá componer de un detector de cruce por cero, un generador a una frecuencia de 120 [kHz], un microcontrolador que procese las instrucciones provenientes de la computadora y envíe la información a través de la red eléctrica y finalmente, una fuente de 5 [V] que alimente a los componentes; a continuación se describirá el funcionamiento del módulo, como se diseñó cada uno de los elementos que lo componen, además de la elección del microcontrolador que se consideró el más adecuado.

3.1.1. Elección del microcontrolador para el Módulo emisor

Antes de explicar la manera en la que se desarrollaron los diferentes componentes del módulo emisor, se mencionará que microcontrolador se eligió para este módulo.

Como se dijo en el capítulo anterior, se pretende que el módulo emisor sea capaz de conectarse a una computadora mediante cable USB y por medio de este realizar la comunicación con el software que enviará las instrucciones. También se requiere que el módulo emisor sea capaz de tener comunicación con un dispositivo móvil mediante un bluetooth y poder desde ahí controlar los diferentes dispositivos.

Por lo que los requisitos para el módulo emisor son los siguientes:

- Conexión USB para realizar la conexión física con la computadora.
- 1 puerto serial, ya que la transferencia de información mediante bluetooth se da mediante RS-232.
- 1 Timer para la salida de frecuencia de 120 [kHz].
- Salidas digitales para colocar LED-s de visualización.
- 1 Interrupción para captar los cruces por cero provenientes de la línea eléctrica.

Además de los requisitos anteriores también debe de tomarse en cuenta el tamaño, costo y lenguaje de programación del microcontrolador a la hora de escogerlo.

Después de investigar varios microcontroladores se supo que se puede hacer la conexión USB mediante software con los microcontroladores PIC, sin la necesidad de agregar ningún otro elemento físico, por lo que se optó por elegir uno de estos, además en estos se puede usar una clase que permite crear un puerto serial virtual y hacer la comunicación mediante esta, lo cual sería una ventaja porque la comunicación con el bluetooth también usa la comunicación serial y con esto a la hora de hacer una interfaz, se mantendría el mismo software para ambos sistemas.

Dado lo anterior se decidió optar por uno de los siguientes 4 microcontroladores que cumplen las características antes mencionadas:

Tabla 6. Tabla de los posibles microcontroladores a elegir para el módulo emisor

<i>Numero de parte</i>	<i>Pines</i>	<i>Precio</i>
PIC18F2450	28	\$100.00
PIC18F2455	28	\$96.00
PIC18F2550	28	\$100.00
PIC18F4550	40	\$88.00

* *Precios al 1° de diciembre de 2012.*

Se optó por elegir el PIC18F4550 debido a que uno de los más económicos, fáciles de conseguir y con mayor cantidad de información disponible. Con este microcontrolador debe lograrse la conexión USB con la computadora y la conexión bluetooth con el dispositivo móvil.

3.1.2 Funcionamiento

El módulo emisor será el encargado de conectarse con el computador y dispositivo móvil, dicho módulo recibirá las instrucciones, las decodificará a bits y los transmitirá por la red eléctrica en cada cruce por cero por medio de pulsos de 120 [kHz]. En diagrama de flujo de la figura 3.2 se muestra lo descrito. Para realizar lo anterior primero debe de haber realizado correctamente la comunicación entre el microcontrolador y la computadora y/o el dispositivo móvil; a continuación se describe como se realizará la comunicación mediante USB y mediante el bluetooth.

3.1.2.1. Comunicación a través del USB

USB son las siglas de *Universal Serial Bus*, se trata de un estándar de la industria, desarrollado a mediados de la década de 1990 con el fin de comunicarse entre computadoras y dispositivos electrónicos.



Fig. 3.1. Imagen de un conector USB macho tipo A [25]

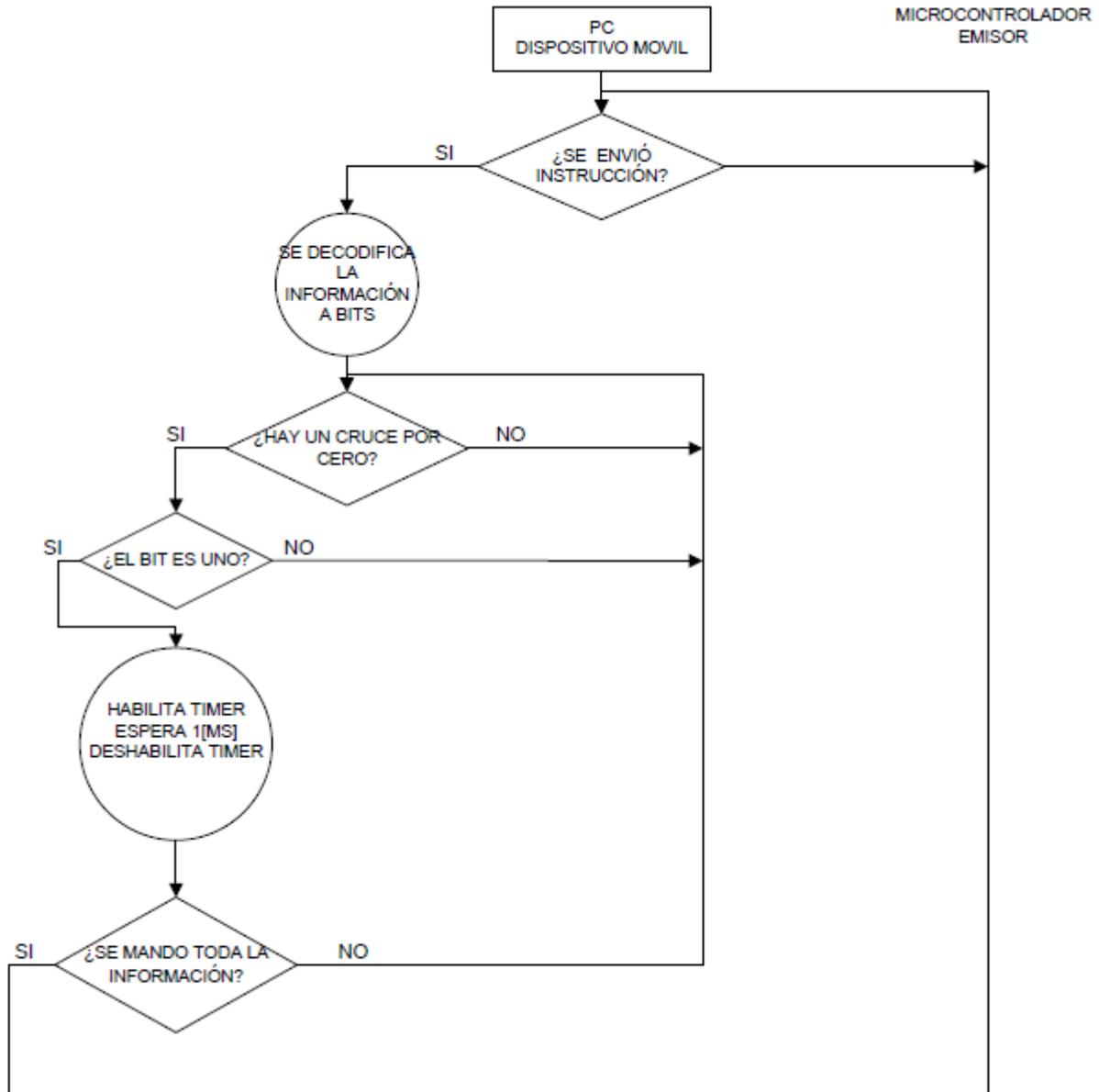


Fig. 3.2. Diagrama de flujo del envío de información en el módulo emisor

USB fue diseñado para estandarizar la conexión de periféricos, como teclados, dispositivos de señalización, cámaras digitales, impresoras, reproductores de medios portátiles, discos duros y adaptadores de red en los ordenadores personales, así mismo se emplea para comunicarse y para suministrar energía eléctrica. Se ha convertido en algo común en otros dispositivos, tales como teléfonos inteligentes, agendas electrónicas de bolsillo y consolas de videojuegos. USB ha reemplazado efectivamente una variedad de interfaces anteriores, como los puertos serie y paralelo, así como cargadores de alimentación independientes para los dispositivos portátiles [24].

La comunicación por puerto USB utiliza 4 cables: uno para alimentación (5 [V]), otro cable para la tierra (GND) y dos cables para datos (D+ y D-). En la siguiente imagen se observan los 2 tipos de conectores USB más usados y el patillaje de cada uno de estos.

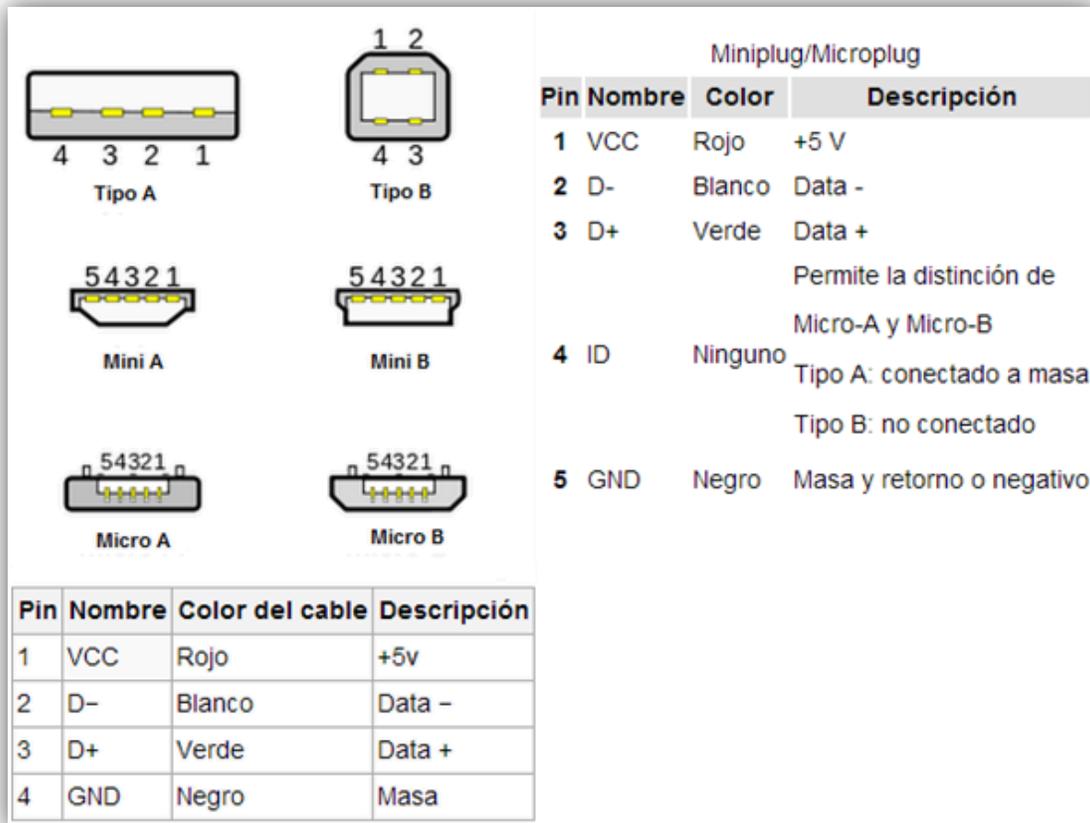


Fig. 3.3. Imagen de los diferentes tipos de conectores USB y su distribución en pines [24]

En la comunicación USB existen 4 tipos de transferencias, estas son:

- **Control:** Se utiliza para enviar datos de configuración, todos los equipos deben de soportar esta clase.
- **Bulk:** Es usada durante la transmisión de datos de forma masiva, es el tipo de transferencia más rápida, las utilizan comúnmente dispositivos como: *pendrives*, discos duros, impresoras etc.
- **Isócronas:** Permite mover grandes cantidades de datos con el tiempo de entrega asegurado, pero no garantiza la integridad de los datos. Se usa por ejemplo en audio y video donde la pérdida de pequeñas porciones de datos no son críticas, ya que es el único tipo que no soporta retransmisión automática de datos recibidos con errores.

- **Interrupción:** Se usa en dispositivos HID (*Human Interface Devices*), porque entrega los paquetes muy rápido (paquetes pequeños) por ejemplo en un ratón o teclado USB no se nota que se retrase cuando escribes o mueves el ratón. La desventaja aquí es que no tiene corrección de errores [26].

Ahora debemos establecer las diferentes clases de comunicación que existen para la comunicación USB. Una clase, especifica el modelo o plantilla que describe el estado y el comportamiento de los objetos que lo comparten. Las clases más utilizadas con microcontroladores en la comunicación USB son:

- **HID** (por sus siglas en inglés, *Human Interface Devices*): Para los dispositivos que utilizan esta clase de comunicación, no se requiere ningún driver específico, ya que este se encuentra almacenado en el sistema operativo, ejemplos de estos son los teclados, ratones, *joysticks* o todo aquel que interactúa directamente con los humanos, toman entradas de humanos, y pueden entregar una salida a los humanos.
- **MSD** (por sus siglas en inglés, *Mass Storage Device Class*): Utilizado por dispositivos de almacenamiento masivo como: *pendrives*, cámaras, discos duros etc. Utiliza transferencias del tipo de control Bulk y por interrupción. No se requieren drivers para utilizar esta clase, ya que generalmente se encuentra en la siguiente ruta: `"C:\Windows\System32\Drivers\usbstor.sys"`.
- **CDC** (Por sus siglas en inglés, *Communications Device Class*): En esta clase se puede incluir más de una interfaz, como una interfaz de control personalizado, interfaz de datos, interfaces de almacenamiento de audio. La clase se usa principalmente para módems, máquinas de fax y aplicaciones de telefonía para realizar llamadas de voz normales [26].

Esta clase se puede utilizar para equipos industriales, tales como maquinaria CNC (*Computer Numerical Control*) y en áreas de la robótica, permitiendo la desinstalación de los cables RS-232 para cambiarlos por cables USB, y seguir manteniendo la compatibilidad del software. El dispositivo se conecta a una línea de comunicaciones RS-232 y el sistema operativo en el lado USB hace que el dispositivo USB aparezca como un tradicional puerto RS-232. Los fabricantes de chips como FTDI, Microchip y Atmel dan ciertas facilidades para el desarrollo de USB mediante el uso de transmisión RS-232. Para usar esta clase se requiere un archivo .inf el cual será usado por el PC para instalar el controlador del dispositivo.

Dado lo anterior, se optó por usar la clase CDC, ya que con esta se da la posibilidad de usar un puerto serial, y a la hora de crear el software se mantendrá la compatibilidad con la manera de enviar los datos entre bluetooth y PC.

Para poder realizar la conexión mediante USB y comunicar el microcontrolador con la computadora en modo de emulador de puerto serial, se utilizará el compilador CCS PCWH, en donde se pueden programar microcontroladores de la familia PIC en lenguaje C. Dentro del compilador existen un par de librerías que será necesario modificar, para que a la hora de conectar el microcontrolador a la computadora el microcontrolador se reconozca correctamente, estas librerías son las siguientes:

- **usb_cdc.h**

Driver de la clase de dispositivo CDC USB que permite la emulación de un puerto serial RS-232.

- **usb_cdc_desc.h**

Descriptor del dispositivo de clase CDC.

Estas librerías controlan el driver USB del microcontrolador, pero para que el dispositivo sea reconocido por la computadora, es necesario que ésta cuente también con un driver USB. El driver de la computadora le permite reconocer el dispositivo USB así como los recursos que requiere para la conexión, este archivo es proporcionado de manera gratuita por Microchip dentro de su firmware de USB. El nombre del archivo es "**Mchpcdc.inf**".

A continuación se mostrará cómo se modificaron los 3 ficheros para desplegar la información deseada de este proyecto. Es importante recalcar que sólo pueden modificarse estos 3 documentos; si se modifica cualquier otro archivo del firmware que no sea alguno de los mencionados a continuación, es posible que el dispositivo USB no funcione adecuadamente.

- **Mchpcdc.inf**

Este archivo permite instalar en la computadora los recursos necesarios para la conexión con el dispositivo USB. Dentro de este archivo puede modificarse:

- Versión y fecha de creación del driver (**DriverVer**)
- Proveedor o creador (**MFGNAME**)
- Descripción y/o nombre (**DESCRIPTION**)
- Servicio (**SERVICE**)

Por lo que el archivo ya modificado en las partes mencionadas será el siguiente:

```
DriverVer=01/01/2013,1.0.2013
```

```
MFGFILENAME="mchpcdc"
```

```
MCHP="Microchip Technology Inc."
```

```
MFGNAME="X-10 USB-RS232"
```

```
DESCRIPTION="X-10 USB-RS232"
```

```
SERVICE="PROTOCOLO X-10"
```

- **Usb_cdc_desc.h**

Este archivo provee los elementos necesarios para describir el dispositivo USB. El código modificable se encuentra al final del archivo. Las cadenas dentro de este código se mostrarán durante la instalación del nuevo dispositivo de hardware, una vez instalado el driver de la computadora, se visualizarán los nombres dentro del archivo con extensión *.inf*. El *string 1* corresponde al nombre del fabricante y el *string 2* corresponde al nombre del producto.

Para calcular el tamaño de estos *string* es necesario utilizar la siguiente fórmula para asegurar un número par:

$$tam_{string} = (tam_{cadena}) * 2 + 2$$

Por lo que el archivo en su parte final queda de esta forma:

```
//string 1 - manufacturer
8, //length of string index
USB_DESC_STRING_TYPE, //descriptor type 0x03 (STRING)
'C', 0,
'C', 0,
'S', 0,
//string 2 - product
38, //length of string index
USB_DESC_STRING_TYPE, //descriptor type 0x03 (STRING)
'X',0,
'-',0,
'1',0,
'0',0,
' ',0,
'U',0,
'S',0,
'B',0,
'-',0
'B',0,
'L',0,
'U',0,
'E',0,
'T',0,
'O',0,
'O',0,
'T',0,
'H',0
```

- **Usb_cdc.h**

Este archivo incluye los descriptores necesarios para configurar el dispositivo como CDC. También provee el código para manejar las peticiones USB y las peticiones CDC, que son procesadas en la interrupción por USB. Para usar un descriptor personalizable es necesario cambiar la siguiente línea al principio del archivo para direccionar el archivo modificado anteriormente:

```
#include "..\include\usb_desc_cdc.h"
```

Ya que se personalizaron los archivos anteriores, ahora será necesario configurar el microcontrolador para que pueda trabajar correctamente con la parte de conexión USB. Lo primero que debe de hacerse es configurar la frecuencia de trabajo del módulo USB del PIC así como la frecuencia de trabajo del microcontrolador, para esto se toma como base la estructura interna del oscilador mostrada en el datasheet del PIC18F4550.

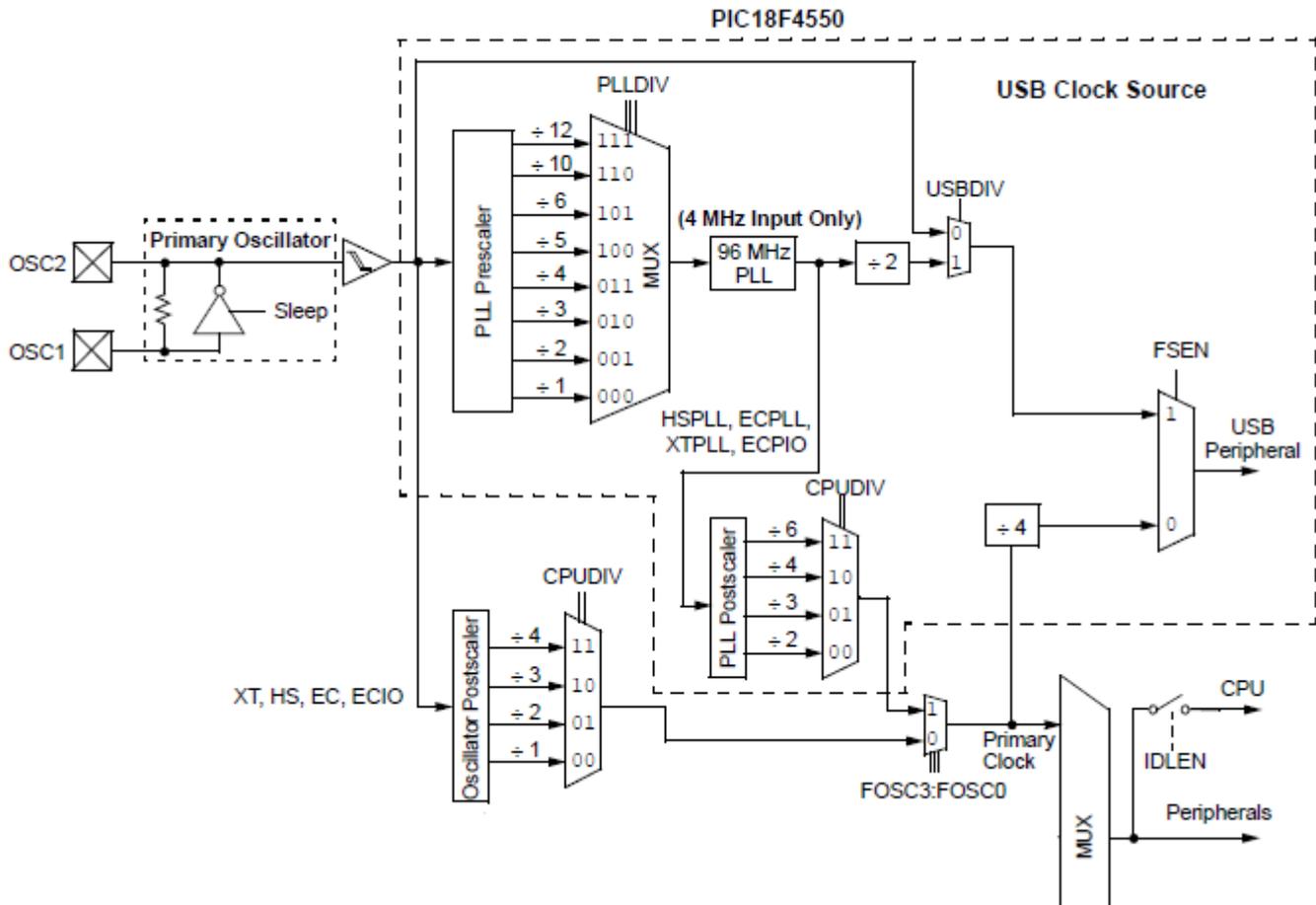


Fig. 3.4. Estructura interna del oscilador del PIC18F4550

El módulo USB del PIC tiene como restricción, que solamente funciona con una frecuencia de 48 [MHz], pero el usuario es libre de elegir una frecuencia diferente a ésta a la entrada del microcontrolador, este funcionará correctamente siempre y cuando también se activen ciertos divisores de frecuencia que se ajusten a la frecuencia de 48 [MHz].

En la figura 3.5 se ha marcado el camino a seguir para obtener la frecuencia necesaria, partiendo de un oscilador externo de 20 [MHz].

El camino marcado en la siguiente imagen, para obtener las frecuencias de trabajo del USB y del microcontrolador, deberá de ser plasmada en código para su funcionamiento, a continuación se explican partes de este en donde se configuran los elementos necesarios, partiendo de que a la entrada existen 20 [MHz].

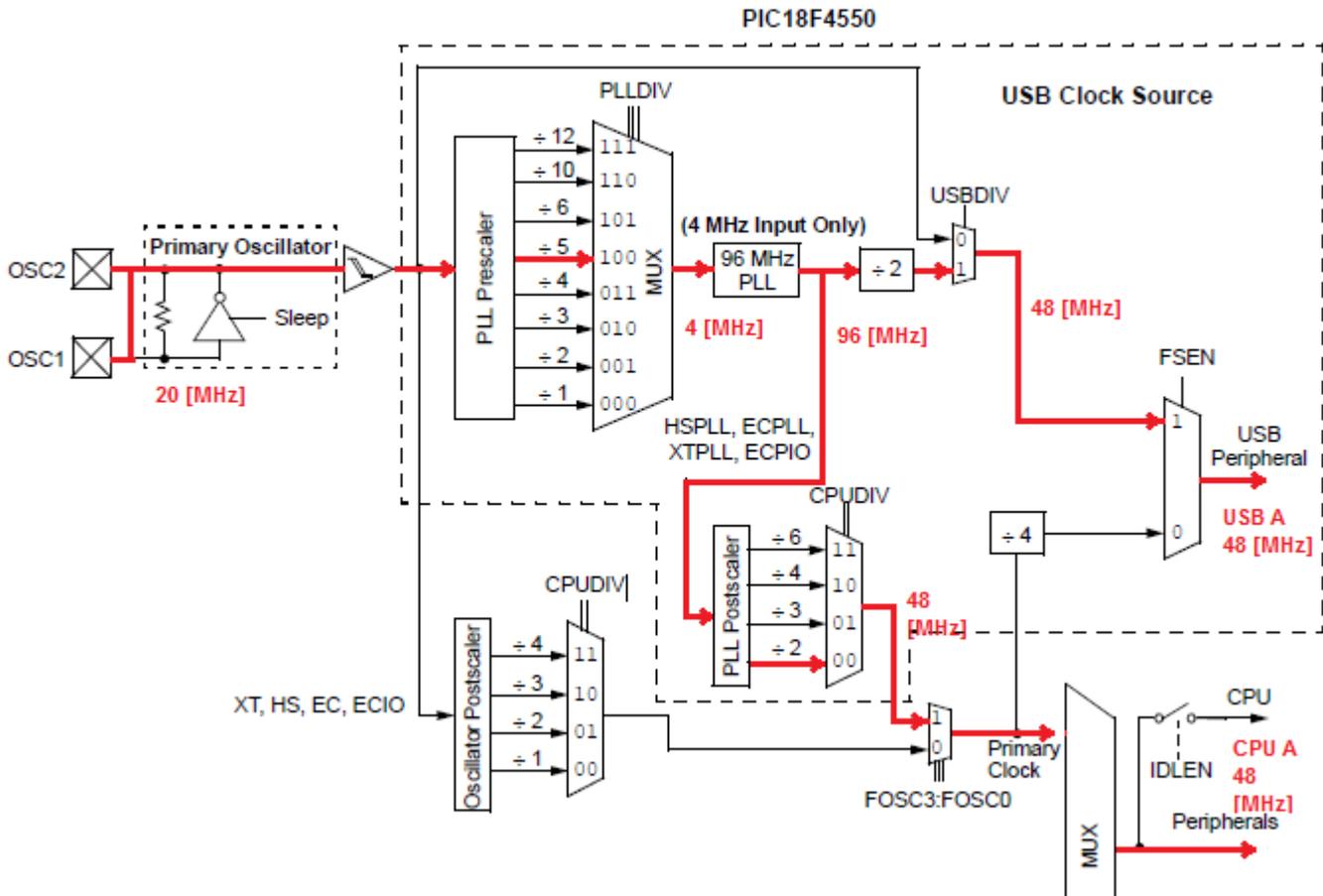


Fig. 3.5. Imagen del camino a seguir para obtener la frecuencia deseada para el USB y el microcontrolador

- **#FUSES HSPLL**

Con esta instrucción se habilita el pre-escalador PLL que va a dividir la frecuencia de entrada y permitir entregarle los 48 [MHz] que el módulo USB necesita.

- **#FUSES USBDIV**

Con esta instrucción, se habilita el divisor de frecuencia para que el módulo USB funcione.

- **#FUSES PLL5**

Como se observa en la figura 3.5 se requiere que a la salida del divisor del PLL existan 4 [MHz]. Vemos que para un reloj de 20 [MHz], se debe utilizar un divisor de 5, esto se realiza colocando *PLL5*. Es por ello, que es necesario tener en cuenta el valor del reloj a utilizar para ajustar el PLL tal como se requiere, de otra forma, la PC no podrá sincronizarse con el microcontrolador y no se logrará la comunicación entre los dispositivos.

- *#FUSES CPUDIV1*

Finalmente con esta línea de código, la frecuencia a la salida del divisor PLL (96 [MHz]) se dividirá por 2, por lo que la frecuencia de trabajo del PIC también será de 48 [MHz].

- *# FUSES VREGEN*

Con esta instrucción se habilita el regulador de 3.3 volts que usa el módulo USB, esto es necesario ya que se usa para alimentar las líneas del par diferencial D+/D- y así facilitar la detección de la velocidad USB. Para funcionar correctamente es necesario conectar un capacitor de aproximadamente 0.47 [µF] a tierra en la terminal V_{USB}.

Una vez realizado todo lo anterior, expliquemos el proceso para tener comunicación con la computadora. Primero que nada, se deberá de inicializar la comunicación en modo CDC y después el USB. A continuación se entra en un bloque en donde se estará comprobando continuamente si las tareas del USB como la enumeración, conexión y trasmisión de datos se están llevando a cabo. Ya cuando se ha realizado la enumeración y conexión por parte de la computadora, el microcontrolador sólo estará esperando por la información proveniente del software.

De esta forma, el enlace USB-PIC está listo para ser usado, por lo que podemos utilizar el software (Con la configuración del puerto predeterminada) y elegir el puerto COM que se le asignó al enlace USB, de manera que cualquier dato que se escriba a través del software lo recibirá el PIC.

El diagrama de la figura 3.7 explica lo antes mencionado.

Es importante mencionar que la computadora se encarga de detectar cualquier dispositivo que se conecte, cuando esto ocurre el host debe comenzar a enumerar el dispositivo mediante la obtención de determinada información, que no es más que el identificador del vendedor VID (*VID*, por sus siglas en inglés *Vendor Identifier*) y el identificador del producto (*PID*, por sus siglas en inglés, *Product Identifier*) que vienen dentro de los archivos 'mchpcdc.inf' y en el archivo 'usb_desc_cdc', por lo que, para que la computadora funcione correctamente, el VID y el PID que se encuentran dentro del mchpcdc (host) y el de la librería usb_desc_cdc deben de coincidir.

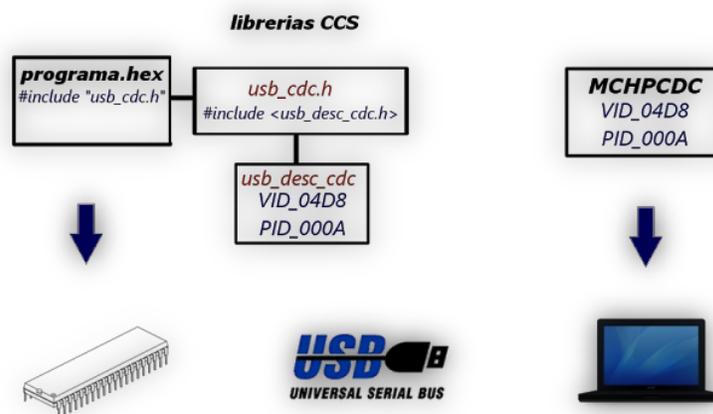


Fig. 3.6. Imagen en donde se muestra la relación entre los archivos de la librería de CCS y el controlador de microchip[27]

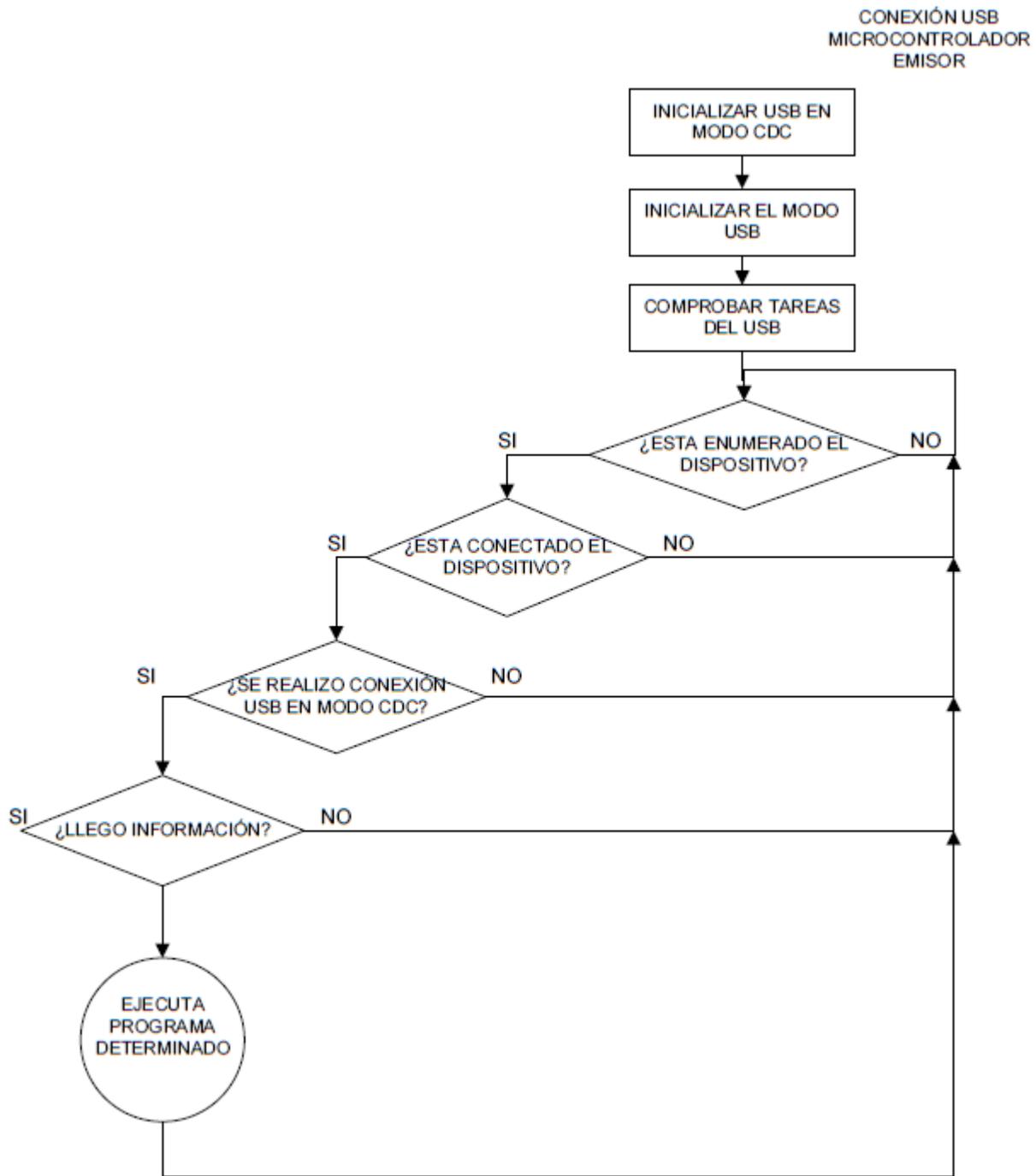


Fig. 3.7. Diagrama de bloques del proceso de conexión USB entre el microcontrolador y la computadora

Los *VID* y *PID* que se usarán en este proyecto son los que proporciona Microchip Technology, ya que no se debe de colocar cualquier valor a estos identificadores porque si se llegan a encontrar unos iguales en la computadora a la hora de la instalación, ésta no instalará el controlador y el dispositivo no funcionará. En caso de querer comercializar un producto es necesario solicitar 2 identificadores únicos a la “*USB Implementers Forum*”.

3.1.2.2. Comunicación a través del Bluetooth

Se denomina Bluetooth al protocolo de comunicaciones diseñado especialmente para dispositivos de bajo consumo, que requieren corto alcance de emisión y están basados en transceptores de bajo costo. Los dispositivos que incorporan este protocolo pueden comunicarse entre ellos cuando se encuentran dentro de su alcance. En este protocolo las comunicaciones se realizan por radiofrecuencia de forma que los dispositivos no tienen que estar alineados y pueden incluso estar en habitaciones separadas si la potencia de transmisión es suficiente [28].

Estos dispositivos se clasifican como "Clase 1", "Clase 2" o "Clase 3" en referencia a su potencia de transmisión, siendo totalmente compatibles los dispositivos de una clase con los de las otras.

Tabla 7. Tabla de los tipos de clases en el protocolo de comunicaciones bluetooth[28]

Clase	Potencia máxima permitida (mW)	Potencia máxima permitida (dBm)	Alcance (aproximado)
Clase 1	100 mW	20 dBm	~100 metros
Clase 2	2.5 mW	4 dBm	~10 metros
Clase 3	1 mW	0 dBm	~1 metro

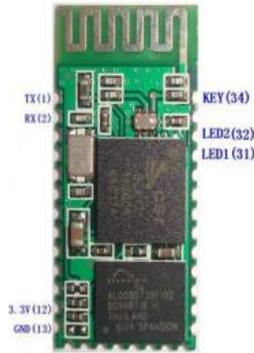
Dado que este proyecto es de domótica y se pretende que el control de los elementos se lleve dentro del mismo hogar, no se considera necesario usar dispositivos bluetooth de clase 1, aunque uno de clase 2, se supone adecuado ya que el alcance está dentro del rango buscado.

Debido a lo anterior, se optó por el módulo Bluetooth HC-05, distribuido en todo el mundo por *Guangzhou HC Information Technology Co.*, este es un dispositivo muy fácil de obtener, económico y sencillo de utilizar.

Con dimensiones tan pequeñas como un conector USB, el módulo Bluetooth HC-05 es una de las piezas de mercado más económicas que se puede encontrar. Una de las principales ventajas del módulo HC-05, además de su pequeño tamaño y sus buenas características de transmisión y recepción que le brindan un alcance muy amplio (por tratarse de un sistema local Bluetooth), es el bajo consumo de corriente que posee, tanto en funcionamiento como en modo de espera, es decir, alimentado con energía, pero sin conexión o enlace a otro dispositivo, por ejemplo, un dispositivo móvil. Otra característica interesante de este módulo es que una vez que ha realizado un enlace con otro dispositivo es capaz de recordarlo en su memoria y no solicita validación alguna (“1234” por defecto), pero si se activa el pin 26 (*KEY*) hacia la tensión de alimentación, esta información se elimina y el módulo solicitará nuevamente la validación del enlace. Otro detalle particular es que su tensión de alimentación desde 3,3 Volts y su bajo consumo (8 [mA] en

transmisión/recepción activa) lo transforman en un dispositivo ideal para trabajar con microcontroladores de la misma tensión de alimentación, logrando de este modo equipos portátiles que pueden ser alimentados durante muchas horas por baterías recargables o alcalinas AA, demostrando características excepcionales en aplicaciones médicas, o para actividades recreativas donde la fuente energética debe ser liviana y portátil.[29]

A continuación una breve descripción de los pines más importantes del módulo bluetooth HC-05.



HC-05

Fig. 3.8. Imagen del módulo bluetooth HC-05 y los pines más importantes[29]

Tabla 8. Tabla de las características del módulo bluetooth HC-05

Número de Pin	Características
1	TXD: Pin de envío de datos desde el bluetooth, debe de conectarse al pin RXD del microcontrolador.
2	RXD: Pin de captación de datos desde el bluetooth, debe de conectarse al pin TXD del microcontrolador.
11	RESET: El RESET del módulo puede activarse colocando un nivel bajo de voltaje, cuando está funcionando, este puede no estar conectado.
12	VCC: Alimentación del módulo, el voltaje estándar es de 3.3 [V], y puede llegar a trabajar hasta 5 [V].
13	GND: Tierra del módulo, debe conectarse a la misma tierra del microcontrolador.
31	LED 1: LED indicador de que el módulo esta encendido y conectado con otro dispositivo.

Con respecto al método de conexión del módulo Bluetooth, utiliza el protocolo SPP (*Serial Port Profile*), para enlazarse con otro dispositivo Bluetooth que también utilice este perfil, el cual establece una conexión serie virtual RS-232 inalámbrica.

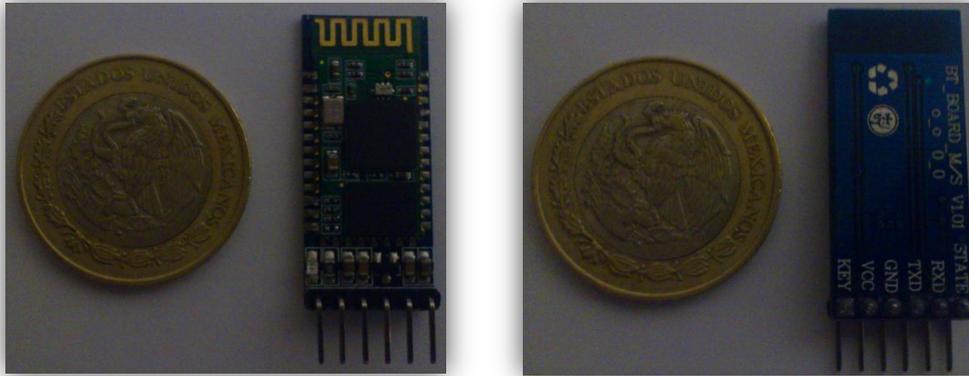


Fig. 3.9. Comparativa de tamaños entre el módulo bluetooth a usar y una moneda

Finalmente, cabe mencionar que las configuraciones por default de este módulo son las siguientes:

- Configuración RS-232 a 9600 [bps], 1 bit de parada, 0 bits de paridad, 8 bits de comunicación.
- Código de paridad=1234.
- Nombre de dispositivo: HC-05.

Dirección MAC (*Media Access Control address*): 00:11:07:19:02:49. Esta dirección es muy importante ya que es el identificador único del dispositivo, que será necesario a la hora de querer establecer comunicación con el dispositivo móvil.

Después de ver todo lo anterior y saber la manera y características con las que se comunica el dispositivo bluetooth con los demás dispositivos, toca comunicar el microcontrolador con la computadora; para poder hacer esto se continuará trabajando bajo el mismo programa que se desarrolló para el módulo USB en el punto pasado.

Antes que nada, se deberá de inicializar la comunicación en modo RS-232 a 9600 baudios, 8 bits, 1 bit de parada y sin paridad, a continuación entrará en una instrucción en donde se estará comprobando continuamente si el microcontrolador ha recibido dato, si esto es así, se procederá a la comprobación del dato mandado y si este concuerda con lo que se espera, se procederá con la decodificación de esta información. De esta forma, el enlace USB-bluetooth estará listo para ser usado, por lo que se podrá utilizar el software de la computadora (Con la configuración del puerto predeterminada) y elegir el puerto COM que se le asignó al enlace bluetooth, de manera que cualquier dato que se escriba a través del software de la computadora lo recibirá el PIC. Ya que el módulo bluetooth emplea una conexión inalámbrica y realiza una transmisión RS-232 con el microcontrolador, éste también estará en la capacidad de leer la información proveniente desde un dispositivo móvil.

A continuación un diagrama de bloques mostrando los pasos a seguir para que el microcontrolador logre comunicación y lectura de datos desde el módulo bluetooth.

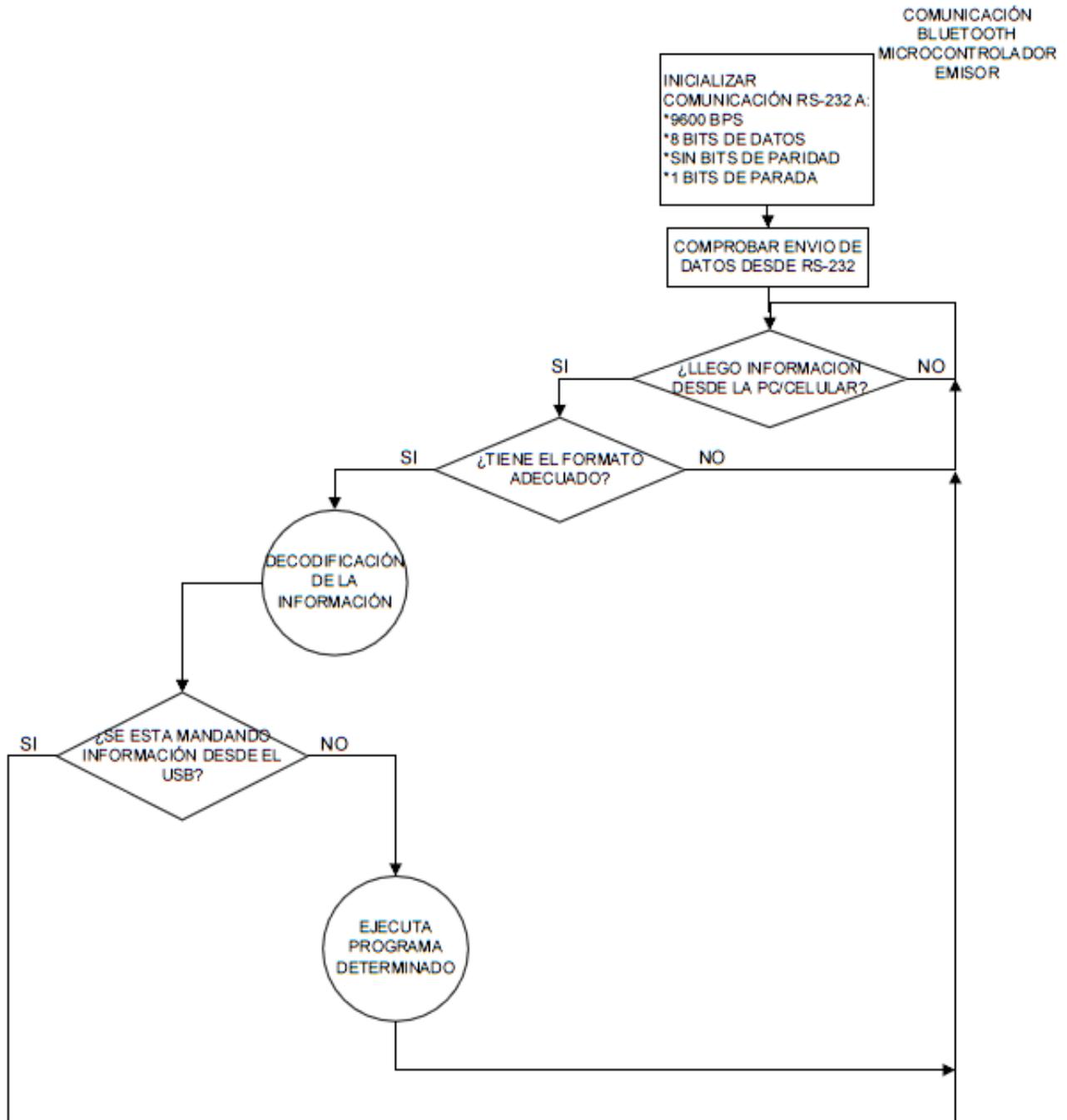


Fig. 3.10. Diagrama de bloques del proceso de conexión bluetooth entre el microcontrolador y la computadora

3.1.3. Fuente de alimentación

Todos los elementos para su funcionamiento necesitan una fuente de alimentación de 5 [V], y aunque de principio se pensó en llevar a cabo el diseño y construcción de esta, finalmente se optó por comprar cargadores para celular con USB incluido, estos tienen la característica de que dan 5

[V] exactos a la salida, proporcionan una corriente suficiente para alimentar a todos los elementos del sistema y son de menor precio que si se construyera uno con esas características.



Fig. 3.11. Cargador USB para alimentar al sistema

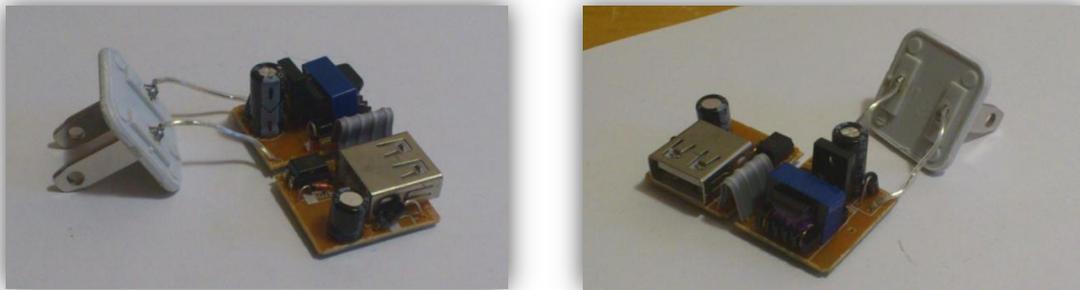


Fig. 3.12. Circuito interno del cargador

3.1.4. Detector de cruce por cero

Como se vio en el capítulo 1, en el protocolo X-10 la información que se envía es sincronizada con los cruces por cero de la línea de corriente alterna, llegando a tener una tolerancia máxima de 200 $[\mu\text{s}]$.

La manera más sencilla de realizar un detector de cruce por cero es creando una interrupción externa en el microcontrolador. Esta interrupción externa provoca que el microcontrolador suspenda cualquier programa que esté llevando a cabo y atienda el cambio que existió en su terminal, esta detección se dará no importando si el cambio de estado es de alto a bajo o viceversa.

Como es conocido, la línea de corriente alterna en México trabaja a 127 $[\text{V}_{\text{AC}}]$, si se decidiera conectar ese voltaje a una terminal del microcontrolador, ésta se dañaría, por lo que es necesario limitar la corriente que la terminal recibirá, para lo cual se tendrá que usar una resistencia en serie que limite esta corriente.

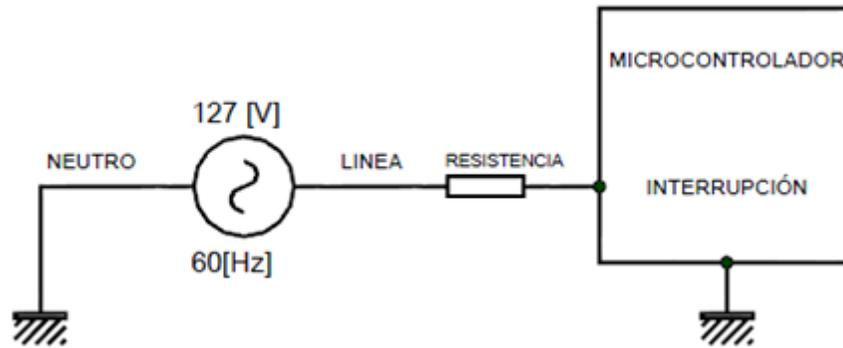


Fig. 3.13. Detector de cruce por cero

Teniendo que el voltaje nominal es de 127 [V]:

$$V_{\text{nom}} = 127 \text{ [V]}$$

Recordemos que para calcular la corriente máxima que llega a la entrada del microcontrolador, es necesario usar el voltaje pico-pico de la línea:

$$V_{\text{pico}} = V_{\text{nom}} * \sqrt{2} = 127 * \sqrt{2} = \sim 180 \text{ [V]}$$

La corriente máxima que un microcontrolador en general puede soportar en alguna de sus terminales cuando está siendo alimentado a 5 [V] es aproximadamente de 25 [mA], por lo que:

$$R_{\text{min}} = \frac{180 \text{ [V]}}{25 \text{ [mA]}} = 7.2 \text{ [k}\Omega\text{]}$$

$$R_{\text{min}} = 7.2 \text{ [k}\Omega\text{]}$$

Este valor de resistencia será el valor mínimo que habrá de colocarse entre el pin del microcontrolador que realice el cruce por cero y la línea de alterna.

La razón por la que se puede realizar lo anterior, es porque los microcontroladores a la entrada/salida de todos los pines digitales tienen diodos de protección, los cuáles detienen cualquier voltaje mayor al voltaje de alimentación del microcontrolador y menor a la tierra de alimentación (ver figura 3.14). Cuando el voltaje de corriente alterna proveniente de la línea está en su media onda positiva, la terminal del microcontrolador lo interpretará como un uno lógico; cuando el voltaje de corriente alterna está en su media onda negativa, la terminal del microcontrolador lo interpretará como un cero lógico.

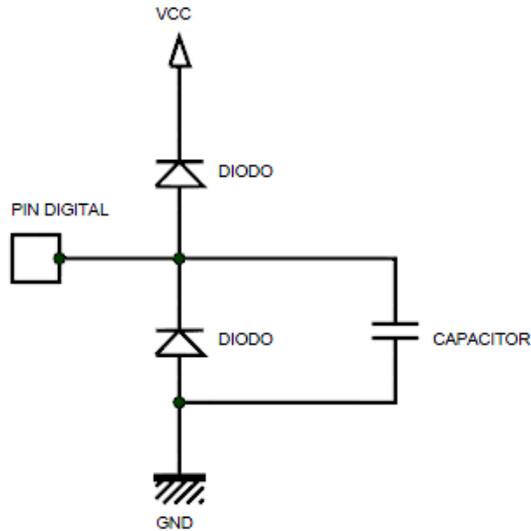


Fig. 3.14. Diodos de protección a la entrada de los pines digitales de un microcontrolador.

Como ya se sabe, en el protocolo X-10 se manda la información en el cruce por cero y si esto no llegara a ser posible se puede tener una tolerancia de 200 [μ s].

Mediante la siguiente ecuación podemos calcular el valor de voltaje en un determinado instante a una cierta frecuencia, aunque para nuestros propósitos queremos saber el tiempo que tarda el llegar a determinado voltaje, podemos despejar este y así saber cuánto tarda el microcontrolador en tener un cambio de estado y así detectar el cruce por cero.

$$V = V_{\text{pico}} * \text{sen}(2 * \pi * f * t)$$

$$t = \frac{\text{sen}^{-1}\left(\frac{V}{V_p}\right)}{2 * \pi * f}$$

Para nuestro caso en particular tenemos:

$$V_{\text{pico}} = 180 \text{ [V]}$$

$$f = 60 \text{ [Hz]}$$

Si tomamos a 3.3 [V] como el voltaje de cambio hacia el estado lógico alto, tenemos que:

$$V = 3.3 \text{ [V]}$$

$$t = \frac{\text{sen}^{-1}\left(\frac{3.3}{180}\right)}{2 * \pi * 60} = 48.633 \text{ [\mu S]}$$

Con esto vemos que al microcontrolador le toma 48.633 [μs] detectar el cruce por cero y por lo tanto empezar a mandar la instrucción; por lo que perfectamente entra en el rango antes mencionado.

3.1.5. Generador de 120 [kHz]

Para obtener la señal de 120 [kHz] con un ciclo de trabajo de 50% se habilitará uno de los timer que contiene el microcontrolador PIC18F4550; de principio, se usará el timer 2 que se encuentra en la mayoría de los PIC's.

El timer 2 es un temporizador de 8 bits que tiene la particularidad de tener un pre-escalador y un post-escalador, además este módulo cuenta con un registro de periodo llamado PR2 que marca el valor máximo que puede alcanzar la cuenta del registro TMR2. A diferencia de los otros temporizadores (timer 0, timer 1), el temporizador timer 2 no incrementa su cuenta hasta llegar a 0xFF y después al desborde, sino que incrementa su cuenta desde 0x00 con cada ciclo de instrucción, hasta que el valor del registro TMR2 coincide con el del registro PR2 y después, en el siguiente ciclo reinicia la cuenta desde 0x00.

En el timer 2 el pre-escalador tiene la misma función que en los otros dos timers y sirve como divisor de frecuencia antes de cada incremento, el post-escalador funciona como un divisor de frecuencia después de cada coincidencia entre los registros TMR2 y PR2. Si el post-escalador es 1:1 la bandera de interrupción TMR2IF se habilitará en cada coincidencia entre TMR2 y PR2, en cambio, si por ejemplo el post-escalador es 1:16, la interrupción se presentará cada 16 coincidencias.

Este temporizador, al igual que el Timer 1, puede habilitarse y deshabilitarse mediante el bit TMR2ON.

La temporización del Timer 2 está dada por la siguiente expresión:

$$\text{TempTMR2} = (\text{Preescalador} * \text{Postescalador} * T_{\text{instrucción}})(\text{PR2} + 1)$$

Donde:

- Pre-escalador = Valor del pre-escalador (1, 2, 4, 8, 16).
- PR2 = Valor cargado al registro PR2.
- Post-escalador = Valor del post-escalador (1, 2, 4, 8, 16).
- $T_{\text{Instrucción}} = 4/\text{Frecuencia del oscilador}$.
- TempTMR2 = Temporización dada en segundos.

Para lograr el tiempo deseado que se da mediante la frecuencia de 120 [kHz] debemos cargar el valor necesario al registro PR2.

$$PR2 = \frac{\text{TempTMR2}}{\text{Preescalador} * \text{Postescalador} * T_{\text{instrucción}}} - 1$$

Si tenemos en cuenta que la frecuencia de trabajo del PIC que se configuró anteriormente es de 48[MHz], tenemos que:

$$T_{\text{Instrucción}} = \frac{4}{\text{Frecuencia del oscilador}} = \frac{4}{48,000,000[\text{Hz}]} = 83.333\text{E}^{-9} [\text{s}]$$

Además si definimos que:

- Pre-escalador = 1
- Post-escalador = 1

Y la temporización que queremos con una frecuencia de 120 [kHz] es de:

$$\text{TempTMR2} = \frac{1}{120000[\text{Hz}]} = 8.333\text{E}^{-6} [\text{s}]$$

Por lo que tenemos que el registro PR2 para lograr una frecuencia de 120 [kHz] tendrá que ser de:

$$PR2 = \frac{8.333\text{E}^{-6}[\text{s}]}{1 * 1 * 83.333\text{E}^{-9}[\text{s}]} - 1 = 100 - 1 = 99$$

Los pulsos generados por el microcontrolador se deben transmitir a través de la red eléctrica para que después los módulos receptores decodifiquen esa información, pero debe de tomarse en cuenta que el pin del microcontrolador no se debe de conectar directamente a la línea de alterna ya que si se hace se quemaría el pin del microcontrolador, por lo que se debe de colocar una etapa de acople entre ambos:

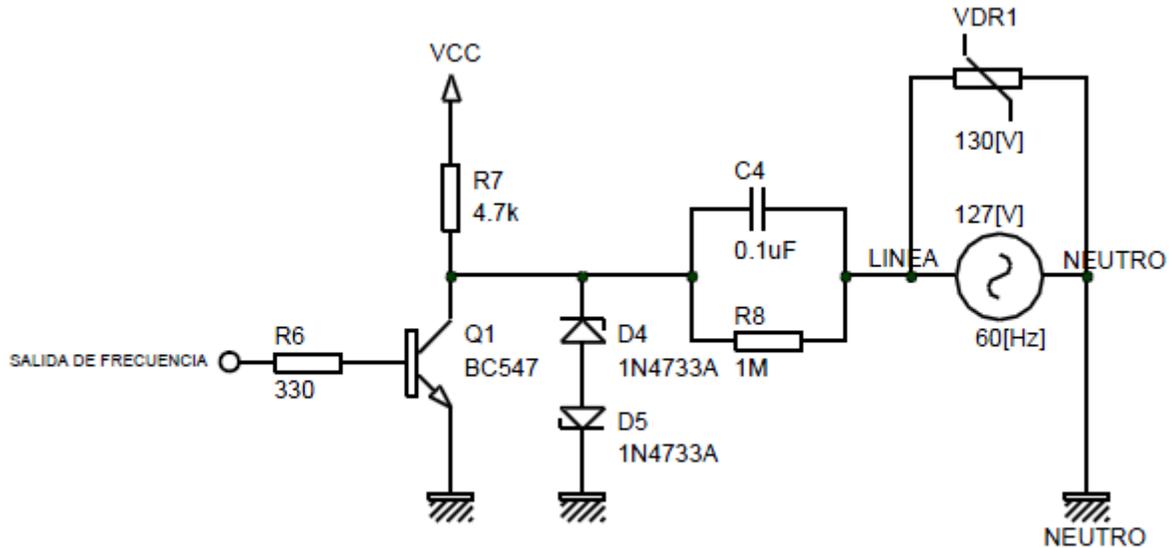


Fig. 3.15. Circuito de acoplo a la salida del microcontrolador

La etapa de acoplamiento está compuesta de un transistor, que servirá para asegurar que la señal de 120 [kHz] a la salida del microcontrolador sea totalmente cuadrada. Para el cálculo de las resistencias de base y de carga del transistor se hace el análisis de un transistor como interruptor, el transistor elegido para esto es el BC547B por ser uno de los más económicos y de propósito general.

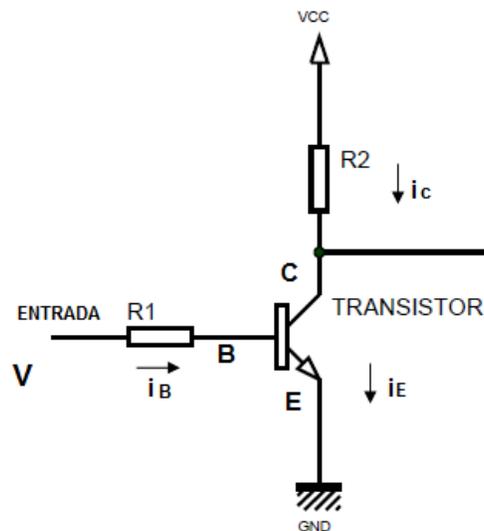


Fig. 3.16. Diagrama del transistor para el cálculo de las resistencias de base y de carga

De la hoja de datos del transistor tenemos que:

La corriente de base es de:

$$i_c = 100[\text{mA}]$$

Y la ganancia del transistor es de:

$$\beta = 200$$

Tenemos también que el voltaje necesario para que el transistor entre en saturación es de 0.7 [V].

$$V_{BE} = 0.7[\text{V}]$$

Además el voltaje de entrada al transistor proveniente del timer es de 5 [V].

$$V_{\text{Entrada}} = 5[\text{V}]$$

Partiendo de las ecuaciones relacionadas al transistor:

Para cuando está en saturación:

$$V_{\text{Entrada}} = V_{R1} + V_{BE} \dots \dots \dots (1)$$

$$V_{R1} = i_{R1} * R_1 \dots \dots \dots (2)$$

Para cuando está en corte:

$$V_{R2} = V_{CC} - V_{\text{Carga}} \dots \dots \dots (3)$$

$$V_{R2} = i_{R2} * R_2 \dots \dots \dots (4)$$

De la ecuación 1 tenemos:

$$V_{\text{Entrada}} = V_{BE} + V_{R1}$$

De la hoja de datos teniendo que para una $I_c = 100[\text{mA}]$ tenemos que tener una $I_B = 5[\text{mA}]$ y un voltaje de activación $V_{BE} = 0.7 [\text{V}]$; por lo que tenemos que:

$$V_{R1} = V_{\text{Entrada}} - V_{BE} = 5[\text{V}] - 0.7 [\text{V}] = 4.3 [\text{V}]$$

De la ecuación 2 tenemos:

$$R_1 = \frac{V_{R1}}{i_{R1}}$$

V_{R1} la calculamos en el paso anterior y del diagrama de la figura 3.16 sabemos que I_{R1} es igual a i_B por lo que:

$$R_1 = \frac{V_{R1}}{i_{R1}} = \frac{4.3[V]}{i_B} \dots\dots\dots (5)$$

Y sustituyendo en 5 tenemos:

$$R_1 = \frac{V_{R1}}{i_{R1}} = \frac{4.3[V]}{i_B} = \frac{4.3[V]}{5[mA]} = 860[\Omega]$$

Este valor de R_1 es el valor máximo que debe de tener para poder saturar el transistor a esa corriente de base calculada.

Para el cálculo de R_2 obtenemos el valor de V_{R2} de la ecuación 3:

$$V_{R2} = V_{CC} - V_{Carga} = 5[V] - 4.9[V] = 0.1[V]$$

Ahora despejamos R_2 de la ecuación 4 y sustituimos:

$$R_2 = \frac{V_{R2}}{i_{R2}} = \frac{0.1[V]}{i_{R2}}$$

Sabemos que i_{R2} es igual a i_c por lo que:

$$R_2 = \frac{V_{R2}}{i_c} = \frac{0.1[V]}{100[mA]} = 10[\Omega]$$

Este valor de R_1 es el máximo que debe de existir para tener una salida de 100[mA] a casi 5[V].

A continuación se colocan 2 diodos encontrados que funcionan como un regulador de voltaje; ya que durante el semiciclo positivo, el diodo de arriba conduce y el de abajo se encuentra en la zona Zener; por tanto, la salida se corta, el nivel de recorte es igual a la tensión Zener (diodo en ruptura) más 0.7 [V] (diodo con polarización directa); durante el semiciclo negativo, la acción se invierte, el diodo de abajo conduce, y el diodo de arriba entra en zona Zener. De esta manera, la salida es casi una onda cuadrada. Cuando mayor sea la onda senoidal de entrada, más perfecta será la onda cuadrada de salida.

Después de esto, se utiliza un circuito RC en paralelo, este funciona como un filtro pasa altas, esto debido a que la impedancia del capacitor crece cuando la frecuencia disminuye, de tal manera que para frecuencias muy bajas, la impedancia es muy grande y por lo tanto la corriente tiende a cero; dando como consecuencia que el circuito RC no deje pasar señales con frecuencias pequeñas.

Teniendo que la impedancia del capacitor es igual a:

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

Donde f es la frecuencia de la señal de entrada y C la capacitancia del capacitor.

Fijando el valor del capacitor en 0.1 [μ F] y el de la resistencia en 1 [M Ω], podemos calcular las impedancias a 60 [Hz] y a 120 [kHz]:

60 [Hz].

$$f = 60 \text{ [Hz]}$$

$$C = 0.1 \text{ [}\mu\text{F]}$$

$$Z_c = \frac{1}{2 * \pi * 60 * 0.1 * 10^{-6}} = 26.525 \text{ [k}\Omega\text{]}$$

120 [kHz].

$$f = 120 \text{ [kHz]}$$

$$C = 0.1 \text{ [}\mu\text{F]}$$

$$Z_c = \frac{1}{2 * \pi * 120 * 10^3 * 0.1 * 10^{-6}} = 13.263 \text{ [}\Omega\text{]}$$

Como podemos ver, la impedancia del capacitor con una frecuencia de 120 [kHz] es de 13 [Ω], sin embargo con una frecuencia de 60 [Hz], la impedancia es de 26.5 [k Ω].

Mediante este circuito de acoplo es posible agrupar la señal de 120 [kHz] con la señal de 60 [Hz] de la línea eléctrica.

Integrando todos los bloques diseñados previamente, se obtiene el módulo emisor, los resultados de este módulo serán presentados posteriormente en el capítulo correspondiente.

3.2. Módulo receptor

El módulo receptor es el encargado de detectar los pulsos de 120 [kHz] y de 1 [ms] que son transmitidos por medio de la red eléctrica en cada cruce por cero, así mismo se encarga de

adaptar esta señal a un nivel de voltaje aceptable para el microcontrolador y finalmente, debe de realizar la decodificación pertinente para generar la señal de control adecuada. El protocolo de comunicación definido en el primer capítulo nos indica el formato de la trama y la manera en cómo es transmitida la información, ahora se debe especificar como la información es reconstruida y el proceso de verificación que se realiza; para esto es necesario contar con las señales de cruce por cero, los códigos de casa y códigos de unidad.

Así, el módulo receptor debe de componerse de una etapa de detección de pulsos, un detector de cruce por cero, un microcontrolador que procese las instrucciones provenientes de la línea eléctrica y finalmente una fuente de 5 [V] que alimente a los componentes; a continuación se describirá como se diseñó cada uno de estos elementos.

3.2.1. Detector de pulsos

Para poder detectar los pulsos de 120 [kHz] presentes en la red eléctrica, el detector de pulsos estará compuesto de varios elementos como los son un circuito de desacoplo, un filtro pasa-altos, un amplificador y una adecuación en la señal antes de entrar al microcontrolador.

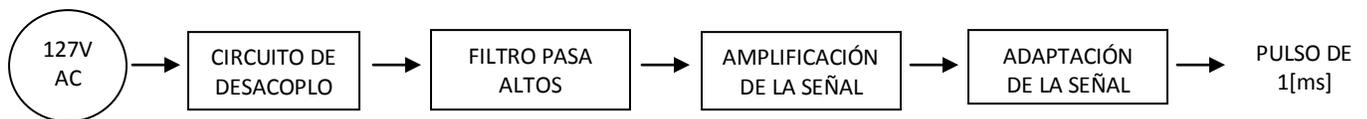


Fig. 3.17. Diagrama de bloques de los elementos que conforman al detector de pulsos

- **Circuito de desacoplo**

El circuito de desacoplo consiste en un circuito RC en paralelo, y que funciona como un desacoplador de impedancias, ya que debido a la configuración planteada y el valor del capacitor, la impedancia de este circuito es muy grande a frecuencias bajas, pero casi inexistente a frecuencias altas; pudiendo así desacoplar 2 señales de diferentes frecuencias.

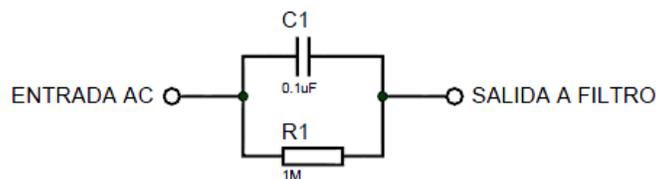


Fig. 3.18. Diagrama del circuito RC de desacoplo

- **Filtro pasa altos**

Después del circuito de desacoplo, se deben filtrar las señales y solamente dejar la señal de 120 [kHz], debido a que la frecuencia de 120 [kHz] es bastante mayor que la de 60 [Hz] de la línea eléctrica se utilizará un filtro pasa-altas tipo RC.

Como sabemos la frecuencia de corte de un filtro paso altas pasivo es:

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

Escogiendo un valor comercial muy pequeño:

$$C = 150 \text{ [pF]}$$

Si tenemos una f_c de 120 [kHz], y despejando de la ecuación el valor de la resistencia:

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

Sustituyendo valores tenemos que:

$$R = \frac{1}{2 * \pi * f_c * C} = \frac{1}{2 * \pi * 120 * 10^3 * 150 * 10^{-12}}$$

$$R = 8.842 \text{ [k}\Omega\text{]}$$

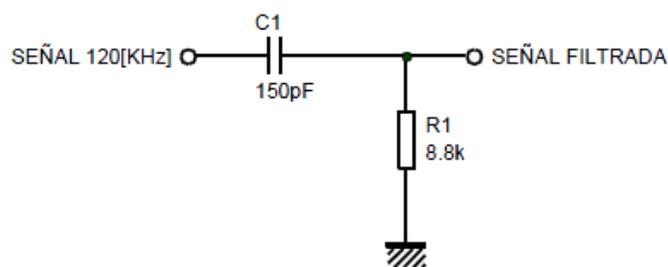


Fig. 3.19. Diagrama del filtro RC pasa altas

Así, finalmente obtenemos los valores de los componentes necesarios para implementar dicho filtro, y con el fin de obtener una señal más limpia, se hará pasar dicha señal por dos filtros de este mismo tipo.

- **Adaptación de la señal**

Una vez obtenida la señal de 120 [kHz], ésta será pasada por un circuito amplificador sintonizado, el cual se encargara de amplificar solamente la señal de 120 [kHz] sin importar el ruido con el que venga la señal, cuando existan frecuencias que no correspondan a la establecida, simplemente las atenúara. En la figura 3.20 se puede apreciar el funcionamiento ideal de un amplificador sintonizado.

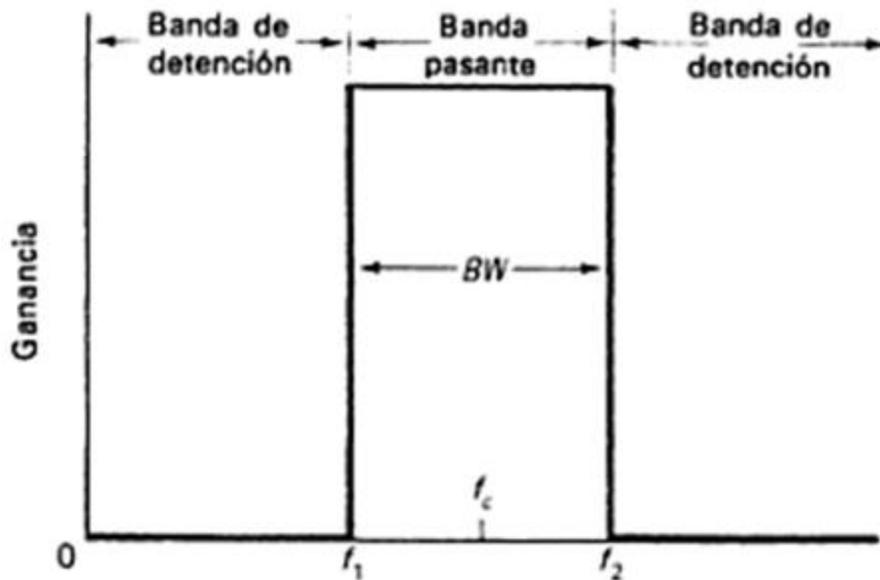


Fig. 3.20. Funcionamiento ideal de un amplificador sintonizado[30]

Este amplificador se escogió porque fue el de resultados más constantes además de que los elementos que lo constituyen son más económicos que las otras opciones que se probaron en el laboratorio (amplificadores de instrumentación, filtros pasa bandas de diferentes órdenes, detector de tonos).

En seguida, la señal obtenida será pasada por un circuito amplificador no sintonizado, el cual sirve para amplificar señales que están por encima de un cierto ancho de banda [31].

Estas dos etapas (Fig. 3.21), han sido consultadas en un manual de "Microchip" (AN236) [32], en el que se desarrolla la implementación de X-10 mediante el PIC16F877.

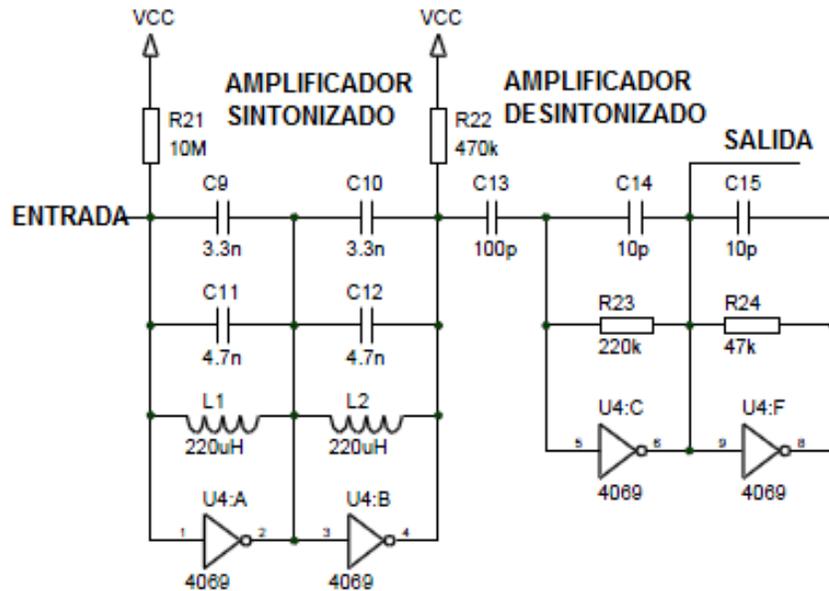


Fig. 3.21. Amplificador sintonizado-no sintonizado

Una vez obtenidas nuestras señales con la frecuencia adecuada y totalmente filtrada, ahora será necesario hacerlas pasar por un transistor con el fin de elevarlas debidamente a niveles lógicos.

A continuación, se deberá procesar estas señales provenientes del amplificador, que son pulsos de 1[ms] que contiene la frecuencia de 120 [kHz] y convertirlas a pulsos lógicos para poder facilitar el procesamiento mediante un microcontrolador. Para ello nuevamente consultamos el documento AN236, el cual proporciona a la salida pulsos en vez de señales con frecuencia, en la figura 3.22 se muestra el diagrama del transistor junto con el detector de envolvente que realizará esta parte.

El detector de envolvente es un circuito eléctrico, que tiene como entrada una señal de alta frecuencia y como salida la envolvente de la señal de entrada, generalmente esta formado por un diodo, un capacitor y una resistencia; en donde el capacitor almacena carga cuando la señal de entrada crece, y se descarga muy lentamente a través del resistor cuando ésta decrece. El diodo conectado en serie asegura que la corriente no circule en sentido contrario del circuito.

A la salida del detector de envolvente se colocará una compuerta inversora, ya que el transistor habrá negado la salida del amplificador sintonizado; esto para mantener la polaridad de la señales originales.

Así es como finalmente obtenemos los pulsos que serán procesados en el microcontrolador.

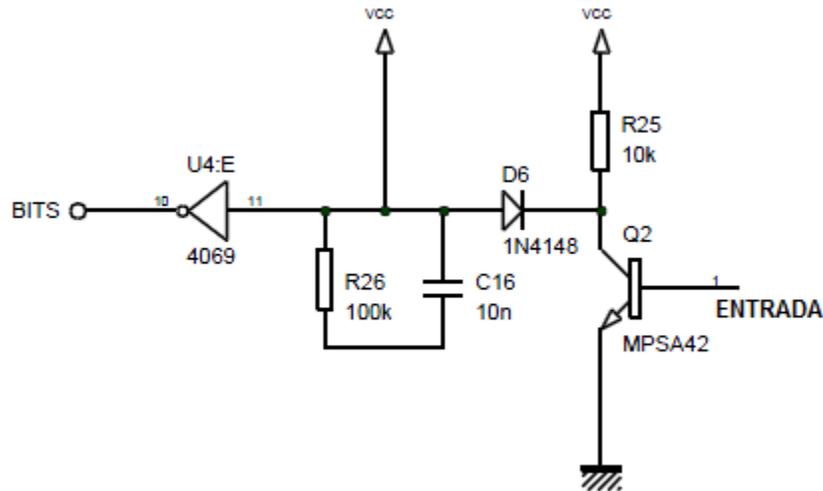


Fig. 3.22. Transistor, detector de envoltorio y compuerta inversora

3.2.2. Detector de cruce por cero

Mismo circuito que se usó en el módulo emisor.

3.2.3. Procesamiento de las señales

El microcontrolador encargado de procesar la información se encontrará esperando a que los datos recibidos tengan el formato del código de inicio; de no cumplirse esto, se asume que la información no es válida, pero cuando se recibe la trama del código de inicio, se procede a leer la trama de la casa, la unidad y la del comando; inmediatamente después se realiza una comparación, entre los códigos transmitidos y los propios de los dispositivos, si no concuerdan, entonces la información recibida no le corresponde y continúa esperando a que le llegue información adecuada; cuando la comparación es exitosa, se genera la señal de control para ejecutar el comando sobre el dispositivo a controlar, el diagrama de la figura 3.23 explica esto.

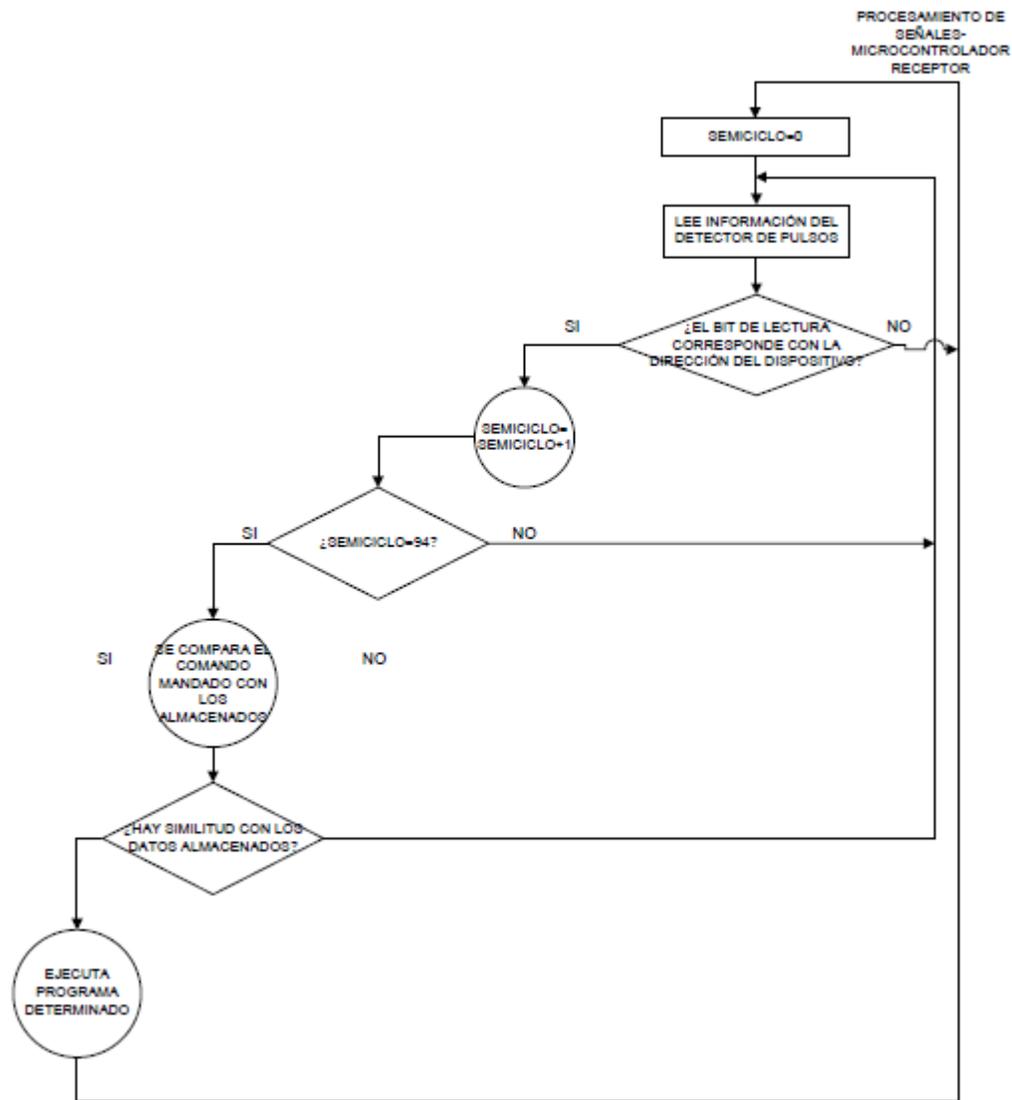


Fig. 3.23. Diagrama de flujo del módulo receptor

3.2.4. Fuente de alimentación

Mismo circuito que se usó en el módulo emisor.

3.2.5. Control de la iluminación

Con base en lo visto en el capítulo 1, se sabe que el protocolo X-10 soporta comandos para la variación de intensidad de luz, por lo que se decidió implementar un módulo que se encargue de esto; este módulo tendrá que ejecutar las siguientes instrucciones provenientes del módulo emisor:

- Encendido
- Apagado
- Iluminación Preestablecida
- Aumentar Intensidad
- Disminuir Intensidad

Para realizar este control de la iluminación se tiene pensado la implementación de un *dimmer* con el cual, se podrá controlar la intensidad de luz desde su ausencia hasta la intensidad máxima que puede alcanzar un foco común.

El funcionamiento de un *dimmer* se basa en el recorte de la tensión de alimentación del foco (127 [V_{AC}]); para ello es importante identificar los cruces por cero, ya que justo en ese momento es donde se corta la señal durante el tiempo que se requiera, según queramos aumentar o disminuir la intensidad de luz.

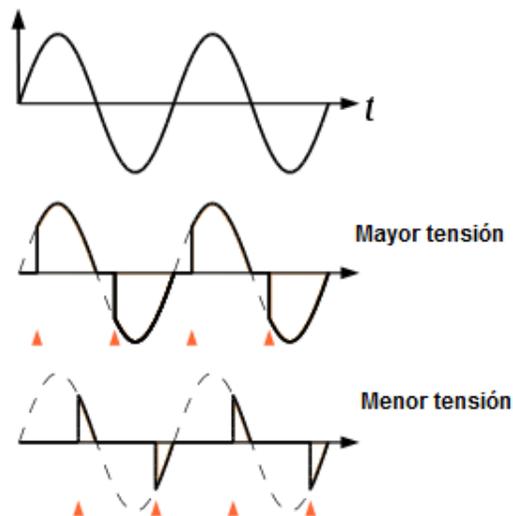


Fig. 3.24. Funcionamiento de un *dimmer* [33]

La disminución del valor eficaz en el foco, se logra recortando la señal en el momento de subida en el punto que se elija, si se corta la señal cuando el ángulo de fase de la onda es pequeño, el foco se encenderá casi al máximo, mientras que si la señal se corta al llegar casi a los 180° se encenderá muy poco.

Para que el microcontrolador ejecute de manera correcta las diferentes instrucciones del 'dimmer', es necesario conocer el tiempo en segundos que dura medio ciclo de la señal de corriente alterna, ya que es esta la que se usa para obtener las diferentes intensidades:

Sabiendo que:

$$1 \text{ [Hz]} = \frac{1}{\text{[s]}}$$

Y teniendo en cuenta que la frecuencia de la línea de corriente alterna es de:

$$f = 60 \text{ [Hz]}$$

Obtenemos que el periodo es igual a:

$$T = \frac{1}{60} \text{ [s]} = 0.016667 \text{ [s]}$$

De lo anterior, tenemos que medio ciclo posee una duración de:

$$\frac{T}{2} = 0.008333 \text{ [s]} = 8333.33 \text{ [\mu s]}$$

Así, entonces el microcontrolador de acuerdo a la instrucción enviada, recortará la señal después de cierto tiempo del cruce por cero, esto permitirá variar los niveles de intensidad en el foco. El diagrama de bloques de la figura 3.25 muestra lo anteriormente descrito.

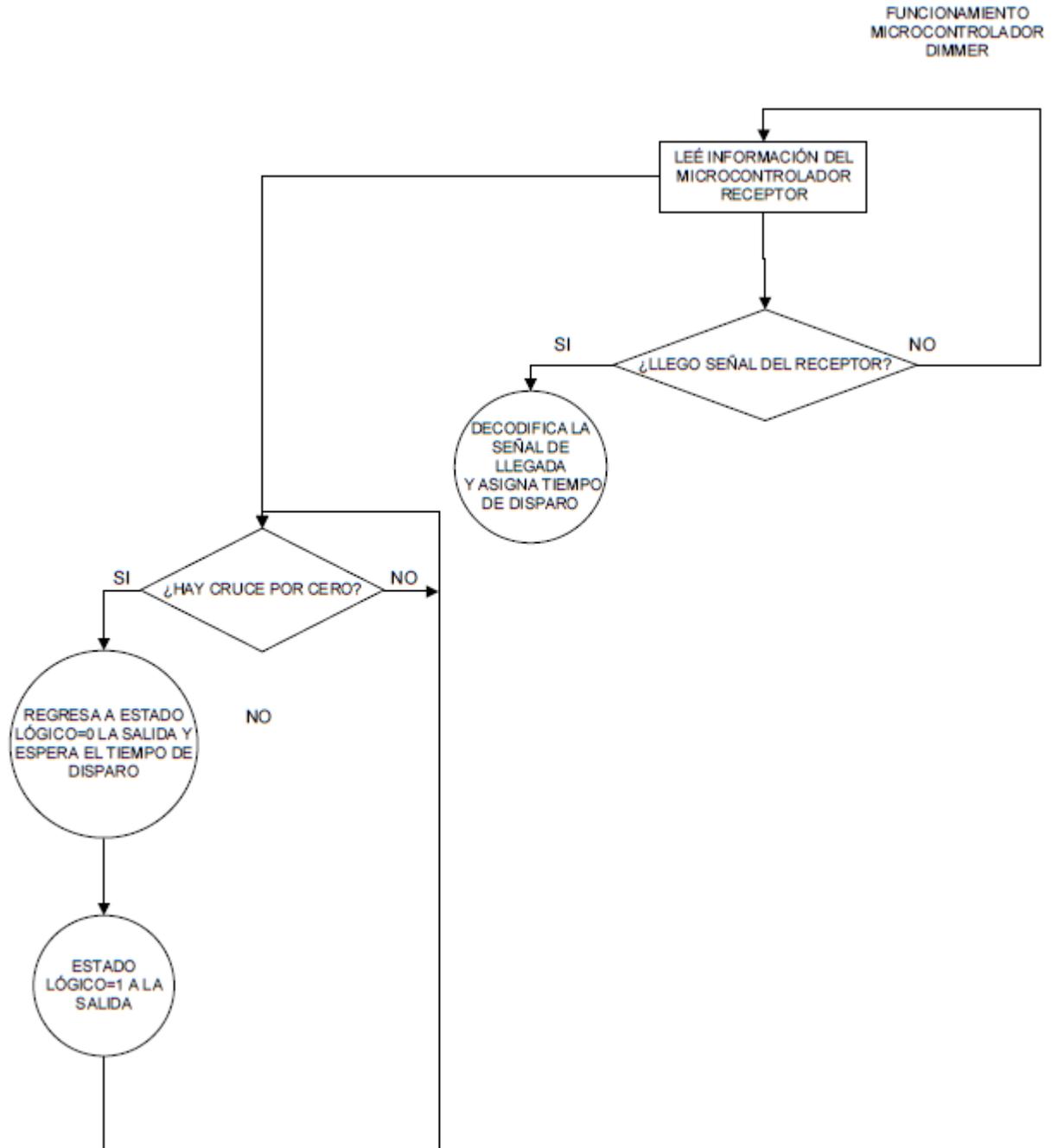


Fig. 3.25. Diagrama de flujo del módulo del dimmer

Para lograr el recorte requerido se empleará un circuito que nos ayudará a controlar mediante señales lógicas el paso de la corriente alterna a través del foco; dicho circuito consta de un TRIAC (Triodo de corriente alterna), un MOC (Optoacoplador) y un microcontrolador. El TRIAC servirá de interruptor capaz de conmutar el paso de la corriente alterna a través del foco, para ello posee dos terminales llamadas MT1 Y MT2 a través de las cuales deja pasar corriente una vez que el MOC se lo ha indicado por medio de una terminal llamada GATE donde es puesto un pulso, así mismo, el MOC también aislará la corriente alterna de los circuitos digitales, finalmente, el microcontrolador se encargará de realizar la lógica del dimmer, indicándole al MOC cuando deberá avisarle al TRIAC que conmute. El circuito antes descrito, se muestra en la figura 3.26.

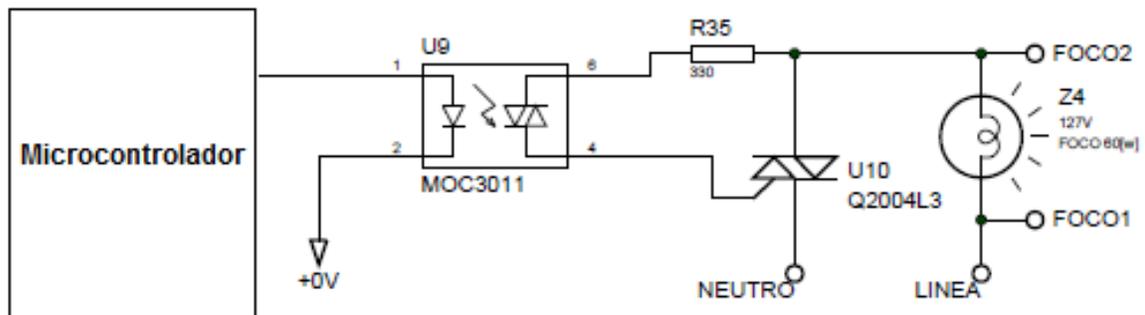


Fig. 3.26. Circuito para control de un dimmer

Capítulo 4: Diseño del software

En este capítulo se muestra como se desarrollaron las aplicaciones tanto para la computadora como para el dispositivo móvil, explicando su contenido y su funcionamiento.

4.1. Aplicación de escritorio

A modo de ejemplo, se desarrolló una aplicación de escritorio que permitirá el control de los diferentes dispositivos de la casa de forma gráfica e intuitiva.

A continuación una breve explicación de su estructura y diseño, así como ejemplos de su funcionamiento.

4.1.1. Elección del entorno de desarrollo

Para la aplicación se decidió utilizar Visual Studio 2010 de Microsoft, programando en C#. Los motivos por los que se eligió este entorno de desarrollo son los siguientes:

- Se considera que Microsoft Visual Studio es la mejor forma de crear aplicaciones para Windows de forma rápida.
- Visual Studio permite desarrollar interfaces gráficas muy agradables para entornos de Windows. Dado que la aplicación, aunque esté desarrollada a modo de ejemplo, debe ser de agradable aspecto y fácil de usar.
- C# es un lenguaje de programación con el que ya se ha trabajado antes, de esta manera no se necesitaría aprender otro entorno de desarrollo.



Fig. 4.1. Icono del lenguaje de programación C#[34]

4.1.2. Estructura del programa

La estructura del programa está pensada de tal manera que el usuario pueda escoger fácilmente el puerto de comunicaciones, la dirección del dispositivo y el comando a ejecutar.

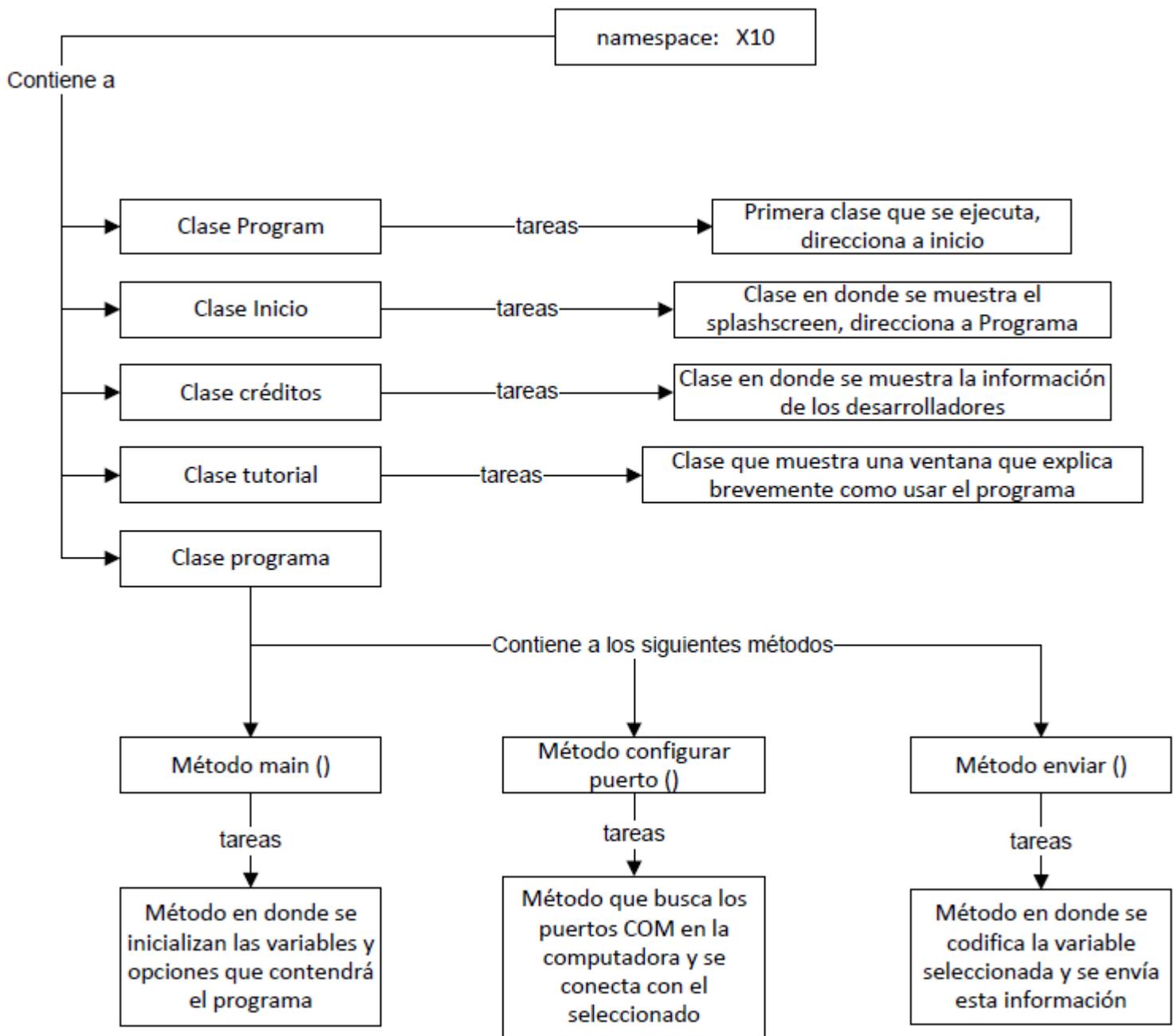


Fig. 4.2. Diagrama de bloques mostrando la estructura del programa de C#

4.1.3. Interfaz gráfica

La interfaz gráfica se divide en zonas, en este caso son 4: La clase inicio, la clase programa, la clase tutorial y la clase créditos, a continuación se explica el funcionamiento de cada una de estas clases

- **Clase inicio**

Al iniciar el programa se muestra una pantalla de inicio, esto sólo es para darle una mejor apariencia al programa. Esta pantalla tiene una duración de 5 segundos.

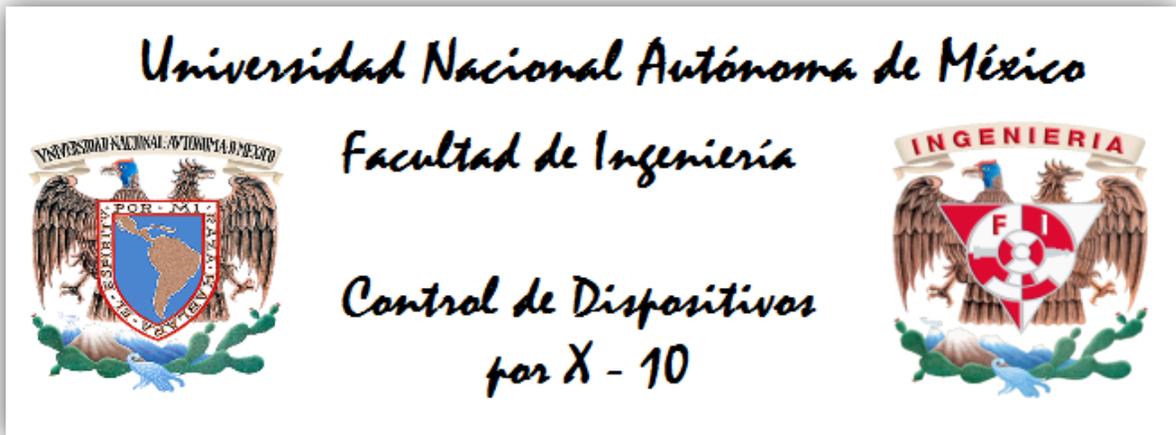


Fig. 4.3. Imagen de la pantalla de inicio del programa

- **Clase programa**

A continuación, se abrirá una ventana como la de la imagen 4.4, esta se encuentra dividida en 3 partes; la barra de herramientas, el bloque de configuración y el bloque de instrucción.

- Barra de herramientas:

Aquí se encuentran botones que ofrecen diversas funciones, tales como: agregar-eliminar una zona, editar las imágenes de cada dispositivo, habilitar-deshabilitar dispositivos, ayuda, salir del programa, cerrar puerto y créditos.

- Bloque de configuración:

Aquí el usuario deberá de seleccionar el puerto serial que pertenece al dispositivo emisor, ya sea para la conexión USB o conexión bluetooth. En dado caso de que no se seleccione el puerto correcto o no se seleccione un puerto y se quiera entrar al bloque de instrucción, se mandará un mensaje de error como el de la imagen 4.6.

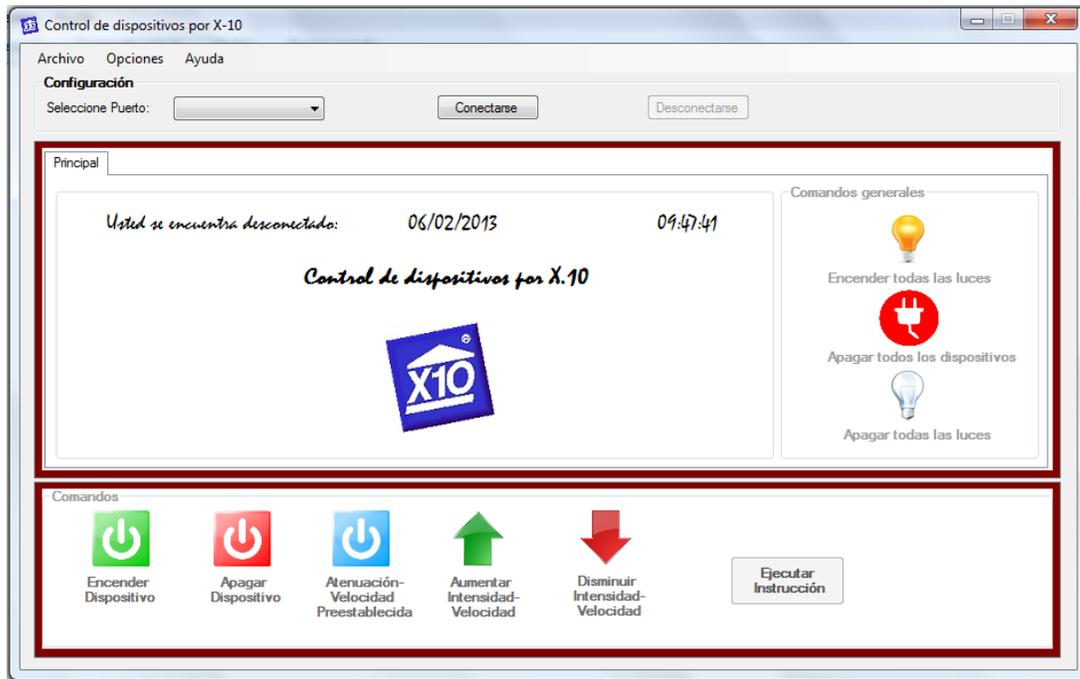


Fig. 4.4. Imagen de la pantalla principal del programa

Cuando se haya seleccionado el puerto correcto, aparecerá un mensaje de conexión exitosa, además el bloque de configuración se deshabilitará y se habilitará el bloque de instrucción, como se muestra en la siguiente imagen.

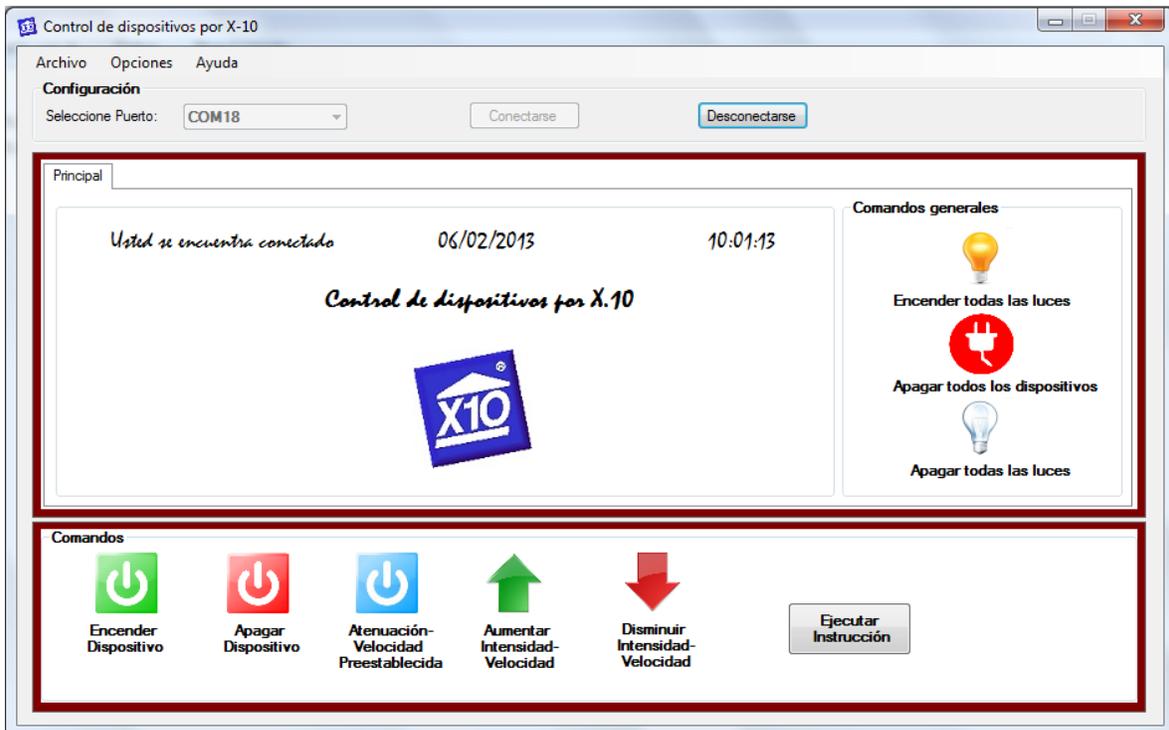


Fig. 4.5. Pantalla principal del programa habilitada para mandar información

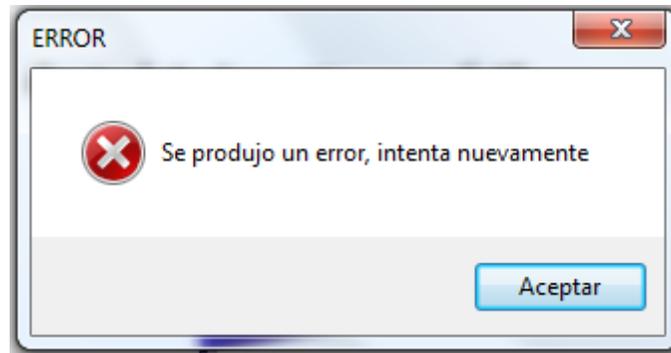


Fig. 4.6. Mensaje de error mandado cuando se conecta al puerto equivocado

- Bloque de instrucción

En este apartado, el usuario es capaz de controlar cualquier dispositivo, lo único que se debe de hacer es seleccionar la zona de las que se quiera tener control y el dispositivo que se quiera controlar, además del comando que se quiera ejecutar. Cuando se haya hecho click en un dispositivo y en un comando se estará disponible para enviar la instrucción.

Esto se muestra en la siguiente imagen.

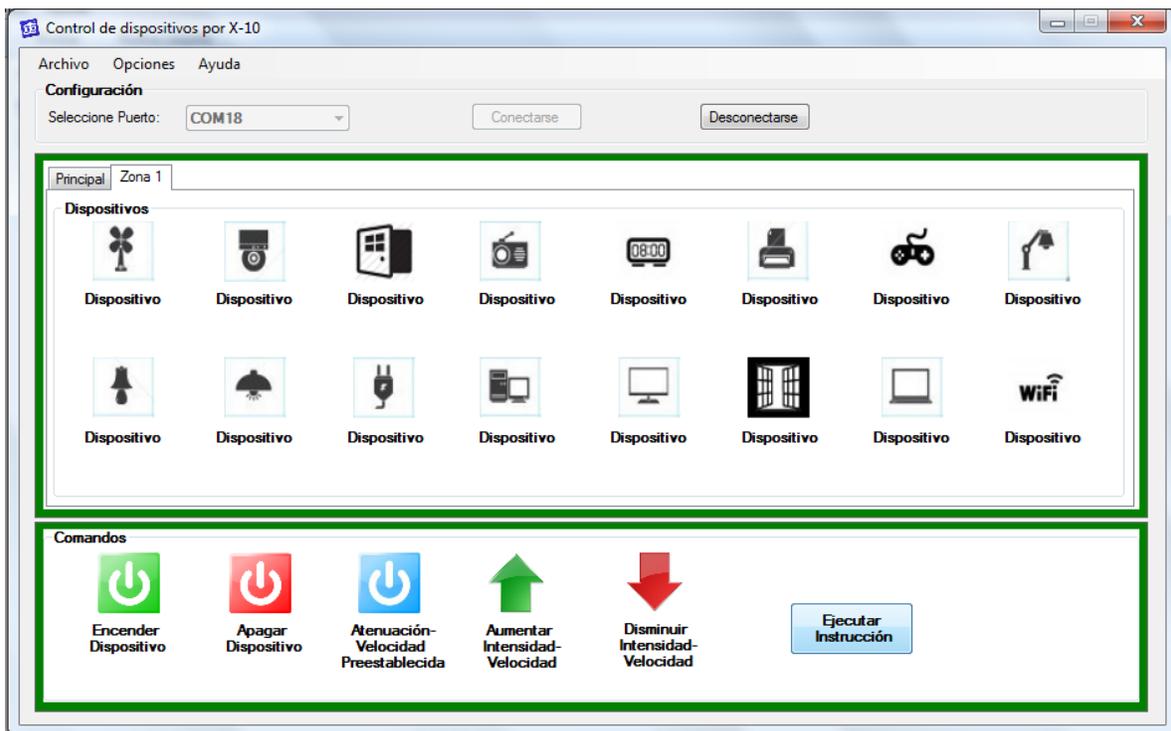


Fig. 4.7. Pantalla principal habiendo elegido el dispositivo y comando

El programa soporta las 16 direcciones de 'Casa' y los 16 dispositivos para cada una de las anteriores (256 dispositivos), además de los comandos más comunes, tales como:

- Apagar todas las luces
- Encender dispositivo
- Atenuación/Velocidad preestablecida
- Encender todos los dispositivos
- Aumentar intensidad/velocidad
- Apagar dispositivo
- Apagar todos los dispositivos
- Disminuir intensidad/velocidad

En el software planteado se puede cambiar los nombres de todos los dispositivos a controlar, así como cambiar las imágenes por las que el usuario crea más conveniente, habilitar y deshabilitar zonas y dispositivos; todos estos cambios serán guardados automáticamente por el software.

Cuando el usuario haya elegido la zona, dispositivo y comando deseado, podrá apretar el botón de 'ejecutar instrucción', en caso de un envío exitoso, el módulo emisor encenderá un led en señal de aviso.

- **Clase tutorial**

En esta clase se puede consultar un pequeño manual de cómo usar el programa.

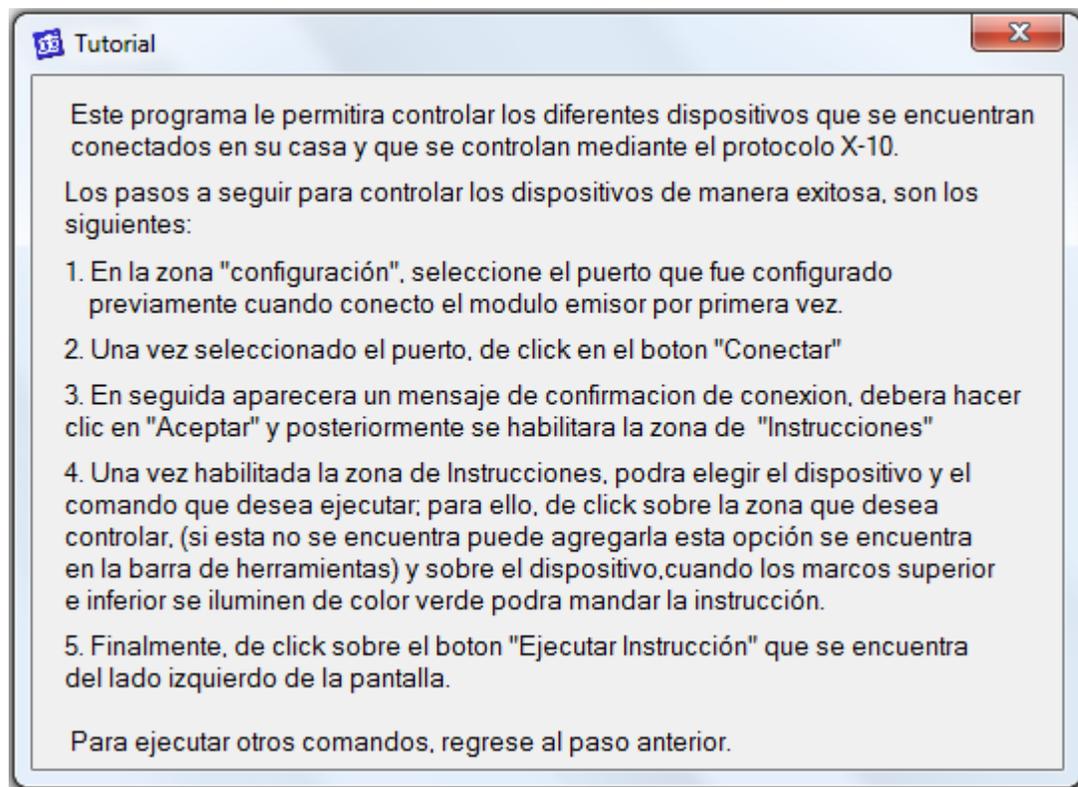


Fig. 4.8. Imagen de la ayuda desplegada por el programa

- **Clase créditos**

En esta clase se despliega información sobre quien ha desarrollado el programa.

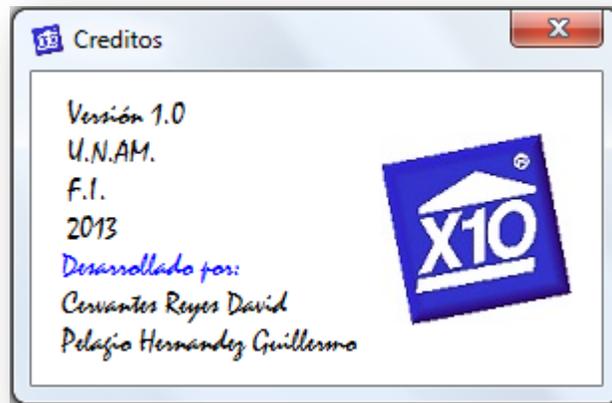


Fig. 4.9. Imagen del programa en donde se despliega información de los desarrolladores

4.1.4. Transferencia de datos entre la aplicación y el módulo emisor

La manera en la que se realiza la transferencia de información entre la aplicación para la computadora y el módulo emisor es la siguiente:

Dado que la computadora reconoce la conexión del módulo emisor como un puerto serial, ya sea usando el módulo con conexión USB o el módulo Bluetooth, lo único que se debe de hacer es mandar en una cadena los datos de 'casa', 'comando' y 'dispositivo' mediante el puerto serial.

```
//MENSAJE QUE SE MANDA AL MÓDULO EMISOR  
serialPort1.Write (zone+device+instruction+"#");
```

'zone', 'device' e 'instruction', se refieren al valor que les fue asignado cuando se seleccionó una opción de botón en el programa. A estas variables les son asignadas caracteres que van desde la 'A' hasta la 'P' y que se refieren a los 16 diferentes valores de 'casa', de dispositivo y de comando. En la cadena enviada se utiliza un identificador del final de esta (en este caso el símbolo #), con el fin de que a la hora en que el microcontrolador esté leyendo la información sepa que ya no se mandarán más datos.

4.2. Aplicación móvil

También a modo de ejemplo, se desarrolló una aplicación móvil para dispositivos Android, la cual permite al igual que la aplicación de escritorio, el control de los dispositivos X-10 de la casa, pero con una interfaz mucho más simple.

4.2.1. Elección del entorno de desarrollo

Las razones por la que se eligió desarrollar una aplicación para Android y no para cualquier otro sistema operativo, son las siguientes:

- Contrariamente a iOS, Blackberry, Symbian o Windows Mobile, Android permite crear y distribuir libremente cualquier aplicación sin necesidad de usar el Android Market, lo que nos ahorra pedir permisos a Google y da mucha facilidad a la hora de desarrollar aplicaciones.
- El lenguaje de programación es Java, lo que nos permite programar la aplicación con suma facilidad.
- Ya se había programado con anterioridad aplicaciones para Android, lo que nos permitió ahorrarnos el aprendizaje del nuevo sistema.
- El entorno de desarrollo para aplicaciones Android es Eclipse, un entorno de desarrollo muy amigable en donde es muy fácil programar.

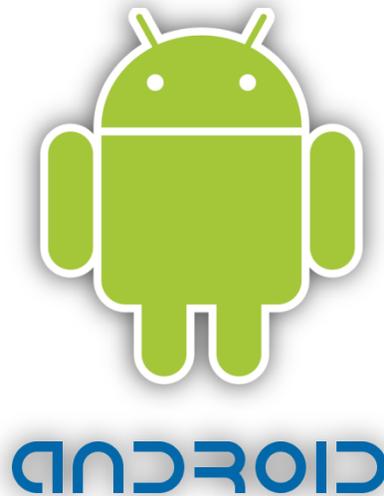


Fig. 4.10. Icono distintivo del sistema operativo Android[35]

4.2.2. Estructura del programa

La estructura del programa está pensada de tal manera que el usuario pueda conectarse fácilmente con el dispositivo bluetooth y además le sea fácil la elección del dispositivo a controlar y el comando a ejecutar.

La aplicación móvil es muy sencilla y su estructura de clases es la siguiente:

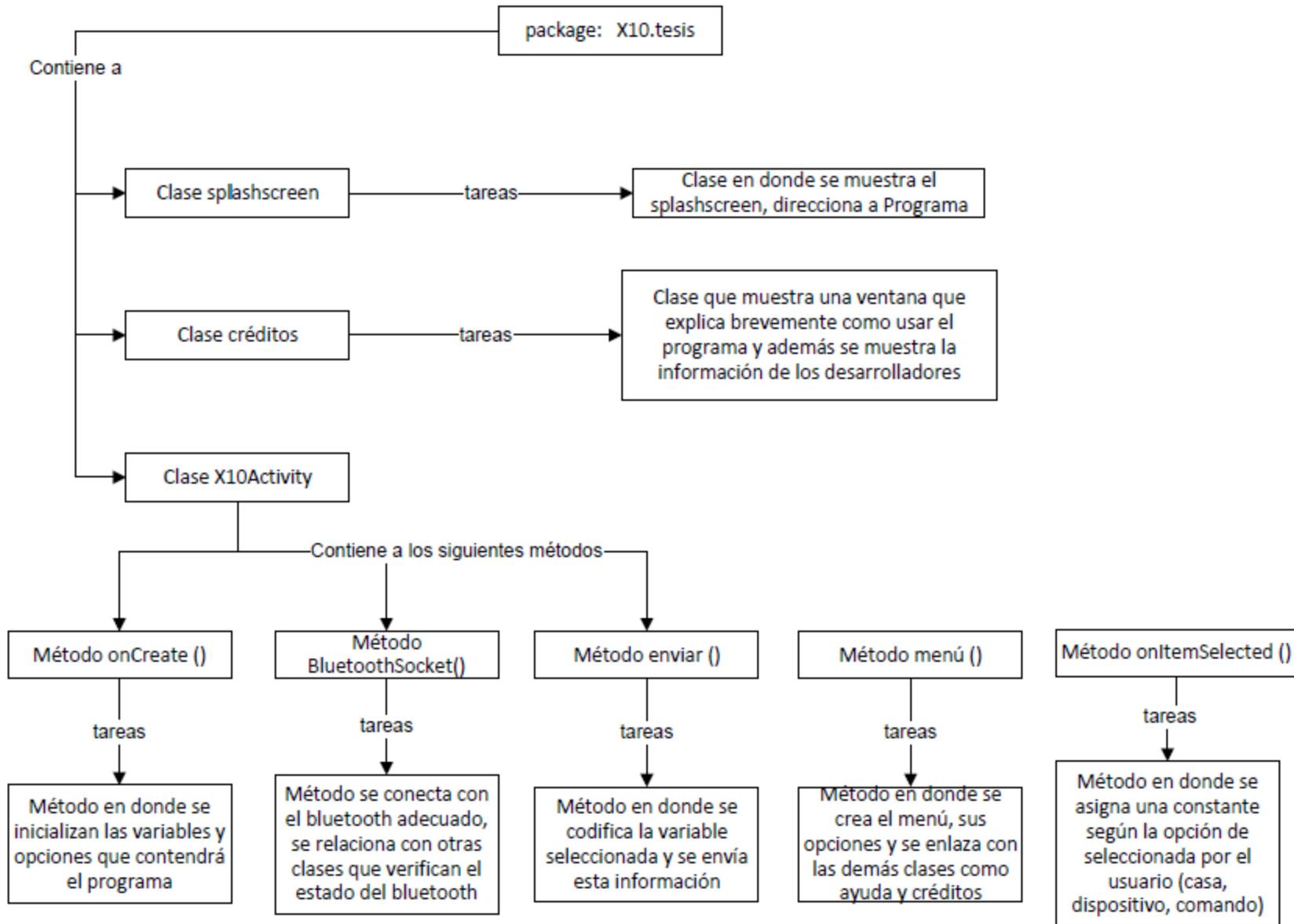


Fig. 4.11. Diagrama de bloques mostrando la estructura del programa de Android

4.2.3. Interfaz gráfica

Para hacer uso de esta interfaz, lo primero que hay que hacer después de haber generado el archivo .apk en eclipse, es instalarlo en el dispositivo móvil deseado, después de haber hecho esto, deberá de aparecer como una aplicación más en el dispositivo.



Fig. 4.12. Imagen de la aplicación ya instalada en el celular

A continuación se ejecutará esta aplicación y lo primero que se verá es una pantalla de inicio donde se desplegará el nombre de la aplicación y su icono.

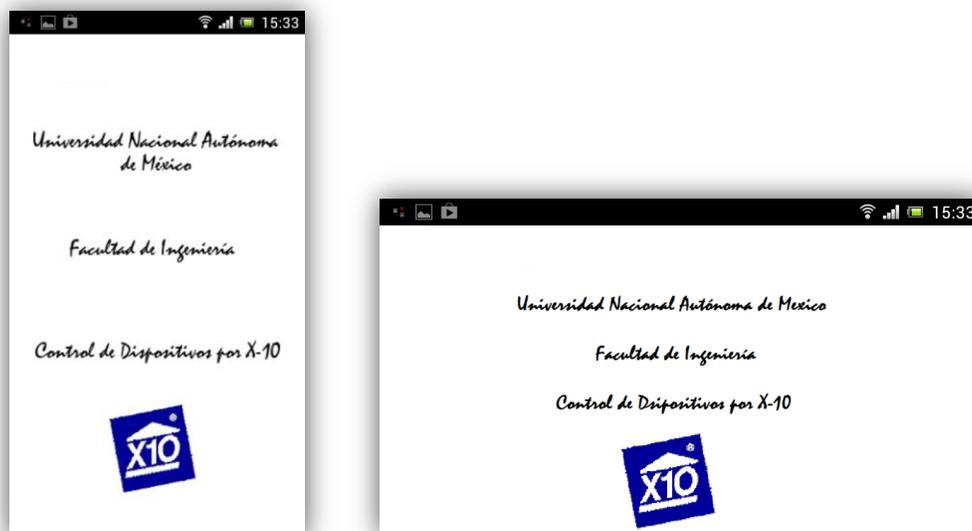


Fig. 4.13. Imagen de la pantalla de inicio de la aplicación

Inmediatamente después se pedirá que se active la comunicación bluetooth del dispositivo, en dado caso de que no se quisiera activar la comunicación en ese instante, la aplicación se cerrará, sin embargo, si se opta por activar el bluetooth, el dispositivo procederá a activarlo y a continuación entrará al programa.

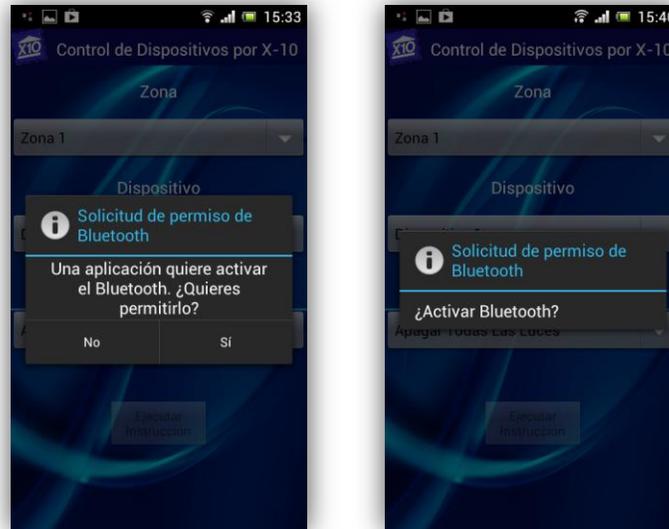


Fig. 4.14. Imagen del proceso que tiene el celular para activar el bluetooth

Ya que se encuentra activado el bluetooth y la aplicación ha entrado en la pantalla principal, la aplicación se dispondrá a conectarse a la dirección MAC perteneciente al bluetooth en el módulo emisor, este logrará la conexión si el módulo emisor esta a una distancia de hasta 10 metros a la redonda del dispositivo móvil y el módulo emisor esta encendido, si no es así, la aplicación continuamente buscará la dirección MAC correspondiente.

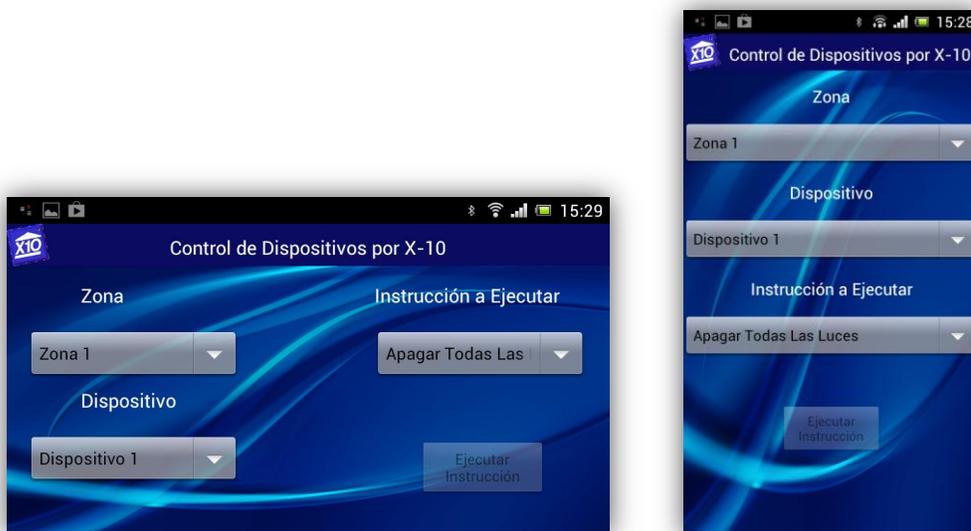


Fig. 4.15. Pantalla principal de la aplicación sin conexión al bluetooth del módulo emisor

Cuando se establezca conexión con el módulo emisor, el botón de 'Ejecutar instrucción' se habilitará, y el software estara listo para mandar la instrucción deseada a cualquiera de los 256 dispositivos que el protocolo soporta.

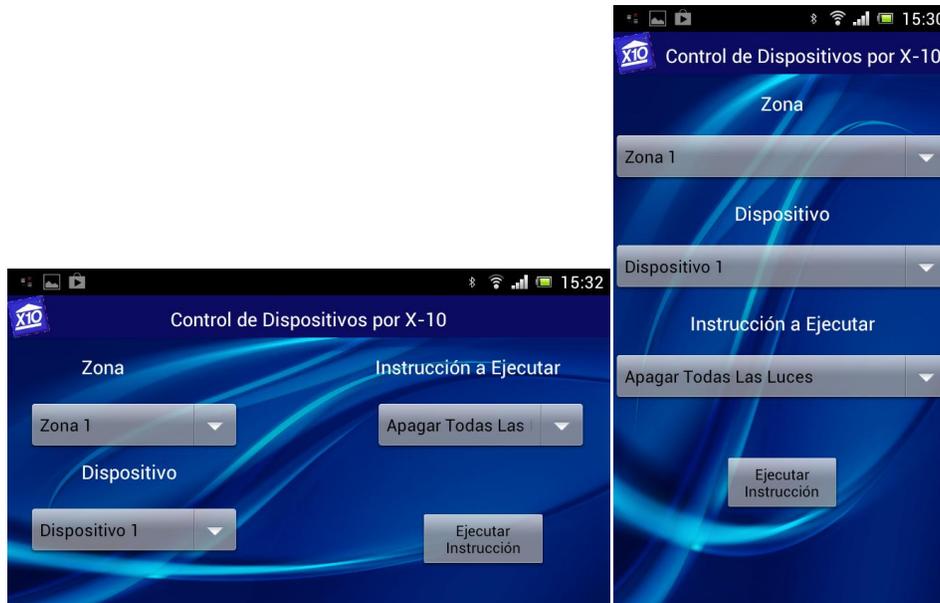


Fig. 4.16. Pantalla principal de la aplicación con conexión al bluetooth del módulo emisor

Al igual que la aplicación para la computadora, esta aplicación también soporta las 16 direcciones de 'Casa' y los 16 dispositivos para cada una de las anteriores (256 dispositivos), además de los comandos más comunes, tales como:

- Apagar todas las luces
- Encender dispositivo
- Atenuación/Velocidad preestablecida
- Encender todos los dispositivos
- Aumentar intensidad/velocidad
- Apagar dispositivo
- Apagar todos los dispositivos
- Disminuir intensidad/velocidad

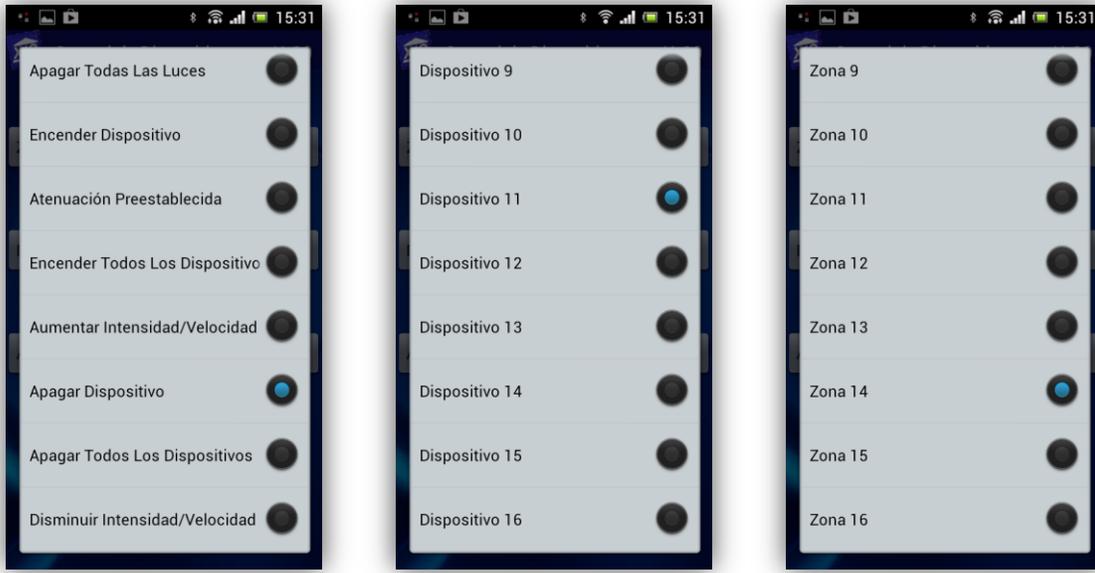


Fig. 4.17. Pantallas de selección para comando, dispositivo y zona.

Cuando el usuario haya elegido la zona, dispositivo, comando deseado y además haya dado clic al botón 'Ejecutar Instrucción', en el módulo emisor se encenderá un led.

De manera similar a lo hecho para la aplicación de la computadora, se decidió colocar un menú en donde se visualicé la ayuda y la información de quien creó la aplicación. Se puede acceder a este menú apretando el botón físico de menú del dispositivo, ahí se desplegará la opción 'Acerca'.

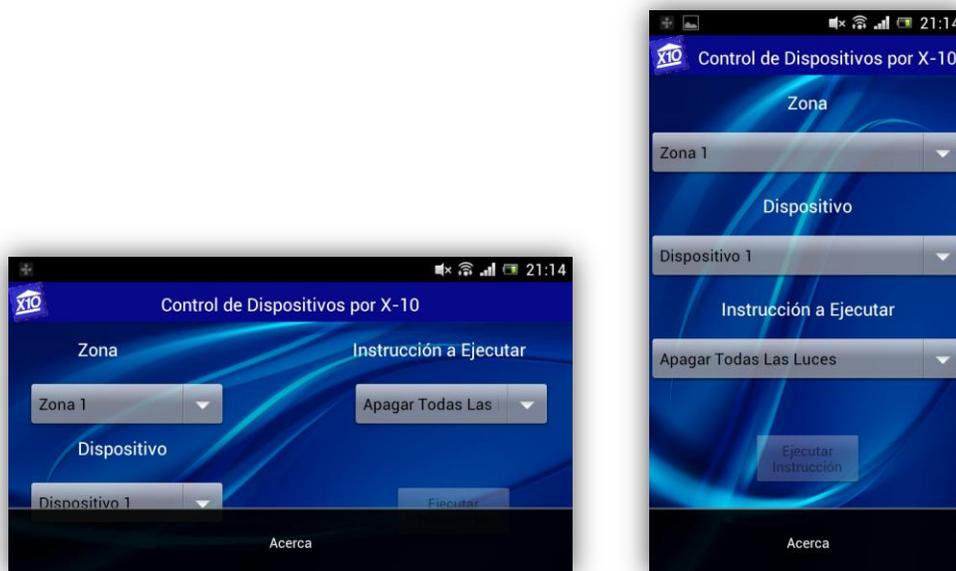


Fig. 4.18. Imagen del menú de la aplicación

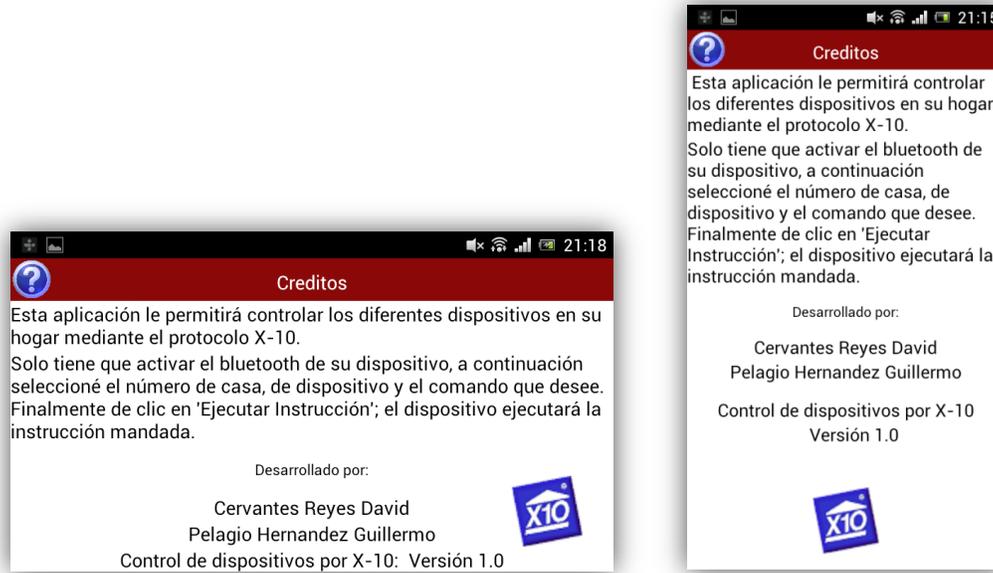


Fig. 4.19. Imagen de la ayuda y los créditos de la aplicación

En esta pantalla, se puede consultar un pequeño manual de cómo usar el programa, además de ver la información sobre quien ha desarrollado el mismo.

4.2.4. Transferencia de datos entre la aplicación y el módulo emisor

La manera en la que se realiza la transferencia de información entre la aplicación móvil y el módulo emisor es la siguiente:

Aunque en este caso la comunicación entre el dispositivo móvil y el módulo emisor se da por bluetooth, el envío de datos es muy similar a lo que se hace por RS-232, primero se debe tener en una cadena todos los valores a mandar, a continuación se convierte dicha cadena con los datos de 'casa', 'comando' y 'dispositivo', en un array, el cual es enviado al bluetooth, quien leerá la información y la transmitirá al microcontrolador como comunicación RS-232.

```
//MENSAJE QUE SE MANDA AL MÓDULO EMISOR
```

```
CADENA=H+D+C+"#";
```

```
byte[] array = CADENA.getBytes();
```

```
write (byte[] array)
```

'H', 'D' y 'C', se refieren a los valores que les fue asignado cuando se seleccionó una opción en las listas del programa. A estas variables les son asignadas caracteres que van desde la 'A' hasta la 'P' y que se refieren a los 16 diferentes valores de 'casa', de dispositivo y de comando. Se utiliza al igual que en el software para la computadora, un identificador de fin de cadena (el carácter '#'), con el objetivo de saber cuándo detener el envío de datos.

Capítulo 5. Pruebas y resultados

Para la realización de pruebas, se implementaron en protoboard los circuitos del emisor y del receptor; se hizo uso del equipo de laboratorio existente en la Facultad y se procedió a verificar el correcto funcionamiento del sistema. Una vez que el sistema mostro un correcto funcionamiento, se prosiguió a elaborar circuitos impresos.

5.1. Pruebas del módulo emisor

Lo primero que se hizo fue verificar la señal de entrada de 127 [V_{AC}] y 60[Hz] en el osciloscopio, dado que no se cuenta con puntas de osciloscopio que atenúen el voltaje, se decidió elaborar un divisor de voltaje con la señal de entrada; el diagrama es el siguiente:

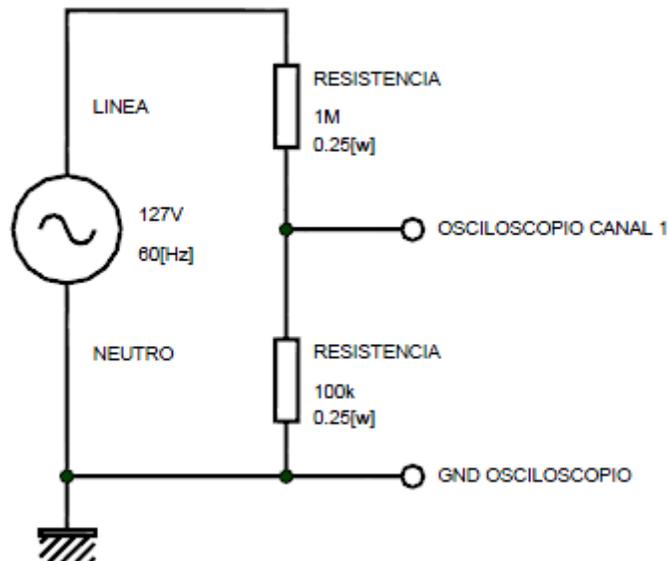


Fig. 5.1. Divisor de voltaje implementado

El voltaje a la salida de este divisor fue calculado con la siguiente ecuación:

$$V_{\text{divisor}} = \frac{127 \text{ [V]} * R_2}{R_1 + R_2} = \frac{127 \text{ [V]} * 100 \text{ [k}\Omega\text{]}}{100 \text{ [k}\Omega\text{]} + 1 \text{ [M}\Omega\text{]}} = 11.545 \text{ [V]}$$

A continuación se muestra el resultado del divisor en el laboratorio :

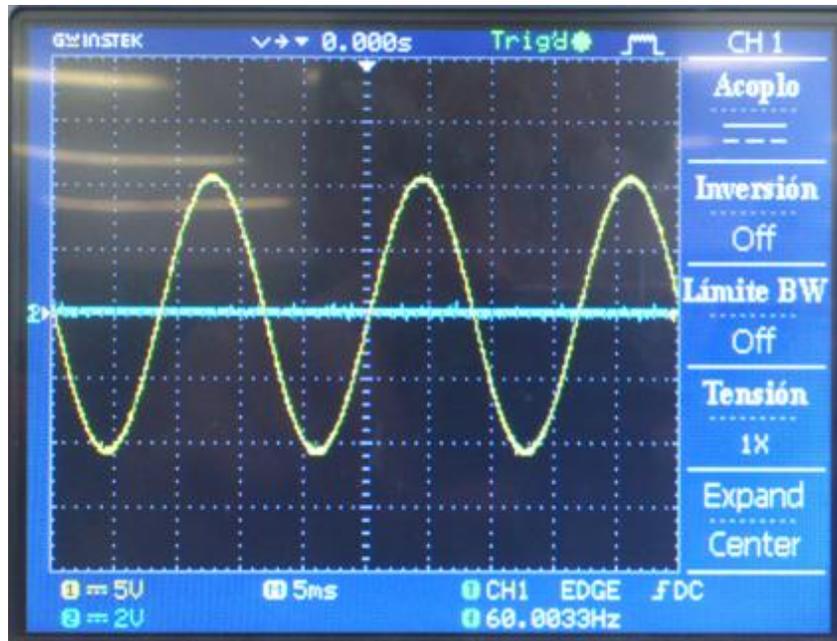


Fig. 5.2. Resultado del divisor de voltaje implementado

A continuación se procedió a medir el voltaje en la entrada del microcontrolador que corresponde a los cruces por cero; además de verificar si el retraso entre cada cambio de estado estaba dentro del rango de lo que se había calculado anteriormente.

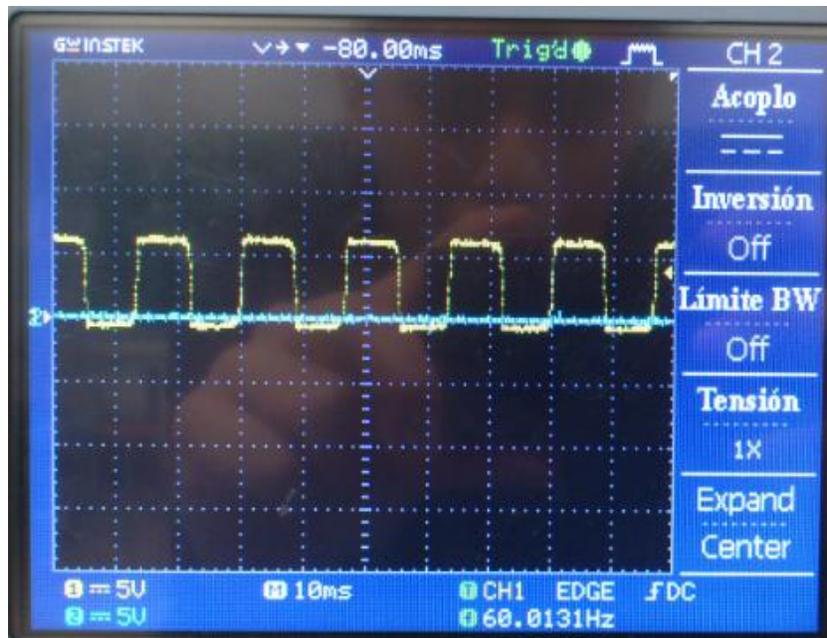


Fig. 5.3. Resultado de los cruces por cero a la entrada del microcontrolador



Fig. 5.4. Retraso real del microcontrolador para pasar de un estado lógico a otro.

Como se ve en la figura anterior el retraso que se observa es de aproximadamente $30[\mu\text{s}]$ contra los $48.633 [\mu\text{s}]$ calculados en el capítulo 3.

Ya que se observó que los cruces por cero se daban de manera correcta y estaban dentro del rango de tiempo, lo que se procedió a hacer fue comunicar en primera instancia el módulo emisor con el software de la computadora mediante el cable USB, después se hizo mediante el propio bluetooth del emisor y finalmente se realizó la comunicación del emisor con el dispositivo móvil.

5.1.1. Comunicación mediante cable USB

Se procedió a conectar el cable USB a la computadora y al módulo emisor, al ser la primera vez que se conectaba nos apareció un mensaje en la que Windows había encontrado un nuevo software y estaba en busca del controlador de dispositivo adecuado.

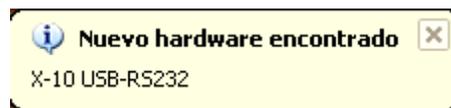


Fig. 5.5. Mensaje de nuevo hardware encontrado por parte de la computadora

Ya que el controlador necesario no se encontró en la computadora, apareció un mensaje en donde se da la opción de instalar el controlador adecuado.

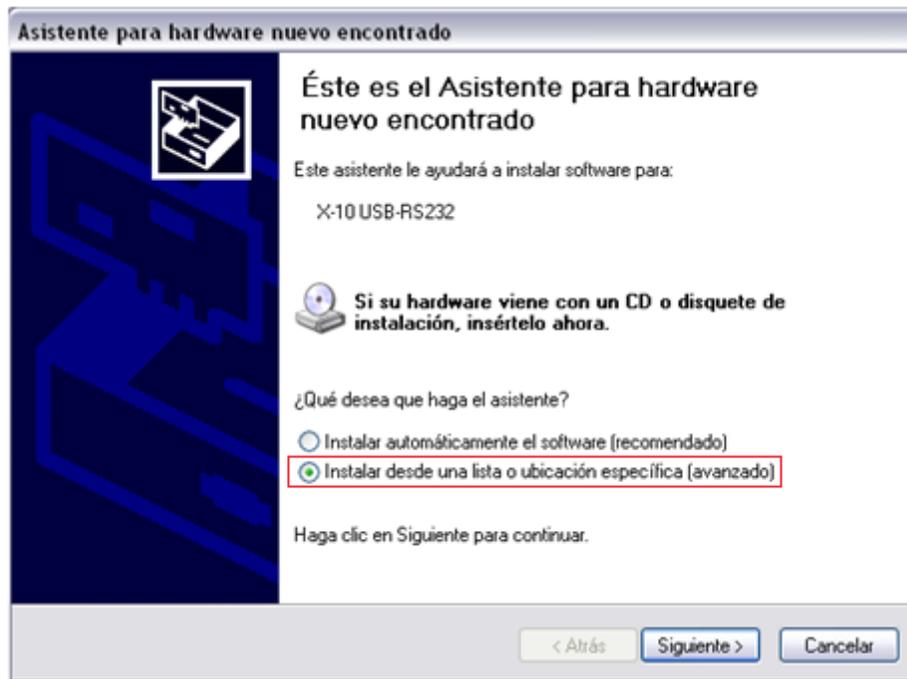


Fig. 5.6. Mensaje de instalación del controlador adecuado

A este mensaje se le dio la opción de 'instalar desde una ubicación específica', y se seleccionó la ubicación en donde se encontraba el controlador que habíamos editado previamente.

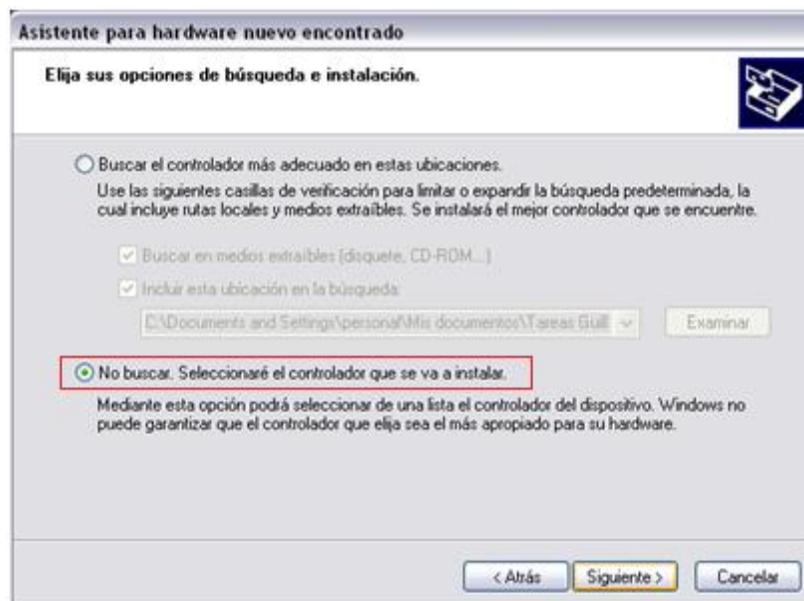


Fig. 5.7. Ventana asistente para la instalación del controlador

A continuación, apareció un mensaje en el que se pedía la confirmación para instalar el controlador, ya que este no superó la compatibilidad con la versión de Windows. Se continuó con la instalación ya que este no causa ningún mal funcionamiento en la computadora.



Fig. 5.8. Notificación de seguridad de Windows

El controlador instaló correctamente y lo siguiente fue verificar en la ventana del administrador de dispositivos que así fuera, lo que implicaba que se pudiera reconocer el dispositivo adecuadamente. Éste apareció ahora en el listado de puertos COM y LTP como era de esperarse.

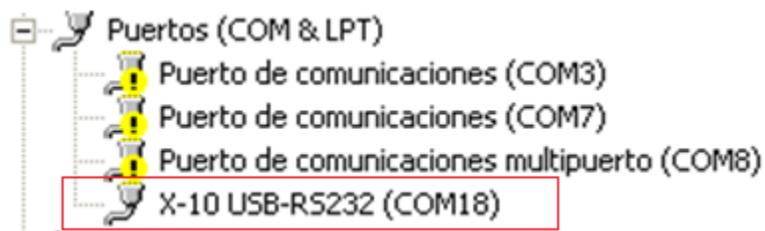


Fig. 5.9. Listado de puertos COM y LPT en donde ya se encontraba instalado el controlador

Para desplegar la información adicional, referente a nuestro dispositivo, como el proveedor del controlador, la versión y el nombre del dispositivo, sólo se tuvo que dar propiedades a la hora de seleccionarlo en el menú del administrador de dispositivos.



Fig. 5.10. Propiedades del controlador instalado

A continuación se procedió a abrir el software, seleccionando el puerto que se le había asignado a la conexión USB, cuando este entró se seleccionó un número de casa, de dispositivo y un comando y se observó que a la salida del circuito emisor se proporcionara la trama correcta, con la frecuencia deseada y un largo de pulso de 1 [ms]; a continuación imágenes de lo que se midió en el circuito emisor:

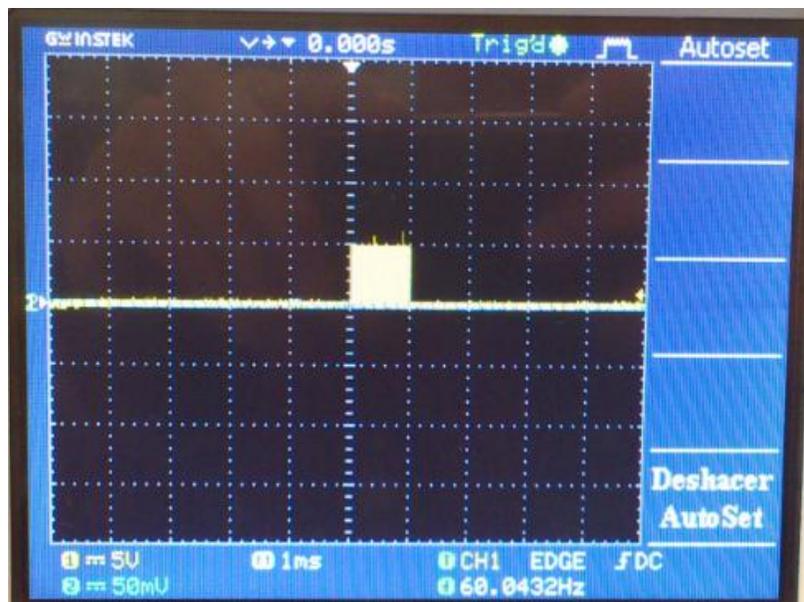


Fig. 5.11. Frecuencia de 120 [kHz] durante 1 [ms]

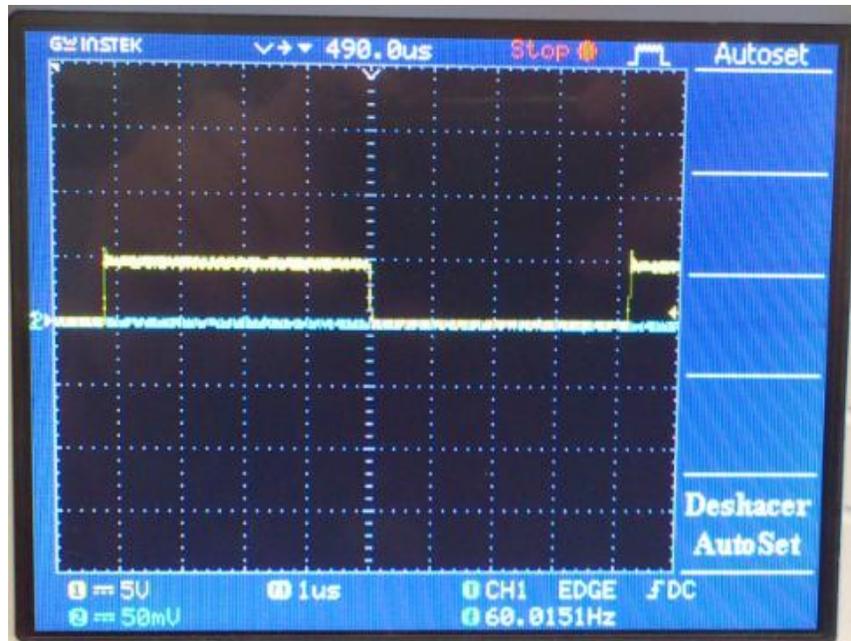


Fig. 5.12. Señal de 120 [kHz] dentro del pulso

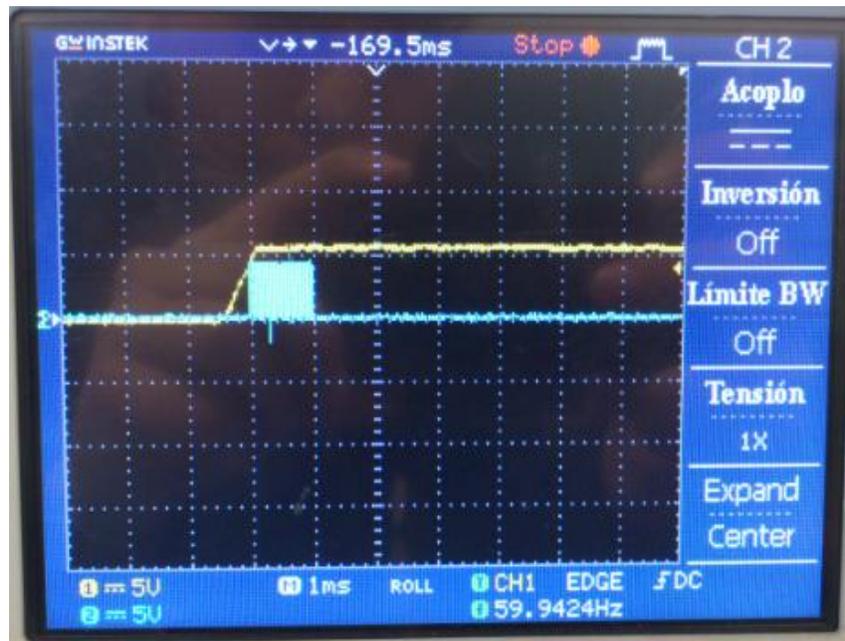


Fig. 5.13. Señal de 120 [kHz] durante 1 [ms] sincronizado en el cruce por cero

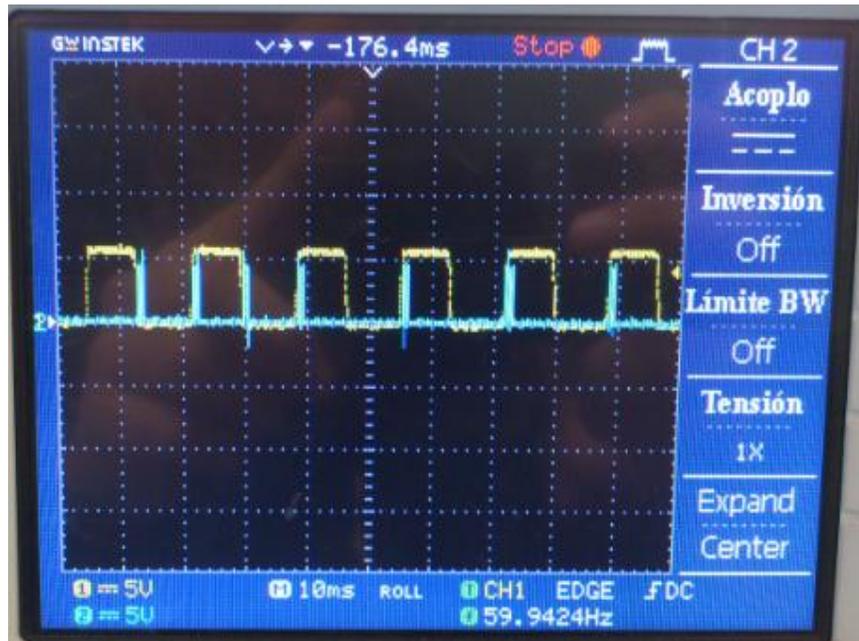


Fig. 5.14. Trama de bits

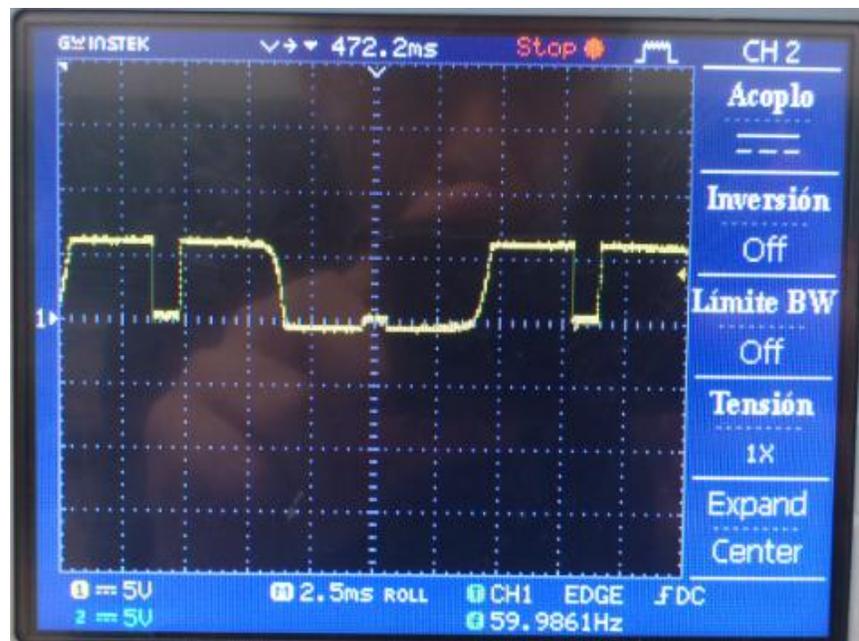


Fig. 5.15. Señal a la salida del circuito emisor

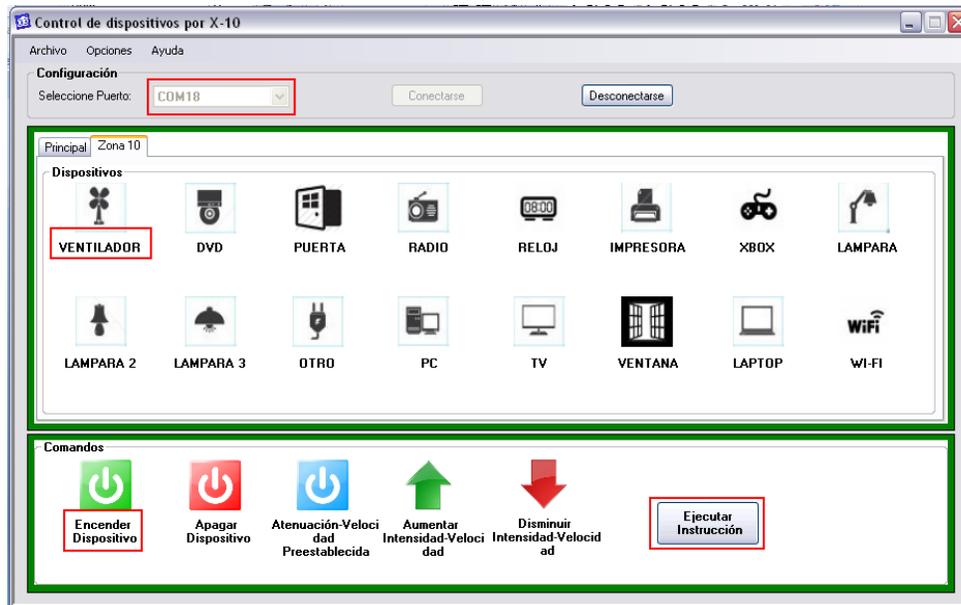


Fig. 5.16. Puerto conectado y comando mandado

5.1.2. Comunicación mediante Bluetooth

Después de verificar que todo funcionaba perfectamente mediante USB, se procedió a realizar la conexión con el software mediante bluetooth, para esto se debió desconectar en el software la conexión USB y después fue necesario conectarse ahora seleccionando el puerto que se la había asignado al módulo bluetooth. Previamente a esto se debió haber identificado de manera correcta el módulo bluetooth en la computadora, además de haberlo emparejado con el código '1234', como se muestra en la siguientes figuras.

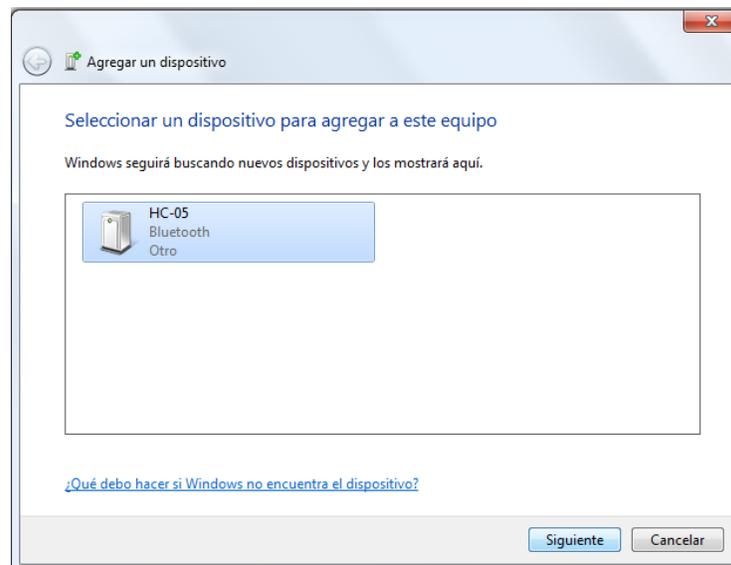


Fig. 5.17. Dispositivo bluetooth identificado en la computadora

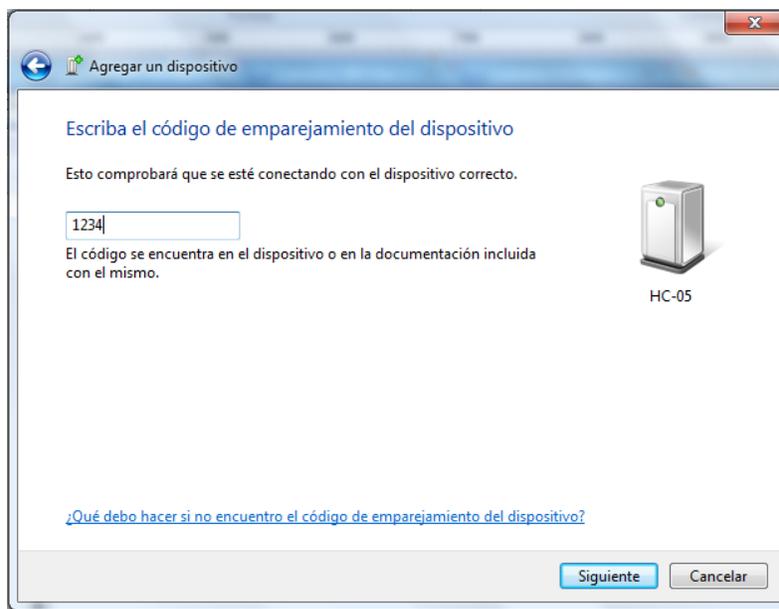


Fig. 5.18. Emparejando el dispositivo con la computadora

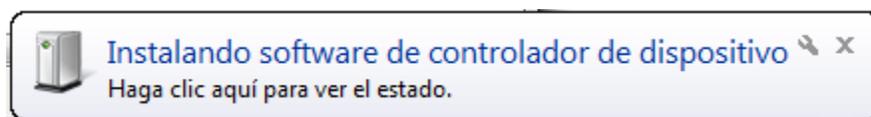


Fig. 5.19. Mensaje de instalación del software

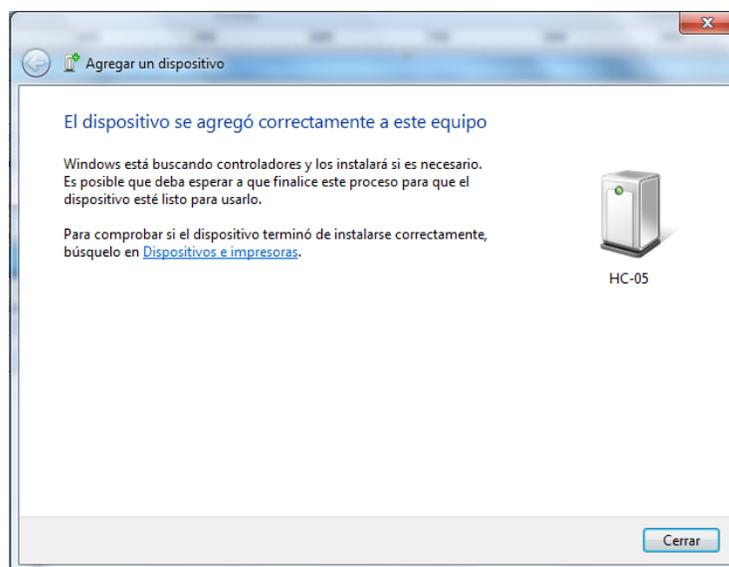


Fig. 5.20. Dispositivo correctamente agregado

Se verificó que puerto COM se le había asignado al bluetooth y se procedió a realizar la conexión con el software.

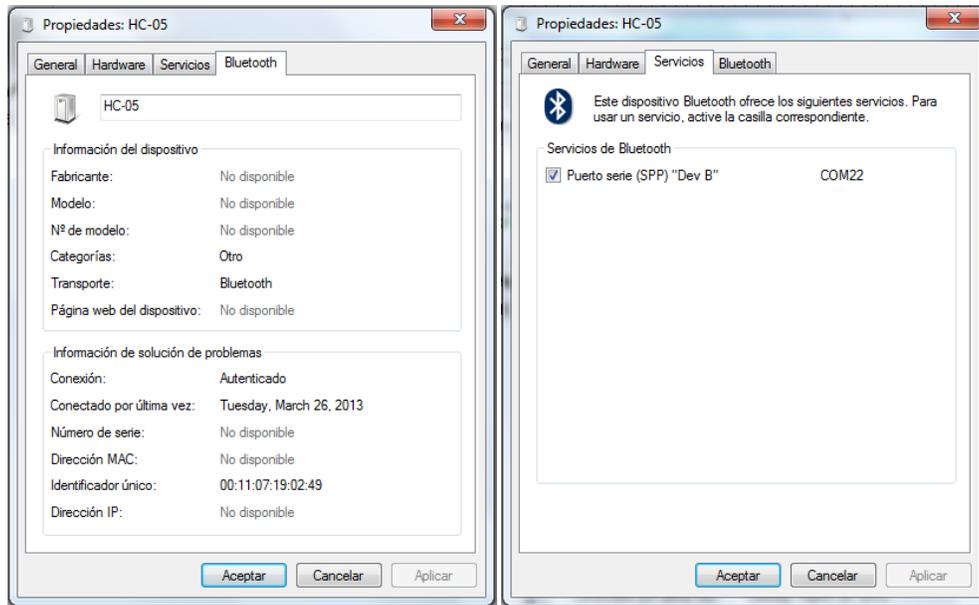


Fig. 5.21. Propiedades del bluetooth usado

Ya aquí, se procedió a seleccionar un número de casa, de dispositivo y un comando y se observó que de la misma manera que con la conexión USB, la salida del circuito emisor proporcionara la trama correcta, con la frecuencia deseada y un largo de pulso de 1[ms]. Los resultados del módulo emisor fueron similares a los mostrados en la parte de conexión mediante el USB.

A continuación algunas imágenes del circuito del módulo emisor en protoboard y en circuito impreso.

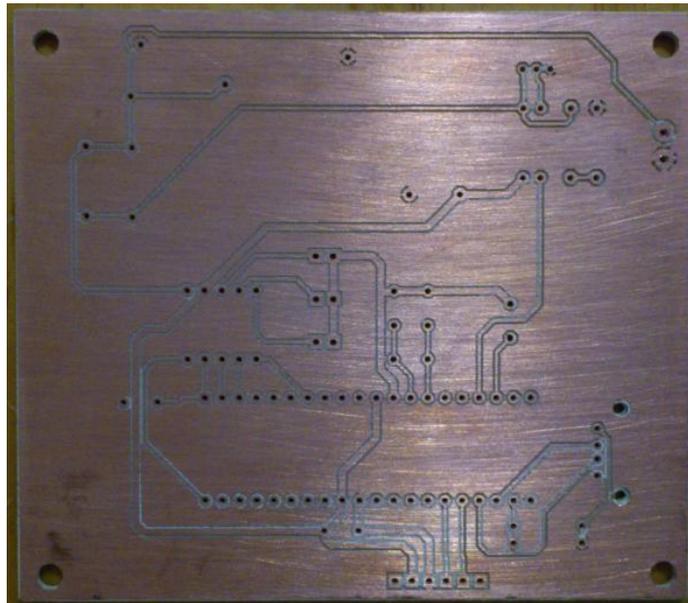


Fig. 5.22. Circuito impreso del módulo emisor

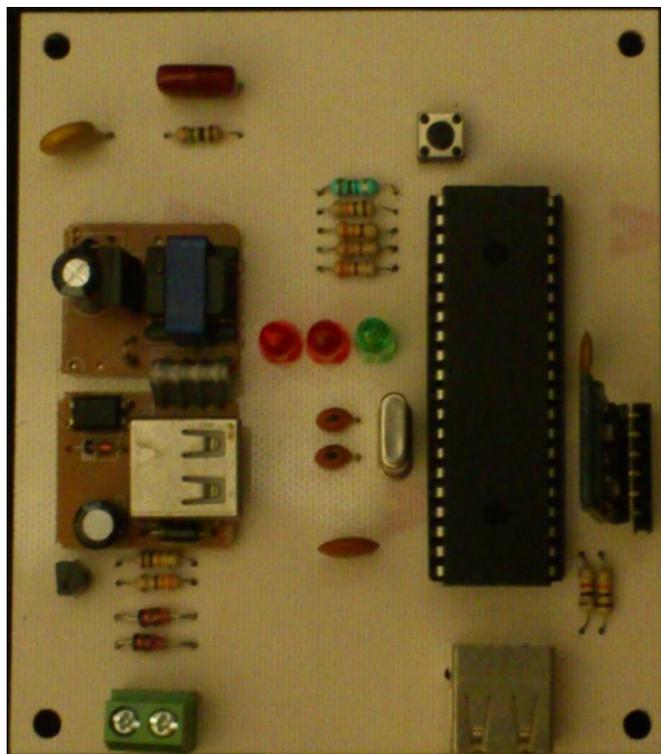


Fig. 5.23. Módulo emisor en circuito impreso

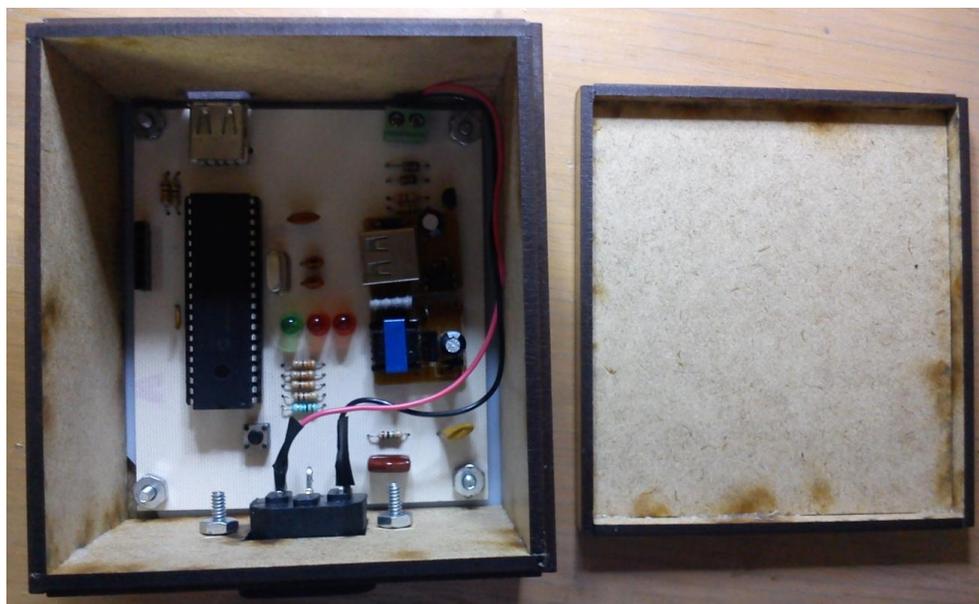


Fig. 5.24. Módulo emisor terminado

5.2. Pruebas del módulo receptor

Ya que se tuvo la certeza de que se estaba mandando correctamente la trama dentro del rango de frecuencia y tiempos, se decidió conectar el módulo receptor y verificar que se recibiera la información que se había mandado por la red eléctrica.

Para esto, se procedió a probar cada una de las etapas de las que se compone el módulo receptor, aquí los resultados:

La primer parte dentro del módulo emisor se refiere al filtrado de la señal de 120 [kHz]. En el diseño que se propuso en el capítulo anterior se colocaron 2 filtros pasivos pasa altas con el fin de solamente dejar pasar la frecuencia de 120 [kHz]. Al momento de conectar y medir en el primer filtro, se notó que el voltaje que dejaba pasar estaba en el rango de los milivolts, dado que la frecuencia de 60[Hz] es demasiado pequeña para no ser eliminada totalmente por el filtro, las señales existentes en milivolts se refieren al ruido existente en la línea debido a los diferentes dispositivos eléctricos conectados a la misma.

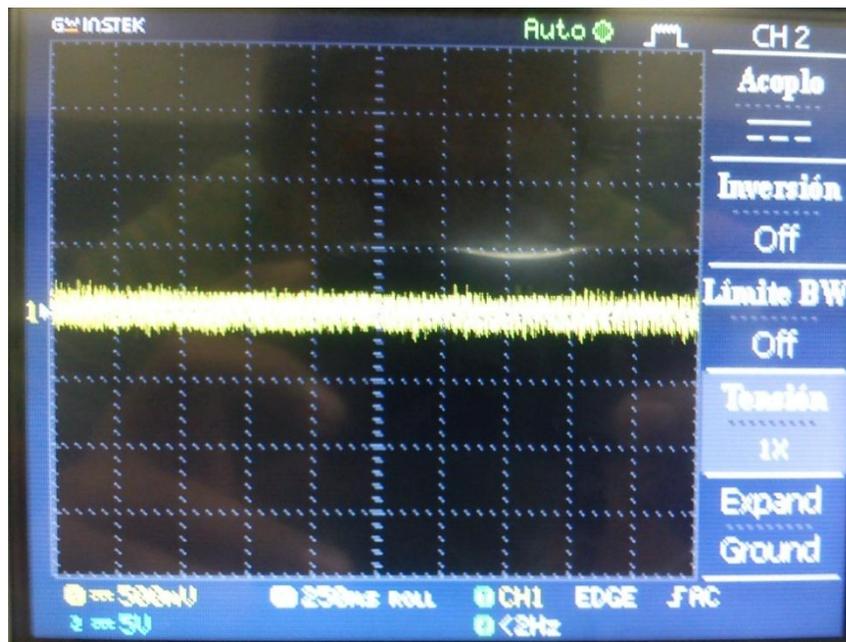


Fig. 5.25. Salida del primer filtro sin transmitir datos

A continuación, se mandó información desde el módulo emisor para ver si el primer filtro dejaba pasar la frecuencia de 120 [kHz]. Como se ve en la siguiente imagen, varios de los bits mandados superan el nivel de ruido original aunque los demás parecen confundirse con el nivel de ruido existente

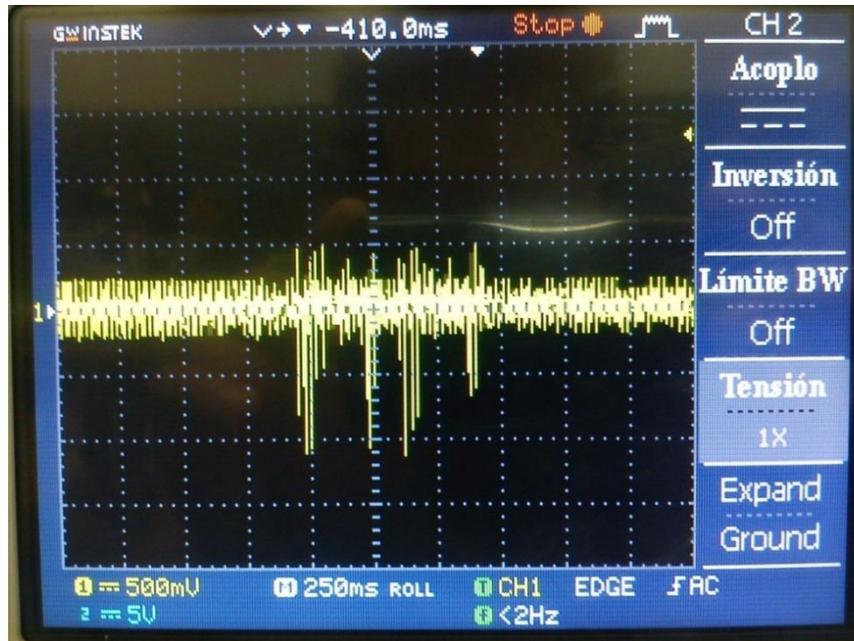


Fig. 5.26. Señal de 120 [kHz] presente a la salida del primer filtro

A continuación se procedió a medir a la salida del segundo filtro, como se esperaba, el nivel de ruido de la señal disminuyó hasta los 100 [mV], por lo que en seguida se mandó la trama para ver la respuesta de este filtro. Las siguientes imágenes del osciloscopio muestran lo visto en el laboratorio.

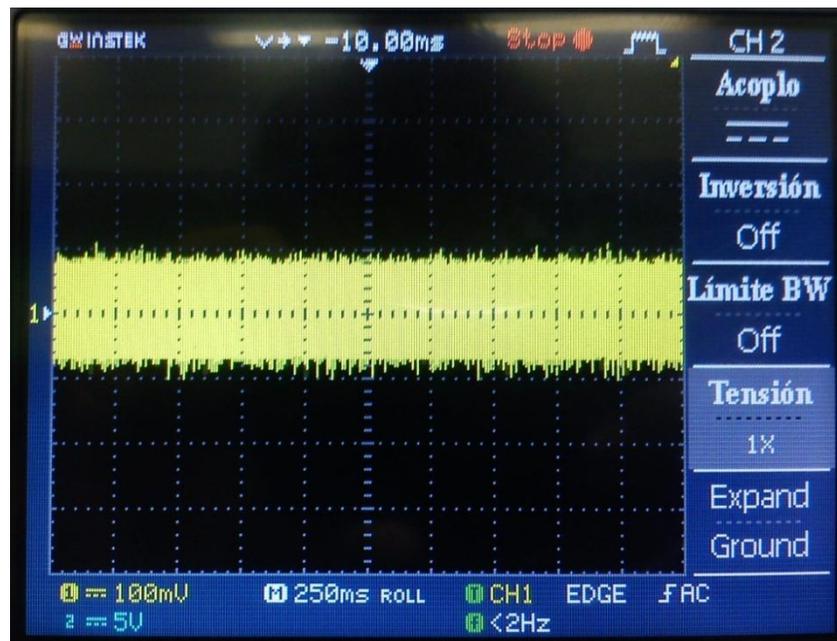


Fig. 5.27. Señal del segundo filtro sin transmitir datos

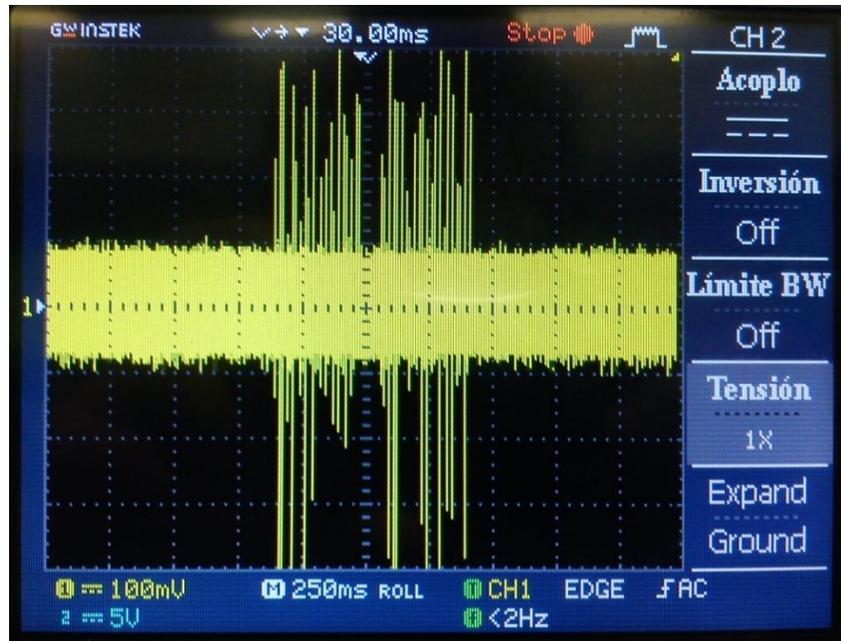


Fig. 5.28. Señal de 120 [kHz] a la salida del segundo filtro

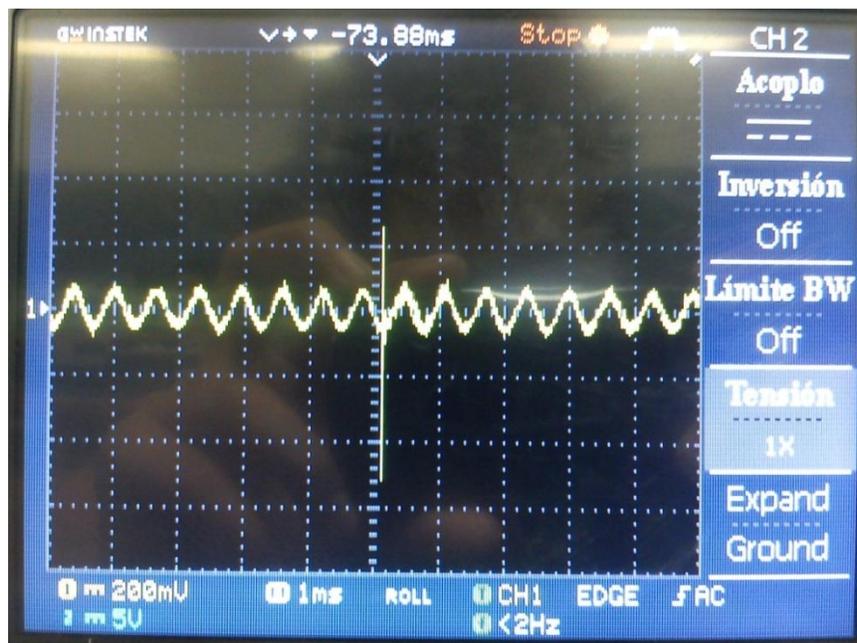


Fig. 5.29. Un bit de la trama transmitida a la salida del segundo filtro

En los resultados se puede ver que mientras el nivel de ruido se mantiene en el rango de los 100 [mV], la mayoría de los bits mandados superan los 200 [mV] llegando algunos tener hasta 0.5 [V].

Ya que se probaron los 2 filtros a la entrada del circuito, se pensó de inicio que no sería necesario aumentar el voltaje para que las señales se pudieran amplificar de manera adecuada en el amplificador sintonizado-desintonizado; pero después de algunas pruebas se notó que este amplificador solo funciona de manera correcta cuando en su entrada hay aproximadamente 0.5 [V], de lo contrario este no amplifica la señal.



Fig. 5.30. Salida del amplificador sin datos transmitidos

Debido a lo anterior, se procedió a amplificar la señal proveniente del segundo filtro para poder trabajar mejor con esta en la siguiente etapa, por lo que se armó el amplificador que se muestra en seguida.

De las pruebas realizadas en el laboratorio, se observó que el voltaje de la señal a la entrada del amplificador sintonizado-desintonizado debe ser por lo menos de 0.5 [V], y teniendo que la mayoría de nuestras señales a la salida de los filtros pasivos tienen un valor de aproximadamente 150 [mV], consideramos que una ganancia de 3 es suficiente.

La configuración que se empleará para amplificar el voltaje, será la de amplificador en tipo no inversor, en esta configuración el voltaje de salida está dado por:

$$V_{\text{salida}} = \left(\frac{R_1}{R_2} + 1 \right) (V_{\text{entrada}})$$

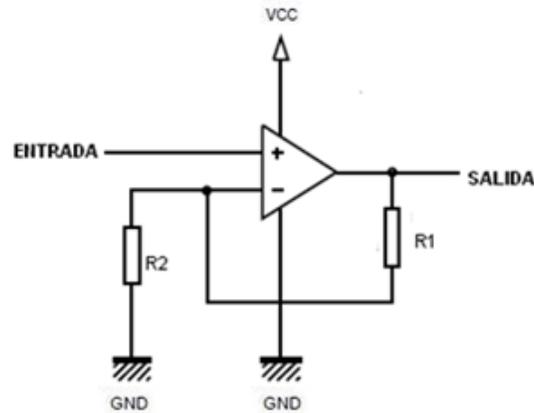


Fig. 5.31. Amplificador no inversor

Para obtener la ganancia deseada, se debe cumplir que:

$$\left(\frac{R_1}{R_2} + 1\right) = 3$$

Por lo que:

$$\left(\frac{R_1}{R_2}\right) = 2$$

$$R_1 = 2R_2$$

El amplificador que se decidió usar fue el TL082 ya que presenta una gran respuesta en frecuencia, un gran ancho de banda y una alta impedancia de entrada, ideal para estas circunstancias en donde no se están manejando señales de grandes amplitudes.

Se procedió a armar el amplificador mostrado y como se ve en la imagen 5.32, después de la amplificación con ganancia de 3, los bits superan sin problemas los 500 [mV] que se necesitan en la siguiente etapa. Si bien el nivel de ruido se amplificó a casi 200 [mV], los bits nunca se llegan a confundir con el ruido de la línea y se distinguen sin problemas.

Cabe mencionar que se probó colocando un amplificador con una ganancia mayor, pero a partir de una ganancia de 3 la salida del amplificador empezaba a amplificar demasiado el ruido y los bits se empezaban a confundir con éste, por lo que no se creyó conveniente aumentar más la ganancia.

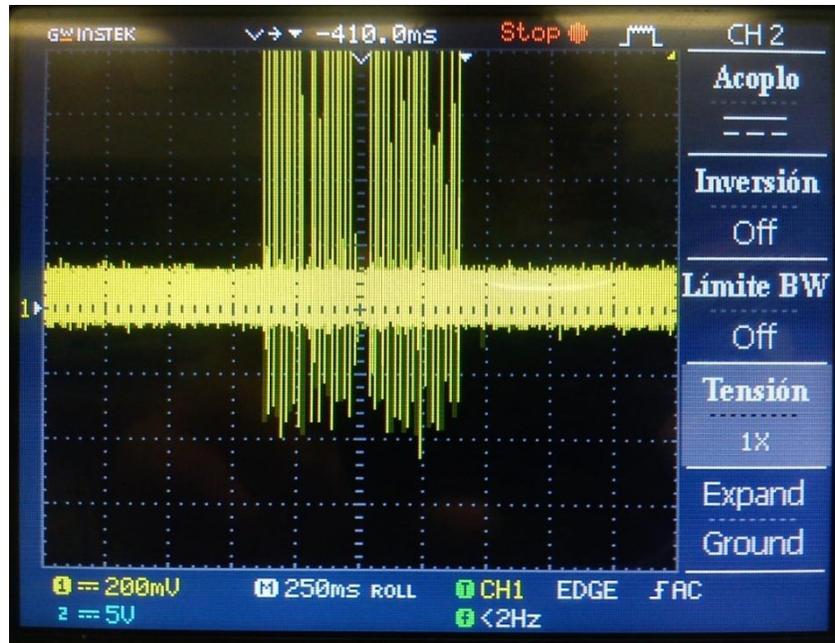


Fig. 5.32. Trama amplificada

Después de lo anterior, se procedió a usar el amplificador sintonizado-desintonizado que como ya se explicó en el capítulo 3, sirve para amplificar señales pero solo dentro de un rango de frecuencia; los resultados se muestran en las siguientes imágenes.

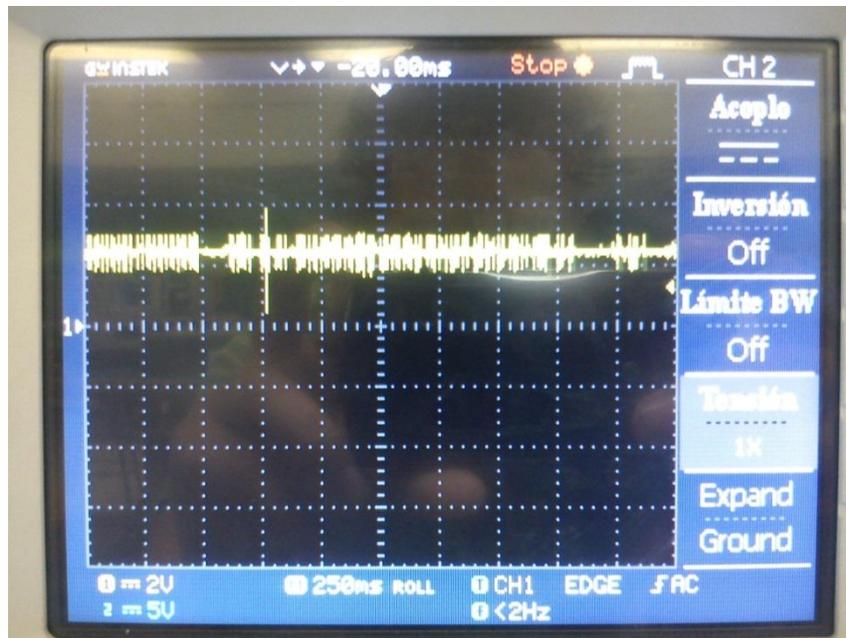


Fig. 5.33. Señal resultante de pasar por el amplificador sintonizado y no sintonizado sin transmitir información

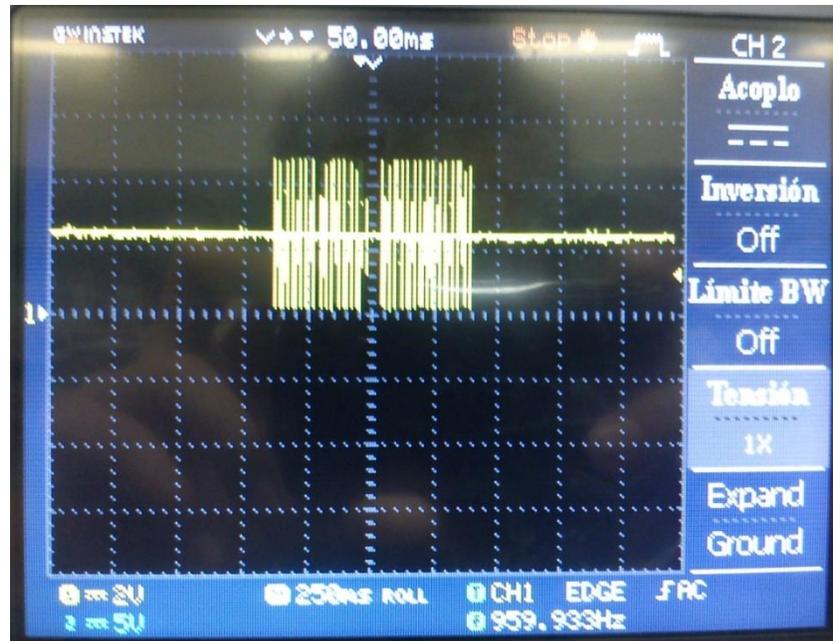


Fig. 5.34. Señal resultante a la salida del amplificador sintonizado y no sintonizado con datos transmitidos

Como se ve en la imagen 5.34, a la salida de este bloque amplificador solo se encuentran los bits mandados con una amplitud de 5 [V], sin ningún tipo de ruido, también se observa que la señal se encuentra con un offset de aproximadamente 3.3 [V].

Dado que la señal de offset cuando no se está mandando ningún tipo de información, se encuentra muy cercano a los 3.3 [V], siendo que en algunas veces este valor se ve superado, no se puede meter esta señal directamente a un microcontrolador, porque éste estaría reconociendo valores lógicos altos a cada rato, sin llegar a ser los bits correctos, por lo que se procedió a eliminar este offset.

La manera más rápida que se pensó para eliminar este offset, es utilizar amplificadores operacionales en su configuración de comparador. Se estableció la referencia del offset en aproximadamente 3.2 [V], así toda señal por arriba de este punto, será reconocida mientras que las que no lo estén, no serán tomadas en cuenta. Inicialmente este comparador se alimentó con 0 [V] y 5 [V], pero debido a que a la salida de éste se seguía presentando un offset de aproximadamente 1 [V] y la amplitud de los bits no superaba 1 [V] se decidió alimentar a este comparador con una alimentación simétrica de 5 [V].

Dicho comparador también se decidió que fuera inversor ya que la señal resultante a la salida del amplificador sintonizado-desintonizado se notó que todos los bits cruzaban hacia abajo del offset pero no todos estos cruzaban hacia arriba de la referencia. En la siguiente imagen se muestra dicho comparador, a este se le colocó un Trimpot para darle la posibilidad de variar la referencia, ya que esta cambia dependiendo del nivel de ruido del lugar.

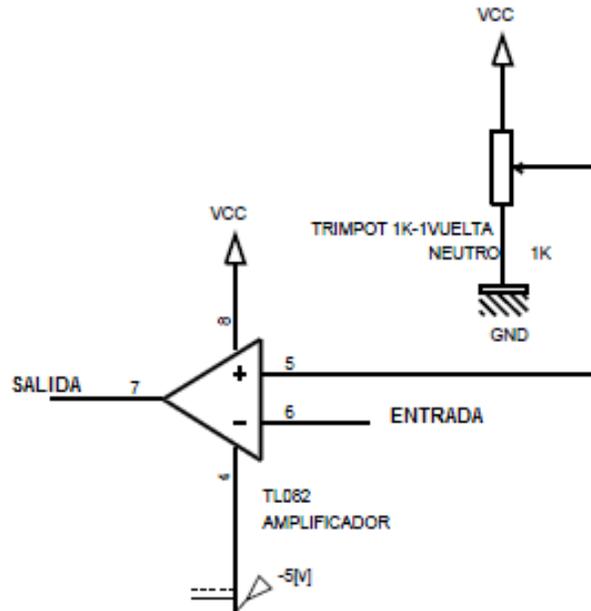


Fig. 5.35. Amplificador en modo comparador con alimentación simétrica

Debido a la alimentación simétrica del comparador antes mencionado, la señal de respuesta ahora tiene su referencia en un voltaje negativo; sin embargo las señales transmitidas de 120 [kHz] rebasan desde dicha referencia a los 0 [V], hecho que nos ayuda para implementar un nuevo comparador con referencia en 0 [V] (Fig. 5.36.), con el objetivo de obtener señales que ahora estén desde los 0 [V].

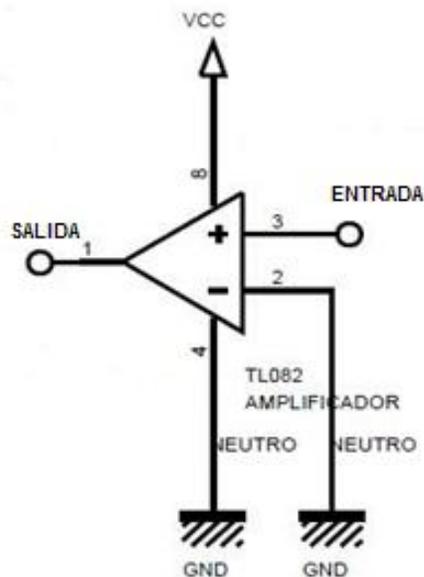


Fig. 5.36. Segundo amplificador en modo comparador

Los resultados de esto último se muestran en las siguientes imágenes:

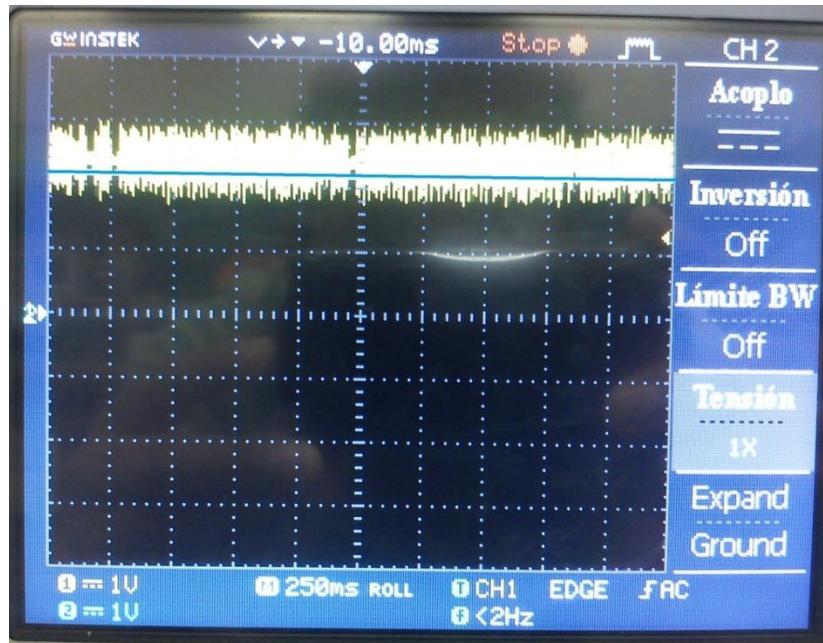


Fig. 5.37. Señal de ruido y la referencia del potenciómetro



Fig. 5.38. Salida del primer comparador sin transmitir datos

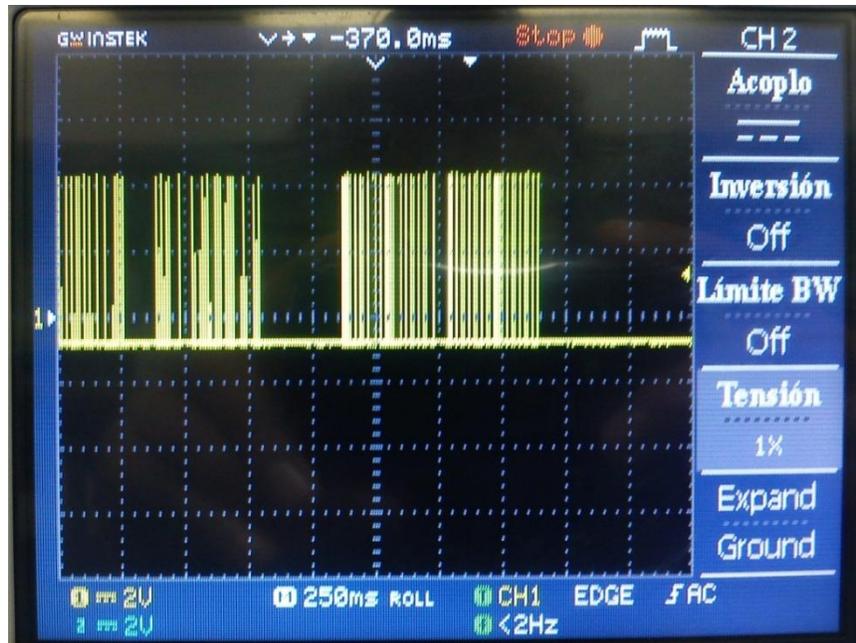


Fig. 5.39. Salida del primer comparador transmitiendo datos

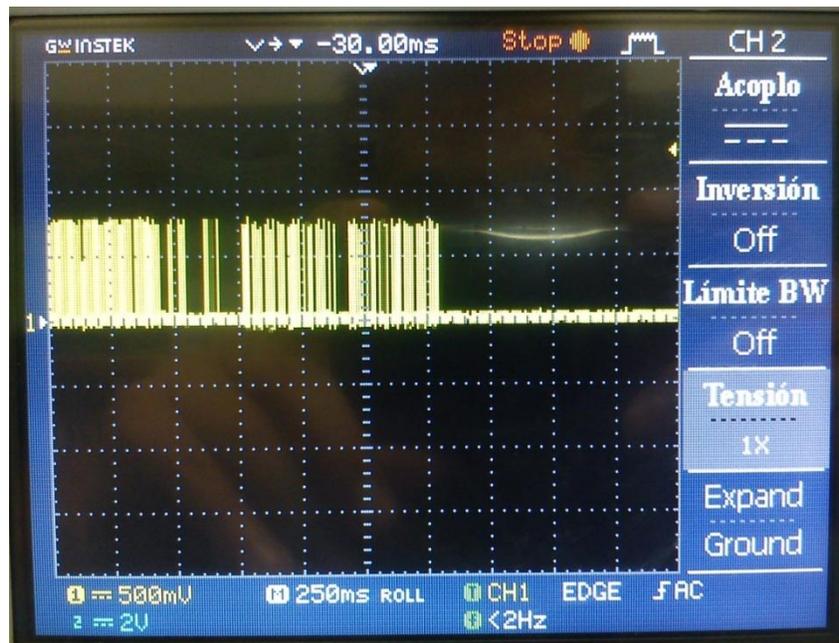


Fig. 5.40. Salida del segundo comparador transmitiendo datos

Como se puede observar de las últimas 3 imágenes, los resultados usando los comparadores son satisfactorios, ya que se eliminó el ruido completamente y al final de estos 2 amplificadores la señal va de 0 [V] a 5 [V], sin tener la pérdida de algún bit.

A continuación se dispuso a conectar el transistor para asegurar que todas las señales de frecuencia se encuentren correctamente en los niveles lógicos. Como ya se mencionó anteriormente, hasta ahora el trabajo se ha realizado con señales de 120 [kHz] y con duración de 1[ms]; pero se necesitan pulsos para poder facilitar la lectura de estas señales en el microcontrolador.

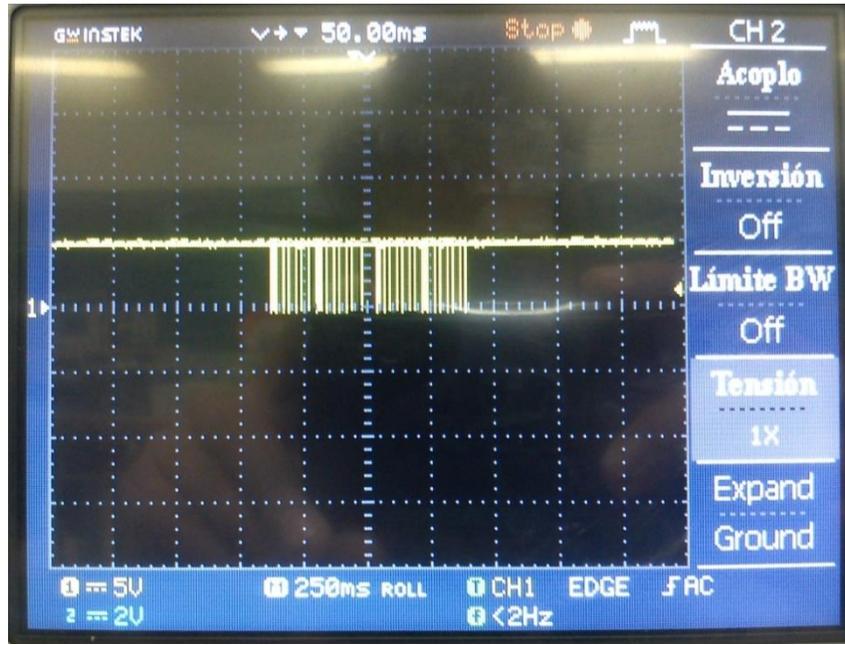


Fig. 5.41. Trama a la salida del transistor

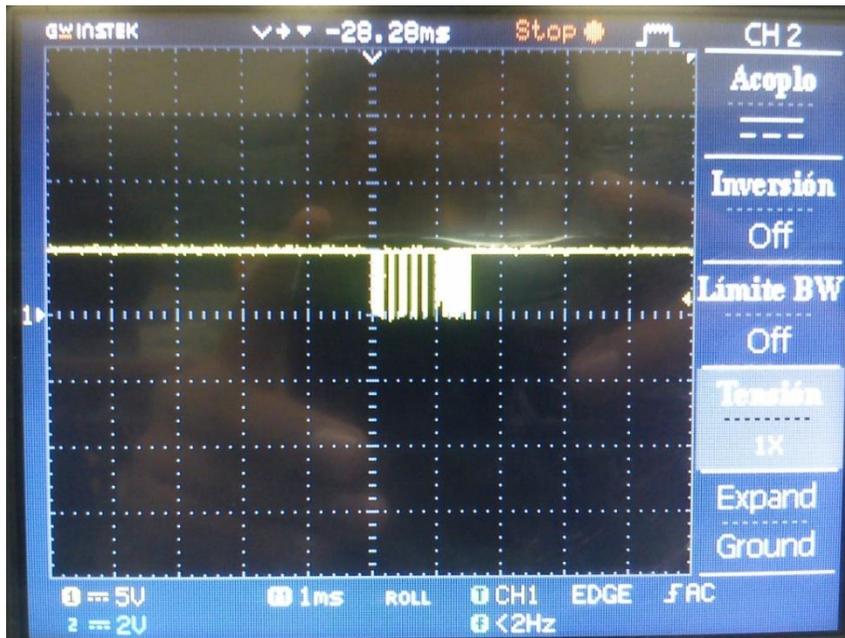


Fig. 5.42. Bit de 1 [ms] y 120 [kHz], a la salida del transistor

Posteriormente se armó el circuito detector de envoltura que se propuso en el capítulo 3, el cual proporciona a la salida del circuito pulsos en vez de señales con frecuencia. En la figura 5.43 se muestra el resultado a la salida de este circuito.

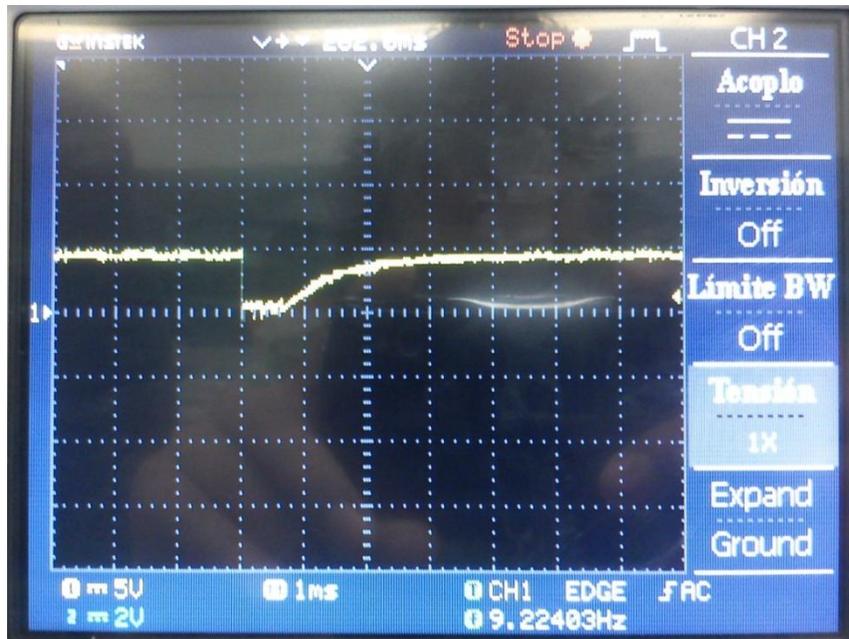


Fig. 5.43. Bit de la trama, a la salida del detector de envoltura

Dado que la señal de salida está invertida y no es exactamente una onda cuadrada se conectó a uno de los pines de la compuerta inversora existente.

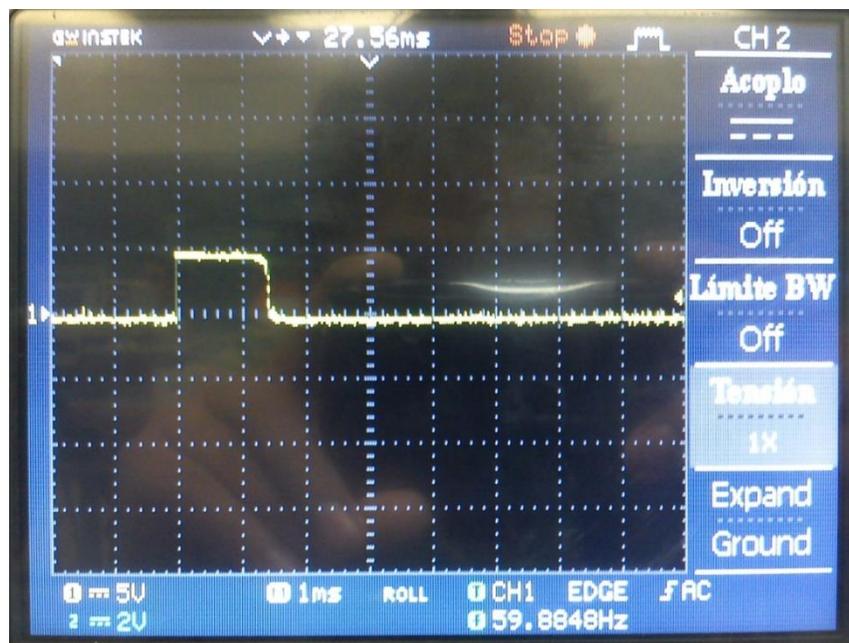


Fig. 5.44. Bit de la trama, a la salida de la compuerta inversora

Así, es como finalmente se obtuvieron los pulsos de 1[ms] que posteriormente serán procesados en el microcontrolador. En la siguiente imagen se muestra parte de la trama recibida en los cruces por cero en el circuito receptor después de todo lo anteriormente descrito.



Fig. 5.45. Bits recibidos en el circuito emisor.

A continuación, algunas imágenes del circuito del módulo receptor en protoboard y en circuito impreso.

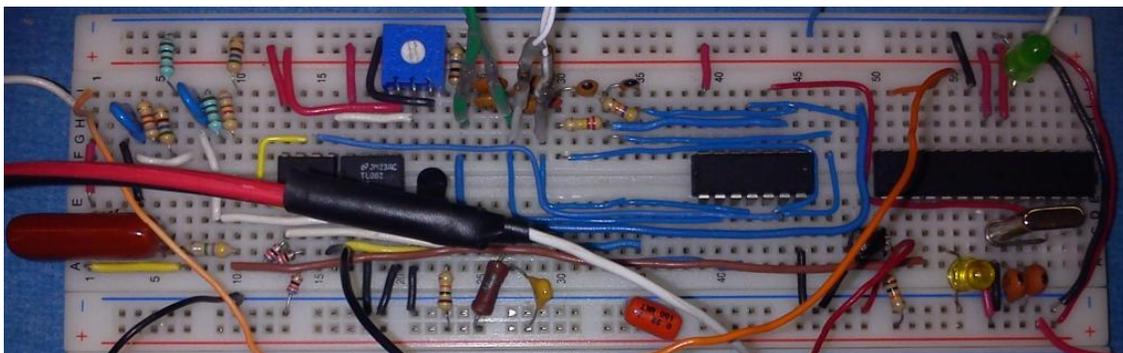


Fig. 5.46. Módulo receptor en protoboard

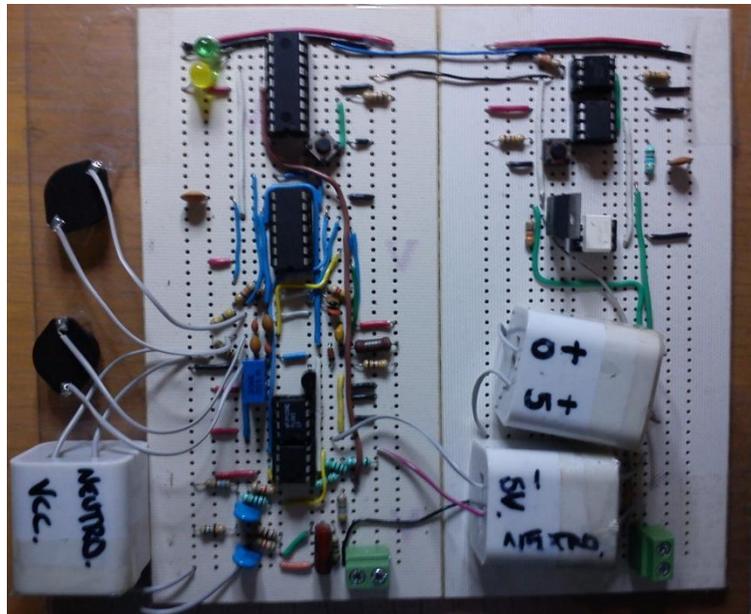


Fig. 5.47. Módulo receptor en placa perforada

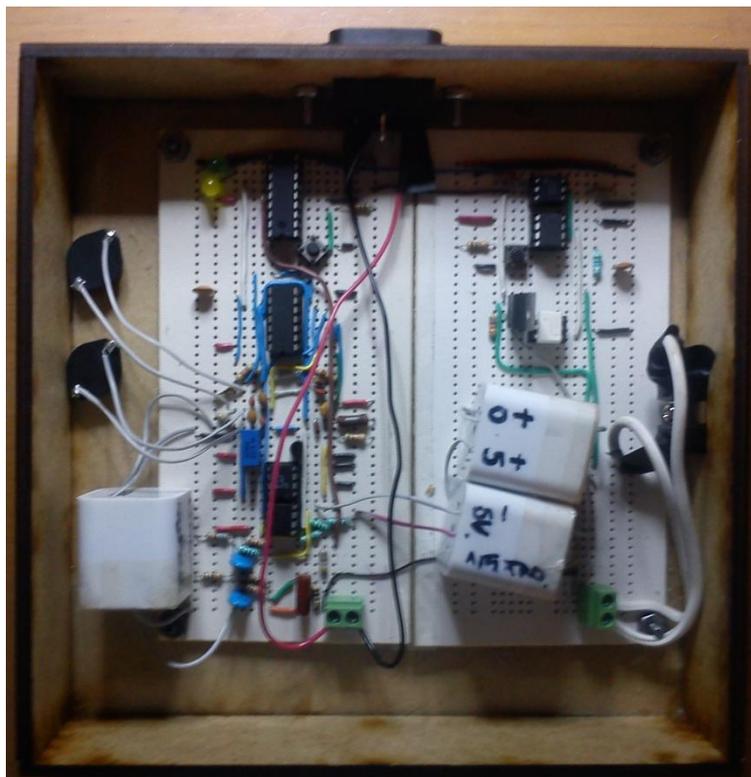


Fig. 5.47. Módulo receptor terminado

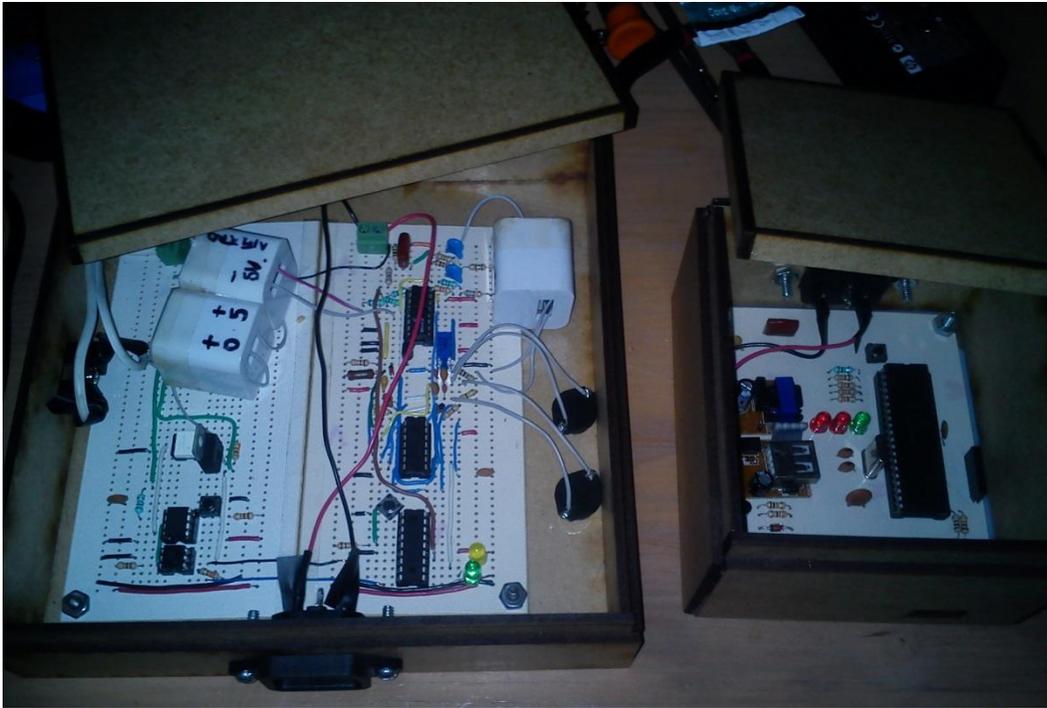


Fig. 5.48. Módulos emisor y receptor

Capítulo 6: 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.

6.1. Conclusiones

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.

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 fabricante. De esta manera se comparten las instrucciones básicas que permiten la completa funcionalidad del protocolo y a la vez se permite tener comandos más avanzados, si la aplicación lo requiere.

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 parte de decodificación de información, deseche la trama y se pierdan datos o comandos.

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 pudieran llegar a interferir en otros sistemas de domótica basados en la transmisión de información mediante cable dedicado o mediante dispositivos inalámbricos.

Es difícil configurar una red X-10 para tareas complicadas ya que la comunicación se realiza a través de un bus compartido. 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.

El protocolo X-10 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 o no.

El uso del protocolo X-10, tal como está planteado, sólo es recomendable para la domótica en hogares y en oficinas donde el número de tareas de control que se requiere realizar es limitado. Es decir, cuando no se requiere enviar más de una trama X-10 por segundo como máximo.

La utilización de la red eléctrica como medio de comunicación supone un ahorro en cuanto a la implementación práctica, debido a que no es necesario realizar otro tipo de inversión adicional en cuanto a la instalación, lo cual hace más atractivo la adquisición de este tipo de sistema.

Como conclusión final del trabajo realizado, se obtiene que el sistema implementado funciona adecuadamente, por lo que el diseño y el desarrollo basado en la tecnología de corrientes portadoras es una técnica que puede ser utilizada para implementar un sistema domótico el cual permita la automatización de vivienda.

6.2. Recomendaciones

Se plantean las siguientes pautas a tomar en cuenta como medida de protección.

Si se planea medir la señal de la red eléctrica se recomienda utilizar puntas de osciloscopio atenuadas por diez, como medida de precaución para no dañarlas, o realizar un divisor de voltaje.

Cuando se trabaje con la señal de la red eléctrica, al momento de realizar las pruebas, es recomendable aislar la tierra del osciloscopio con la del resto del circuito, porque se puede producir un cortocircuito entre la línea viva y la tierra digital.

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.

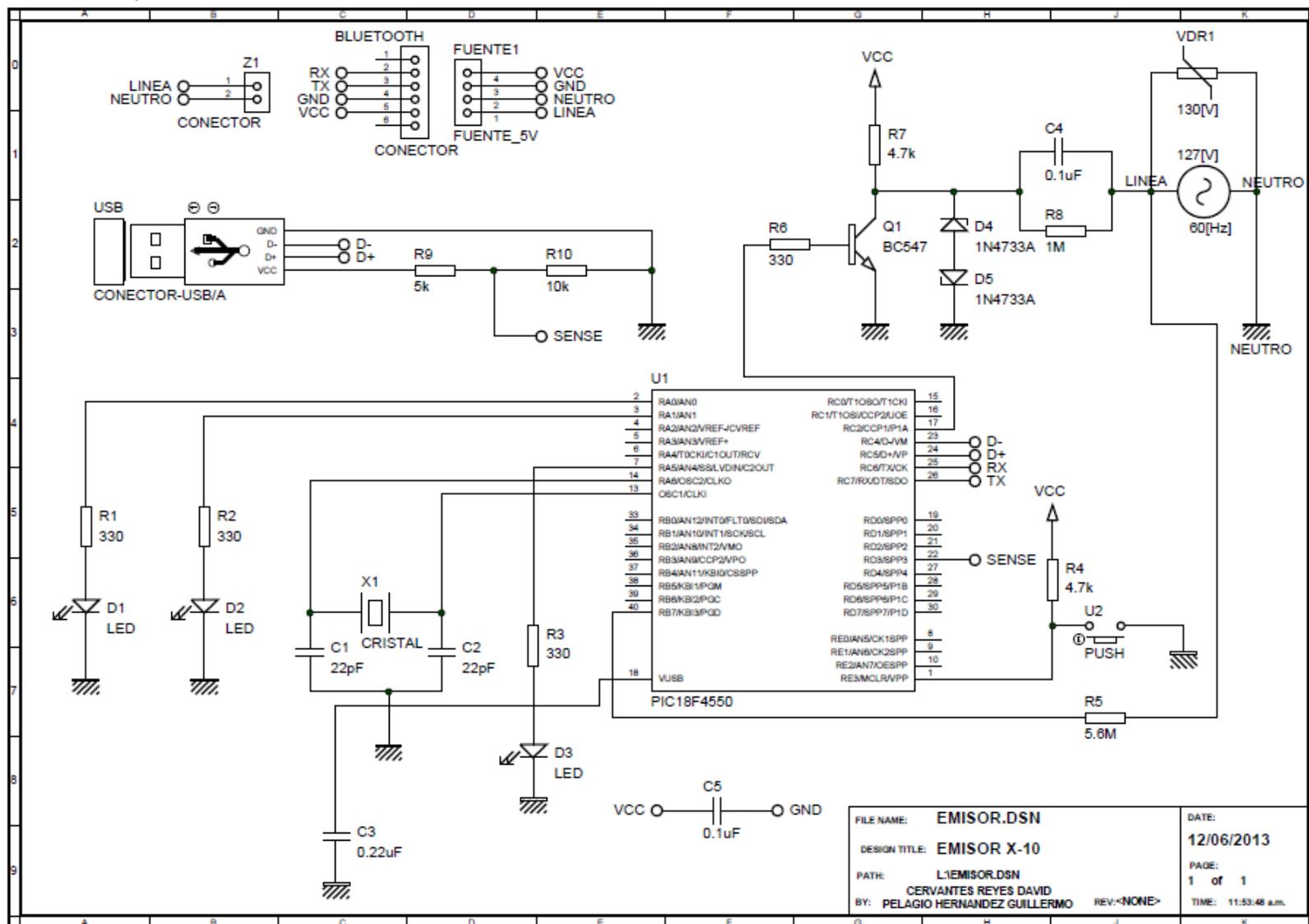
Si el hogar cuenta con alimentación bifásica o trifásica, es conveniente instalar puentes de señal para poder controlar dispositivos conectados a una fase distinta a la cual está conectada el módulo de control.

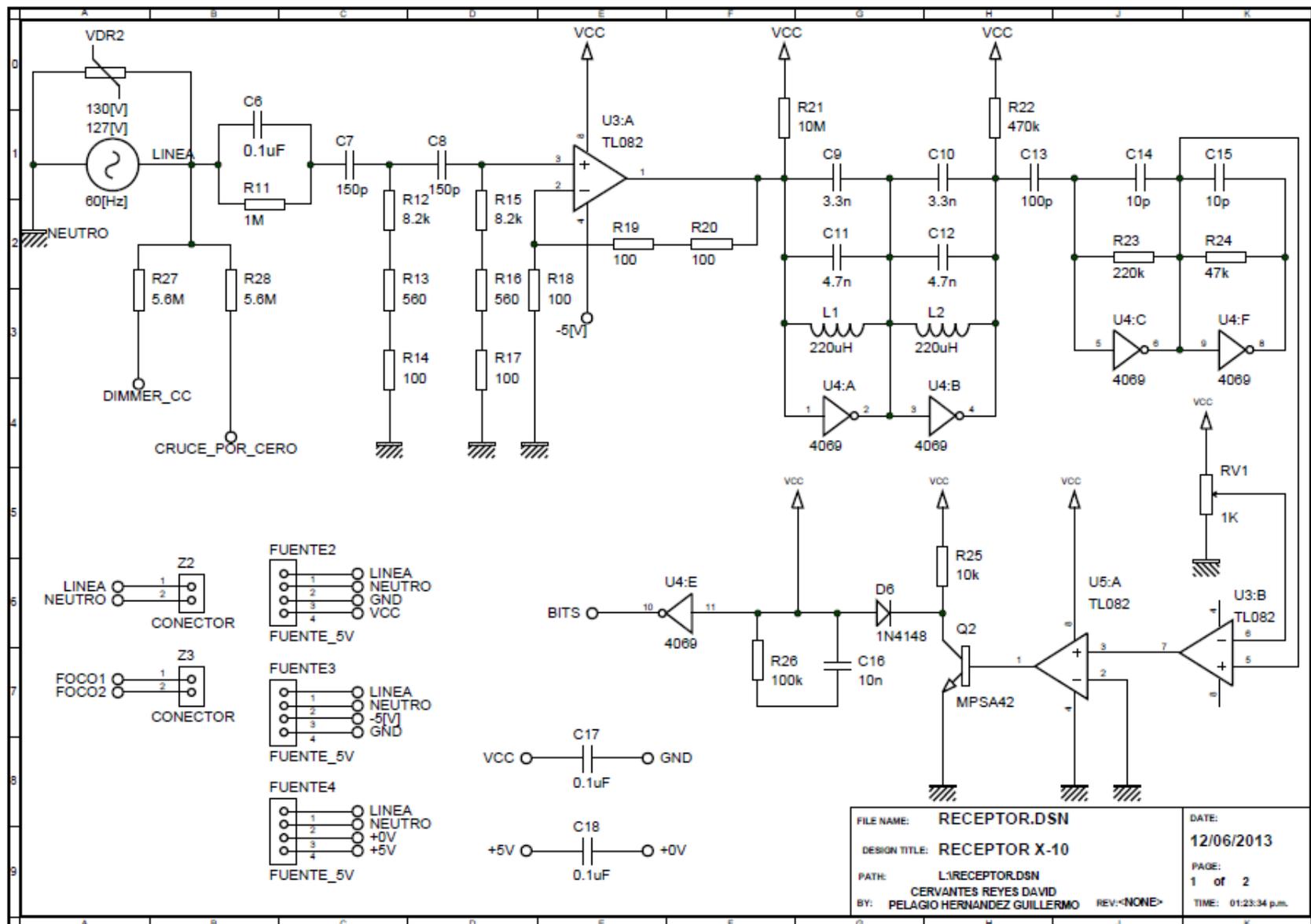
Capítulo 7. Trabajo a futuro

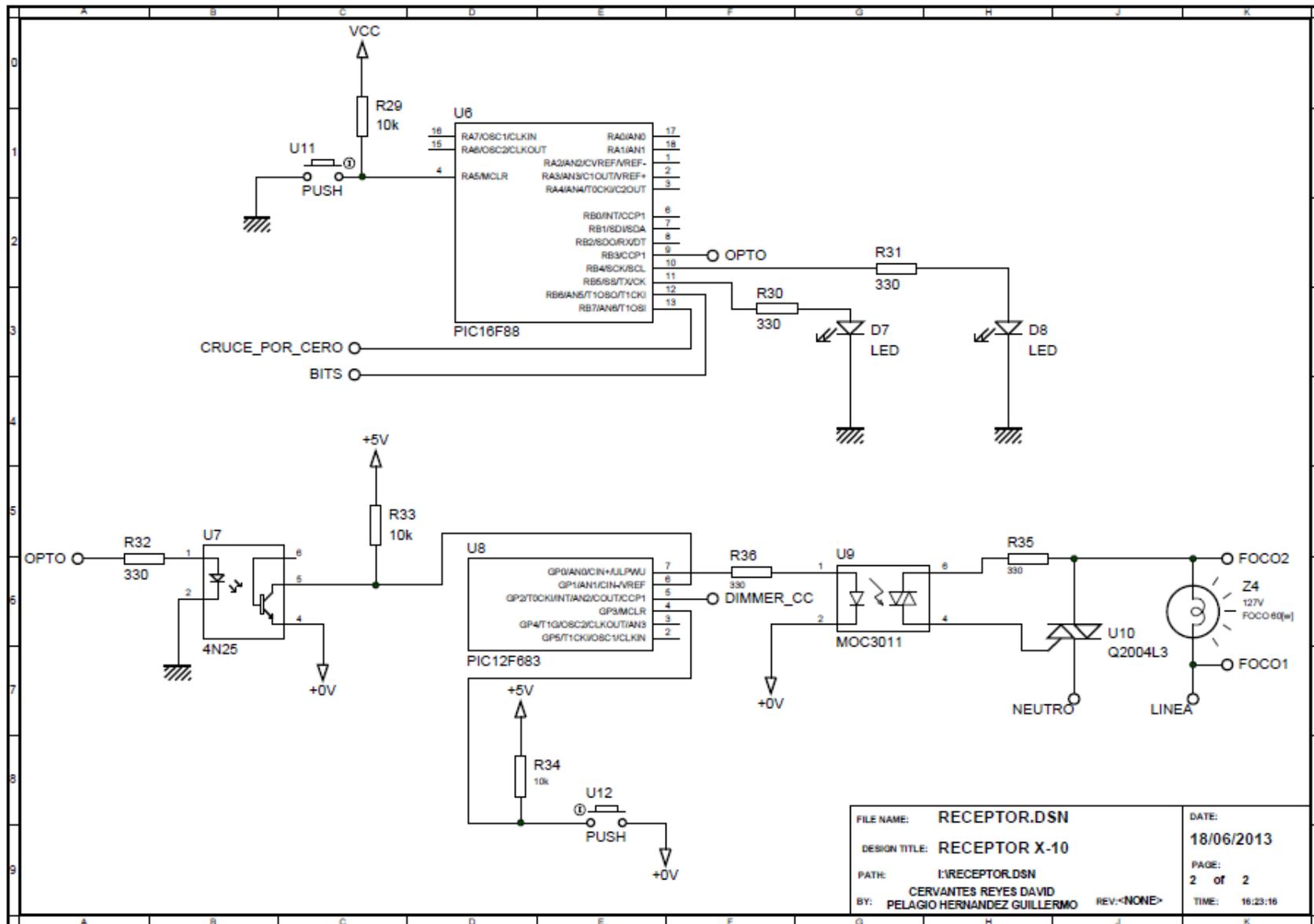
Una vez analizado el resultado obtenido, nos percatamos que el proyecto puede llegar a mejorar en varios aspectos por lo que proponemos los siguientes puntos para que en un futuro se puedan desarrollar:

- La creación de un sistema bidireccional mediante la unión de ambos dispositivos, emisor y receptor, en un solo aparato. De esta manera se podría dotar al proyecto de una mayor robustez e incluso se podrían agregar sistemas de control.
- También desarrollar la bidireccionalidad entre el módulo emisor, la computadora y dispositivo móvil, así se podría monitorizar el estado de toda la casa sin tener que moverse del o la computadora, de esta manera se podría desarrollar la posibilidad de que el emisor se mantenga enviando las órdenes que sean necesarias para realizar la operación dictaminada por el operador hasta que el receptor le responda dándole la confirmación.
- La posibilidad que el usuario pueda asignar según su criterio las direcciones de cada módulo y que éstas se actualicen de modo automático en el software (PC-dispositivo móvil), evitando así, que la asignación de direcciones sea solamente mediante la programación de los receptores.
- Otro aspecto a seguir desarrollando son las interfaces (computadora y dispositivo móvil) en donde se podría enfocar en hacerlas más dinámicas y gráficas, en las que sea posible agregar dispositivos cuando el usuario lo desee, evitando así, que la interfaz tenga dispositivos predeterminados que podrían no usarse.
- Ya que en caso de presentarse una emergencia, es indispensable que se cuente con acciones casi inmediatas para otorgar más posibilidades de acceso a los usuarios. Se propone implementar un componente que permita tener acceso al sistema y control total de éste, a través de internet, mediante un módulo Wi-Fi, con lo que prácticamente podríamos tener interacción con nuestra casa en cualquier momento.
- Considerando que los sistemas domóticos serían de gran utilidad en lugares como por ejemplo cárceles, guarderías o institutos geriátricos, donde la seguridad y la supervisión son políticas fundamentales para prestar un buen servicio. Se propone desarrollar el diseño de un sistema domótico enfocado más a estos lugares mencionados.
- Se pueden agregar más opciones a la programación del microcontrolador. Se podría usar una pantalla LCD y un teclado matricial para crear una interfaz más amigable para el usuario, además de poder agregar comandos que se ejecuten a determinada hora del día.

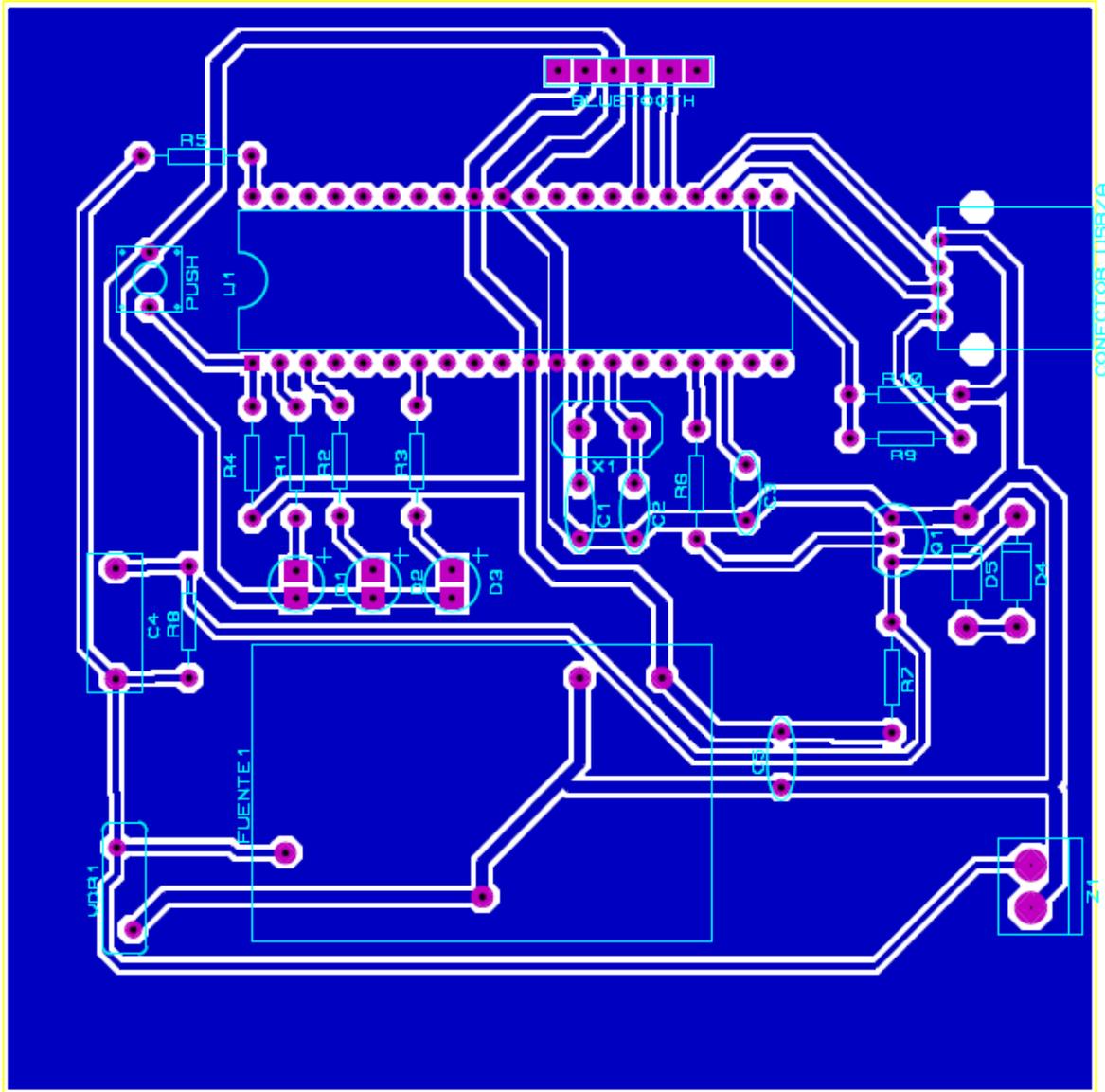
Anexo A: Esquemáticos







Anexo B: Circuitos impresos



Circuito impreso del módulo emisor

Referencias

1. *Domótica*. 2012 [cited Octubre, 2012; Available from: <http://es.wikipedia.org/wiki/Domotica>.
2. Mejía, E. *Domótica: un beneficio para el medio ambiente y para todos*. 2012; Available from: <http://www.magia.fm/sitio/?p=10864>.
3. CASADOMO. *Intro Domotica*. 1999 [cited 2012; Available from: <http://www.casadomo.com/noticiasDetalle.aspx?c=14>.
4. *Automatización Integral de Edificios*. [cited 2012; Available from: <http://isa.uniovi.es/docencia/AutomEdificios/transparencias/Generalidades2.pdf>.
5. Sthepanie. *Cableado de la red*. 2010 [cited 2012; Available from: <http://cruzlanda.blogspot.mx/2010/04/cableado-de-la-red-hay-dos-tipos-de.html>.
6. León, E.E.U. *Cableado*. 2005 [cited 2012; Available from: <http://www.monografias.com/trabajos30/cableado/cableado.shtml>.
7. Juancitux. *Fibra Optica, que es y como funciona*. 2010 [cited 2012; Available from: <http://blog.soporteti.net/documentacion-tecnica/fibra-optica-que-es-y-como-funciona/>.
8. Fuentes, S. *Bluetooth: un problema de privacidad*. 2008 [cited 2012; Available from: <http://www.xatakamovil.com/seguridad/bluetooth-un-problema-de-privacidad>.
9. *Wi-Fi*. 2012 [cited 2012; Available from: <http://es.wikipedia.org/wiki/Wi-Fi>.
10. keloko03. *Wi-Fi*. 2009 [cited 2012; Available from: <http://keloko03.blogspot.es/1259097360/>.
11. Mark. *ZigBee Celebrates 10 Years delivering The Internet of Things*. 2012 [cited 2012; Available from: <http://www.vesternet.com/blog/2012/10/zigbee-celebrates-10-years-delivering-the-internet-of-things>.
12. *Sensors: Different Types of Sensors*. 2012 [cited 2012; Available from: <http://www.engineersgarage.com/articles/sensors?page=4>.
13. Plata, O.A.C., *Estudio De Factibilidad De La Domótica En Una Casa-Habitación En México*, in *Facultad De Estudios Superiores Cuatitlan* 2005, Universidad Nacional Autónoma De México. p. 29, 30.
14. Plata, O.A.C., *Estudio De Factibilidad De La Domótica En Una Casa-Habitación En México*, in *Facultad De Estudios Superiores Cuatitlan* 2005, Universidad Nacional Autónoma De México. p. 31, 32.
15. Rodrigo Acuña Agost, J.A.O., Mariela Avendaño Arriagada. *Capítulo 3. Protocolos*. 2001 [cited 2012; Available from: <http://www2.udec.cl/~racuna/domotica/x10.htm>.
16. *Communication Protocols*. [cited 2012; Available from: <http://www.web177824.clarahost.fr/HomeTech/protocols.html>.
17. *LonWorks*. 2012 [cited 2012; Available from: <http://www.eltam-eh.com/LonWorks%C2%AE.html>.
18. *Estándares de control domótico*. 2012 [cited 2012; Available from: http://www.construnario.com/notiweb/tematicos_resultado.asp?id=149&informe=1.
19. *BatiBUS*. [cited 2012; Available from: <http://www.cwct.co.uk/ibcwindow/ibc/fieldbus/batibus.html>.

20. Martínez, L. *X-10: Comunicaciones a través de la red eléctrica (1/2)*. 2012 [cited 2012; Available from: <http://www.securitybydefault.com/2012/06/x-10-comunicaciones-traves-de-la-red.html>].
21. biblioman. *X - 10 - Consideraciones de Instalación*. 2011 [cited 2012; Available from: <http://www.aquihayapuntos.com/x-10.html?start=2>].
22. Quintero, L.F.H. *Viviendas inteligentes (Domótica)*. Ingeniería e Investigación, May/Aug. 2005. Vol.25 no.2.
23. biblioman. *X - 10 - Principio de funcionamiento del protocolo X-10*. 2011 [cited 2012; Available from: <http://www.aquihayapuntos.com/x-10.html?start=1>].
24. Wikipedia. *Universal Serial Bus*. 2012 [cited 2013; Available from: http://es.wikipedia.org/wiki/Universal_Serial_Bus].
25. labidi, r., 2012.
26. *Techno-Hall Electronica, Telecomunicaciones y Tecnologias de la Informacion*, 2012.
27. gvz, *comunicacion PIC por USB CDC y VB 2010*, 2012.
28. Wikipedia. *Bluetooth*. 2013 [cited 2013; Available from: <http://es.wikipedia.org/wiki/Bluetooth>].
29. Sacco, M., *Módulo Bluetooth HC-06 (Android)*, 2011.
30. Michael M. Cirovic, J.n.F.n.F., *Electrónica fundamental: dispositivos, circuitos y sistemas*, B. Reverté, Editor 1982. p. 599.
31. Rakshit, D.C.P.S., *Electronics: fundamentals and applications*, N.D.N.A. Publishers, Editor 2006. p. 648.
32. Jon Burroughs, M.T.I., *AN236, X-10® Home Automation Using the PIC16F877A*, Microchip, Editor 2002.
33. Sacco, M. *Electrónica Básica: Dimmer*. 2012 [cited 2012; Available from: <http://www.neoteo.com/electronica-basica-dimmer>].
34. Proteco. *C# Básico*. 2013 [cited 2013; Available from: http://proteco.fi-b.unam.mx/inicio.php?dest=cursos&id_curso=34&id_paquete=9].
35. Wikipedia. *Android*. 2013 [cited 2013; Available from: <http://es.wikipedia.org/wiki/Android>].
36. J.A.Sirgo, J.M.E. *Redes para los edificios y la industria. Domótica e Inteligencia Ambiental*. Redes Inalámbricas industriales 2008 [cited 2012; Available from: <http://isa.uniovi.es/~sirgo/doctorado/>].
37. *EL SISTEMA EIB*. [cited 2012; Available from: <http://bibing.us.es/proyectos/abreproy/11371/fichero/Volumen1%252FCapitulo3.pdf>].
38. *Protocolos de control domótico*. [cited 2012; Available from: <http://saber.ucv.ve/jspui/bitstream/123456789/746/3/%5BAnexo%20No%202%5D.pdf>].
39. *CEBus*. 2002 [cited 2012; Available from: <http://odisea.ii.uam.es/esp/recursos/Cebus.htm>].
40. Jesus Diaz-Alejo Roper, F.J.M.L. *Comunicación red Domótica*. EHS 2005 [cited 2012; Available from: [http://www.info-ab.uclm.es/labelec/solar/Otros/Domotica\(2005\)/indexprotocolos.htm#EHS](http://www.info-ab.uclm.es/labelec/solar/Otros/Domotica(2005)/indexprotocolos.htm#EHS)].
41. Martinez, D.I.T., *Desarrollo e Implementación de un sistema domótico del estado de Colima*, in *Facultad de Telemática*2004, Universidad de Colima. p. 105.
42. Velásquez, V.H.G., *Tecnología Para Automatización De Edificios*, in *Facultad De Ingeniería*2008, Universidad De San Carlos De Guatemala. p. 107.

43. Rodríguez, P.A.C.H., *Estudio Para Reducción De Consumo Energético En Colombia Basado en Tecnología Domótica*, in *Facultad De Ingenierías* 2011, Fundación Universitaria Konrad Lorenz. p. 96.
44. *Bus de Instalación Europeo*. 2012 [cited 2012; Available from: http://es.wikipedia.org/wiki/Bus_de_Instalaci%C3%B3n_Europeo.
45. *Instabus*. 2012 [cited 2012; Available from: http://en.wikipedia.org/wiki/European_Installation_Bus.
46. *EHS (European Home System)*. [cited 2012; Available from: <http://odisea.ii.uam.es/esp/recursos/ehs.htm>.
47. *Protocolos de control EHS*. [cited 2012; Available from: http://www.info-ab.uclm.es/labelec/solar/domotica/estandares_protocolos_ehs.html.
48. *Limitador Zener*. 2000 [cited 2012; Available from: <http://hyperphysics.phy-astr.gsu.edu/hbasees/hframe.html>.
49. *Protocolo LonWorks*. 2012 [cited 2012; Available from: <http://en.wikipedia.org/wiki/LonWorks>.
50. Alterach, G.Á. *Calculador de filtro RC*. 2011 [cited 2012; Available from: <http://gzaloprgm.com.ar/filtrorc/>.
51. *Temporizador Timer 2 en los microcontroladores PIC*. [cited 2012; Available from: <http://www.circuitoselectronicos.org/2011/04/temporizador-timer-2-en-los.html>.
52. migsantiago. *PWM en 877A a 120kHz*. 2005 [cited 2012; Available from: <http://www.todopic.com.ar/foros/index.php?topic=4910.0>.
53. *Cables, Filtros y Acondicionadores de red eléctrica*. Modos de conexión, tipos de cables y modos de conexión 2007 [cited 2012; Available from: <http://www.duiops.net/hifi/cine-en-casa-conexiones-filtrosacondicionadores.html>.
54. 2010, c.P.p.U.C.y.V. *comunicacion PIC por USB CDC y VB 2010*. 2008 [cited 2013; Available from: http://www.i-micro.com/pdf/articulos/usb_imicro.pdf.
55. Villafuerte, S. *PIC + USB para principiantes*. 2009 [cited 2013; Available from: http://www.migsantiago.com/index.php?option=com_content&view=article&id=9&Itemid=10.
56. PIC, D. *Proyecto 1 USB con PIC (XP)*. 2011 [cited 2013; Available from: <http://www.desarrollopic.com/proyectos/proyecto-1-usb-con-pic-xp/>.
57. Trasto, M., *Emulación CDC RS-232 USB*.
58. *HC Serial Bluetooth Products User Instructional Manual* G.H.I.T. Co, Editor.
59. Sacco, M., *Bluetooth + Android + PIC + LED = "Hola Mundo"*, 2011.
60. A.V, K., *Data transfer between Android and Arduino via Bluetooth*, 2013.
61. Wikipedia. *Perfil Bluetooth*. 2013 [cited 2013; Available from: http://es.wikipedia.org/wiki/Perfil_Bluetooth#Serial_Port_Profile_.28SPP.29.
62. Wikipedia. *Protocolos Bluetooth*. 2013 [cited 2013; Available from: <http://es.wikipedia.org/wiki/RFCOMM#RFCOMM>.
63. SOLEC, S.E.y.d.A. *Electronica*. 2013; Available from: <http://www.solecmexico.com/electronica/TensionAlterna.pdf>.
64. Tecnopía. *Creating Context Menus in C#*. 2012 [cited 2013; Available from: http://www.techotopia.com/index.php/Creating_Context_Menus_in_C_Sharp.

65. *Add ContextMenu to ListBox*. 2012 [cited 2013; Available from: http://www.java2s.com/Tutorial/CSharp/0460_GUI-Windows-Forms/AddContextMenutoListBox.htm].
66. Microsoft. *Check/Uncheck on MenuStrip options*. 2013; Available from: <http://social.msdn.microsoft.com/Forums/en-US/csharpgeneral/thread/c01aba2f-5601-4736-a341-4cec0a0230b2/>.
67. stackoverflow. *How to hide TabPage from TabControl*. 2009 [cited 2013; Available from: <http://stackoverflow.com/questions/552579/how-to-hide-tabpage-from-tabcontrol>].
68. CDFlik. *Cargar imagen c#*. 2012 [cited 2013; Available from: <http://csharpmaniacs.blogspot.mx/2012/04/cargar-imagen-en-picturebox-usando.html>].
69. Microsoft. *Using Settings in C#*. 2013 [cited 2013; Available from: <http://msdn.microsoft.com/en-us/library/aa730869%28VS.80%29.aspx>].
70. ttony, *Guardar configuracion previa en C#*, 2010.

[36-70]