



UNIVERSIDAD NACIONAL DE CHIMBORAZO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES

**“Trabajo previo a la obtención del título de Ingeniero en
Electrónica y Telecomunicaciones”**

TRABAJO DE GRADUACIÓN

**“DISEÑO E IMPLEMENTACIÓN DE UN ENTRENADOR DE
MICROCONTROLADORES PARA LA COMUNICACIÓN ZIGBEE”**

Autores: Nelson Rolando Conya Yautibug

Cristopher David Lema Yépez

Director. ING. Fabián Gunsha

RIOBAMBA – ECUADOR

AÑO 201

CERTIFICACIÓN

tribunal de Graduación del proyecto de investigación de título:
"DISEÑO E IMPLEMENTACIÓN DE UN ENTRENADOR DE
MICROCONTROLADORES PARA LA COMUNICACIÓN" presentado por
Nelson Rolando Conya Yautibug, Christopher David Lema Yépez, dirigido por el
Ing. Fabián Celso Gunsha Maja.

Certifico que el siguiente trabajo de investigación previo a la obtención del título de grado de Ingeniero en Electrónica y Telecomunicaciones con el tema "DISEÑO E IMPLEMENTACIÓN DE UN ENTRENADOR DE MICROCONTROLADORES PARA LA COMUNICACIÓN ZIGBEE" el cual ha sido elaborado por: Nelson Rolando Conya Yautibug y Cristopher David Lema Yépez mismo que ha sido revisado y analizado al cien por ciento con el asesoramiento de mi persona en calidad de Tutor, el cual se encuentra apto para su presentación y defensa respectiva ”


Ing. Carlos Peñafiel
Presidente del tribunal



FIRMA

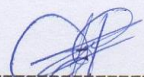
Es todo en cuanto puedo informar en honor a la verdad


Ing. Fabián Gunsha
Director del Proyecto



FIRMA

Atentamente:



Ing. Fabián Gunsha
TUTOR
Miembro del Tribunal

FIRMA

Los miembros del tribunal de Graduación del proyecto de investigación de título: “DISEÑO E IMPLEMENTACIÓN DE UN ENTRENADOR DE MICROCONTROLADORES PARA LA COMUNICACIÓN” presentado por Nelson Rolando Conya Yautibug, Cristopher David Lema Yépez, dirigido por el Ing. Fabián Celso Gunsha Maja.

Una vez escuchada la defensa oral y revisado el informe final del proyecto de investigación con fines de graduación escrito en el cual se ha constado el cumplimiento de las observaciones realizadas, remite el presente para uso y custodia en la biblioteca de la Facultad de Ingeniería de la UNACH.


Para la constancia de lo expuesto firman:

Ing. Carlos Peñafiel
Presidente del tribunal




FIRMA

Ing. Fabián Gunsha
Director del Proyecto



FIRMA

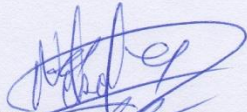
Ing. Deysi Inca
Miembro del Tribunal



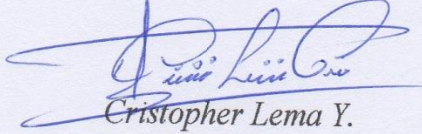
FIRMA

AUTORÍA DE LA INVESTIGACIÓN

“La responsabilidad del contenido de este Proyecto de Graduación, nos corresponde exclusivamente a: Nelson Rolando Conya Yautibug, Cristopher David Lema Yépez e Ing. Fabián Celso Gunsha Maji, el patrimonio intelectual de la misma a la Universidad Nacional de Chimborazo.



Nelson Conya Y
0604233197



Cristopher Lema Y.
0604142018

DEDICATORIA I

Este trabajo está dedicado a Dios por haberme permitido llegar hasta este punto y haberme dado salud para lograr mis objetivos, además de su infinita bondad y amor. A mi madre Carmita por haberme apoyado en todo momento, por sus consejos, sus valores, por la motivación constante y por su gran amor. A mi padre José por los ejemplos de perseverancia y constancia que lo caracteriza y que me ha infundido siempre y sobre todo por su gran amor. A mi hermana Silvia y mi medio Hermano Manuel que siempre me han brindado su gran apoyo y su gran amor.

Cristopher Lema

DEDICATORIA II

Dedico a Mercedes Yautibug mi querida madre, por ser mi soporte y ejemplo a seguir, sin ella jamás hubiese llegado a saborear estos momentos tan importantes de mi vida.

A mis hermanos y hermanas políticas, quienes con sus palabras de aliento impulsaron, motivaron para que cumpla con este anhelado sueño.

A toda mi familia por brindarme el amor y la calidez apoyándome en momentos difíciles.

Nelson Conya

AGRADECIMIENTO I

Agradezco la Universidad Nacional de Chimborazo y en especial a la Facultad de Ingeniería de la Escuela de Electrónica y Telecomunicaciones por permitir ser parte de una generación de triunfadores y gente productiva para el País.

A mis queridos maestros gracias por su tiempo, su apoyo, su amistad, por compartir sus experiencias sus conocimientos en el desarrollo de nuestra formación profesional.

Un agradecimiento especial al Ing. Fabián Gunsha quien con sus grandes y amplios conocimientos nos tuteló en el desarrollo de este trabajo de investigación y de igual forma a los distinguidos Miembros del Tribunal.

Cristopher Lema

AGRADECIMIENTO II

Agradezco a Dios al ser supremo que me dio la vida, fuerzas, fe y has llenado de bendiciones en todo este tiempo, a él que con su infinito amor me ha dado la sabiduría suficiente para culminar esta carrera universitaria.

De igual manera al Ing. Fabián Gunsha como director de mi tesis, me ha orientado, apoyado y corregido en esta labor científica con un interés y una entrega que ha sobrepasado.

A mi madre, quien es responsable de esta historia, quien cumplió el duro valor de ser padre y madre a la vez, gracias por su sacrificio y paciencia que demostraste todos estos años.

A mis compañeros, quienes sin esperar nada a cambio compartieron sus conocimientos.

Nelson Conya

ÍNDICE GENERAL

TRABAJO DE GRADUACIÓN	I
AUTORÍA DE LA INVESTIGACIÓN	¡Error! Marcador no definido.
DEDICATORIA I.....	V
DEDICATORIA II.....	VI
AGRADECIMIENTO I.....	VII
AGRADECIMIENTO II.....	VIII
ÍNDICE GENERAL	IX
ÍNDICE DE CUADROS.....	XIII
ÍNDICE DE GRÁFICOS	XIV
RESUMEN	XVI
SUMMARY	¡Error! Marcador no definido.
INTRODUCCIÓN	1
CAPITULO I	2
1 FUNDAMENTACIÓN TEÓRICA.....	2
1.1 DOMÓTICA	2
1.1.1 Servicios que gestiona la domótica.....	2
1.1.1.1 Gestión de seguridad.....	2
1.1.1.2 Gestión de comunicaciones.....	2
1.1.1.3 Gestión de accesibilidad.....	3
1.1.1.4 Gestión de energía.....	3
1.2 TECNOLOGÍA ZIGBEE.....	3
1.2.1 Características importantes de ZigBee.....	3
1.3 MÓDULOS XBEE	4
1.3.1 Utilidades del Xbee.....	4
1.3.2 Ventajas y Desventajas de Xbee	4
1.3.2.1 Ventajas.....	4
1.3.2.2 Desventajas	5
1.3.3 Modos de operación de los módulos Xbee	5
1.3.3.1 Modo Transmisor/Receptor	5
1.3.3.2 Modo comandos AT	5
1.3.3.3 Modo API.....	6
1.3.3.4 Distribución de pines del Xbee	6

1.3.3.5	Canalización.....	7
1.3.4	Software X-CTU.....	8
1.4	CIRCUITO INTEGRADO DS1307	8
1.4.1	Características	9
1.4.2	Frecuencia de operación.....	9
1.5	SENSOR DE TEMPERATURA LM35	9
1.5.1	Características del LM35	10
1.6	PIR SENSOR	10
1.6.1	Características del PIR SENSOR.....	10
1.7	MICROCONTROLADOR	11
1.7.1	PIC 16F877A	11
1.7.2	Características	12
1.7.3	Comunicación USART	12
1.7.4	ATmega 328P	13
1.7.5	Características	13
	CAPITULO II.....	14
2	METODOLOGÍA	14
2.1	Tipo de estudio.....	14
2.2	POBLACIÓN Y MUESTRA.....	14
2.2.1	Población.....	14
2.2.2	Muestra	14
2.3	Hipótesis	15
2.4.	OPERACIONALIZACION DE VARIABLES	16
2.5	PROCEDIMIENTOS	17
2.5.1	Diseño de circuito y PCB de comunicaciones	17
2.5.1.1	Diseño de circuito electrónico de comunicaciones	17
2.5.1.2	Diseño de PCB de comunicaciones	19
2.5.2	Diseño de circuito y PCB de sensores	20
2.5.2.1	Diseño de circuito uno de sensores	21
2.5.2.2	Diseño de PCB uno de sensores.....	23
2.5.2.3	Diseño de circuito dos de sensores	25
2.5.2.4	Diseño de PCB dos de sensores	27
2.5.4	Diseño del sistema domótico con entrenador de comunicaciones	29
2.5.4.1	Diseño del coordinador	29

2.5.4.2	Diseño del Router-A	32
2.5.4.3	Diseño del Router-B	36
2.6	COMPROBACIÓN DE LA HIPÓTESIS	39
2.6.1	Planteamiento de hipótesis.....	39
2.6.2	Establecimiento de nivel de significancia.....	39
2.6.3	Obtención de datos.....	39
2.6.4	Calculo estadístico utilizando software SPSS.....	40
CAPÍTULO III.....		42
3	RESULTADOS.....	42
3.1	PRUEBA DE COMUNICACIÓN DE RED MESH.....	42
3.1.1	Comunicación.	42
3.2	ANÁLISIS FINANCIERO	46
CAPÍTULO IV.....		47
4	Discusión.....	47
CAPÍTULO V.....		48
5	CONCLUSIONES Y RECOMENDACIONES.....	48
5.1	Conclusiones	48
5.2	Recomendaciones	48
CAPITULO VI.....		49
6	PROPUESTA.....	49
CAPITULO VII		52
7	BIBLIOGRAFÍA	52
ANEXOS		53
ANEXO 1. CIRCUITO COMPLETO DE PLACA COMUNICACIONES.....		54
ANEXO 2. CIRCUITO COMPLETO DE PLACA UNO DE SENSORES		55
ANEXO 3. CIRCUITO COMPLETO DE PLACA DOS DE SENSORES.....		56
ANEXO 4. PROGRAMACIÓN DEL ATMEGA 328P		57
ANEXO 5. PROGRAMACION DEL MICROCONTROLADOR 16F877A – ROUTER- A.....		71
ANEXO 6. PROGRAMACIÓN DEL MICROCONTROLADOR 16F877A COMO ROUTER-B.....		73
ANEXO 7. DISEÑO DE CIRCUITOS ELÉCTRICOS DEL COORDINADOR		75
ANEXO 8. CIRCUITOS ELÉCTRICOS DE LOS DISPOSITIVOS UTILIZADOS EN EL ROUTER-A.		76

ANEXO 9. CIRCUITOS ELÉCTRICOS DE LOS DISPOSITIVOS UTILIZADOS EN EL ROUTER-B.....	77
ANEXO 10. HOJA DE DATOS DEL MÓDULO XBEE	78
ANEXO 11. HOJAS DE DATOS DEL MICROCONTROLADOR 16F877A.....	81
ANEXO 12. HOJA DE DATOS DEL LM35	85
ANEXO 13. HOJA DE DATOS DEL DS1307	89
ANEXO 14. HOJA DE DATOS DEL PIR SENSOR.....	92
ANEXO 15. PRÁCTICA DE LABORATORIO 1.1: CONFIGURACIÓN DE LOS MÓDULOS XBEE CON TOPOLOGÍA MESH.	95
ANEXO 16. IMÁGENES OBTENIDAS EN LA COMPROBACIÓN DE HIPÓTESIS	107
ANEXO 17. DESCRIPCIÓN DE LA PLACA DE COMUNICACIONES	108
ANEXO 18. PRACTICA PARA LA COMPROBACION DE HIPOTESIS SIN ENTRENADOR	111
ANEXO 19. PRACTICA PARA LA COMPROBACION DE HIPOTESIS CON ENTRENADOR	124

ÍNDICE DE CUADROS

Tabla 1.- Principales características del estándar ZigBee.....	3
Tabla 2.- Comandos AT.....	6
Tabla 3.- Descripción detallada de los pines del módulo Xbee.....	7
Tabla 4.- Frecuencia de oscilación.....	9
Tabla 5.- Características del LM35.....	10
Tabla 6.- Características del PIR SENSOR.....	10
Tabla 7.- Características el 16F877A.....	12
Tabla 8.- Características del ATmega 328P.....	13
Tabla 9.- Operacionalizacion de las variables independiente y dependientes.....	16
Tabla 10.- Dimensiones de pistas.....	27
Tabla 11.- Declaración de variables coordinador.....	30
Tabla 12.- Configuración del módulo Xbee coordinador.....	32
Tabla 13.- Declaración de variables Router-A.....	35
Tabla 14.- Configuración del módulo Xbee Router-A.....	35
Tabla 15.- Declaración de variables Router-B.....	38
Tabla 16.- Configuración del módulo Xbee Router-B.....	38
Tabla 17.- Datos estadísticos.....	40
Tabla 18.- Estadísticos de muestras relacionadas.....	41
Tabla 19.- Pruebas de muestras relacionadas.....	41
Tabla 20.- Prueba de comunicaciones.....	45
Tabla 21.- Presupuesto total del proyecto.....	46

ÍNDICE DE GRÁFICOS

Figura 1.- Elementos de interconexión domótica.....	2
Figura 2.- Dispositivo Xbee.....	4
Figura 3.- Correspondencia del Buffer.....	5
Figura 4.- Canalización de la banda 2.4GHz.....	8
Figura 5.- Circuito integrado DS1307.....	8
Figura 6.- Sensor de temperatura LM35.....	9
Figura 7.- PIR SENSOR.....	10
Figura 8.- Representación esquemática de un microcontrolador.....	11
Figura 9.- Pic 16F877A.....	11
Figura 10.- Comunicación Usart.....	12
Figura 11.- ATmega 328P.....	13
Figura 12.-Fuentes de alimentación.....	17
Figura 13.- Circuito de módulo Xbee.....	18
Figura 14.- Circuito de módulo Bluetooth.....	18
Figura 15.- Posicionamiento de elementos, placa de comunicación.....	19
Figura 16.- Diseño de pistas, placa de comunicaciones.....	20
Figura 17.- Módulo Xbee en 3D, placa de comunicaciones.....	20
Figura 18.- Circuito de sensor LM35.....	21
Figura 19.- Circuito de sensor encoder.....	22
Figura 20.- Circuito integrado DS1307.....	22
Figura 21.- Circuito de sensor MPU6050.....	22
Figura 22.- Circuito de MLX90614.....	23
Figura 23.- Posicionamiento de elementos, placa uno.....	23
Figura 24.- Diseño de pistas, placa uno.....	24
Figura 25.- Placa uno de sensores en 3D.....	24
Figura 26.-Circuito de sensor temperatura y humedad.....	25
Figura 27.- circuito de sensor de corriente.....	25
Figura 28.- circuito sensor de movimiento	26
Figura 29.- Circuito sensor ultrasónico.....	26
Figura 30.- Circuito sensor de alcohol.....	26
Figura 31.- Posicionamiento de elementos, placa dos.....	27
Figura 32.-Diseño de pistas, placa dos.....	28

Figura 33.- Placa dos de sensores en 3D.....	28
Figura 34.- Diagrama esquemático del proyecto.....	29
Figura 35.- Diagrama de bloques del coordinador.....	29
Figura 36.- Circuito del coordinador.....	30
Figura 37.- Flujo grama como coordinador.....	31
Figura 38.- Diagrama de bloques del Router-A.....	32
Figura 39.- Circuito Router-A.....	33
Figura 40.- Circuito de potencia.....	33
Figura 41.- Flujo grama, Router-A.....	34
Figura 42.- Diagrama de bloques del Router-A.....	36
Figura 43.- Circuito Router-B.....	36
Figura 44.- Flujo grama, Router-B.....	37
Figura 45.- Domicilio con el sistema instalado.....	42
Figura 46.- Circuito coordinador.....	42
Figura 47.- Dispositivo Router-B.....	43
Figura 48.- Dispositivo Router-A.....	43
Figura 49.- Activación del foco.....	44
Figura 50.- Control del calefactor.....	44
Figura 51.- Sistema de seguridad.....	45
Figura 52.- Diagrama del sistema organizacional del proyecto.....	51

RESUMEN

El presente proyecto de titulación, tiene como objetivo diseñar e implementar un entrenador electrónico para microprocesadores basado en las comunicaciones ZigBee, debido a la demanda que requiere las asignaturas para realizar sistemas embebidos y que el estudiante mejore el rendimiento académico optimizando el tiempo y la economía.

Para realizar el presente proyecto se estructuro siete capítulos partiendo por la descripción de las generalidades, fundamentos teóricos, la ampliación de los conocimientos respecto a las comunicaciones ZigBee, su funcionalidad, características, protocolos y la configuración en diferentes topologías.

Laplacade comunicaciones y sensores fueron diseñadas utilizando el software Proteus (Isis, ares) y los elementos en 3D en el software skethup, implementación del entrenador didáctico tomando en cuenta los parámetros requeridos para su correcto funcionamiento, el cual permite desarrollar sistemasdomóticos que consiste en diseñar el control automático de; iluminación, climatización y seguridad.

La comprobación dela hipótesis realizada con los estudiantes de sexto semestre de la Facultad de Ingeniería, Escuela de Electrónica y Telecomunicaciones de la Universidad Nacional de Chimborazo con datos reales así demostrando la eficiencia del entrenador, realizar prácticas con los entrenadores para la verificación de su correcto funcionamiento.La última parte del proyecto consiste en presentar: conclusiones, recomendaciones, propuestas y los anexos que conllevo a realizar este proyecto.



UNIVERSIDAD NACIONAL DE CHIMBORAZO

FACULTAD DE INGENIERÍA

CENTRO DE IDIOMAS



Dp.S. Byron Soria

18 de Diciembre del 2015

SUMMARY

This Project previous to get engineering degree, it aims to design and implement an electronic trainer for microprocessors based on the Zigbee communications, due to demand requiring subjects for embedded systems, and Students their academic performance optimizing time and the economy.

To make this project there was seven chapters starting with the description of the generalities was structured, theoretical foundations, the expansion of knowledge about the Zigbee communications, functionality, features, and configuration protocols in different topologies.

The communications board and sensors were designed using the Proteus software (Isis, ares) and 3D elements in the skethup software, the implementation of instructional instructor taking into account the parameters required for proper operation, which permits to develop automation systems consisting of automatic control design; lighting, air conditioning and security.

Hypothesis survey was accomplished with the sixth semester students of the Faculty of Engineering, School of Electronics and Telecommunications at the National University of Chimborazo with real data, so demonstrating the efficiency of the trainer, and performing practices with the trainers for verifying proper operation. In the last part of the project it is about: conclusions, recommendations, proposals and annexes that led to this project.

CENTRO DE IDIOMAS



COORDINACION

INTRODUCCIÓN

El propósito fundamental de este proyecto de investigación es diseñar un entrenador para desarrollar prácticas de laboratorio en el área de microprocesadores, ya que hoy en día el estudiante universitario ha invertido tanto en recursos económicos como en tiempo, y sus fallos constantes por la fatiga del armado del circuito han generado factores que impide el desarrollo eficiente de los estudiantes.

Actualmente las comunicaciones ZigBee se ha convertido en un factor común para diversas aplicaciones, por lo tanto el estudiante debe estar acorde a las necesidades de la sociedad y así brindar una solución inmediata y eficiente.

Para integrar el desarrollo del diseño del entrenador se eligió ZigBee que contiene un conjunto de protocolos de alto nivel de comunicación, como una de las características es de bajo costo, y pueden estar conectados con 65535 dispositivos a la vez. Esta comunicación es utilizada en muchas áreas como es las industrias, domóticas, seguridad, edificios, entre otras.

El uso del entrenador permitirá una mayor facilidad en el desarrollo de las prácticas, que integran; sensores, circuitos básicos de electrónica, display, teclados, matriz de led y componentes electrónicos con capacidades básicas de comunicaciones, que permiten realizar sistemas embebidos de acuerdo a las necesidades que requieran las asignaturas a fines.

La metodología de esta investigación es empírica y cuasi experimental, pretende proporcionar entrenadores que faciliten el desarrollo de las prácticas en menor costo y menor tiempo debido que son didácticos y versátiles.

CAPITULO I

1 FUNDAMENTACIÓN TEÓRICA

1.1 DOMÓTICA

La domótica no es más que un conjunto de sistemas capaces de automatizar y controlar algunas de las funciones básicas de las instalaciones de una casa, estas funciones se puede ejecutar por medio de redes interiores o exteriores de comunicación, puede ser mediante cable o inalámbricamente, recogen toda la información posible para procesar diferentes órdenes.

Los elementos de interconexión domóticas se muestran en la **Figura 1**.



Figura1. Elementos de interconexión domótica

Fuente:<http://rayuelaradio.com/domus-automatica-domotica/.jpg>

1.1.1 Servicios que gestiona la domótica

La domótica ofrece algunos servicios de gran utilidad en una vivienda.

1.1.1.1 Gestión de seguridad

El sistema de seguridad es el encargado de proteger los bienes patrimoniales, seguridad personal, alarmas contra incendios, y presencia de persona extrañas etc.

1.1.1.2 Gestión de comunicaciones

Es el encargado de intercambiar información ya sea entre los usuarios o entre los mismos dispositivos que se están utilizando.

1.1.1.3 Gestión de accesibilidad

Es accesible para las aplicaciones o instalaciones de control remoto del entorno que favorece la autonomía personal de personas que limitan sus funciones o con discapacidad.

1.1.1.4 Gestión de energía

Se encarga de controlar el sistema de iluminación como también de los electrodomésticos existentes en la vivienda.

1.2 TECNOLOGÍA ZIGBEE

ZigBee, es una tecnología con protocolo inalámbrico normalizado para la conexión de una red de área personal inalámbrica o WPAN.

Esta tecnología a diferencia de las otras tecnologías inalámbricas ya han sido diseñadas para soportar diversas aplicaciones con una conectividad sostenida, son específicamente diseñados para redes domóticas, que realizan trabajos con sensores, actuadores, etc. Es ideal para realizar comunicaciones punto a punto, punto a multipunto y comunicación en red mesh, opera en una banda libre de 2.4 GHZ, con una baja tasa de transferencia de datos.

1.2.1 Características importantes de ZigBee

Las características importantes del estándar ZigBee, se identifican en la **Tabla 1**.

CARACTERÍSTICAS	RANGO
Bandas de Frecuencia	2400 – 2483.5 MHz (Todo el mundo)
Alcance	10 – 75 m
Retardo (Latencia)	Menor a 15ms
Número de canales	2400 -2483.5 MHz: 16 canales
Rango de trasmisión de datos	2400 – 248.5 MHz: 250 kbps
Direccionamiento	Adicional: corto de 16 bits
Nodos por red	Hasta 65536 Hasta 255 por subred

Tabla 1. Principales características del estándar ZigBee

Fuente: Los Autores

1.3 MÓDULOS XBEE

Los módulos Xbee son soluciones integradas que brindan un medio de comunicación inalámbrico para la interconexión entre dispositivos, creando redes punto a punto, punto a multipunto, y red mesh. Fueron diseñados para aplicaciones que requieren de un alto tráfico de datos, baja latencia, menor consumo de potencia y una sincronización de comunicación predecible, **Figura 2**.

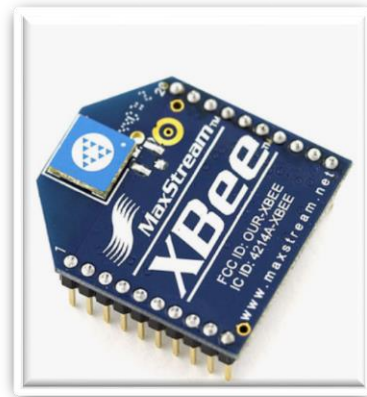


Figura 2.Dispositivo Xbee

Fuente:<http://programacionits.blogspot.com/2012/05/zigbee-bluetooth-rn-42-wi-fi.html>

1.3.1 Utilidades del Xbee

- Automatización de hogares.
- Sistemas de seguridad.
- Monitoreo de sistemas remotos.
- Aparatos domésticos.
- Alarmas contra incendios.
- Aplicaciones en industrias

1.3.2 Ventajas y Desventajas de Xbee

1.3.2.1 Ventajas

- Bajo consumo de energía.
- Bajo Costo comparado con el resto de módulos de comunicaciones.
- Facilidad de configuración.
- Soporte de múltiples tecnologías de red.
- Reduce el tiempo en el envío y recepción de datos.
- Baja latencia.

1.3.2.2 Desventajas

- Tasa de transferencia de datos muy baja.
- Tiene menor cobertura, porque pertenece a redes inalámbricas de tipo WPAN¹.
- Menor ancho de banda.

1.3.3 Modos de operación de los módulos Xbee

Los módulos Xbee, pueden trabajar en 3 modos: Transmisor/Receptor, Comandos AT y API.

1.3.3.1 Modo Transmisor/Receptor

Este modo permite realizar transmisión y recepción de datos en forma directa e indirecta a la dirección del destino, esto significa que se encarga de recibir paquetes RF² (Modo receptor) y transmitir paquetes RF (Modo Transmisor) a través de la antena del módulo **Figura 3**.

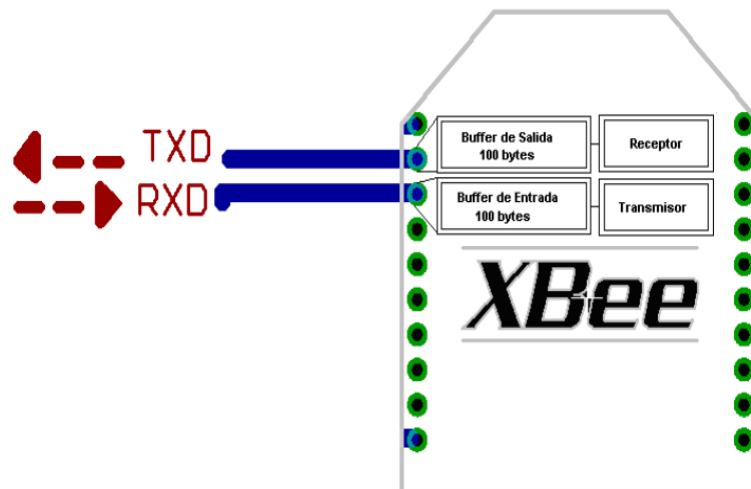


Figura 3. Correspondencia del Buffer

Fuente: http://www.olimex.cl/pdf/Wireless/ZigBee/XBee-Guia_Usuario.pdf

1.3.3.2 Modo comandos AT

Este modo permite ingresar comandos al módulo Xbee, para configurar los distintos parámetros de acuerdo a las necesidades del usuario, es necesario utilizar el hyper terminal de Windows para tener conectividad entre el módulo Xbee y la PC, ver **Tabla 2**.

¹Red inalámbrica de área personal

²Radio Frecuencia

COMANDOS AT	DESCRIPCIÓN
+++	Entrar en modo de comandos AT
AT	Intento de ingresar en modo AT
ATID	Ingreso dirección PAN
ATSH/ATSH	Ingresar dirección del dispositivo TX/RX
ATND	Verificar, módulos conectados a las Red
ATCN	Salir de modo AT

Tabla 2. Comandos AT

Fuente: Los Autores

1.3.3.3 Modo API

Este modo es más complejo, porque utiliza tramas con cabecera para asegurar la transmisión y recepción de datos al estilo TCP³. Esta configuración hace que la información que sale y entra al módulo sea empaquetada en tramas, que definen operaciones y eventos dentro del módulo.

Así una trama de transmisión de información incluye:

- Trama de información RF transmitida

Trama de recepción de información incluye:

- Tramas de información RF recibida.
- Comandos de respuesta ACK⁴

Opciones que permite la API

- Transmitir información a múltiples destinatarios.
- Direccionar el destino de Transmisión y recepción de datos

1.3.3.4 Distribución de pines del Xbee

La distribución de los pines de los módulos de Radiofrecuencia Xbee se describe en la **Tabla 3**.

³Protocolo de control de transmisión

⁴Acuse de recibo

Pin	Nombre	Dirección	Descripción
1	VCC	-	Alimentación
2	Dout	Salida	Salida UART
3	DIN/	Entrada	Entrada UART
4	DIO12	E/S	Digital E/S12
5	RTS	Entrada	Módulo de reinicio
6	PWM0/RSSI/DIO10	E/S	PWM Salida 0 Indicador de la Intensidad de la Señal Recibida Digital E/S
7	DIO11	E/S	Digital E/S 11
8	Reservado	E/S	No conectar
9	SLEEP_RQ/DIO8	E/S	Pin Sleep, Control en línea o Digital E/S 8
10	GND	-	Tierra
11	DIO4	E/S	Digital E/S 4
12	DIO7	E/S	Control de flujo Clear-to-Send o Digital E/S 7. CTS, si se encuentra Habilitado este es una salida
13	ON	Salida	Indicador del estado del módulo o Digital E/S 9
14	VREF	Entrada	Conectado a GND o VCC
15	DIO6	E/S	Digital E/S
16	DIO6	E/S	Control de Flujo E/S
17	AD3/DIO3	E/S	Entrada analógica 3 o Digital E/S
18	AD2/DIO2	E/S	Entrada analógica 2 o Digital E/S
19	AD1/DIO1	E/S	Entrada analógica 1 o Digital E/S
20	AD0/DIO0	E/S	Entrada analógica 0 o Digital E/S

Tabla 3. Descripción detallada de los pines del módulo Xbee

Fuente: <http://www.andresduarte.com/arduino-y-xbee>

1.3.3.5 Canalización

Consta de 16 canales de 2 MHz como se muestra en la **Figura 4.** La distribución de canales en el espectro de frecuencia, trabaja dentro del rango de 2.4 y 2.4835 GHz, con un amplio espacio entre canales de 5 MHz, lo que facilita el requerimiento de filtrado en la transmisión y recepción de datos, el espaciado de los canales provee slots en las que hay superposición con las bandas WIFI, Bluetooth, permitiendo diseñar alrededor de 65535 nodos.

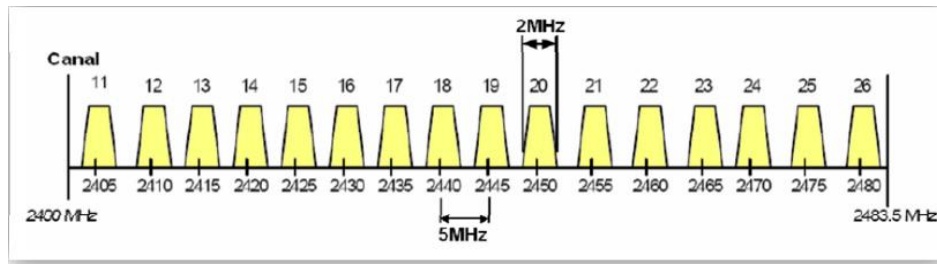


Figura 4. Canalización de la banda 2.4 GHz

Fuente: http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/archundia_p_fm/capitulo4.pdf

1.3.4 Software X-CTU

Software gratuito que se utiliza como interfaz para configurar el módulo Xbee a través de la PC, que proviene de la empresa MaxStream.

Partes importantes de software.

- Configuraciones del PC.- Sirve para seleccionar el puerto serial por la cual se comunica el módulo Xbee con la PC.
- Opción terminal.- Pantalla que sirve para leer datos recibidos y enviados por el resto de módulos que están conectados.
- Opción de configuración del Modem.- Sirve para configurar o leer ciertos parámetros.

1.4 CIRCUITO INTEGRADO DS1307

El DS1307, **Figura 5**, es un Reloj en tiempo real con comunicación I²C, de bajo consumo de energía, con código binario decimal, la dirección de datos de integrado son transmitidos a través de 2 hilos serie, el SCL⁵ y SDA⁶, este reloj/calendario provee información de segundos, minutos, horas, día, fecha, mes y año,

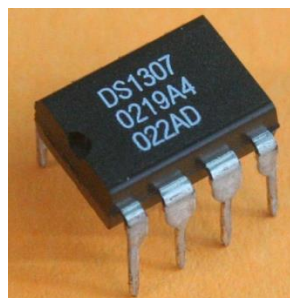


Figura 5. Circuito Integrado DS1307

Fuente: Los Autores

⁵Define Pin de Reloj I2C

⁶Define Pin de datos I2C

1.4.1 Características

Las características más relevantes del DS1307:

- 56-Byte con batería de respaldo,
- RAM de uso general para almacenamiento de datos.
- Interfaz Serial I²C.
- Señal de salida de Onda-Cuadrada.
- Detector Automático de Fallo de energía y circuito de conmutación.
- Consumo menos de 500nA
- Rango de temperatura Industrial Opcional: -40°C a +85°C.
- Disponible en empaquetado de 8-Pines

1.4.2 Frecuencia de operación

La **Tabla 4**, muestra el control de frecuencia de oscilación de la onda cuadrada a la salida del pin del integrado.

BIT 1	BIT 0	FRECUENCIA DE OSCILACIÓN
0	0	1Hz
0	1	4.096KHz
1	0	8.192KHz
1	1	32.768KHz

Tabla 4. Frecuencia de oscilación
Fuente: DataSheet

1.5 SENSOR DE TEMPERATURA LM35

El LM35 es un sensor de temperatura que brinda una salida de voltaje linealmente proporcional a la temperatura en grados centígrados a su pin 2, con su rango de temperatura a censar entre -55 a 150 grados ver **Anexo 12**. Su funcionamiento es muy simple ver **Figura 6**, el pin 1 debe ser alimentado con una fuente de voltaje aproximadamente de 4 a 20 voltios, mientras el pin 3 debe ir conectado a tierra.

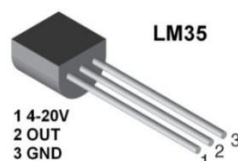


Figura 6. Sensor de temperatura LM35
Fuente: DataSheet

1.5.1 Características del LM35

Características principales del sensor se detallan en la **Tabla 5**.

PARÁMETROS	VALORES
Calibración directa en grados centígrados	°C
Factor de escala	+10.0mv/°C
Precisión	0.5 a 25 °C (ambiente)
Rango	-55 a +150 °C
Voltaje operación	4 a 20V
Corriente operación	<=60 uA

Tabla 5.Características del LM35

Fuente: DataSheet

1.6 PIR SENSOR

El PIR⁷ Sensor, **Figura 7**, es un dispositivo electrónico modular que detecta cualquier tipo de movimiento y permite un amplio rango de detección gracias a su lente de Fresnel, dando como salida una señal digital al pin 2, requiere una alimentación de 3.3v a 5v en su pin 1, mientras el pin 3 debe ir conectado a tierra ver **Anexo 14**.



Figura 7.PIR SENSOR

Fuente:DataSheet

1.6.1 Características del PIR SENSOR

La **Tabla 6**, describes las características más importantes del sensor.

PARÁMETROS	VALORES
Rango de detección	6m
Salida	1/0
Tamaño	Pequeño
Alimentación	3.3 a 5 VDC
Corriente de operación	>=3 mA

Tabla 6. Características del PIR SENSOR

Fuente: Los Autores

⁷Sensor infrarrojo pasivo

1.7 MICROCONTROLADOR

El microcontrolador, no es más que un circuito integrado que posee alta escala de integración y en su interior posee algunas características de un computador.

- CPU (Unidad de procesamiento Central).
- Memoria RAM (Memoria de acceso aleatorio)
- EEPROM (ROM programable, borrable)
- Entradas y salidas.

Estos microcontroladores no tienen fuente interna por lo que requieren de una alimentación externa, necesita configurar la señal externa de oscilación para su funcionamiento, tienen puertos de entrada y salida, memoria, temporizadores **Figura 8.** (Reyes 2006).

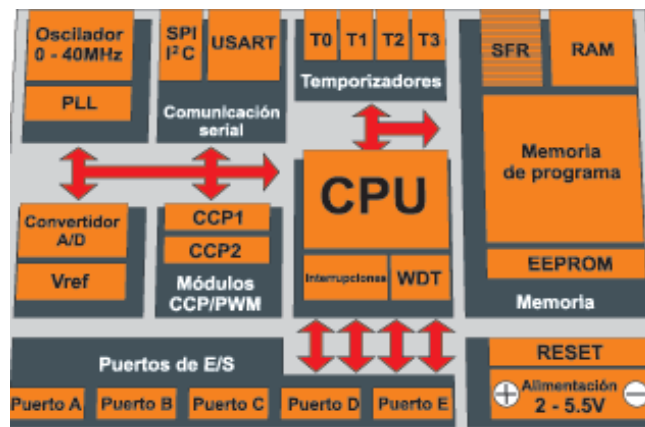


Figura 8. Representación esquemática de un microcontrolador
Fuente: <http://perso.wanadoo.es/pictob/microprg.htm>

1.7.1 PIC 16F877A

El PIC 16F877A, **Figura 9**, posee 40 pines de E/S, dispositivo de 8 bits, tiene una alta velocidad, dispone de un oscilador interno de 8 MHz programable, bajo consumo de potencia, alimentación en bajo voltaje (5v) y dispone de herramientas para su programación.

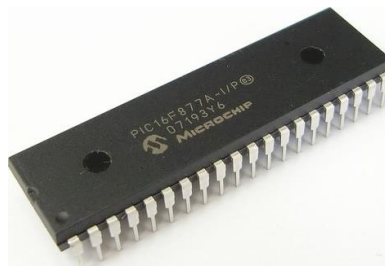


Figura 9. PIC 16F877A
Fuente: Los Autores

1.7.2 Características

La **Tabla 7**, muestra las características más relevantes del microcontrolador.

PARÁMETROS	VALORES
Frecuencia de operación	DC-20MHz
Memoria Programable	8 KB
RAM	368
Memoria EEPROM	256
Numero de pines	40
Puertos	A,B,C,D
Timers	3
Canales Analógicos	6
Canales de comunicación	2
Interrupciones	15
Set de instrucciones	35
Puertos PWM	2
Reset	PWRT-OST

Tabla 7. Características del 16F877A
Fuente: DataSheet

1.7.3 Comunicación USART

Los microcontroladores tienen incorporado el módulo USART (Transmisor – Receptor), que realiza comunicaciones con otros dispositivos como computadoras o entre microcontroladores utilizando el protocolo RS232 **Figura 10**.

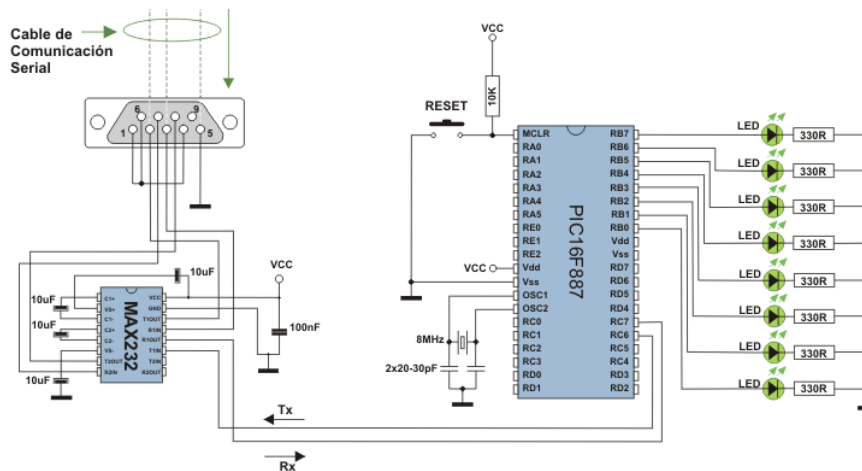


Figura 10. Comunicación Usart
Fuente: Los Autores

1.7.4 ATmega 328P

El ATmega 328P, **Figura 11**, es un circuito integrado de alto rendimiento con una capacidad de memoria que puede leer/escribir al mismo tiempo, bajo consumo de potencia, alimentación entre 1.8y 5.5vdc, posee 28 pines de E/S, puede realizar programaciones de comunicaciones Usart, interrupciones internas y externas, lectura de señales analógicos, PWM⁸, etc. a una buena velocidad de proceso.

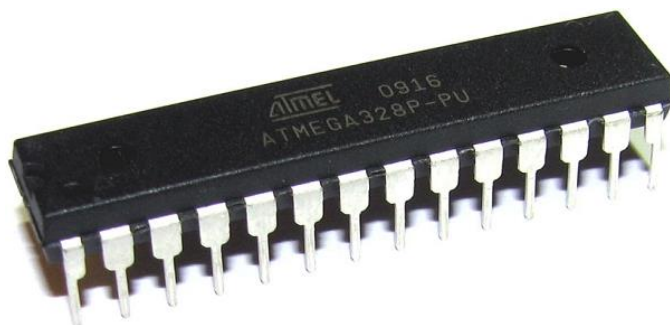


Figura 11. ATmega 328P

Fuente: Los Autores

1.7.5 Características

Las características más importantes del microcontrolador se describen en la **Tabla**

8.

PARÁMETROS	VALORES
Voltaje de operación	5V
Oscilador externo	hasta 20 MHz
EEPROM	1 kB
USART	Full duplex
canales PWM	6
Memoria de programa flash	32 kB
Pins Digital I/O	14
Pins analógica	8
SRAM	2 kB
Velocidad max	CPU 20 MIPS
ADC	10 bits y 6 canales
Serial Interface	2-Wire, I2C compatible
interrupción	24
Corrientes de operación	40mA
CPU	8-bit AVR

Tabla 8: Características del ATmega 328P

Fuente: DataSheet

⁸Modulación por ancho de pulso

CAPITULO II

2 METODOLOGÍA

Este capítulo presenta la metodología utilizada para el desarrollo del proyecto de investigación.

2.1 Tipo de estudio

El tipo de estudio que se va realizar es Empírica, Cuasi Experimental, y su metodología es la medición, esta investigación propone soluciones a una situación determinada que implica Investigar, Diseñar, Explicar, e Implementar.

Empírica: Para la realización de esta investigación se utilizó este tipo de investigación, porque está basado en experimentos y observaciones, mediante los cuales se puede realizar la comprobación de la hipótesis.

Cuasi experimental: Al aplicar este tipo de investigación se expone el entrenador y el manual de usuario a la disposición de los estudiantes para que puedan realizar sus prácticas y se pruebe la hipótesis.

Medición: La metodología que se utiliza para la investigación es la medición, porque se va usar, cuantificar el tiempo de armado.

2.2 POBLACIÓN Y MUESTRA

2.2.1 Población

El proyecto será implementado en la escuela de electrónica y telecomunicaciones de la Universidad Nacional de Chimborazo exclusivamente para los estudiantes de sexto semestre para que puedan realizar sus prácticas de Microcontroladores aplicadas a las comunicaciones. Para el estudio y el análisis se realizó el armado de un circuito con/sin el entrenador con los estudiantes, para determinar el tiempo en el desarrollo de las prácticas

2.2.2 Muestra

Debido a la menor cantidad de población se toma como muestra el total de los alumnos.

$N=n=25$

2.3 Hipótesis

“Diseño e implementación de un entrenador de Microcontroladores para comunicación ZigBee, mejora el aprendizaje y facilita el desarrollo de prácticas en el área de las comunicaciones para los estudiantes de electrónica y telecomunicaciones”

La hipótesis planteada es de tipo correccional, ya que involucran dos variables y declara que el sistema domótica diseñado e implementado, mejorara el tiempo en el desarrollo de prácticas.

2.4. OPERACIONALIZACION DE VARIABLES

La **Tabla 9**. Muestra la operacionalización de las variables independiente y dependiente.

VARIABLE	CONCEPTO	CATEGORIA	INDICADOR	TECNICAS/INSTRUMENTOS
VARIABLE INDEPENDIENTES Módulos complementarios de comunicación ZigBee.	Es un estándar que define un conjunto de protocolos para el armado de redes inalámbricas de corta distancia y baja velocidad de datos. Opera en las bandas de 868 MHz, 915 MHz y 2.4 GHz y puede transferir datos hasta 250Kbps.	<ul style="list-style-type: none"> • Investigación y desarrollo • Implementación de nuevas tecnologías • Telemática y telecomunicaciones 	<ul style="list-style-type: none"> ➤ cambio ➤ eficiencia ➤ administrable 	<ul style="list-style-type: none"> ➤ Manual de prácticas
VARIABLE DEPENDIENTE Entrenador de microcontroladores	El entrenador es un dispositivo electrónico que sirve para el desarrollo de prácticas en el área de Microcontroladores.	<ul style="list-style-type: none"> • confiabilidad • económico • calidad • servicio 	<ul style="list-style-type: none"> ➤ tranquilidad ➤ comodidad ➤ reducción de costos 	<ul style="list-style-type: none"> ➤ Observación ➤ Análisis de los componentes. ➤ Medición de tiempo

Tabla 9. Operacionalización de las variables independiente y dependiente.

Fuente: Los Autores

2.5 PROCEDIMIENTOS

Describe los pasos que se realiza durante el desarrollo, los mismo que serán ampliamente explicados, diseños de circuitos electrónicos, diseño de placas en PCB. La práctica aplicada a la domótica con los entrenadores formando una topología mesh.

2.5.1 Diseño de circuito y PCB de comunicaciones

Para este diseño se eligió dos módulos importes de comunicaciones dentro de las redes WPAN (Bluetooth, y Xbee).

2.5.1.1 Diseño de circuito electrónico de comunicaciones

El diseño de esta placa consta de tres circuitos fundamentales, circuito de alimentación, circuito Xbee y circuito Bluetooth. El circuito de alimentación de la **Figura 12**, tiene un regulador de voltaje LM 7805 y LM7833 que incorpora capacitores para filtrar las corrientes parasitas y ruido eléctrico, dando una salida un voltaje de 5V y 3.3V respectivamente ver **Anexo 1**.

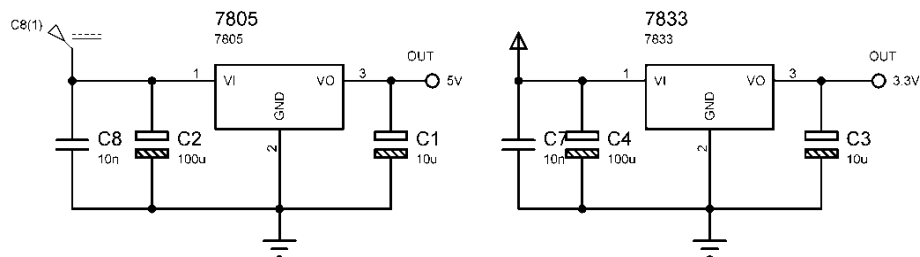


Figura 12.Fuentes de alimentación
Fuente: Los Autores

El módulo Xbee consta de 20 pines, requiere una alimentación de 3.3V ver **Anexo 10**, necesita un divisor de voltaje extra hacia el pin 3 del módulo, este divisor se utiliza cuando se tiene conexión con cualquier tipo de microcontrolador, dicho dispositivo envía 5V, tiene conectado al pin 1 un capacitor de 100nf para filtrar el ruido eléctrico e incorpora leds de encendido del módulo y Recepción de datos ver **Figura 13**.

- Calculo de la resistencia para el divisor de voltaje

Datos:

$V_{pin}=5v$

$V_{xbee}=3.3$

$$R_a = 10\text{kohm}$$

$$R_b = ??$$

$$V_{xbee} = \frac{V_{pin} * R_b}{R_a + R_b}$$

$$V_{xbee} * (R_a + R_b) = V_{pin} * R_b$$

$$V_{xbee} * R_a + V_{xbee} * R_b = V_{pin} * R_b$$

$$V_{xbee} * R_a = V_{pin} * R_b - V_{xbee} * R_b$$

$$V_{xbee} * R_a = (V_{pin} - V_{xbee})R_b$$

$$R_b = \frac{V_{xbee} * R_a}{V_{pin} - V_{xbee}}$$

$$R_b = \frac{3.3v * 10\text{kohm}}{5v - 3.3v}$$

$$R_b = 19.411 \approx 20\text{kohm}$$

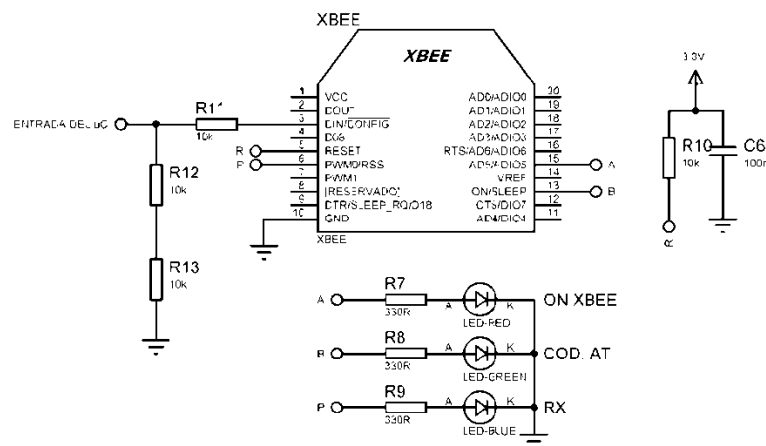


Figura 13. Circuito de módulo Xbee
Fuente: Los Autores

El módulo Bluetooth, **Figura 14**, tiene incorporado dos resistencias pull-up de 10KΩ que sirve para evitar la corriente excesiva fluya a través del circuito cuando necesita un nivel lógico bajo, un capacitor de 100nf para eliminar el ruido.

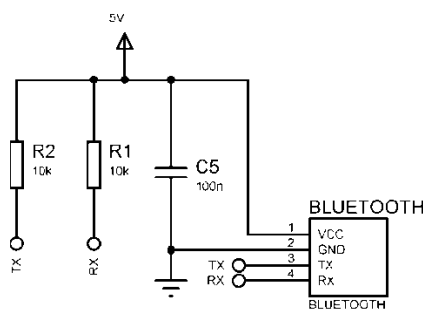


Figura 14. Circuito de módulo Bluetooth
Fuente: Los Autores

2.5.1.2 Diseño de PCB de comunicaciones

ARES⁹ es un software incorporado de Proteus que sirve para la edición y ruteo avanzado para el diseño de placas.

Para realizar el rutado de PCB¹⁰, seleccionar el trazado de bordes de placa con dimensiones de 10*10 cm, poseionar los componentes diseñados en ISIS de una manera ordenada para optimizar espacios entre elementos, **Figura 15**, configuración de pistas a T30, T40 y V60 para la fuente de alimentación, T25, T30 y V40 para las señales de entrada y salida, de esta manera finalizando el ruteo de forma automática como se muestra en la **Figura 16**.

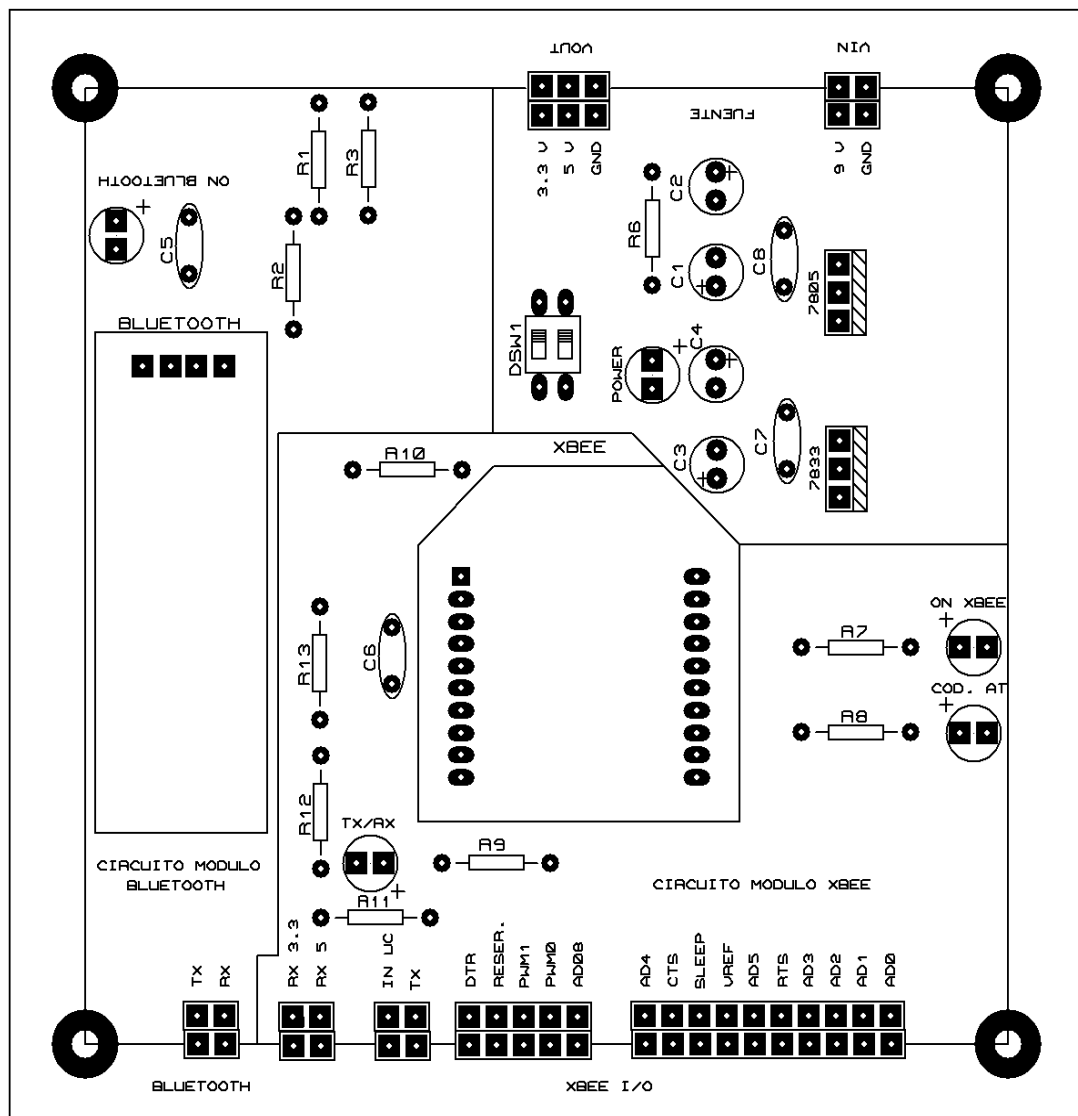


Figura 15. Posicionamiento de elementos, placa de comunicaciones
Fuente: los Autores

⁹Software de edición y ruteo avanzado

¹⁰Sensor infrarrojo

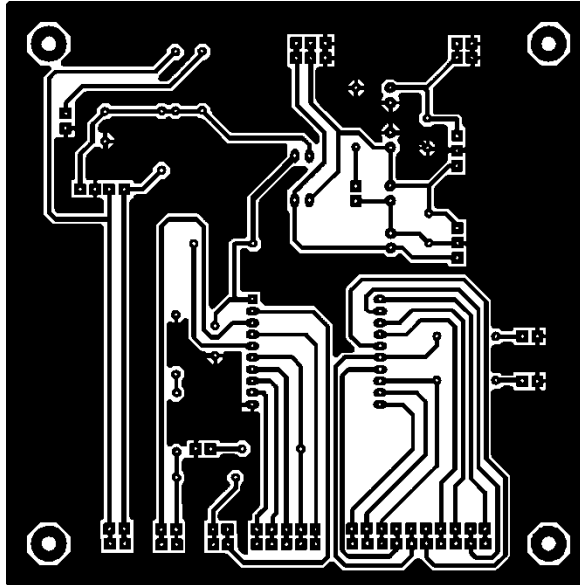


Figura 16. Diseño de pistas, placa de comunicaciones
Fuente: Los Autores

Resulta extremadamente útil disponer información tridimensional, porque interesa conocer la altura y la separación real de cada uno de los componentes de diseño o bien con fines estéticos para adornar de mejor forma la presentación de la placa final, **Figura 17.**

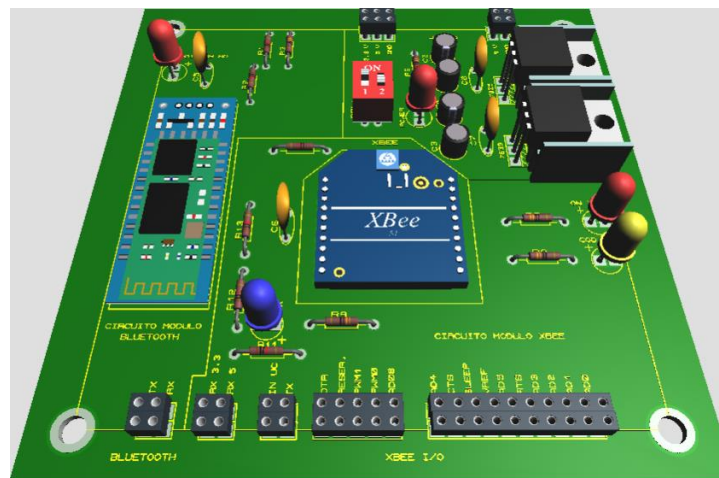


Figura 17. Módulo Xbee en 3D, placa de comunicaciones
Fuente: Los Autores

2.5.2 Diseño de circuito y PCB de sensores

Los sensores son dispositivos eléctricos o mecánicos capaces de detectar magnitudes físicas o químicas y transformarlas en señales eléctricas para realizar diferentes funciones de acuerdo a sus aplicaciones. Está dividida en dos circuitos: Circuito uno, Sensores LM35, MLX 90614, MPU 6050, encoder y el circuito integrado DS1307.

Circuito dos, sensores MQ-3, PIR SESNOR, HC-SR04, ACS712 y DTH11.

2.5.2.1 Diseño de circuito uno de sensores

Esta placa está conformada por cuatro sensores y un circuito integrado DS1307, ver **Anexo 2**.

Al LM35 **Figura 18**, se implementa un seguidor de voltaje para acoplar impedancias, un circuito amplificador no inversor con el LM358 para amplificar la salida de voltaje dandouna ganancia decuatro.

- Calculo de ganancia del amplificador no inversor.

$$G = 1 + \frac{R2}{R1}$$

$$G = 1 + \frac{12K\Omega}{3.9K\Omega}$$

$$G = 1 + 3,07$$

$$G \cong 4$$

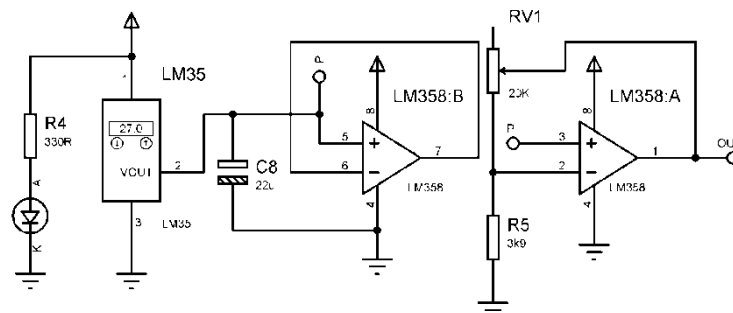


Figura 18.Circuito de sensor LM35
Fuente: Los Autores

El sensor ECG3100, **Figura 19**, se alimenta con 5V, tiene conectado una resistencia de 330Ω al pin 3 para limitar el paso de corriente al led.

- Cálculo del valor de resistencia.

$$V = R * I$$

$$R = \frac{V}{I}$$

$$R = \frac{5V}{15mA}$$

$$R = 333.33\Omega$$

$$R = 330\Omega$$

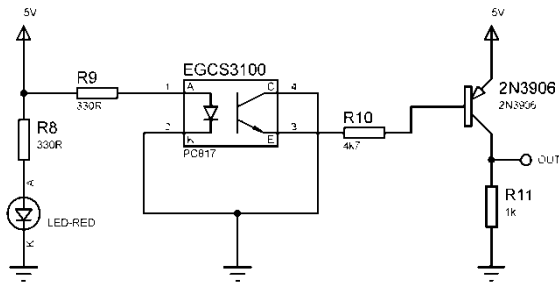


Figura 19.Circuito de sensor encoder
Fuente: Los Autores

El DS1307, **Figura 20**, es un circuito integrado que requiere dos fuentes de alimentación, 5v para el funcionamiento del circuito y 3.3vdc al Pin 3 para mantener funcionando el reloj calendario mientras el circuito principal está apagado ver **Anexo 13**, un led segundero, un cristal de 32.768KHz y dos resistencias contestadas al SCL y SDA de 100kΩ para generar un estado lógico alto.

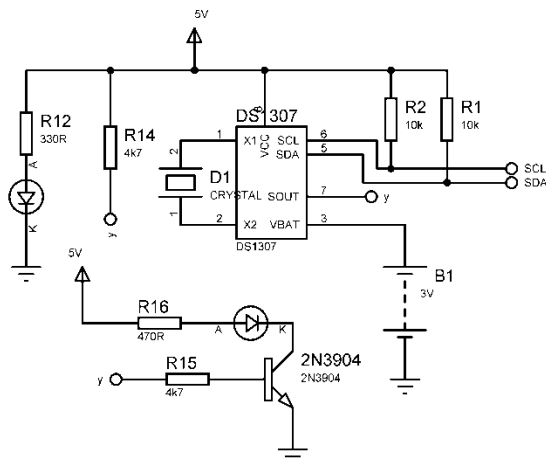


Figura 20.Circuito Integrado DS1307
Fuente: Los Autores

El MPU6050, **Figura 21**, es un sensor de movimiento que posee un acelerómetro y un giroscopio que maneja comunicación I²C y no necesita de ningún circuito extra para su funcionamiento excepto una alimentación de 5vdc.

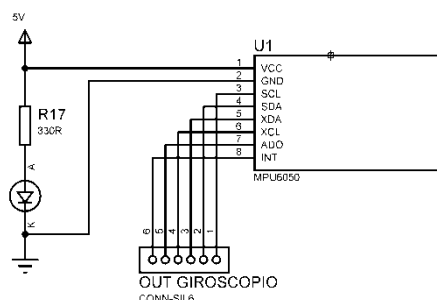


Figura 21.Circuito desensor MPU6050
Fuente: Los Autores

El MLX90614, **Figura 22**, es un sensor infrarrojo de temperatura que tiene conectado resistencias de $10K\Omega$ al SCL y CDA para generar un estado lógico alto.

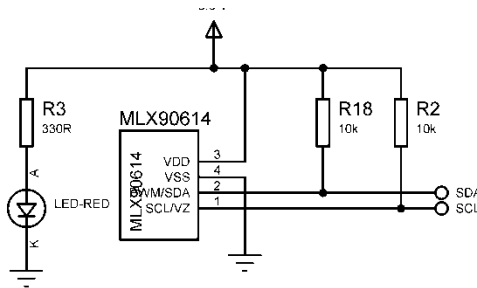


Figura 22. Circuito de MLX90614
Fuente: Los Autores

2.5.2.2 Diseño de PCB uno de sensores

Las configuraciones necesarias para el diseño de esta placa se describen en el numeral 2.5.1.2.

Dando como resultado la **Figura 23** y **Figura 24**.

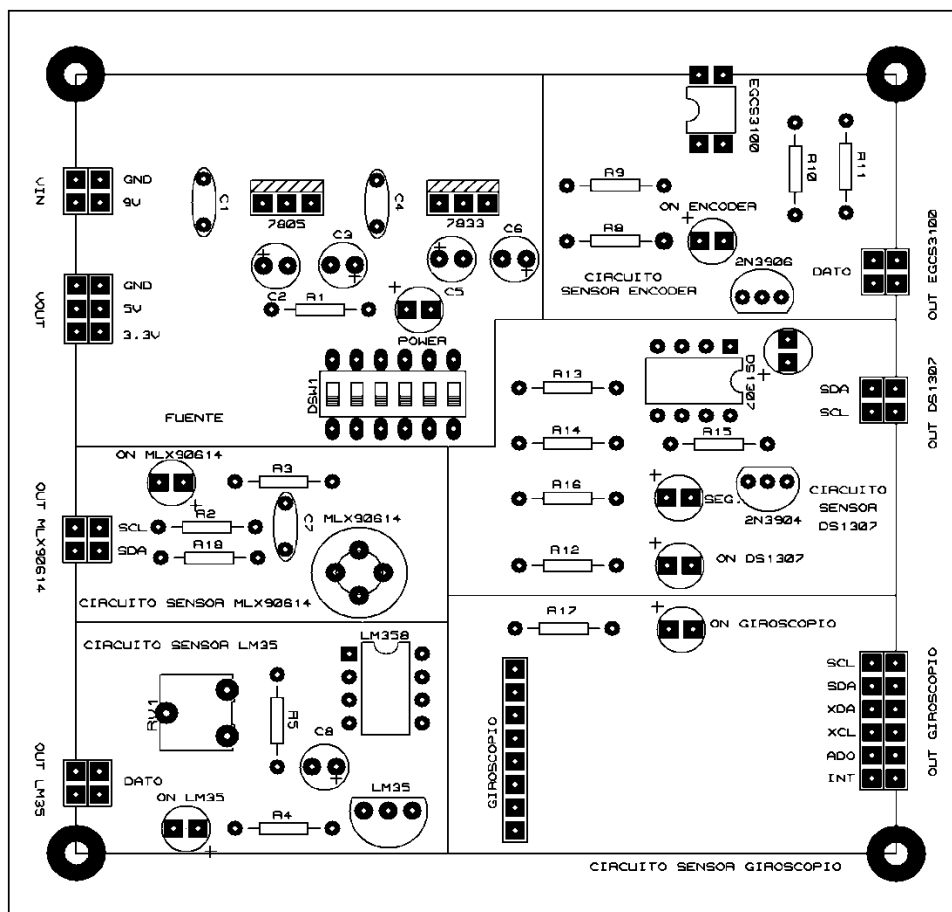


Figura 23. Posicionamiento de elementos, placa uno
Fuente: Los Autores

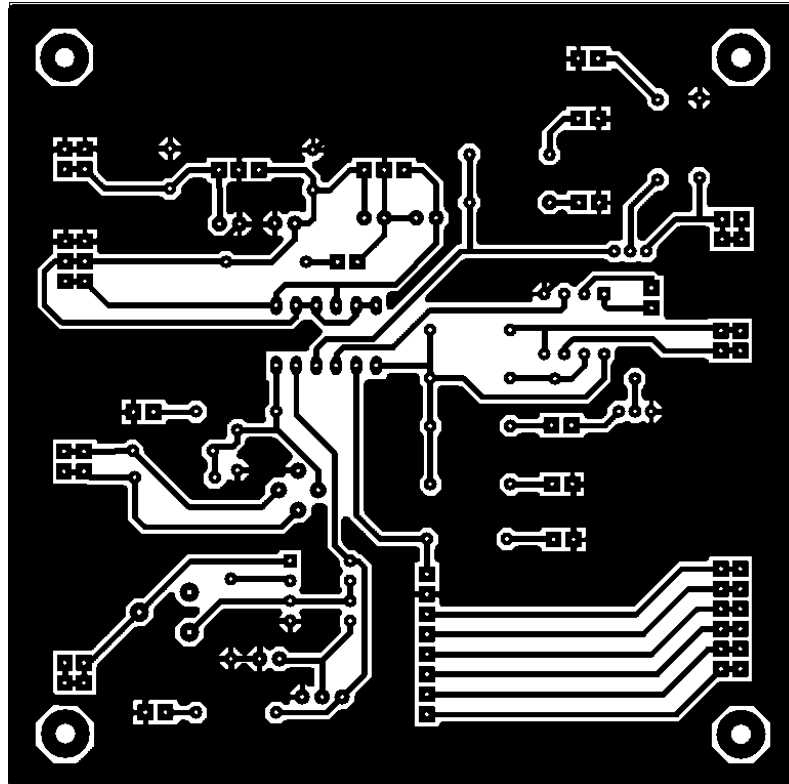


Figura 24. Diseño de pistas, placa uno
Fuente: Los Autores

La librería de Proteus no cuenta con todos los componentes en tercera dimensión por lo que es necesario diseñar utilizando el software SketchUp que sirve para el diseño de cualquier tipo de elementos en 3D, se muestra la **Figura 25**. La placa en tercera dimensión.

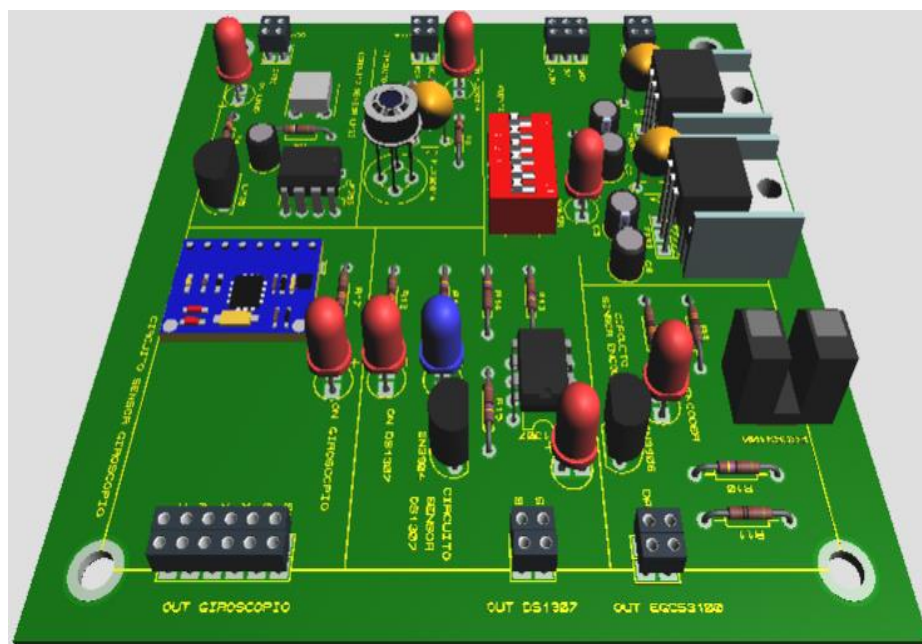


Figura 25. Placa uno de sensores en 3D
Fuente: Los Autores

2.5.2.3 Diseño de circuito dos de sensores

Los sensores se alimentan con 5v, por lo tanto es necesario diseñar un sistema electrónico que permita regular el voltaje, **Figura 12.**

El **DHT11** es un sensor digital de temperatura y humedad, tiene conectada una resistencia de **4.7 K Ω** de manera que cuando está inactivo, su estado es alto y no permite las fluctuaciones eléctricas, se alimenta con 5v al pin uno. **Figura 26.**

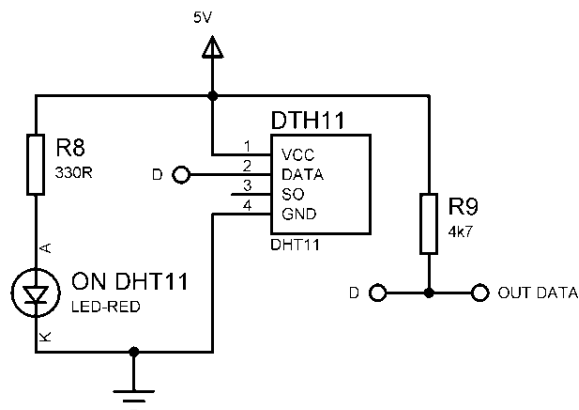


Figura 26. Circuito de sensor temperatura y humedad
Fuente: Los Autores

La **Figura 27.** Muestra el circuito electrónico del sensor de corriente **ACS712**, soporta 20A AC/DC en sus terminales, se alimenta con 5v y tiene conectado un led de power.

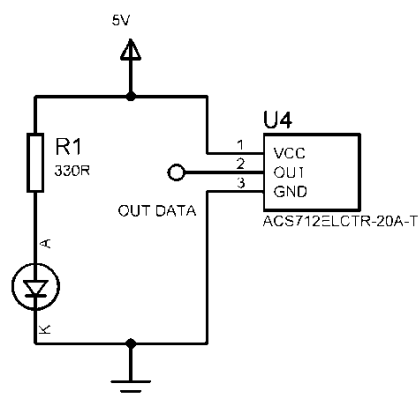


Figura 27. Circuito del sensor de corriente
Fuente: Los Autores

El circuito electrónico del PIR SENSOR sirve para detectar cualquier tipo de movimiento, tiene conectado al pin dos una resistencia de 10k Ω Pull-Down generando un estado lógico bajo, cuando el módulo está inactivo se alimenta con 5v, **Figura 28.**

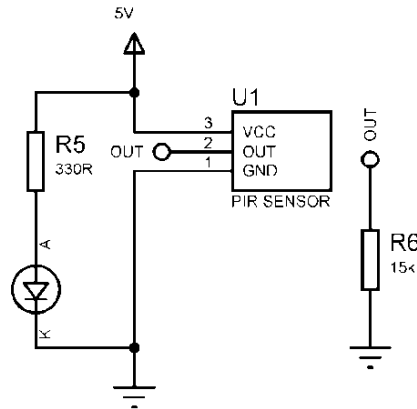


Figura 28. Circuito de sensor de movimiento
Fuente: Los Autores

El **HC-SR04** es un sensor de distancias, capaz de detectar objetos en un rango de 2 a 450 cm, se alimenta con 5v, **Figura 29.**

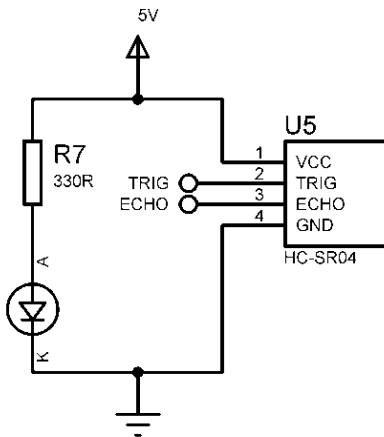


Figura 29. Circuito de sensor ultrasónico
Fuente: Los Autores

El **MQ-3**, es un sensor de alcohol, que se conecta una resistencia de carga de 10KΩ al pin B1, se alimenta con 5v a los pines A1, H1, A2, y su salida de dato es el pin B2, **Figura 30.**

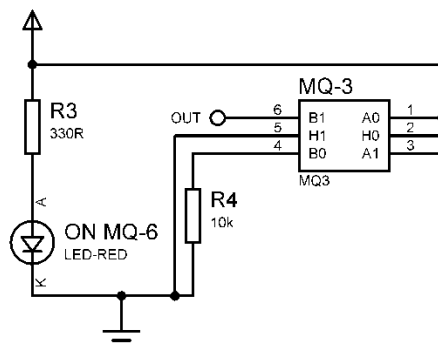


Figura 30. Circuito de sensor de alcohol
Fuente: Los Autores

2.5.2.4 Diseño de PCB dos de sensores

La configuración para el ruteo de pistas se describe en la **Tabla 10**.

PISTAS	DIMENSIONES
Power	T30, T40 y V60
señale	T25, T30 Y v40

Tabla 10. Dimensiones de pistas

Fuente: Los Autores

El consume de corriente del sensor **MQ-3** es de aproximadamente 167mA, por lo tanto requiere un grosor de pistas de mayor dimensión, T40.

Aplicando todas estas dimensiones para el ruteo de las pistas da con resultado la **Figura 31**. Y **Figura 32**.

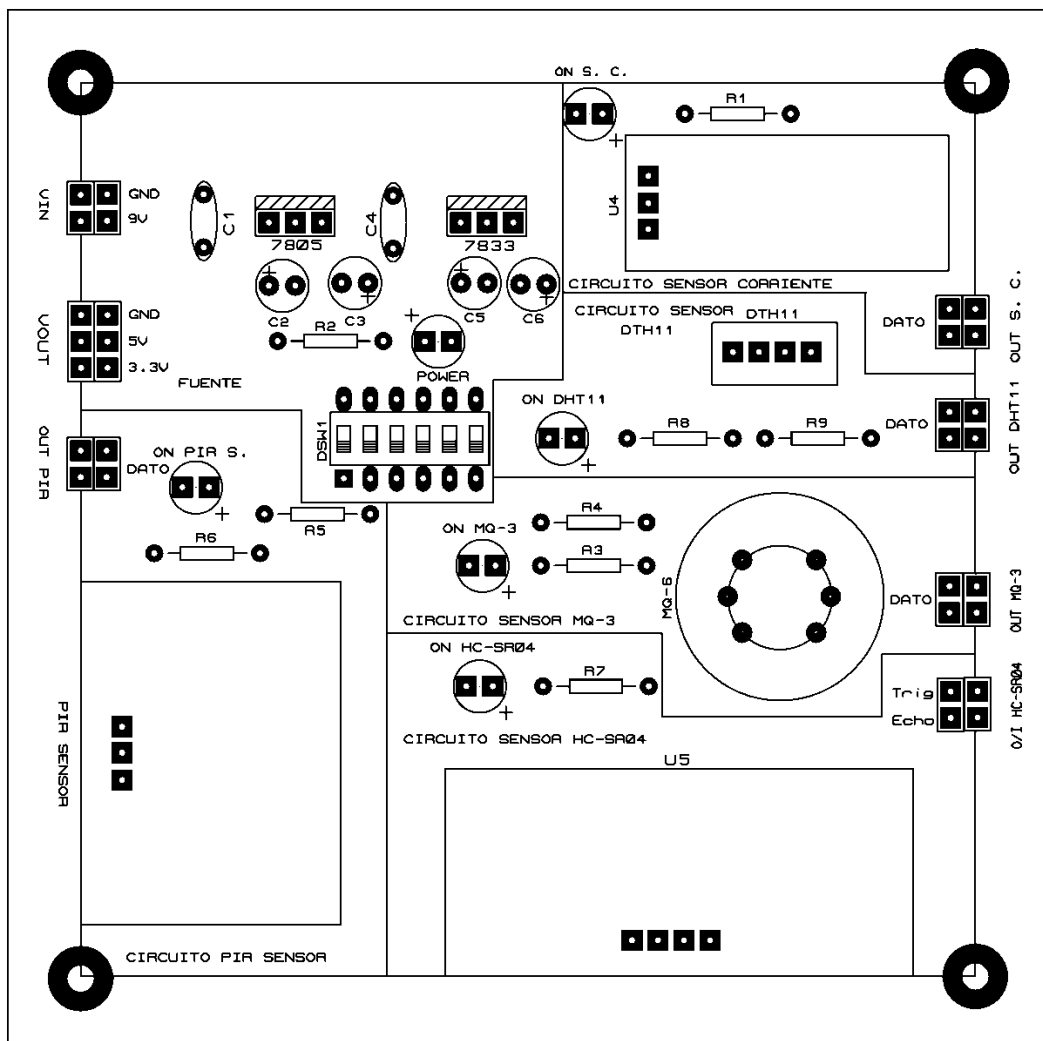


Figura 31. Posicionamiento de elementos, placa dos

Fuente: Los Autores

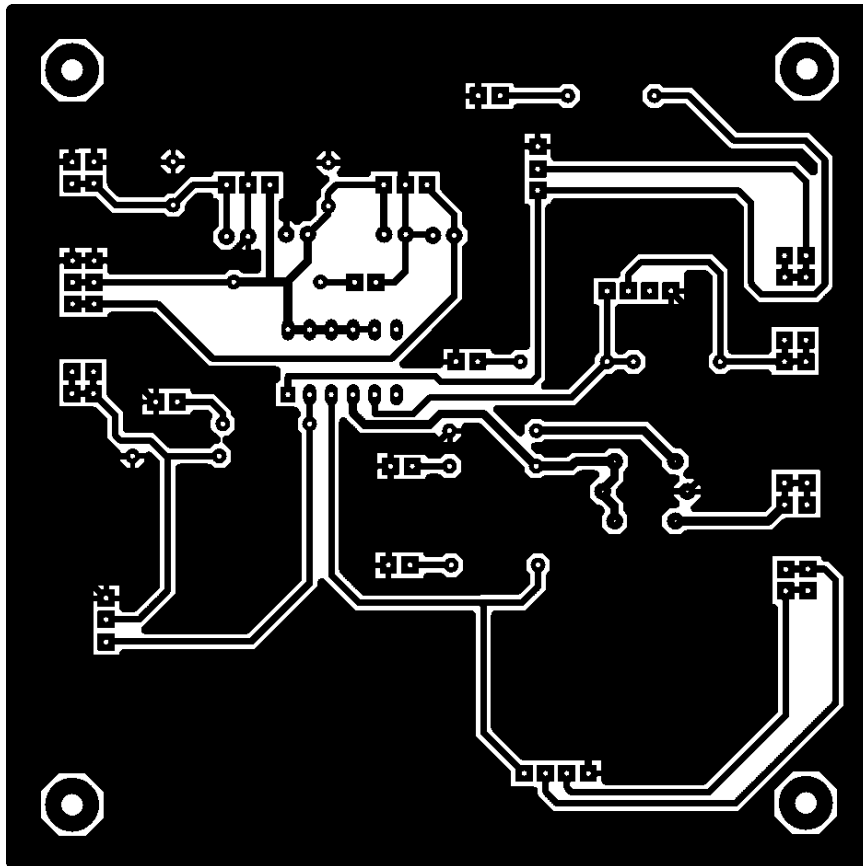


Figura 32. Diseño de pistas, placa dos
Fuente: Los Autores

El software Proteus permite la activación de librería en tercera dimensión para una mejor visualización ver **Figura 33**.

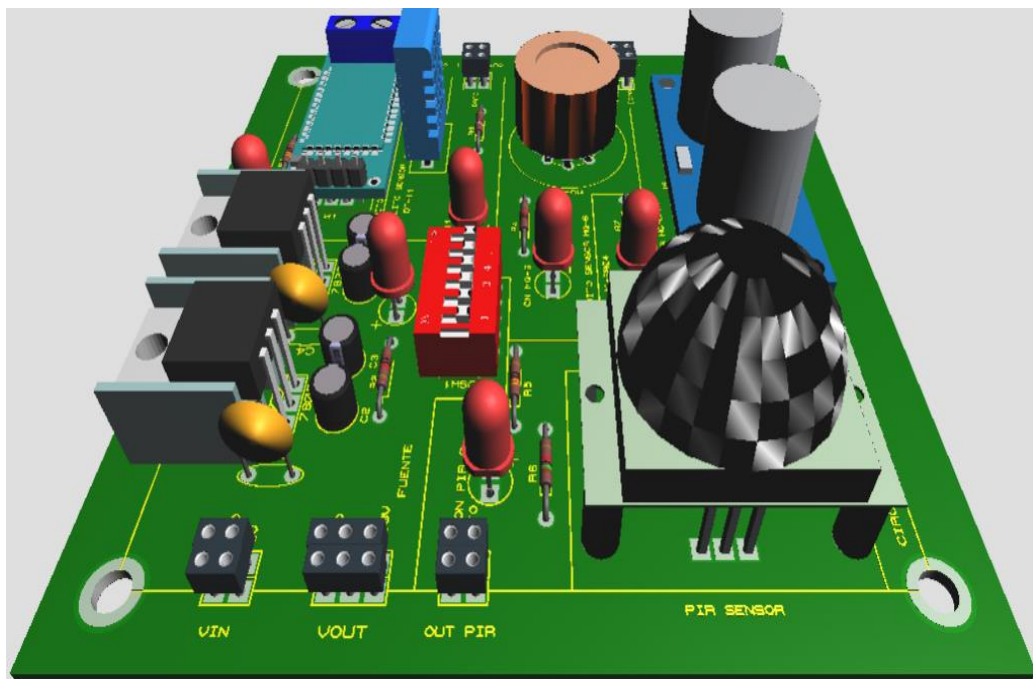


Figura 33.Placa dos de sensores en 3D
Fuente: Los Autores

2.5.4 Diseño del sistema domótico con entrenador de comunicaciones

Este diseño es una aplicación a la domótica, que consta de tres partes fundamentales: coordinador, Router A y Router B, formando una topología mesh, con sus respectivos dispositivos de comunicación, programación, sensores y actuadores que presenta un diagrama de bloques en la **Figura 34**.

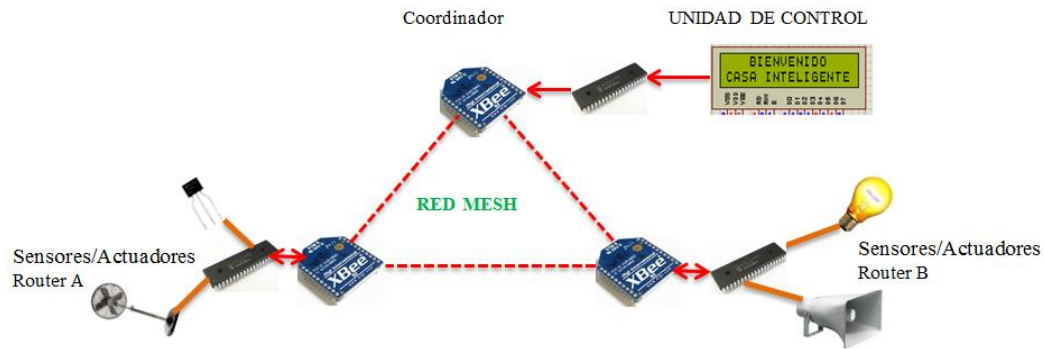


Figura 34. Diagrama esquemático del proyecto
Fuente: Los Autores

2.5.4.1 Diseño del coordinador

La **Figura 35**, muestra un diagrama general del funcionamiento del coordinador.

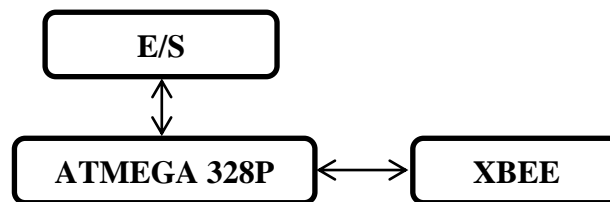


Figura 35. Diagrama de bloques del nodo coordinador
Fuente: Los Autores

E/S: El periférico E/S, permite visualizar e ingresar la información del mismo coordinador como también de los módulos que están conectados a la red.

ATmega 328P: Realiza el procesamiento de información de los nodos que están conectados a su red y sus periféricos de entrada y salida.

Xbee: Es el encargado de transmitir y recibir información con los demás nodos.

- **Diseño eléctrico del coordinador**

El circuito coordinador, requiere dos alimentaciones: uno de 5v para alimentar el microcontrolador y otros componentes, 3.3v para alimentar el módulo de comunicaciones.

El ATmega 328P **Figura 36.** Tiene conectado una resistencia de 10K Ω al Pin uno para el MClear, un lcd de 16x2 al puerto C, cinco pulsadores a los Pines (4, 5, 6, 11,12) con resistencias de 4.7K Ω para eliminar rebotes producidas por los pulsadores, Pin (2/Rx) y Pin (3/Tx) conectados al módulo Xbee ver**Figura 36.** Pin (14/SCL) y Pin (15/SDA) conectado al DS1307 **Anexo 7.**

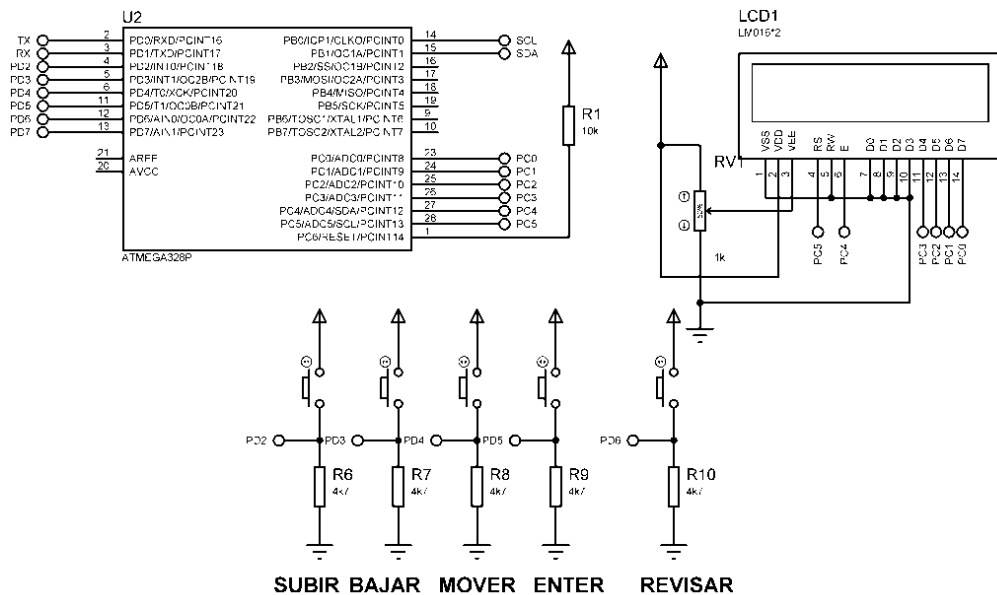


Figura 36. Circuito del coordinador
Fuente: Los Autores

- **Programación del ATmega 328P**

El flujo grama de la **Figura 37.** Proporciona un interfaz gráfica de la programación ATmega 328P, definición de variables **Tabla 11**, lectura de los registro del DS1307, salida de datos al lcd, programación de datos en la memoria EEPROM del control automático mediante los pulsadores, transmisión y recepción de datos a través de los módulos Xbee **Anexo 4**, para ver programa completo.

DESCRIPCIÓN	VARIABLES
DS1307	Hora, Minu, Segu, Dias, Mes y Anio
EEPROM	B, I
INTERRUPCION SERIAL	Texto11, K, W, C1, D1, F1, E1

Tabla 11.Declaración de variables coordinador
Fuente: Los Autores

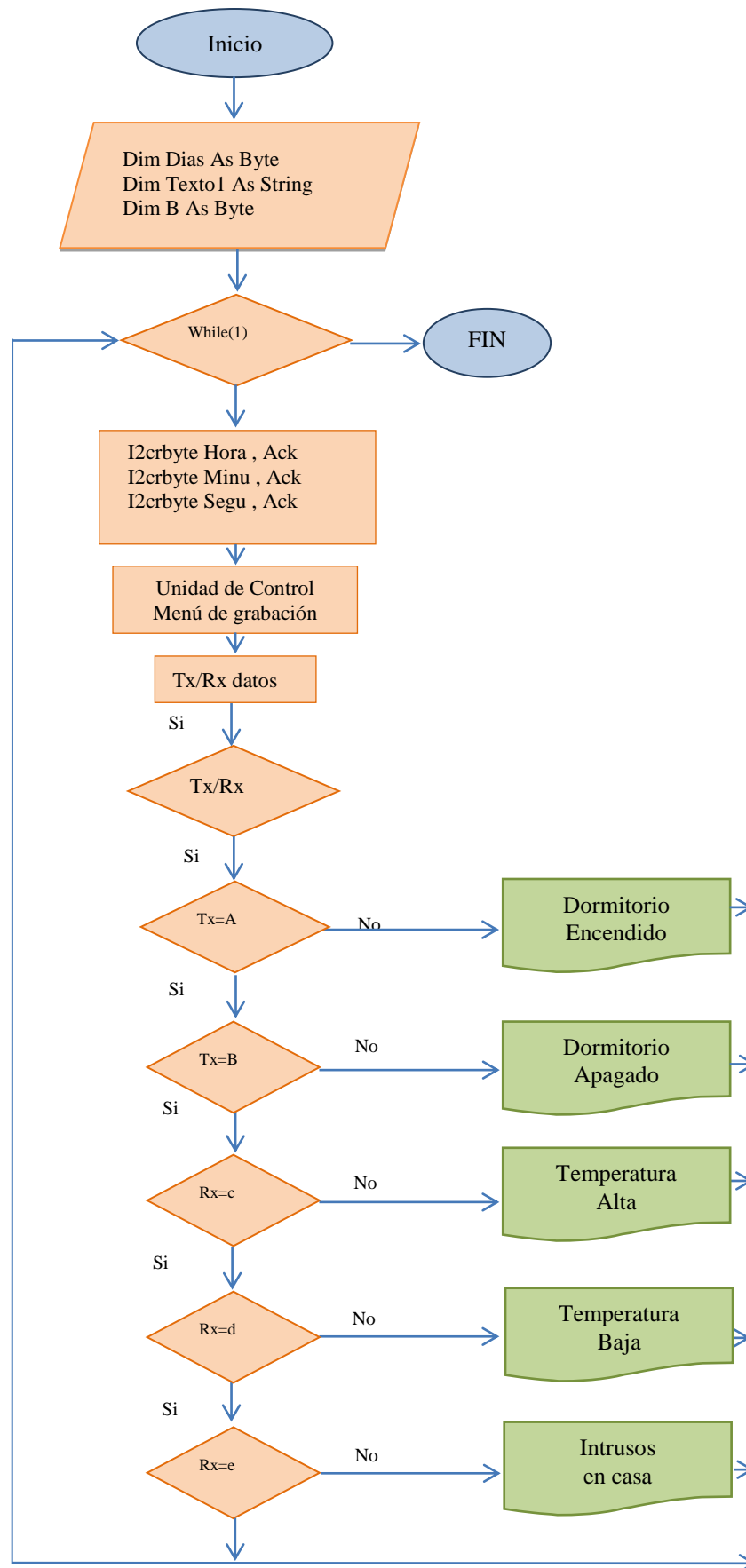


Figura 37. Flujo grama, coordinador
Fuente: Los Autores

- **Configuración del Xbee como coordinador**

Este sistema utiliza una topología mesh, por lo tanto se configura al coordinador utilizando el software X-CTU ver **Anexo 15**, donde se ingresan los valores descritos en la **Tabla 12**.

INDICADOR	NOMBRE	VALOR
NI	Nodo identificador	Coordinador
ID	PAN ID	17
MY	Dirección de red	10
SD	Dirección de escaneo	3
CH	Canal de operación	11
SC	Examina solo el canal elegido	40
DH	Dirección de destino en alto	0
DL	Dirección de destino en bajo	FFFF
SH	Número de serie en alto	13A200
SL	Número de serie en bajo	

Tabla 12. Configuración del módulo Xbee coordinador

Fuente: Los Autores

2.5.4.2 Diseño del Router-A

El diagrama de bloques de la **Figura 38**, describe el funcionamiento completo del Router-A.

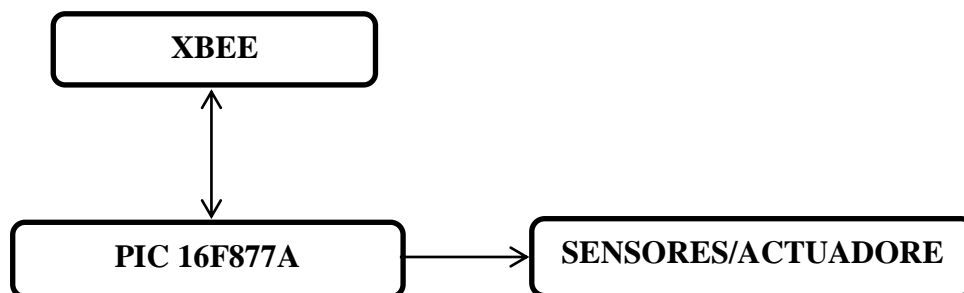


Figura 38. Diagrama de bloque del Router-A

Fuente: Los Autores

Xbee: Dispositivo encargado de tener comunicación con el resto de nodos conectados a su red.

Pic 16F877A: Es el encargado de procesar, ejecutar información de acuerdo a los datos recibidos por el módulo Xbee y sus características se observa en **Anexo 9**.

Actuadores:Dispositivo electrónico que acciona un mecanismo al recibir señalesde control proporcionado por el microcontrolador.

- **Circuito eléctrico del Router-A**

El Pic 16F877A **Figura 39**, tiene conectado un lcd 16 *2 al puerto B, un sensor de temperatura LM35 **Figura 18**, al Pin 2, capacitores de 22pf y un cristal de oscilación de 20MHz a los pines (13,14), circuito de potencia a los pines (19,20) para el control automático del calefactor y una sirena de seguridad, una resistencia de 10kΩ al Pin 1 MClear, Pin (25/Tx) Pin (26/Rx) al módulo Xbee ver **Anexo 8**.

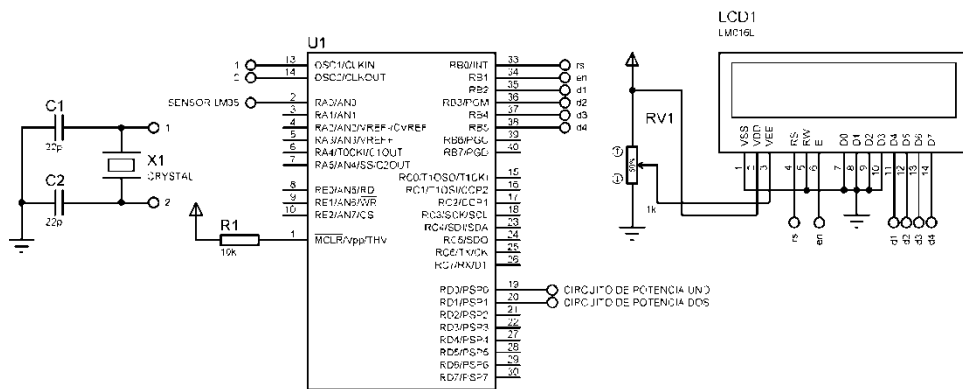


Figura 39.Circuito Router-A

Fuente: Los Autores

- **Circuito relé de estado solido**

El circuito relé de estado sólidoes un dispositivo electrónico que conmuta el paso de electricidad cuando es aplicado un pequeña corriente en sus terminales de control, tiene conectado un led de aviso de entrada de señal, un transistor 2N3904,OptoTriac MOC3023 con detección cruce por ceropara activar al Triac y evitar que regresen los picos de corriente producidos por la carga al microcontrolador, fusible y un Triac Q4015L5 dispositivo de potencia para trabajar con cargas de corriente alterna actuando como interruptor ver **Figura 40**.

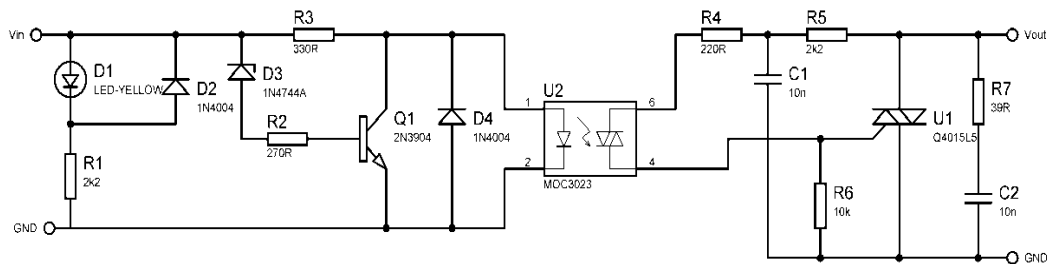


Figura 40.Circuito de potencia

Fuente: Los Autores

- Programa del Pic16F877A

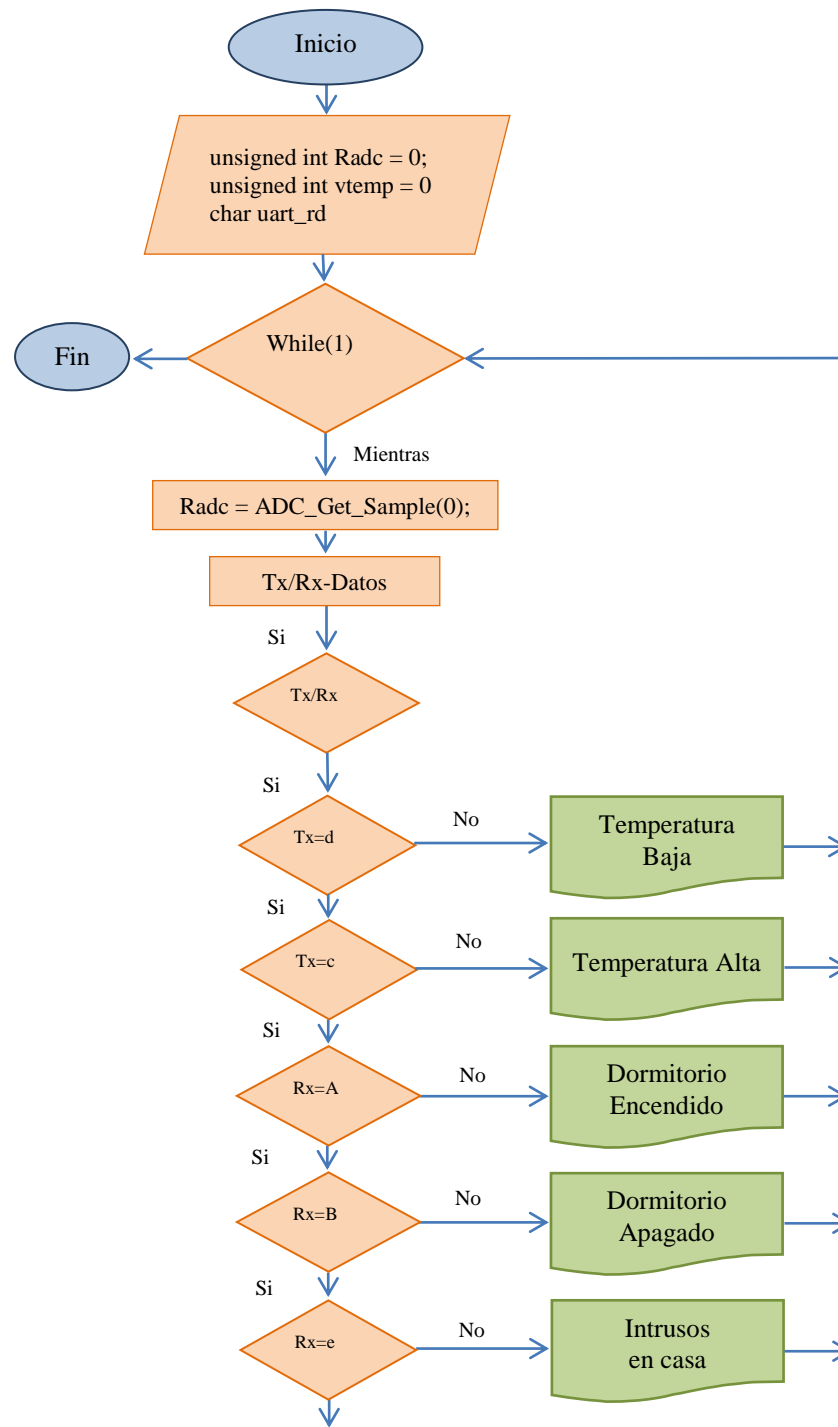


Figura 41. Flujo grama, Router-A
Fuente: Los Autores

La **Figura 41** describe el flujo grama donde muestra la descripción de variables **Tabla 13**, lectura de datos por el puerto A0 (ADC_Get_Sample), obtención del valor real de la conversión ((float) (Radc * 0.12)), para convertir de valor float a tipo char (FloatToStr (temp1, Text)) e imprimir el dato Text en el lcd,

comparación de temperatura con el comando IF para la activación y desactivación del calefactor.

Los comandos (UART_WRITE, UART_READ) para la comunicación con el coordinador y Router B para enviar el nivel de temperatura, y recibir señales detectadas por el PIR SENSOR, utilizando como interfaz de comunicaciones los módulos Xbee, programa completo **Anexo 4**.

DESCRIPCIÓN	VARIABLES
ADC/CANAL-0	Radc = 0
APUNTADOR VOLTAJE	*tempC = "000.0"
UART	uart_rd
CADENA CONVERSIÓN	Text[15]

Tabla 13.Declaración de variables Router-A.

Fuente: Los Autores

- **Configuración del Xbee como Router-A**

Al dispositivo Xbee se configura como Router-A direccionándolo al nodo coordinador y al Router-Bver la configuración completa en el **Anexo 15**.

En la **Tabla 14**se describe todos los parámetros de configuración y los valores asignados al dispositivo.

INDICADOR	NOMBRE	VALOR
NI	Nodo identificador	Router-A
ID	PAN ID	17
MY	Dirección de Red	10
SD	Duración de escaneo	3
CH	Canal de Operación	11
SC	Examina solo el canal elegido	40
DH	Dirección de destino en alto	0
DL	Dirección de destino en bajo	FFFF
SH	Número de serie en alto (Viene por defecto)	13A200
SL	Número de serie en bajo (Viene por defecto)	

Tabla 14.Configuración del módulo Xbee Router-A

Fuente: Los Autores

2.5.4.3 Diseño del Router-B

El diagrama de bloques de la **Figura 42**, describe las partes elementales del circuito Router-B.

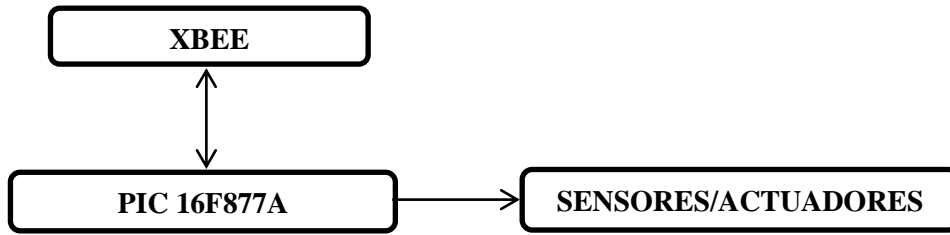


Figura 42. Diagrama de bloques del Router-B
Fuente: Los Autores

Xbee: Transmitir y recibir información de los nodos que están conectados en la red.

Pic 16F877A: Dispositivo encargado de procesar información del PIR SENSOR, Transmitir y Recibir información de los módulos conectados a la red.

Actuadores: será el encargado de realizar conmutaciones de las cargas conectadas a sus terminales.

- **Circuito eléctrico del Router-B**

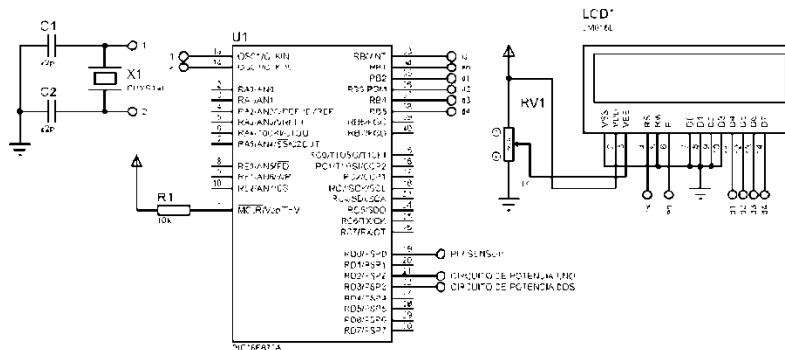


Figura 43. Circuito Router-B
Fuente: Los Autores

La **Figura 43** muestra los componentes elementales del Router-B. Está diseñado por un Pic 16F877A que se alimenta con 5v, periférico de salida lcd 16*2 conectado al puerto B, los pines (21,22) conectados al circuito de potencia para la conmutación del foco y el ventilador, una resistencia de 10KΩ al Pin uno de MClear, PIR SENSOR al Pin 19 **Figura 28**, dos capacitores de 22pf y cristal de 20MHz a los pines (13,14), los pines (25/Tx) y (26/Rx) a los módulos Xbee ver **Anexo 9**.

- **Circuito relé de estado solido**

El microcontrolador será el encargado de enviar una señal a los terminales del circuito de potencia (relé de estado sólido) **Figura 40**, para la conmutación automática del encendido y apagado del foco y el ventilador.

- **Programación del Pic 16F877A**

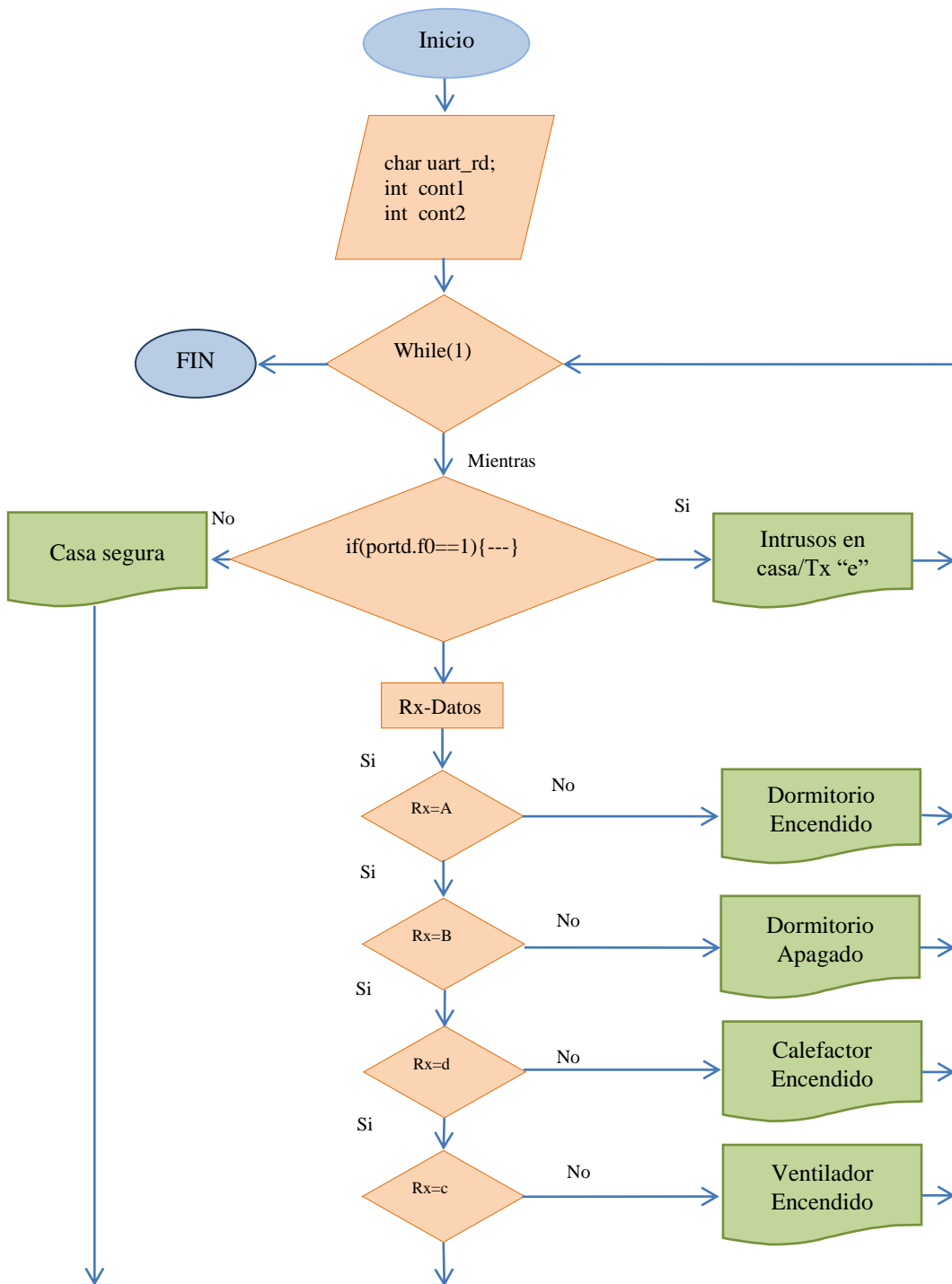


Figura 44. Flujo grama del Router-B
Fuente: Los Autores

Es importante tener una idea clara y precisa para el desarrollo del programa, utilizando el flujo grama de la **Figura 44** se tiene la siguiente lógica de programación, declarando variables **Tabla 15** para la lectura digital del PIR SENSOR, comunicación UART etc. Transmisión y recepción de datos utilizando módulos Xbee, programación completa en el **Anexo 6**.

DESCRIPCIÓN	VARIABLES
COMUNICACIÓN SERIAL	uart_rd
CADENA CONVERSIÓN	Text[4]
Contador	cont1

Tabla 15.Declaración de variables Router-B.

Fuente: Los Autores

- **Configuración del Xbee como Router-B**

Este dispositivo debe estar configurado para que pueda comunicarse con el coordinador y el Router-A.

En la **Tabla 16** se citan los parámetros con de configurado del módulo Xbee para que tenga comunicación con el resto de nodos para observar la configuración completa ver **Anexo 15**.

INDICADOR	NOMBRE	VALOR
NI	Nodo identificador	R-B/Actuador
ID	PAN ID	17
MY	Dirección de Red	10
SD	Duración de escaneo	3
CH	Canal de Operación	11
SC	Examina solo el canal elegido	40
DH	Dirección de destino en alto	0
DL	Dirección de destino en bajo	0
SH	Número de serie en alto (Viene por defecto)	13A200
SL	Número de serie en bajo (Viene por defecto)	

Tabla 16.Configuración del módulo Xbee Router-B.

Fuente: Los Autores

2.6 COMPROBACIÓN DE LA HIPÓTESIS

En el presente apartado se verifica la validez o el rechazo de la hipótesis utilizando datos estadísticos con el software SPSS.

Se realizó la práctica de una comunicación Punto a Punto utilizando módulos Xbee sin el entrenador, **Anexo 18** y con el entrenador, **Anexo 19**, realizando la lectura del sensor de temperatura LM35, **Anexo 16**.

2.6.1 Planteamiento de hipótesis

Se toman dos factores muy importantes de afirmación (H1) y negación (H0):

H0 No existe diferencia estadística significativa en el tiempo de armado, las prácticas con el entrenador de microcontroladores para la comunicación ZigBee.

H1 Existe diferencia estadística significativa en el tiempo de armado, las prácticas con el entrenador de microcontroladores para la comunicación ZigBee.

2.6.2 Establecimiento de nivel de significancia

El nivel de significancia que se establece normalmente es $\alpha=0.05$, representando un 95% de confianza.

2.6.3 Elección de la prueba de la hipótesis

Es una prueba aplicada a un mismo número de estudiantes, el armado de prácticas con el entrenador y sin el entrenador, por lo tanto es óptimo usar el método de T-Student con muestras relacionadas, su función es comparar dos mediciones de puntuación, para su debida comprobación de hipótesis.

Si P-valor $\leq \alpha$ se rechaza H_0 y se acepta H_1 .

Si P-valor $> \alpha$ se rechaza H_1 y se acepta H_0 .

2.6.3 Obtención de datos

Datos obtenidos en la Universidad Nacional de Chimborazo Facultad de Ingeniería, escuela de Electrónica y Telecomunicaciones estudiantes de sexto semestre, la **Tabla 17**. Describe los resultados del tiempo que se demora en la realización de la práctica con el entrenador y sin el entrenador.

NUMERO DE ESTUDIANTES	TIEMPO (minutos)	
	CON ENTRENADOR	SIN ENTRENADOR
1	50	100
2	48	110
3	56	98
4	57	115
5	55	99
6	44	108
7	49	105
8	50	120
9	55	99
10	48	114
11	53	102
12	56	100
13	45	101
14	48	106
15	57	108
16	44	109
17	60	105
18	45	98
19	49	109
20	57	110
21	48	107
22	59	104
23	58	103
24	55	120
25	51	115

Tabla 17.Datos estadísticos.

Fuente: Los Autores

2.6.4 Calculo estadístico utilizando software SPSS

La media efectuada en el armado de la práctica sin el entrenador es ($\bar{X} = 51.88$) y con entrenador es ($\bar{X} = 106.6$) como se muestra en la **Tabla 18**, por lo tanto existe diferencias en cuanto a la optimización de tiempo.

		Media	N	Desviación típ.	Error típ. de la media
Par 1	PRA_C_ENTRE	51,88	25	5,003	1,001
	PRA_S_ENTRE	106,60	25	6,526	1,305

Tabla 18. Estadísticos de muestras relacionadas.

Fuente: Los Autores

		Diferencias relacionadas				t	gl	Sig. (bilateral)	
		Media	Desviación típ.	Error típ. de la media	95% Intervalo de confianza para la diferencia				
					Inferior				Superior
Par 1	PRA_C_ENTRE - PRA_S_ENTRE	-54,720	8,498	1,700	-58,228	-51,212	-32,197	24	,000

Tabla 19. Prueba de muestras relacionadas.

Fuente: Los Autores

El programa SPSS arroja resultados de grado de libertad (crítica) $gl = 24$ el índice de t (calculada) $t = -32.197$ y el valor de significancia p ($p = 2.1019815869E-4$ es muy bajo y tiene a 0, aproximadamente $p = 0$) **Tabla 19.** Por lo tanto la t calculada es menor que la t crítica y el valor $p < \alpha$, por lo cual rechaza la hipótesis nula H_0 y acepta la hipótesis alternativa H_1 .

Lo que significa que existe diferencia estadísticamente significativa en el tiempo de armado de las prácticas con el entrenador de microcontroladores para la comunicación ZigBee.

CAPÍTULO III

3 RESULTADOS.

En este capítulo se detalla las pruebas realizadas entre dispositivos Xbee configurados como red mesh.

3.1 PRUEBA DE COMUNICACIÓN DE RED MESH

3.1.1 Comunicación.

Las pruebas de comunicación se realiza dentro del domicilio **Figura 45**, tomando lugares amplios como la sala o local para tener la facilidad de ubicar los módulos en cada esquina.



Figura 45. Domicilio con el sistema instalado
Fuente: Los autores

El coordinador se instaló en un extremo del local donde haya facilidad de acceso para las personas que se encargan del mantenimiento y control del dispositivo **Figura 46**.

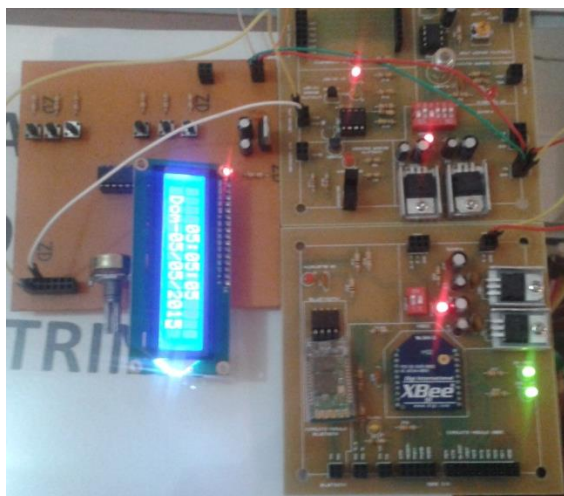


Figura 46. Circuito Coordinador
Fuente: Los Autores

El módulo Xbee coordinador será el encargado de realizar el control de conmutación del sistema de iluminación en cualquier día de la semana, en cualquier instante de tiempo, dependiendo el gusto del usuario.

El foco está conectado al ROTER-B que se encuentra en la puerta principal del local conjuntamente con el PIR SENSOR y el Ventilador **Figura 47**.

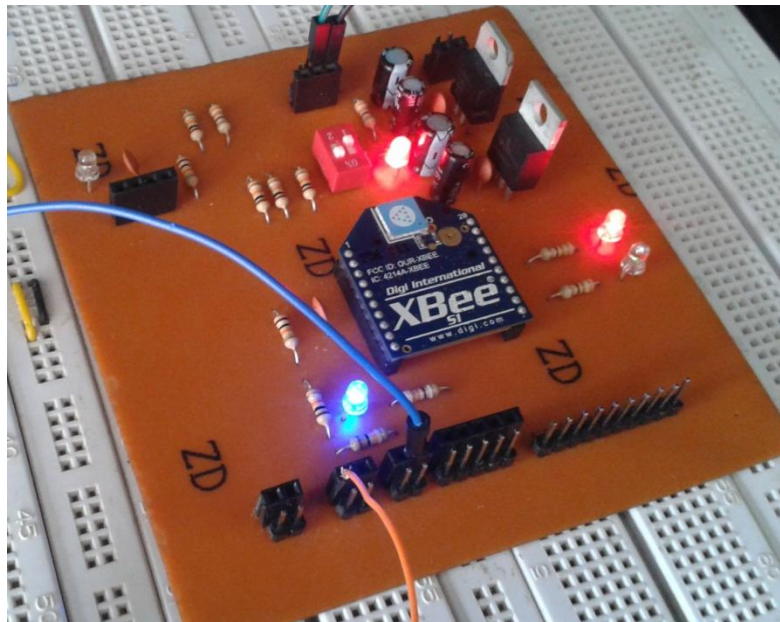


Figura 47. Dispositivo Router-B
Fuente: Los Autores

El módulo Router-A/Activadorencargado de leer el LM35 (Sensor de temperatura), realizar conmutaciones del calefactor y alarma de seguridad se encuentra ubicado en la mitad de local de aproximadamente 2m de distancia a cada módulo **Figura 48**.

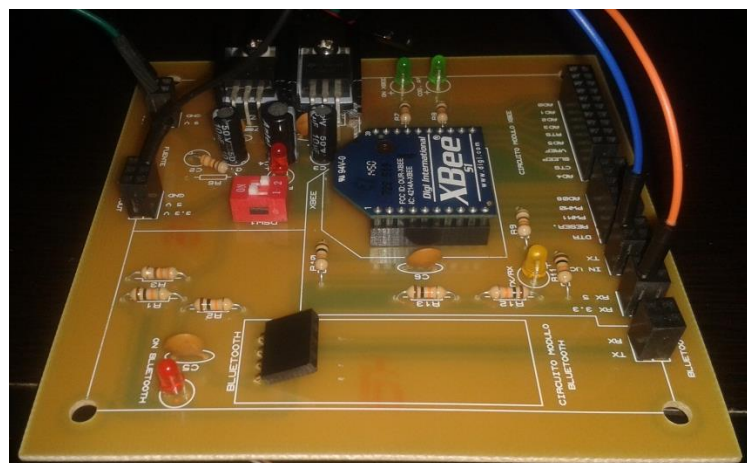


Figura 48.Dispositivo Router-A
Fuente: Los Autores

La primera prueba realizada fue de manera automática, la activación del foco **Figura 49** en el Router-B/Activador.



Figura 49. Activación del foco
Fuente: Los Autores

La siguiente prueba, fue realizada con el LM35 sensor de temperatura, conmutación del calefactor ver **Figura 50**.



Figura 50. Control, calefactor
Fuente: Lo Autores

Otra de las pruebas realizadas fue el control de seguridad del hogar, **Figura 51**, que es una prioridad para el usuario, este dispositivo se encuentra en la puerta principal del hogar y está en comunicación con los dos módulos Xbee que se encuentran conectado a su red



Figura 51. Sistema de seguridad
Fuente: Los Autores

Para verificar la calidad de comunicación y la efectividad de los módulos se realizó pruebas dentro del domicilio dando movilidad a diferentes distancias a los Routers/Activadores excepto el coordinador porque ese va ser nuestro punto de referencia.

La **Tabla 20** muestra una descripción de las diferentes distancias tomadas por los módulos de las pruebas realizadas en el interior del hogar.

Nº pruebas	conmutaciones	Distancia (metros)	Estado
1	Iluminación	2	Exitoso
2	Iluminación	5	Exitoso
3	Iluminación	10	Exitoso
4	Iluminación	15	Exitoso
5	Iluminación	20	Exitoso
6	Iluminación	38	Exitoso
7	Ventilador	2	Exitoso
8	Ventilador	5	Exitoso
9	Ventilador	10	Exitoso
10	Ventilador	15	Exitoso
11	Ventilador	20	Exitoso
12	Ventilador	40	Fallido
13	Alarma	2	Exitoso
14	Alarma	5	Exitoso
15	Alarma	10	Exitoso
16	Alarma	15	Exitoso
17	Alarma	20	Exitoso
18	Alarma	40	Fallido

Tabla 20. Prueba de comunicaciones

Fuente: Los Autores

Haciendo un análisis entre los exitosos y fallidos se establece el siguiente porcentaje de error producido en las pruebas.

$$\text{Total de Pruebas realizadas } 18 = 100\%$$

$$\text{Total perdida, datos } 2 = X$$

$$X = \frac{2 * 100\%}{18} = 11.11\%$$

3.2 ANÁLISIS FINANCIERO

La **Tabla 21** muestra un costo total empleado en el desarrollo del proyecto de investigación.

Costos	Descripción	Valor Usd.
Hardware	Computador	0
	GLCD	320
	Módulos ZigBee	80
	Módulos IP	100
	Módulos GPS celular	150
	Módulos Wi - fi	130
	Módulos Bluetooth	40
	Módulos Transeiver	100
	Microcontroladores	40
	Pulsadores	5
	LCD	8
	Protoboard	45
	Resistencias	5
	Condensadores	5
	Cristal de cuarzo	2
	Led's - RGB's	10
	Potenciómetros	5
	Cables	5
	Teclado	15
	Trabajos en Plaquetas de cobre	400
Trabajos en acrílico	200	
Software	MicroCode	0
	Proteus 8 Professional	0
	Micro C PRO for Pic	0
	AVR Studio	0
Proyecto	Transporte	100
varios	Cables, baquelita, estaño, pasta. Cautín.	300
Subtotal		1170
Imprevistos 10% subtotal		117
Total		2265

Tabla 21. Presupuesto total del proyecto

Fuente: Los Autores

CAPÍTULO IV

4 Discusión

En el mercado existe varias dispositivos electrónicos que pueden ser utilizados para realizar comunicaciones inalámbricas, pero a través de una investigación y recolección de información de nuevas posibles, técnicas proyectadas para el control domótica, se logró encontrar los módulos adecuados que permitan el desarrollo de un mecanismo estable que brinde a los estudiantes mayor comodidad en el desarrollo de sus prácticas por su versatilidad para adaptarse a múltiples usos.

Esta investigación tiene como objetivo primordial mejorar el rendimiento académico de los estudiantes que cursan la asignatura de microprocesadores, por lo tanto se realizó las pruebas para sustentar el proyecto, se tomó como muestra 25 alumnos de sexto semestre teniendo en cuenta que la muestra es el promedio de los que realizan prácticas en dicha área durante la clase práctica.

Durante el desarrollo de la prueba se pudo observar que el entrenador es un dispositivo eficiente, preciso, alcanzando una velocidad de 250 Kbps como dice el estándar de la tecnología ZigBee, de las 25 pruebas realizadas se puede hablar un 99% de efectividad y un 1% de error que fue cometida por los estudiantes debido a las conexiones erradas. Otra de las pruebas que se considero es el tiempo que se emplea para armar el circuito electrónico con o sin el entrenador, obteniendo resultados favorables como se observa en la **Tabla 17**, que la media de tiempo de armado sin el entrenador es de 106.6 minutos y la media de tiempo de armado con el entrenador es de 51.88 minutos, es evidente la optimización de tiempo en el armado del circuito, por lo tanto se hizo uso de la herramienta software de SPSS realizando la prueba de t Student para comprobar estadísticamente el rechazo de la hipótesis nula y la aceptación de la hipótesis alternativa.

Y como propuesta se planteó realizar entrenadores similares para las diferentes asignaturas que así pueda el estudiante optimizar recursos, explorar nuevos sistemas o nuevas tecnologías ya que hoy en día la tecnología da pasos agigantados y el estudiante tiene la obligación de ir a la par con la tecnología.

CAPÍTULO V

5 CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- La efectividad de los Xbee es proporcional al medio donde se está realizando la transmisión.
- Los módulos Xbee brindan mayor facilidad de configuración para realizar comunicaciones dentro de la red PAN.
- Estos dispositivos son utilizados en control de industrias, principios de domótica, automatización, porque son efectivos en la transmisión de paquete de datos y son de bajo costo.
- Los módulos Xbee son dispositivos de alta funcionalidad a pesar de su tamaño reducido que permiten un ahorro significativo de energía.
- Estos dispositivos son muy sensibles a las fluctuaciones en la alimentación ya que los picos de voltaje de la fuente afectaron al dispositivo a tal punto de dañarlos.

5.2 Recomendaciones

- Verificar el voltaje de salida de la fuente con la que van a alimentar los módulos Xbee.
- Siempre utilizar un regulador de voltaje para la alimentación del módulo y no hacerlo directamente de una fuente debido a los picos presentes en la fuente.
- Seleccionar el puerto correcto que le asigna la computadora al cable serial USB para la configuración del módulo con el programa X-CTU.
- Si el módulo Xbee trabaja conjuntamente con un microcontrolador en una transmisión es muy importante realizar un divisor de tensión, debido a la salida de voltaje que proporciona el PIC, (5V) y el voltaje que soporta el Xbee, (3.3V).
- En caso de trabajar con dos fuentes de alimentación se debe hacer referencia al GND de las dos fuentes para el correcto funcionamiento del circuito.

CAPITULO VI

6 PROPUESTA

TÍTULO DE PROPUESTA

Diseño e implantación de un Entrenador de Microcontroladores para la comunicación ZigBee.

INTRODUCCIÓN:

Para el funcionamiento adecuado de los dispositivos se necesita analizar las funciones que se va a realizar en el proyecto domótico, para su correcto funcionamiento en la transmisión y recepción de datos en lugares externos e internos, a cortas y largas distancias. Una vez estudiados los requerimientos y parámetros que permita cumplir con todos los requisitos necesarios para la comunicación y control domótico.

Para efectuar todos los objetivos planteados en el desarrollo del proyecto de investigación, se deberá desarrollar el equipo necesario con un esquema de diseño, para que de esta manera se pueda lograr el correcto funcionamiento del sistema y pueda cumplir con todos los requerimiento necesarios del usuario: comunicación en red mesh, conmutaciones automáticas del foco, control automático de temperatura y sobre todo el sistema de seguridad del hogar que de gran prioridad para la sociedad.

Con el estudio anticipado realizado lo que se logra es buscar el tipo de tecnología que puede ayudar a cumplir los objetivos plateados, de una manera eficiente y sobre todo pueda corregir los errores que se presenten en cada una de las etapas del desarrollo y de esa manera conseguir un a buna culminación del proyecto de investigación.

Dentro de todas las tecnologías se pudo notar que la tecnología Xbee que utiliza el estándar ZigBee pueden trabajar de una manera eficiente ya sea solo o con cualquier tipo de microcontrolador, solo depende de la configuración del módulo y el tipo de red en la que desea trabajar ya se punto a punto, punto a multi-punto, o de tipo mesh.

OBJETIVOS

OBJETIVO GENERAL

- Diseñar e implementar un entrenador para microcontroladores con comunicación ZigBee.

OBJETIVOS ESPECÍFICOS

- Demostrar la comunicación ZigBee mediante el control de un sistema domótico con sensores.
- El entrenador enfocara al desarrollo de sistemas embebidos.
- Diseñar un entrenador con módulos Xbee para microcontroladores.
- Realizar un manual de prácticas del manejo del módulo Xbee para que los estudiantes puedan realizas sus prácticas sin ningún tipo de problema.

FUNDAMENTACIÓN CIENTÍFICA-TÉCNICA

Para el desarrollo y la implementación de los dispositivos domóticos es importante considerar algunos aspectos que forman parte del desarrollo, los cuales cumplen un papel muy importante en sistema domotico, el cual se detalla a continuación.

- **Módulo Xbee.**

Los módulos Xbee son soluciones integradas que brindan un medio inalámbrico para la interconexión y comunicación entre dispositivos que utiliza el estándar ZigBee.

ZigBee es una nueva tecnología inalámbrica de corto alcance y bajo consumo originaria de la antigua alianza HomeRF y que se definió como una solución inalámbrica de baja capacidad para aplicaciones en el hogar como la seguridad y la automatización. ZigBee es un protocolo de comunicaciones basado en el estándar de comunicaciones para redes inalámbricas IEEE_802.15.4. Creado por ZigBee Alliance, una organización, teóricamente sin fines de lucro.

Las comunicaciones ZigBee se realizan en la banda libre de 2.4GHz. A distancia diferencia de Bluetooth, este protocolo no utiliza FHSS (Frequency hopping), sino que realiza las comunicaciones a través de una sola frecuencia, es decir, de un

canal. Normalmente puede escogerse un canal de entre 16 posibles. El alcance depende de la potencia de transmisión del dispositivo así como también del tipo de las antenas utilizadas (cerámicas, dipolos, etc.) EL alcance del módulo Xbee en línea de vista, está en 100 metros con una y en interiores esta de unos 30m con una potencia de 1mW. La velocidad de trasmisión de datos de una red ZigBee es de hasta 256kbps. Una red ZigBee la puede formar teóricamente hasta 65535 equipos, es decir, el protocolo está preparado para poder controlar en la misma red esta cantidad enorme de dispositivos.

DESCRIPCIÓN DE LA PROPUESTA

La propuesta se fundamentó en el estudio de las tecnologías de difusión de datos, para lograr comunicaciones inalámbricas entre dispositivos domóticos, los mismo que serán empleados por estudiantes de la UNIVERSIDAD NACIONAL DE CHIMBORAZO de la ESCUELA DE ELECTRÓNICA Y TELECOMUNICACIONES, para su mejor entendimiento, de manera que puedan desarrollar prácticas que ayuden a edificar conocimientos en microcontroladores con aplicaciones de comunicaciones.

DISEÑO ORGANIZACIONAL

A continuación se muestra la estructura funcional de la unidad administrativa con la que se ejerció el proyecto **Figura 52**.

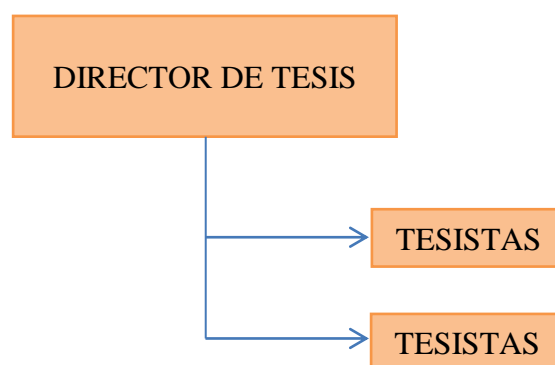


Figura 52. Diagrama del sistema organizacional del proyecto
Fuente: Los Autores

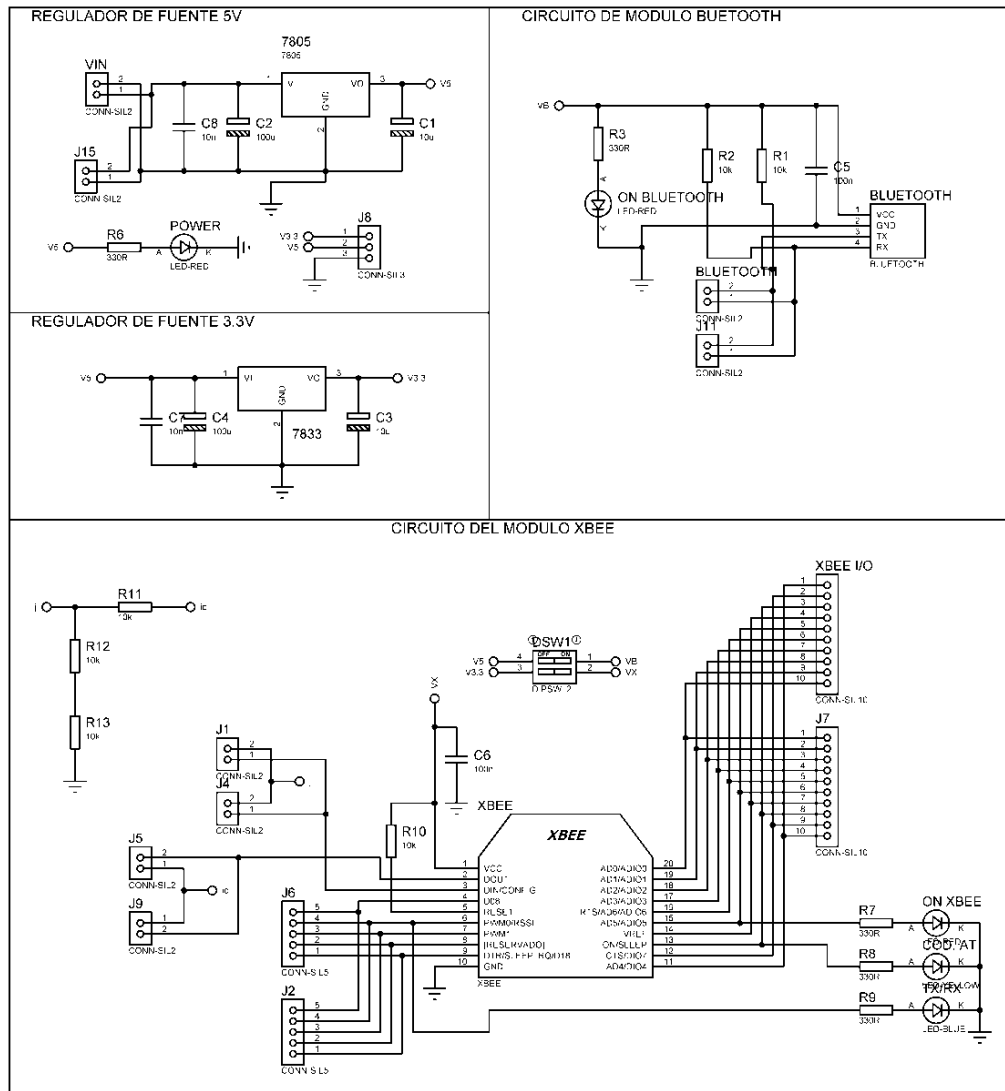
CAPITULO VII

7 BIBLIOGRAFÍA

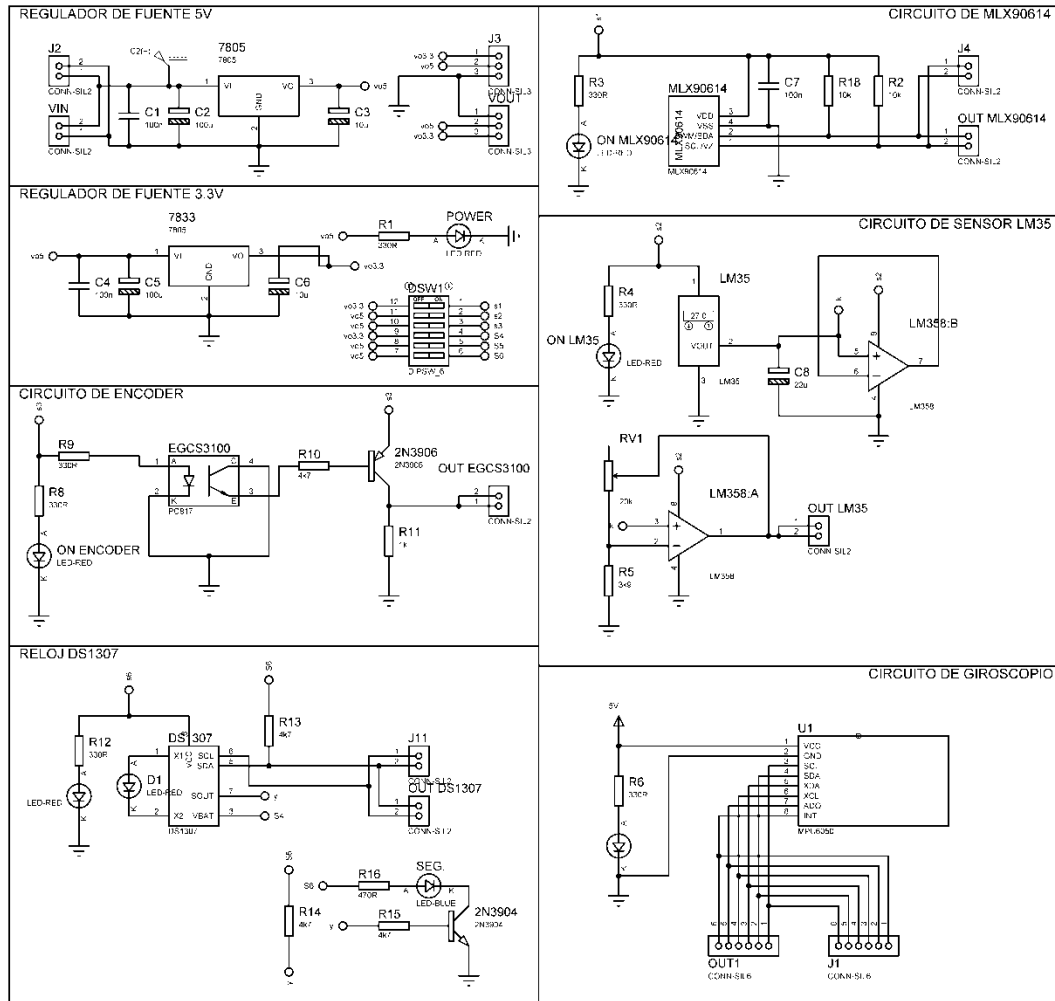
- (Manual_usuario_módulos_xbee)http://postgrado.info.unlp.edu.ar/Carreras/Especializaciones/Redes_y_Seguridad/Trabajos_Finales/Dignanni_Jorge_Pablo.pdf.
- Reyes, C. A. (2006). Microcontroladores PIC Programación en Basic. En C. A. Reyes. Microcontroladores PIC Programación en Basic, Quito: RISPGRAF.
- Ramiro Valencia, MICROTEL, Aplicaciones Electrónicas con Microcontroladores, lenguaje Basic (BASCOM AVR).
- Guía de usuario-XBEE SERIES 1/http://www.olimex.cl/pdf/Wireless/ZigBee/XBee-Guia_Usuario.pdf
- J. Dignani, S. Drangosh, “Interconectando sistemas de domótica”. WICC 2008.
- Adams, Jon; Bob Heile (2005-10). Busy as a ZigBee. [IEEE]. Compare with Other Technologies. Bluetooth SIG.

ANEXOS

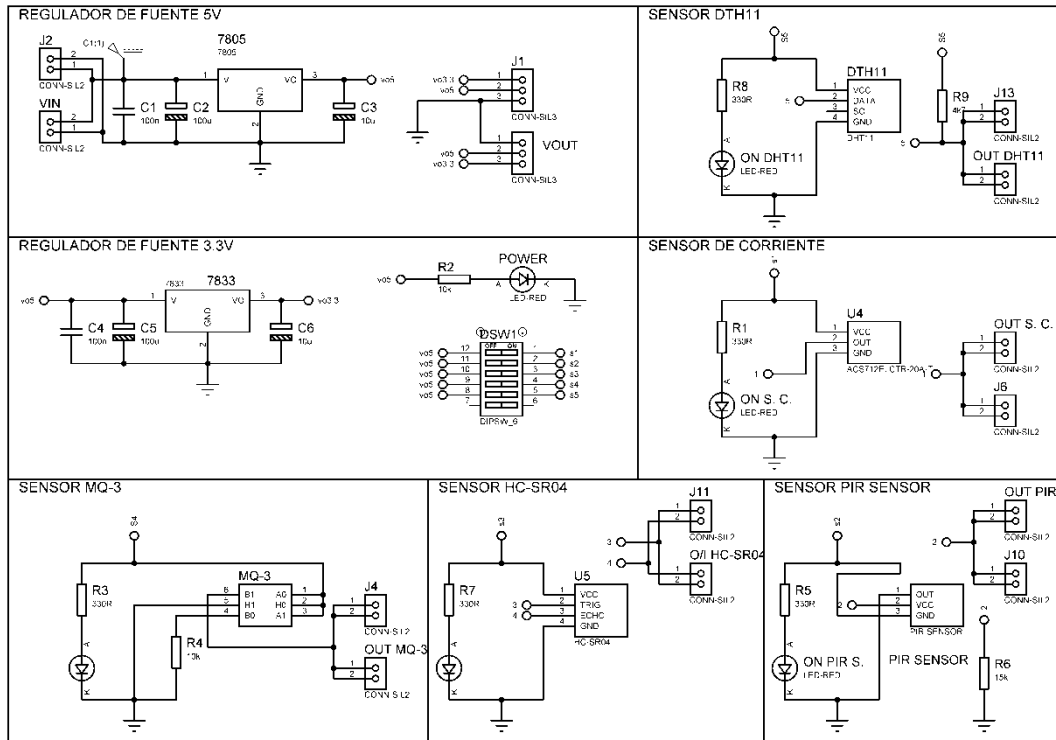
ANEXO 1. CIRCUITO COMPLETO DE PLACA COMUNICACIONES



ANEXO 2. CIRCUITO COMPLETO DE PLACA UNO DE SENSORES



ANEXO 3. CIRCUITO COMPLETO DE PLACA DOS DE SENSORES



ANEXO 4. PROGRAMACIÓN DEL ATMEGA 328P

```
*****
DEFINICION MICROCONTROLADOR, CRISTAL Y VELOCIDAD SERIAL
*****
$regfile = "m328pdef.dat"                'MICROCONTROLADOR A
UTILIZAR
$crystal = 8000000                       'VELOCIDAD DEL CRISTAL
$baud = 9600                             'VELOCIDAD COMUNICACION X-BEE

Config Com1 = Dummy , Synchrone = 0 , Parity = None , Stopbits = 1 , Databits = 8 ,
Clockpol = 0

***** LCD ALFANUMERICO*****
Config Lcdpin = Pin , Rs = Portc.5 , E = Portc.4 , Db4 = Portc.3 , Db5 = Portc.2 , Db6 =
Portc.1 , Db7 = Portc.0
Config Lcd = 16 * 2

***** INTERRUPCION de serial *****
On Urxr Rec_isr                          'Define subrutina de Interrupcion

*****
'*          LED ROJO                      *
*****
Ddrb.2 = 1
Portb.2 = 0
Ledrojo Alias Portb.2
*****
'*          BOTON1                        *
*****
Ddrd.2 = 0
Portd.2 = 1
Tsubir Alias Pind.2
*****
'*          BOTON2                        *
*****
Ddrd.3 = 0
Portd.3 = 1
Tbajar Alias Pind.3
*****
'*          BOTON3                        *
*****
Ddrd.4 = 0
Portd.4 = 1
Tmover Alias Pind.4
*****
'*          BOTON4                        *
*****
Ddrd.5 = 0
Portd.5 = 1
Tenter Alias Pind.5
*****
'*          BOTON5                        *
*****
Ddrd.6 = 0
```

```

Portd.6 = 1
Boton5 Alias Pind.6
*****
*           BOTON6           *
*****

Ddrd.7 = 0
Portd.7 = 1
Boton6 Alias Pind.7
'++++++

***** DS1307 RTC *****
Config Sda = Portb.1           'Configura I2C pin Datos
Config Scl = Portb.0          'Configura I2C pin Reloj

'Direccion del DS1307
Const Ds1307w = &HD0           'Constante escritura DS1307
Const Ds1307r = &HD1           'Constante lectura DS1307

'Declara variables necesarias para el DS1307
Dim Dias As Byte
Dim Segu As Byte
Dim Minu As Byte
Dim Hora As Byte
Dim Diam As Byte
Dim Mes As Byte
Dim Anio As Byte
Dim Dias1 As String * 3
Dim Texto1 As String * 3 , Texto22 As String * 2 , Texto3 As String * 2
Dim Texto4 As String * 30 , Texto5 As String * 30
Dim Tempod As Byte
Dim Tempou As Byte
Dim Tempo1 As Byte
Dim Senial As Byte
Dim Tempo11 As Byte
Dim Tempo22 As Byte
Dim Tempo33 As Byte
Dim Tempo44 As Byte
Dim Tempo55 As Byte
Dim Tempo66 As Byte
Dim Tempo77 As Byte
Dim Control As Byte
Dim Breve As Byte
Dim Tecla As Byte
Dim Tempo As Byte

Dim B As Byte
Dim I As Byte
Dim I2 As Byte
Dim V As Byte

Dim Flag1 As Bit
Dim Flag2 As Bit
Dim Flag3 As Bit
Dim Flag4 As Bit
Dim Pos As Byte

```

```

Dim Temporal As String * 1
Dim Temporal1 As String * 1
Dim Temporal2 As String * 1

Dim Clave1(5) As String * 1
Dim Clavea1(5) As String * 1
Dim Clavea2(5) As String * 1
Dim Serial1 As String * 10

Const Tiempo = 120                'Tiempo en seg
Const Tiempo1 = 30                'Tiempo en seg
Const Mil = 1000                  '1000
Const Retardo = 3                 'Tiempo en seg.
Const Vermensaje = 2              'Tiempo en seg.

Const Delay1 = 80
Const Posfila1 = 1                'variable para Posicion del cursor
Const Posfila2 = 2
Const Poscol1 = 5
Const Poscol2 = 2
Const Retardo1 = 4

Waitms 500
Cursor Off
Cls
Gosub Set1hz

'Gosub Presentacion

***** Animación Inicial *****
'For I = 0 To 9
' Ledrojo = 1
' Waitms 100
'Ledrojo = 0
'Waitms 100
'Next I

Cls
Enable Interrupts
Enable Urx

Locate 1 , 4
Lcd "BIENVENIDO"
Locate 2 , 1
Lcd "CASA INTELIGENTE"
Wait 4
Cls

***** Programa Principal *****
Do
'-----
  Gosub Getdatetime
  Locate Posfila1 , Poscol1
  Lcd Texto4

```

```

Locate Posfila22 , Poscol22
Lcd Texto5

If Tenter = 1 Then
  Waitms Delay1
  While Tenter = 1
    Wend
  Waitms Delay1
  Gosub Igualarrtc
End If
If Flag2 = 1 Then
  Cls
  Select Case Temporal1
    Case "a"
      Flag2 = 0
      Lcd " DORMITORIO "
      Lowerline
Lcd " ENCENDIDO "
      Wait Retardo1
    Case "b"
      Flag2 = 0
      Lcd " DORMITORIO "
      Lowerline
      Lcd " APAGADO "
      Wait Retardo1
    Case "c"
      Flag2 = 0
      Lcd " TEMPERATURA "
      Lowerline
      Lcd " ALTA "
      Wait Retardo1
    Case "d"
      Flag2 = 0
      Lcd " TEMPERATURA "
      Lowerline
Lcd " BAJA "
      Wait Retardo1
    Case "e"
      Flag2 = 0
      Lcd " INTRUSOS "
      Lowerline
      Lcd " EN CASA "
      Wait Retardo1
  End Select
  Cls
End If

Gosub Toggle_ledrojo

'-----
'-----

Loop
*****
*****

```



```

*****
Rec_isr:
  Disable Interrupts
  Disable Urx
  Texto1 = ""
  Input Texto1 Noecho
  Temporal1 = Mid(texto1 , 1 , 1)
Cls
Select Case Temporal1
  Case "a"
    Flag2 = 1
  Case "b"
    Flag2 = 1
  Case "c"
    Flag2 = 1
  Case "d"
    Flag2 = 1
  Case "e"
    Flag2 = 1
  Case Else
    Flag2 = 0
End Select
Cls
Do
  B = Inkey()
Loop Until B = 0
Enable Interrupts
Enable Urx
Return
*****
*****
Toggle_ledrojo:
  Toggle Ledrojo
  For I2 = 1 To 50
    Waitms 1
  Next I2
Return
*****
*****

***** Subrutina Cambiar el Tiempo
*****

Igualarrtc:
  Cls
  Gosub Getdatetime
  Locate Posfila1 , Poscol1
  Lcd Texto4
  Locate Posfila22 , Poscol22
  Lcd Texto5
  Locate 3 , 1
  Lcd "1=" ; Chr(7) ; " 2=" ; Chr(6) ; " 3=" ; Chr(126)
  Locate 4 , 1
  Lcd "D=GRABAR  *=SALIR"
  Flag3 = 1
  Pos = 0

```

```

Breve = Poscol11 + 1
Locate Posfila11 , Breve
Cursor On

While Flag3 = 1
  If Tsubir = 1 Then
    Waitms Delay1
    While Tsubir = 1
    Wend
    Waitms Delay1
    Select Case Pos
Case 0
  Tempo = Hora
Tempo = Makedec(tempo)
  Tempo = Tempo + 1
If Tempo = 24 Then
  Tempo = 0
End If
Tempo = Makebcd(tempo)
  Hora = Tempo
  Breve = Poscol11
  Locate Posfila11 , Breve
  Lcd Hex(hora)
  Breve = Poscol11 + 1
  Locate Posfila11 , Breve
Case 1
  Tempo = Minu
  Tempo = Makedec(tempo)
Tempo = Tempo + 1
If Tempo = 60 Then
  Tempo = 0
End If
Tempo = Makebcd(tempo)
  Minu = Tempo
  Breve = Poscol11 + 3
  Locate Posfila11 , Breve
  Lcd Hex(minu)
  Breve = Poscol11 + 4
  Locate Posfila11 , Breve
Case 2
  Tempo = Segu
  Tempo = Makedec(tempo)
Tempo = Tempo + 1
If Tempo = 60 Then
  Tempo = 0
End If
Tempo = Makebcd(tempo)
  Segu = Tempo
  Breve = Poscol11 + 6
  Locate Posfila11 , Breve
  Lcd Hex(segu)
  Breve = Poscol11 + 7
  Locate Posfila11 , Breve
Case 3
Dias = Dias + 1

```



```

        If Dias = 8 Then
            Dias = 1
        End If
        Select Case Dias
            Breve = Poscol22
            Locate Posfila22 , Breve
            Case 1
                Lcd "Dom"
            Case 2
Lcd "Lun"
            Case 3
                Lcd "Mar"
Case 4
                Lcd "Mie"
            Case 5
                Lcd "Jue"
            Case 6
                Lcd "Vie"
            Case 7
                Lcd "Sab"
        End Select
        Breve = Poscol22 + 2
        Locate Posfila22 , Breve
        Case 4
        Tempo = Diam
            Tempo = Makedec(tempo)
            Tempo = Tempo + 1
        If Tempo = 32 Then
            Tempo = 1
        End If
        Tempo = Makebcd(tempo)
            Diam = Tempo
            Breve = Poscol22 + 4
            Locate Posfila22 , Breve
            Lcd Hex(diam)
            Breve = Poscol22 + 5
            Locate Posfila22 , Breve
        Case 5
            Tempo = Mes
            Tempo = Makedec(tempo)
        Tempo = Tempo + 1
        If Tempo = 13 Then
            Tempo = 1
        End If
        Tempo = Makebcd(tempo)
            Mes = Tempo
            Breve = Poscol22 + 7
            Locate Posfila22 , Breve
            Lcd Hex(mes)
            Breve = Poscol22 + 8
            Locate Posfila22 , Breve
        Case 6
            Tempo = Anio
            Tempo = Makedec(tempo)
        Tempo = Tempo + 1

```

```

If Tempo = 100 Then
    Tempo = 0
End If
Tempo = Makebcd(tempo)
Anio = Tempo
Breve = Poscol22 + 10
Locate Posfila22 , Breve
Lcd "20"
Breve = Poscol22 + 12
Locate Posfila22 , Breve
Lcd Hex(anio)
Breve = Poscol22 + 13
    Locate Posfila22 , Breve
End Select
End If
'-----
If Tbajar = 1 Then
    Waitms Delay1
    While Tbajar = 1
    Wend
    Waitms Delay1
    Select Case Pos
Case 0
        Tempo = Hora
        Tempo = Makedec(tempo)
Tempo = Tempo - 1
If Tempo = 255 Then
    Tempo = 23
End If
Tempo = Makebcd(tempo)
    Hora = Tempo
    Breve = Poscol11
    Locate Posfila11 , Breve
    Lcd Hex(hora)
    Breve = Poscol11 + 1
    Locate Posfila11 , Breve
Case 1
    Tempo = Minu
    Tempo = Makedec(tempo)
Tempo = Tempo - 1
If Tempo = 255 Then
    Tempo = 59
End If
Tempo = Makebcd(tempo)
    Minu = Tempo
    Breve = Poscol11 + 3
    Locate Posfila11 , Breve
    Lcd Hex(minu)
    Breve = Poscol11 + 4
    Locate Posfila11 , Breve
Case 2
    Tempo = Segu
    Tempo = Makedec(tempo)
Tempo = Tempo - 1
If Tempo = 255 Then

```

```

        Tempo = 59
    End If
Tempo = Makebcd(tempo)
    Segu = Tempo
    Breve = Poscol11 + 6
    Locate Posfila11 , Breve
    Lcd Hex(segu)
    Breve = Poscol11 + 7
    Locate Posfila11 , Breve
Case 3
Dias = Dias - 1
    If Dias = 0 Then
        Dias = 7
    End If
    Select Case Dias
        Breve = Poscol22
        Locate Posfila22 , Breve
        Case 1
            Lcd "Dom"
        Case 2
Lcd "Lun"
        Case 3
            Lcd "Mar"
        Case 4
            Lcd "Mie"
        Case 5
            Lcd "Jue"
        Case 6
            Lcd "Vie"
        Case 7
            Lcd "Sab"
    End Select
    Breve = Poscol22 + 2
    Locate Posfila22 , Breve
Case 4
Tempo = Diam
    Tempo = Makedec(tempo)
    Tempo = Tempo - 1
    If Tempo = 0 Then
        Tempo = 31
    End If
Tempo = Makebcd(tempo)
    Diam = Tempo
    Breve = Poscol22 + 4
    Locate Posfila22 , Breve
    Lcd Hex(diam)
    Breve = Poscol22 + 5
    Locate Posfila22 , Breve
Case 5
    Tempo = Mes
    Tempo = Makedec(tempo)
Tempo = Tempo - 1
    If Tempo = 0 Then
        Tempo = 12
    End If

```

```

Tempo = Makebcd(tempo)
    Mes = Tempo
    Breve = Poscol22 + 7
    Locate Posfila22 , Breve
    Lcd Hex(mes)
    Breve = Poscol22 + 8
    Locate Posfila22 , Breve
Case 6
    Tempo = Anio
    Tempo = Makedec(tempo)
Tempo = Tempo - 1
If Tempo = 255 Then
    Tempo = 99
End If
Tempo = Makebcd(tempo)
    Anio = Tempo
    Breve = Poscol22 + 10
    Locate Posfila22 , Breve
    Lcd "20"
    Breve = Poscol22 + 12
    Locate Posfila22 , Breve
    Lcd Hex(anio)
Breve = Poscol22 + 13
    Locate Posfila22 , Breve
End Select
End If
'-----
If Tmover = 1 Then
    Waitms Delay1
    While Tmover = 1
    Wend
    Waitms Delay1
    Pos = Pos + 1
    If Pos = 7 Then
        Pos = 0
    End If
    Select Case Pos
    Case 0
        Breve = Poscol11 + 1
Locate Posfila11 , Breve
    Case 1
        Breve = Poscol11 + 4
        Locate Posfila11 , Breve
    Case 2
        Breve = Poscol11 + 7
        Locate Posfila11 , Breve
    Case 3
        Breve = Poscol22 + 2
        Locate Posfila22 , Breve
    Case 4
        Breve = Poscol22 + 5
        Locate Posfila22 , Breve
    Case 5
        Breve = Poscol22 + 8
        Locate Posfila22 , Breve

```

```

Case 6
Breve = Poscol22 + 13
Locate Posfila22 , Breve
End Select
End If
'-----
If Tenter = 1 Then
Waitms Delay1
While Tenter = 1
Wend
Waitms Delay1
Dias = Makedec(dias) 'dia de la
semana(D=1,L=2,M=3,M=4,J=5,V=6,S=7)
Diam = Makedec(diam)
Mes = Makedec(mes)
Anio = Makedec(anio)
Gosub Setdate
Segu = Makedec(segu)
Minu = Makedec(minu)
Hora = Makedec(hora)
Gosub Settime
Flag3 = 0
Gosub Getdatetime
Cursor Off
End If
'-----
Wend
Return
***** Subrutina BCD a Binario *****
Bcdabin:
Tempou = Tempo1 And $0f
Shift Tempo1 , Right , 4
Tempod = Tempo1 * 10
Tempo1 = Tempod + Tempou
Return
***** Subrutina Binario a BCD *****
Binabcd:
Tempod = Tempo1 \ 10
Shift Tempod , Left , 4
Tempou = Tempo1 Mod 10
Tempo1 = Tempod + Tempou
Return
*****
Setdate:
Dias = Makebcd(dias)
Diam = Makebcd(diam)
Mes = Makebcd(mes)
Anio = Makebcd(anio)
I2cstart 'Genera inicio de I2C
I2cwbyte Ds1307w 'Envia Constante escritura del DS3232
I2cwbyte 3 'Direccion del dia del mes del DS3232
I2cwbyte Dias 'Nuevo dia de la semana
I2cwbyte Diam 'Nuevo dia del mes
I2cwbyte Mes 'Nuevo mes del año
I2cwbyte Anio 'Nuevo Año

```

I2cstop
Return

Settime:

Segu = Makebcd(segu)

Minu = Makebcd(minu)

Hora = Makebcd(hora)

I2cstart

'Genera inicio de I2C

I2cwbyte Ds1307w

'Envia Constante escritura del DS3232

I2cwbyte 0

'Direccion de segundos del DS3232

I2cwbyte Segu

'Nuevo Segundo

I2cwbyte Minu

'Nuevo Minuto

I2cwbyte Hora

'Nueva Hora

I2cstop

Return

Getdatetime:

I2cstart

'Genera inicio de I2C

I2cwbyte Ds1307w

'Envia Constante escritura del DS1307

I2cwbyte 0

'Direccion Inicial de memoria del DS1307

I2cstart

'Genera inicio de I2C

I2cwbyte Ds1307r

'Envia Constante lectura del DS1307

I2crbyte Segu , Ack

'Lee los segundos

I2crbyte Minu , Ack

'Lee los minutos

I2crbyte Hora , Ack

'Lee la hora

I2crbyte Dias , Ack

'Lee el dia de la

semana(D=1,L=2,M=3,M=4,J=5,V=6,S=7)

I2crbyte Diam , Ack

'Lee dia del mes

I2crbyte Mes , Ack

'Lee el mes del año

I2crbyte Anio , Nack

'Lee el año

I2cstop

Select Case Dias

Case 1

Dias1 = "Dom"

'Mostrar Día De La Semana

Case 2

Dias1 = "Lun"

'Mostrar Día De La Semana

Case 3

Dias1 = "Mar"

'Mostrar Día De La Semana

Case 4

Dias1 = "Mie"

'Mostrar Día De La Semana

Case 5

Dias1 = "Jue"

'Mostrar Día De La Semana

Case 6

Dias1 = "Vie"

'Mostrar Día De La Semana

Case 7

Dias1 = "Sab"

'Mostrar Día De La Semana

End Select

Texto1 = Hex(hora)

Texto22 = Hex(minu)

Texto3 = Hex(segu)

Texto4 = Format(texto1 , "00") + ":" + Format(texto22 , "00") + ":" + Format(texto3 , "00")

```

    Texto1 = Hex(ano)
    Texto22 = Hex(diam)
    Texto3 = Hex(mes)
    Texto5 = Dias1 + "-" + Format(texto22 , "00" ) + "/" + Format(texto3 , "00" ) + "/" +
"20" + Format(texto1 , "00" )    'Dias1 + "-" +
Return
*****
*****
Set1hz:
    Senial = &B10010000
    I2cstart                                'Genera inicio de I2C
I2cwbyte Ds1307w                            'Envía Constante escritura del DS1307
    I2cwbyte 7                              'Dirección de control del DS1307
I2cwbyte Senial                            'Nuevo Segundo
    I2cstop
Return
*****
End

```

ANEXO 5. PROGRAMACION DEL MICROCONTROLADOR 16F877A – ROUTER-A

```
char uart_rd;

sbit LCD_RS at RB0_bit;
sbit LCD_EN at RB1_bit;
sbit LCD_D4 at RB2_bit;
sbit LCD_D5 at RB3_bit;
sbit LCD_D6 at RB4_bit;
sbit LCD_D7 at RB5_bit;
sbit LCD_RS_Direction at TRISB0_bit;
sbit LCD_EN_Direction at TRISB1_bit;
sbit LCD_D4_Direction at TRISB2_bit;
sbit LCD_D5_Direction at TRISB3_bit;
sbit LCD_D6_Direction at TRISB4_bit;
sbit LCD_D7_Direction at TRISB5_bit;

unsigned int Radc = 0;           // Guarda los datos de la lectura del ANO
unsigned int vtemp = 0;         // Voltaje temporal tipo entero
char *tempC = "000.0";         // Apuntador, Voltaje final
char Text[15];                  // Cadena para conversion FloatToStr
char text1[] = "Temperatura:";
float temp1 = 0.0;              // Variable para procesos
int t,h;
void tiempo(){
  Delay_ms(9000);
  Lcd_Cmd(_LCD_CLEAR);}
void tiempo1(){
  delay_ms(500);
  Lcd_Cmd(_LCD_CLEAR);         // Clear display
}

void LM35(){
  if(vtemp < 10000){
    tempC[0] = ' ';
    tempC[1] = (vtemp/1000)%10 + 48;
    tempC[2] = (vtemp/100)%10 + 48;
    tempC[4] = (vtemp/10)%10 + 48;
    Lcd_Out(2,5,tempC);
    Delay_ms(20);
  } else{
    tempC[0] = (vtemp/10000)%10 + 48;
    tempC[1] = (vtemp/1000)%10 + 48;
    tempC[2] = (vtemp/100)%10 + 48;
    tempC[4] = (vtemp/10)%10 + 48;
    Lcd_Out(2,5,tempC);
    Delay_ms(20);
  }
}

void main() {
  UART1_Init(9600);           // Initialize UART module at 9600 bps
  Delay_ms(100);              // Wait for UART module to stabilize
  Lcd_Init();                  // Initialize LCD
```



```

    Lcd_Cmd(_LCD_CLEAR);          // Clear display
    Lcd_Cmd(_LCD_CURSOR_OFF);
    TRISA = 0x0D;                // AN0, AN2, AN3 como entradas
    ADCON1.F4 = 1;              // VREF+ 1 = externo 0 = VDD
    ADCON1.F5 = 1;              // VREF- 1 = externo 0 = VSS
    ADC_Init();                 // Iniciar módulo ADC

do{
    Lcd_Out(1,3,text1);         // Muestra el texto en el LCD
    Lcd_Chr(2,10,223);         // Simbolo de grado
    Lcd_Chr(2,11,'C');         // Mostrar la C para centigrados
    Radc = ADC_Get_Sample(0);   // Lectura del Canal AN0
    temp1 = (float)(Radc * 0.12); // Obtencion del valor real de la conversión
    vtemp = (temp1*100);        // Para mas informacion ir al articulo del blog
    FloatToStr(temp1,Text);     // Se convierte de float a char
    LM35();
    Delay_ms(400);             // Regula la velocidad en la cual se realizara

if (temp1 <= 20){
    tiempo1();
    Lcd_out(1,5,"TEMP. BAJA");
    UART1_Write_Text("3");
    tiempo();
}
if (temp1 >= 30){
    tiempo1();
    Lcd_out(1,5,"TEMP. ALTA");
    UART1_Write_Text("4");
    tiempo();}

if (UART1_Data_Ready()) { // If data is received,
    uart_rd = UART1_Read();
if( (uart_rd)=='1' ) {
    Lcd_Cmd(_LCD_CLEAR);          // Clear display
    Lcd_Cmd(_LCD_CURSOR_OFF);    // Cursor off
    Lcd_out(1,5,"DORMITORIO");
    Lcd_out(2,6,"ENCENDIDO");
    Delay_ms(9000);
    Lcd_Cmd(_LCD_CLEAR);}

if( (uart_rd)=='2' ) {
    Lcd_Cmd(_LCD_CLEAR);          // Clear display
    Lcd_Cmd(_LCD_CURSOR_OFF);    // Cursor off
    Lcd_out(1,5,"DORMITORIO");
    Lcd_out(2,7,"APAGADO");
    Delay_ms(9000);
    Lcd_Cmd(_LCD_CLEAR);}

if( (uart_rd)=='5' ) {
    Lcd_Cmd(_LCD_CLEAR);          // Clear display
    Lcd_Cmd(_LCD_CURSOR_OFF);    // Cursor off
    Lcd_out(1,1,"INTRUSOS EN CASA");
    Delay_ms(9000);
    Lcd_Cmd(_LCD_CLEAR);}
}
} while(1);

```

ANEXO 6. PROGRAMACIÓN DEL MICROCONTROLADOR 16F877A COMO ROUTER-B

```
//Configuracion del lcd
char uart_rd;
sbit LCD_RS at RB0_bit;
sbit LCD_EN at RB1_bit;
sbit LCD_D4 at RB2_bit;
sbit LCD_D5 at RB3_bit;
sbit LCD_D6 at RB4_bit;
sbit LCD_D7 at RB5_bit;
sbit LCD_RS_Direction at TRISB0_bit;
sbit LCD_EN_Direction at TRISB1_bit;
sbit LCD_D4_Direction at TRISB2_bit;
sbit LCD_D5_Direction at TRISB3_bit;
sbit LCD_D6_Direction at TRISB4_bit;
sbit LCD_D7_Direction at TRISB5_bit;

int cont1, cont2, cont3, cont4;
void tiempo(){
Delay_ms(4000);
Lcd_Cmd(_LCD_CLEAR);}
void tiempo1(){
delay_ms(500);
Lcd_Cmd(_LCD_CLEAR);          // Clear display
}
void ToneA() {
Sound_Play(1000, 100);
portb.f3 = 1;
delay_ms(50);
Sound_Play(1000, 100);
portb.f3 = 0;
delay_ms(50);
}
void main(){
trisd=0b00000001;
portd=0;
cont1=0;
cont2=1;
cont3=2;
cont4=3;
UART1_Init(9600);          // Initialize UART module at 9600 bps
Delay_ms(100);           // Wait for UART module to stabilize
Lcd_Init();              // Initialize LCD
Lcd_Cmd(_LCD_CLEAR);     // Clear display
Lcd_Cmd(_LCD_CURSOR_OFF);
Sound_Init(&PORTD,2);
do {
//////////PIR SENSOR//////////
if (portd.f0==1){        ///Lectura del PIR sensor
ToneA();                //Tono de alarma
tiempo1();
Lcd_out(1,1,"INTRUSOS EN CASA"); //Imprime SMS
UART1_Write_Text("e");  //Trasmisión
```

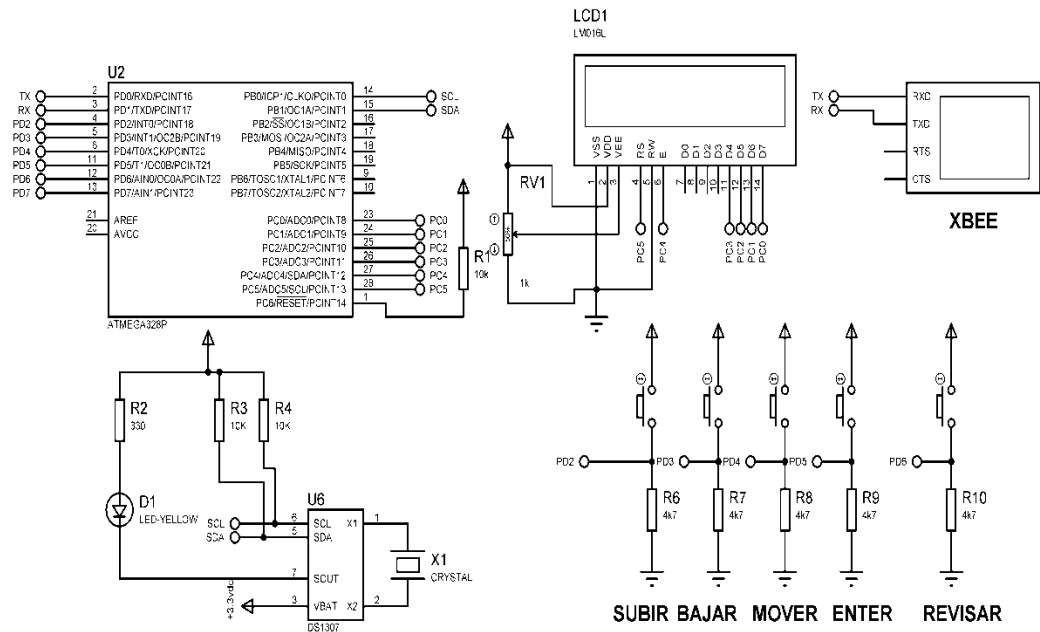
```

        tiempo();}
else {
    Lcd_Cmd(_LCD_CLEAR);          // Clear display
    Lcd_Out(1,3,"CASA SEGURA");
    delay_ms(1000);
}
if (UART1_Data_Ready()) { // If data is received,
    uart_rd = UART1_Read();
if( (uart_rd)=='A' ) {
    cont1++;
    if (cont1==1){
        Lcd_Cmd(_LCD_CLEAR);          // Clear display
        Lcd_out(1,5,"DORMITORIO");
        Lcd_out(2,6,"ENCENDIDO");
        portd.f1=1;
        Delay_ms(4000);
        Lcd_Cmd(_LCD_CLEAR);
        cont1=0;}
if( (uart_rd)=='B' ) {
    cont2++;
    if (cont2==2){
        Lcd_Cmd(_LCD_CLEAR);          // Clear display
        Lcd_out(1,5,"DORMITORIO");
        Lcd_out(2,7,"APAGADO");
        portd.f1=0;
        Delay_ms(4000);
        Lcd_Cmd(_LCD_CLEAR);
        cont2=1;}
if (uart_rd=='d'){
    Lcd_Cmd(_LCD_CLEAR);          // Clear display
    Lcd_Cmd(_LCD_CURSOR_OFF);    // Cursor off
    Lcd_out(1,5,"TEMP. BAJA");
    Lcd_out(2,1,"CALEFACTOR ACTIVO");
    Delay_ms(4000);
    Lcd_Cmd(_LCD_CLEAR);
    }
if( (uart_rd)=='c' ) {
    cont3++;
    if (cont3=3){
        Lcd_Cmd(_LCD_CLEAR);          // Clear display
        Lcd_Cmd(_LCD_CURSOR_OFF);    // Cursor off
        Lcd_out(1,5,"TEMP. ALTA");
        Lcd_out(2,1,"VENTILADOR ACTIVO");
        portd.f6=1;
        Delay_ms(4000);
        Lcd_Cmd(_LCD_CLEAR);}
        cont3=2;}
    }
if (uart_rd =='6'){
    Delay_ms(100);
    portd.f6=0;}

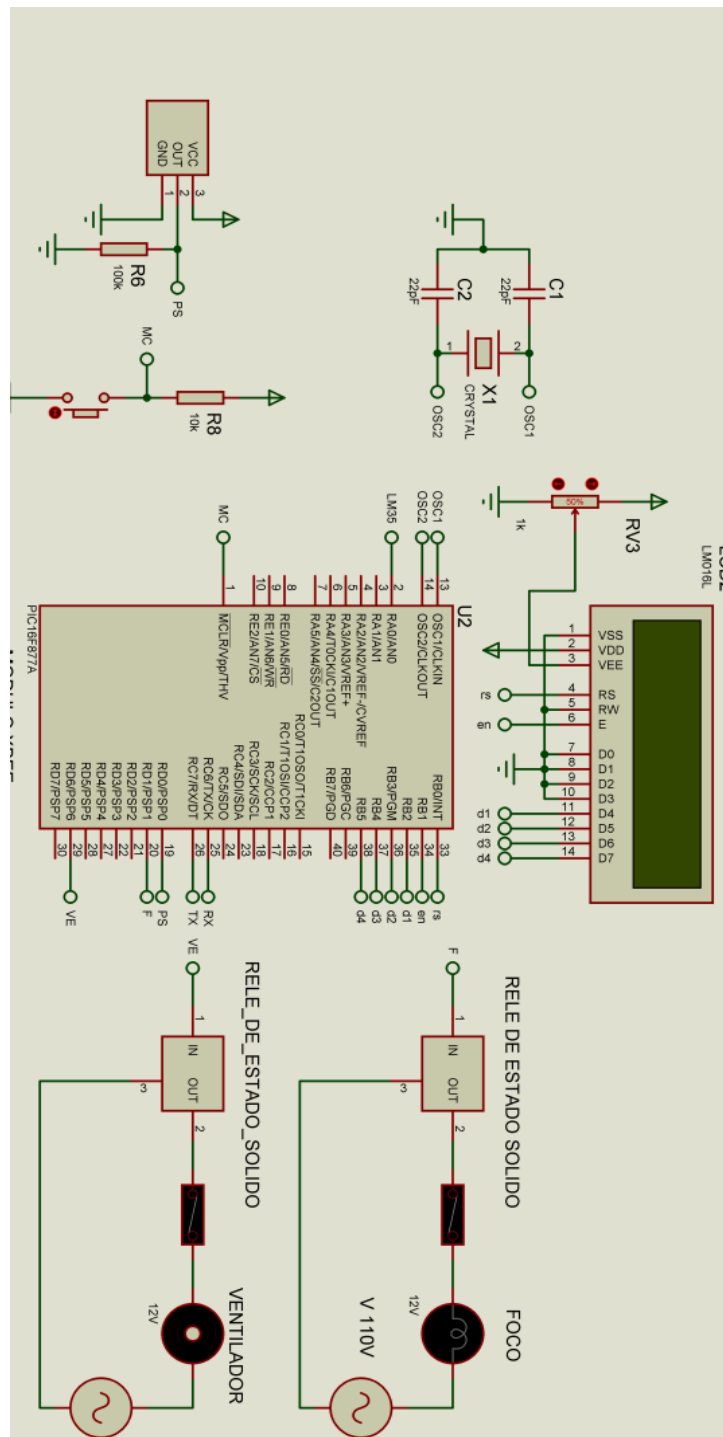
    }while (1);
}

```

ANEXO 7. DISEÑO DE CIRCUITOS ELÉCTRICOS DEL COORDINADOR



ANEXO 9. CIRCUITOS ELÉCTRICOS DE LOS DISPOSITIVOS UTILIZADOS EN EL ROUTER-B.



ANEXO 10. HOJA DE DATOS DEL MÓDULO XBEE



EMBEDDED RF
MODULES FOR OEMS



XBEE® 802.15.4 RF MODULES

Low-cost, easy-to-deploy modules provide critical end-point connectivity to devices and sensors

XBee 802.15.4 RF modules provide OEMs with a common footprint shared by multiple platforms, including multipoint and ZigBee/Mesh topologies, and both 2.4 GHz and 900 MHz solutions. OEMs deploying the XBee can substitute one XBee for another, depending upon dynamic application needs, with minimal development, reduced risk and shorter time-to-market.

XBee 802.15.4 RF modules are ideal for applications requiring low latency and predictable communication timing. Providing quick, robust communication in point-to-point, peer-to-peer, and multipoint/star configurations, XBee 802.15.4 products

enable robust end-point connectivity with ease. Whether deployed as a pure cable replacement for simple serial communication, or as part of a more complex hub-and-spoke network of sensors, XBee 802.15.4 RF modules maximize performance and ease of development.

XBee 802.15.4 modules seamlessly interface with compatible gateways, device adapters and range extenders, providing developers with true beyond-the-horizon connectivity.

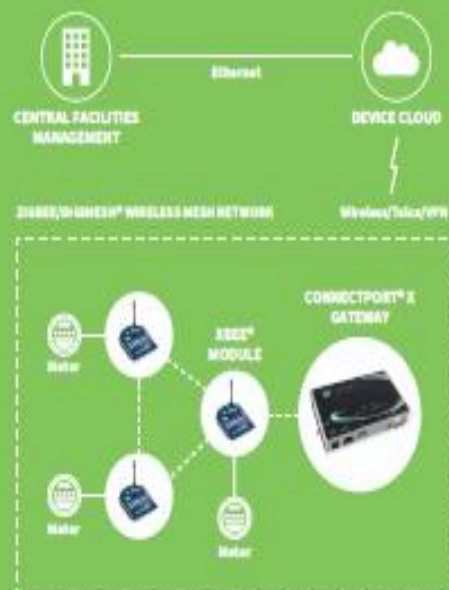
BENEFITS

- Point-to-multipoint network topology
- 2.4 GHz for worldwide deployment
- 900 MHz for long-range deployment
- Fully interoperable with other other Digi networking products, including gateways, device adapters and range extenders
- Common XBee footprint for a variety of RF modules
- Low-power sleep modes
- Multiple antenna options
- Industrial temperature rating (-40° C to 85° C)

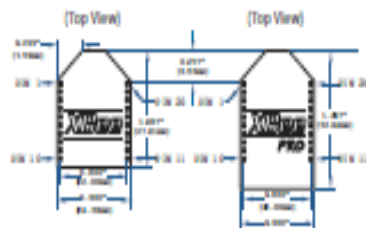
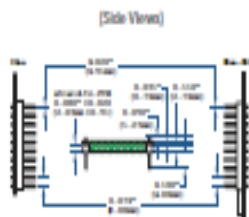
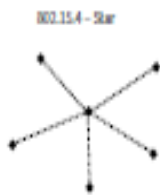
RELATED PRODUCTS



APPLICATION EXAMPLE



SPECIFICATIONS	XBee® 802.15.4	XBee-PRO® 802.15.4
PERFORMANCE		
RF DATA RATE	250 kbps	250 kbps
INDOOR/URBAN RANGE	100 ft (30 m)	300 ft (100 m)
OUTDOOR/RF LINE-OF-SIGHT RANGE	300 ft (100 m)	1 mi (1.6 km)
TRANSMIT POWER	1 mW (+0 dBm)	60 mW (+18 dBm)*
RECEIVER SENSITIVITY (1% PER)	-92 dBm	-100 dBm
FEATURES		
SERIAL DATA INTERFACE	3.3V CMOS UART	
CONFIGURATION METHOD	API or AT Commands, local or over-the-air	
FREQUENCY BAND	2.4 GHz	
INTERFERENCE IMMUNITY	DSSS (Direct Sequence Spread Spectrum)	
SERIAL DATA RATE	1200 bps - 250 kbps	
ADC INPUTS	(0) 10-bit ADC Inputs	
DIGITAL I/O	8	
ANTENNA OPTIONS	Chip, Wire Whip, U.F.L., & RP-SMA	
NETWORKING & SECURITY		
ENCRYPTION	128-bit AES	
RELIABLE PACKET DELIVERY	Retries/Acknowledgments	
IDS AND CHANNELS	PAN ID, 64-bit IEEE MAC, 16 Channels	
POWER REQUIREMENTS		
SUPPLY VOLTAGE	2.8 - 3.4VDC	2.8 - 3.4VDC
TRANSMIT CURRENT	45 mA @ 3.3VDC	215 mA @ 3.3VDC
RECEIVE CURRENT	50 mA @ 3.3VDC	55 mA @ 3.3VDC
POWER-DOWN CURRENT	<10 uA @ 25°C	
REGULATORY APPROVALS		
FCC (USA)	OUR XBEE	OUR XBEEPRO
IC (CANADA)	4214A-XBEE	4214A-XBEEPRO
ETSI (EUROPE)	Yes	Yes - Max TX 30 mW
C-TICK AUSTRALIA	Yes	
TELEC (JAPAN)	Yes	



PART NUMBERS	DESCRIPTION
XXK2-KT-WWC	Wireless Connectivity Kit w/ Xbee BQ2.15.4
XXK2-KT-WWG	Xbee / Arduino Compatible Coding Platform
XED4-AM1-001	Xbee BQ2.15.4 low-power module w/ wire antenna
XED4-AP1-001	Xbee BQ2.15.4 low-power module w/ PCB antenna
XED4-AU1-001	Xbee BQ2.15.4 low-power module w/ U.FL connector
XED4-AS1-001	Xbee BQ2.15.4 low-power module w/ RP-SMA connector
XEP24-AM1-001	Xbee-PRO BQ2.15.4 extended-range module w/ wire antenna
XEP24-AM1-001J	Xbee-PRO BQ2.15.4 extended-range module w/ wire antenna (International)
XEP24-AU1-001	Xbee-PRO BQ2.15.4 extended-range module w/ U.FL connector
XEP24-AU1-001J	Xbee-PRO BQ2.15.4 extended-range module w/ U.FL connector (International)
XEP24-AS1-001	Xbee-PRO BQ2.15.4 extended-range module w/ RP-SMA connector
XEP24-AS1-001J	Xbee-PRO BQ2.15.4 extended-range module w/ RP-SMA connector (International)
XEP24-AP1-001	Xbee-PRO BQ2.15.4 extended-range module w/ PCB antenna
XEP24-AP1-001J	Xbee-PRO BQ2.15.4 extended-range module w/ PCB antenna (International)

FOR MORE INFORMATION
PLEASE VISIT WWW.DIGI.COM

DIGI SERVICE AND SUPPORT / You can purchase with confidence knowing that Digi is always available to serve you with expert technical support and our industry leading warranty. For detailed information visit www.digi.com/support.

© 1996-2015 Digi International Inc. All rights reserved.
All trademarks are the property of their respective owners.

91001412
C1/915

DIGI INTERNATIONAL WORLDWIDE HQ
877-912-3444 / 952-912-3444 / www.digi.com

DIGI INTERNATIONAL FRANCE
+33-1-55-61-99-99 / www.digi.fr

DIGI INTERNATIONAL JAPAN
+81-3-5428-0261 / www.digi-intl.co.jp

DIGI INTERNATIONAL SINGAPORE
+65-6213-5380

DIGI INTERNATIONAL CHINA
+86-21-50492199 / www.digi.com.cn

DIGI

ANEXO 11. HOJAS DE DATOS DEL MICROCONTROLADOR 16F877A

Pin Diagrams (Continued)

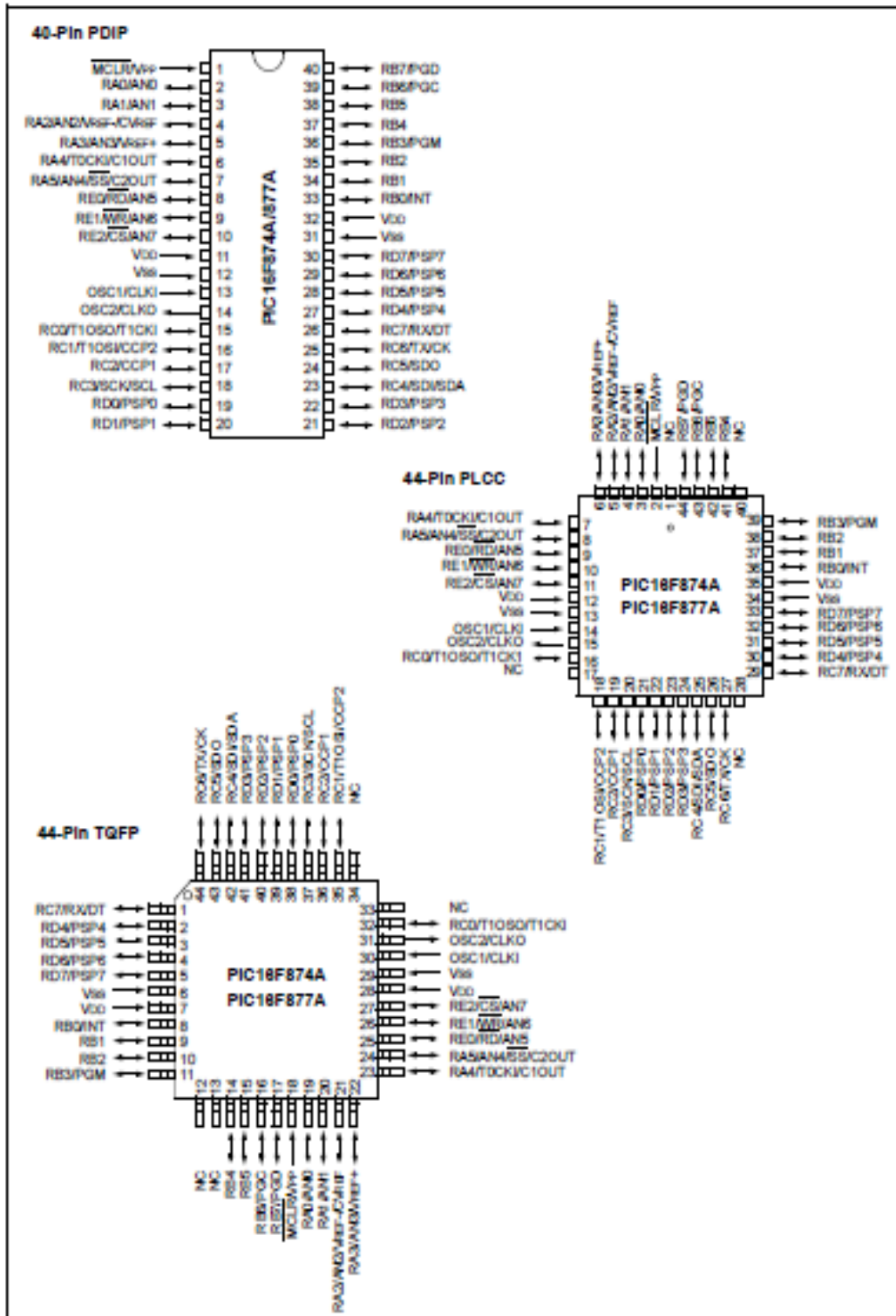


FIGURE 1-1: PIC16F873A/876A BLOCK DIAGRAM

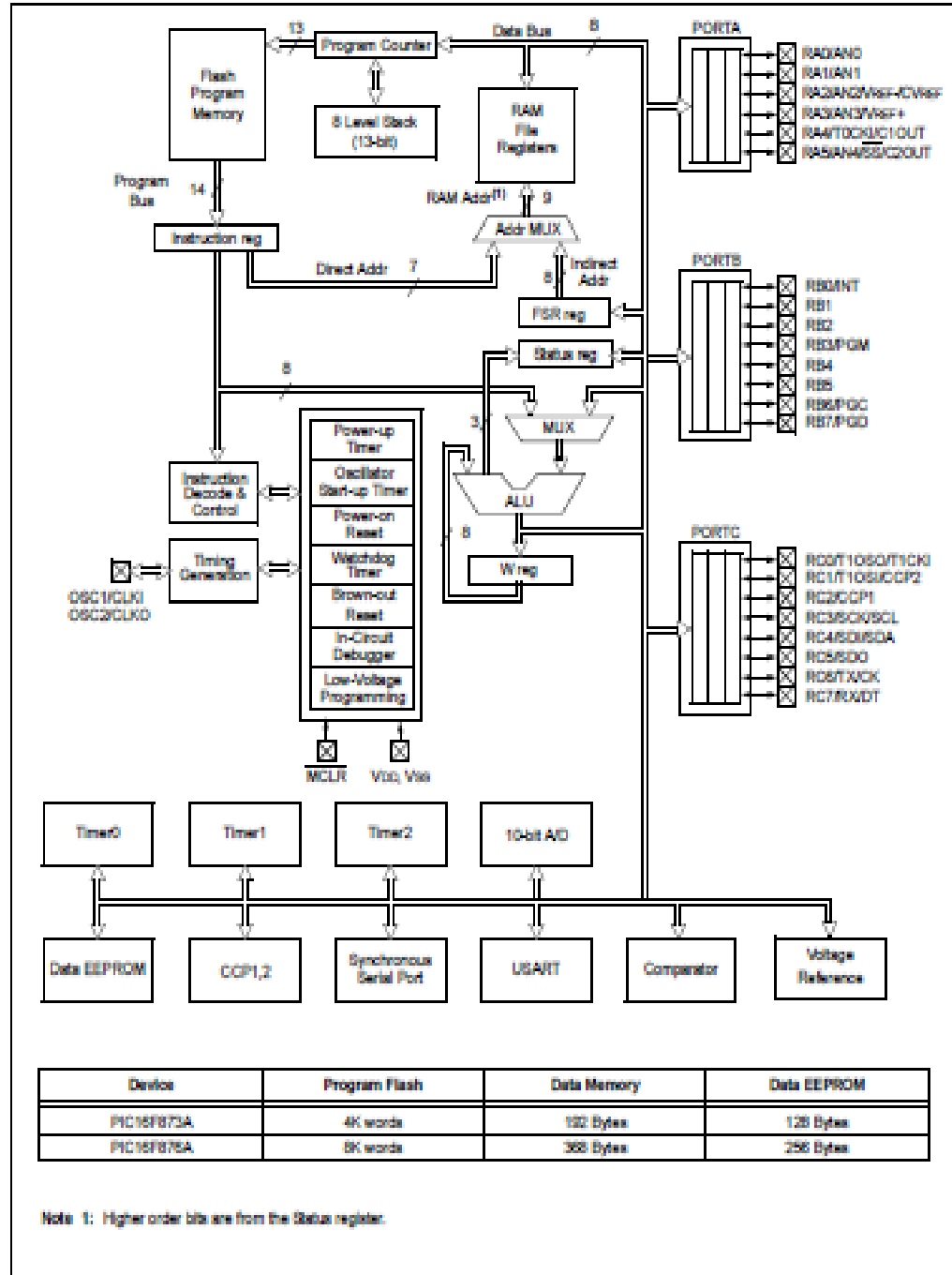


TABLE 1-2: PIC16F873A/876A PINOUT DESCRIPTION (CONTINUED)

Pin Name	PDIP, SOIC, SSOP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
RB0/INT RB0 INT	21	18	I/O I	TTLUST ⁽¹⁾	PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs. Digital I/O. External interrupt.
RB1	22	19	I/O	TTL	Digital I/O.
RB2	23	20	I/O	TTL	Digital I/O.
RB3/PGM RB3 PGM	24	21	I/O I	TTL	Digital I/O. Low-voltage (single-supply) ICSP programming enable pin.
RB4	25	22	I/O	TTL	Digital I/O.
RB5	26	23	I/O	TTL	Digital I/O.
RB6/PGC RB6 PGC	27	24	I/O I	TTLUST ⁽²⁾	Digital I/O. In-circuit debugger and ICSP programming clock.
RB7/PGD RB7 PGD	28	25	I/O I/O	TTLUST ⁽²⁾	Digital I/O. In-circuit debugger and ICSP programming data.
RC0/T1OSO/T1CKI RC0 T1OSO T1CKI	11	8	I/O O I	ST	PORTC is a bidirectional I/O port. Digital I/O. Timer1 oscillator output. Timer1 external clock input.
RC1/T1OSI/CCP2 RC1 T1OSI CCP2	12	9	I/O I I/O	ST	Digital I/O. Timer1 oscillator input. Capture2 input, Compare2 output, PWM2 output.
RC2/CCP1 RC2 CCP1	13	10	I/O I/O	ST	Digital I/O. Capture1 input, Compare1 output, PWM1 output.
RC3/SCK/SCL RC3 SCK SCL	14	11	I/O I/O I/O	ST	Digital I/O. Synchronous serial clock input/output for SPI mode. Synchronous serial clock input/output for I ² C mode.
RC4/SDI/SDA RC4 SDI SDA	15	12	I/O I I/O	ST	Digital I/O. SPI data in. I ² C data I/O.
RC5/SDO RC5 SDO	16	13	I/O O	ST	Digital I/O. SPI data out.
RC6/TXCK RC6 TX CK	17	14	I/O O I/O	ST	Digital I/O. USART asynchronous transmit. USART1 synchronous clock.
RC7/RXD/DT RC7 RX DT	18	15	I/O I I/O	ST	Digital I/O. USART asynchronous receive. USART synchronous data.
Vss	8, 19	5, 6	P	—	Ground reference for logic and I/O pins.
Vdd	20	17	P	—	Positive supply for logic and I/O pins.

Legend: I = input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
3: This buffer is a Schmitt Trigger input when configured in RC Oscillator mode and a CMOS input otherwise.

2.2.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers are registers used by the CPU and peripheral modules for controlling the desired operation of the device. These registers are implemented as static RAM. A list of these registers is given in Table 2-1.

The Special Function Registers can be classified into two sets: core (CPU) and peripheral. Those registers associated with the core functions are described in detail in this section. Those related to the operation of the peripheral features are described in detail in the peripheral features section.

TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Details on page:		
Bank 0													
00h ⁽²⁾	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)									0000 0000	31, 150	
01h	TMR0	Timer0 Module Register									xxxx xxxx	55, 150	
02h ⁽²⁾	PCL	Program Counter (PC) Least Significant Byte									0000 0000	30, 150	
03h ⁽²⁾	STATUS	RP	RP1	RP0	TO	PD	Z	DC	C	0001 1xxxx	22, 150		
04h ⁽²⁾	FSR	Indirect Data Memory Address Pointer									xxxx xxxx	31, 150	
05h	PORTA	—	—	PORTA Data Latch when written; PORTA pins when read								--0x 0000	43, 150
06h	PORTB	PORTB Data Latch when written; PORTB pins when read									xxxx xxxx	45, 150	
07h	PORTC	PORTC Data Latch when written; PORTC pins when read									xxxx xxxx	47, 150	
08h ⁽⁴⁾	PORTD	PORTD Data Latch when written; PORTD pins when read									xxxx xxxx	48, 150	
09h ⁽⁴⁾	PORTE	—	—	—	—	—	RE2	RE1	RE0	---- -xxx	49, 150		
0A ^(1,2)	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter						---0 0000	30, 150	
0Bh ⁽²⁾	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	24, 150		
0Ch	PIR1	PSP1P ⁽²⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	28, 150		
0Dh	PIR2	—	CMF	—	EEF	BCLIF	—	—	CCP2IF	-0-0 0--0	28, 150		
0Eh	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register									xxxx xxxx	80, 150	
0Fh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register									xxxx xxxx	80, 150	
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	--00 0000	57, 150		
11h	TMR2	Timer2 Module Register									0000 0000	82, 150	
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	--00 0000	81, 150		
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register									xxxx xxxx	79, 150	
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	82, 82, 150		
15h	CCPR1L	Capture/Compare/PWM Register 1 (LSB)									xxxx xxxx	83, 150	
16h	CCPR1H	Capture/Compare/PWM Register 1 (MSB)									xxxx xxxx	83, 150	
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	84, 150		
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	112, 150		
19h	TXREG	USART Transmit Data Register									0000 0000	118, 150	
1Ah	RCREG	USART Receive Data Register									0000 0000	118, 150	
1Bh	CCPR2L	Capture/Compare/PWM Register 2 (LSB)									xxxx xxxx	83, 150	
1Ch	CCPR2H	Capture/Compare/PWM Register 2 (MSB)									xxxx xxxx	83, 150	
1Dh	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	--00 0000	84, 150		
1Eh	ADRESH	A/D Result Register High Byte									xxxx xxxx	133, 150	
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON	0000 00--0	127, 150		

Legend: x = unknown, u = unchanged, c = value depends on condition, - = unimplemented, read as '0', r = reserved.
Shaded locations are unimplemented, read as '0'.

- Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8>, whose contents are transferred to the upper byte of the program counter.
- Note 2: Bits PSP1E and PSP1F are reserved on PIC18F873A/876A devices; always maintain these bits clear.
- Note 3: These registers can be addressed from any bank.
- Note 4: PORTD, PORTE, TRISD and TRISE are not implemented on PIC18F873A/876A devices, read as '0'.
- Note 5: Bit 4 of EEADRH implemented only on the PIC18F876A/877A devices.

ANEXO 12. HOJA DE DATOS DEL LM35



LM35

SNIS159E—AUGUST 1999—REVISED JANUARY 2015

LM35 Precision Centigrade Temperature Sensors

1 Features

- Calibrated Directly in Celsius (Centigrade)
- Linear + 10-mV/°C Scale Factor
- 0.5°C Ensured Accuracy (at 25°C)
- Rated for Full -55°C to 150°C Range
- Suitable for Remote Applications
- Low-Cost Due to Wafer-Level Trimming
- Operates from 4 V to 30 V
- Less than 60-µA Current Drain
- Low Self-Heating, 0.08°C in Still Air
- Non-Linearity Only ±¼°C Typical
- Low-Impedance Output, 0.1 Ω for 1-mA Load

2 Applications

- Power Supplies
- Battery Management
- HVAC
- Appliances

3 Description

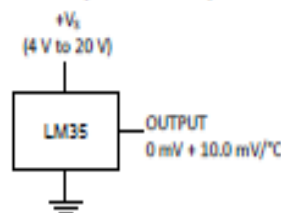
The LM35 series are precision integrated-circuit temperature devices with an output voltage linearly-proportional to the Centigrade temperature. The LM35 device has an advantage over linear temperature sensors calibrated in Kelvin, as the user is not required to subtract a large constant voltage from the output to obtain convenient Centigrade scaling. The LM35 device does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±¼°C over a full -55°C to 150°C temperature range. Lower cost is assured by trimming and calibration at the wafer level. The low-output impedance, linear output, and precise inherent calibration of the LM35 device makes interfacing to readout or control circuitry especially easy. The device is used with single power supplies, or with plus and minus supplies. As the LM35 device draws only 60 µA from the supply, it has very low self-heating of less than 0.1°C in still air. The LM35 device is rated to operate over a -55°C to 150°C temperature range, while the LM35C device is rated for a -40°C to 110°C range (-10° with improved accuracy). The LM35-series devices are available packaged in hermetic TO transistor packages, while the LM35C, LM35CA, and LM35D devices are available in the plastic TO-92 transistor package. The LM35D device is available in an 8-lead surface-mount small-outline package and a plastic TO-220 package.

Device Information⁽¹⁾

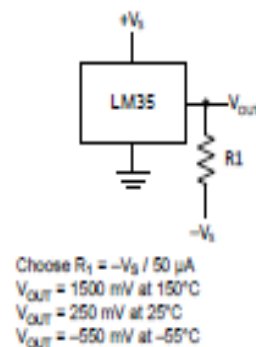
PART NUMBER	PACKAGE	BODY SIZE (NOM)
LM35	TO-CAN (3)	4.699 mm × 4.699 mm
	TO-92 (3)	4.30 mm × 4.30 mm
	SOIC (8)	4.90 mm × 3.91 mm
	TO-220 (3)	14.986 mm × 10.16 mm

(1) For all available packages, see the orderable addendum at the end of the datasheet.

Basic Centigrade Temperature Sensor
(2°C to 150°C)



Full-Range Centigrade Temperature Sensor



An IMPORTANT NOTICE at the end of this data sheet addresses availability, warranty, changes, use in safety-critical applications, intellectual property matters and other important disclaimers. PRODUCTION DATA.

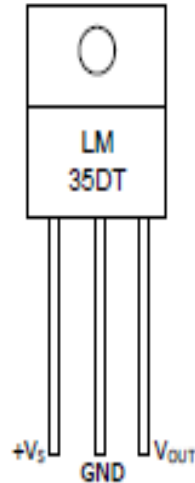
5 Pin Configuration and Functions

NDV Package
3-Pin TO-CAN
(Top View)



Case is connected to negative pin (GND)

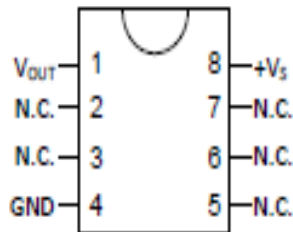
NEB Package
3-Pin TO-220
(Top View)



Tab is connected to the negative pin (GND).

NOTE: The LM35DT pinout is different than the discontinued LM35DP

D Package
8-PIN SOIC
(Top View)



N.C. = No connection

LP Package
3-Pin TO-92
(Bottom View)



Pin Functions

NAME	PIN				TYPE	DESCRIPTION
	TO48	TO92	TO220	SO8		
V _{OUT}	—	—	—	1	O	Temperature Sensor Analog Output
N.C.	—	—	—	2	—	No Connection
	—	—	—	3		
GND	—	—	—	4	GROUND	Device ground pin, connect to power supply negative terminal
N.C.	—	—	—	5	—	No Connection
	—	—	—	6		
	—	—	—	7		
+V _S	—	—	—	8	POWER	Positive power supply pin

6.7 Electrical Characteristics: LM35, LM35C, LM35D Limits

Unless otherwise noted, these specifications apply: $-55^{\circ}\text{C} \leq T_J \leq 150^{\circ}\text{C}$ for the LM35 and LM35A; $-40^{\circ}\text{C} \leq T_J \leq 110^{\circ}\text{C}$ for the LM35C and LM35CA; and $0^{\circ}\text{C} \leq T_J \leq 100^{\circ}\text{C}$ for the LM35D. $V_B = 5\text{ Vdc}$ and $I_{LOAD} = 50\ \mu\text{A}$, in the circuit of [Full-Range Centigrade Temperature Sensor](#). These specifications also apply from 2°C to T_{MAX} in the circuit of [Figure 14](#).

PARAMETER	TEST CONDITIONS	LM35			LM35C, LM35D			UNIT
		TYP	TESTED LIMIT ⁽¹⁾	DESIGN LIMIT ⁽²⁾	TYP	TESTED LIMIT ⁽¹⁾	DESIGN LIMIT ⁽²⁾	
Accuracy, LM35, LM35C ⁽³⁾	$T_A = 25^{\circ}\text{C}$	± 0.4	± 1		± 0.4	± 1		$^{\circ}\text{C}$
	$T_A = -10^{\circ}\text{C}$	± 0.5			± 0.5		± 1.5	
	$T_A = T_{MAX}$	± 0.8	± 1.5		± 0.8		± 1.5	
	$T_A = T_{MIN}$	± 0.8		± 1.5	± 0.8		± 2	
Accuracy, LM35D ⁽³⁾	$T_A = 25^{\circ}\text{C}$				± 0.6	± 1.5		$^{\circ}\text{C}$
	$T_A = T_{MAX}$				± 0.9		± 2	
	$T_A = T_{MIN}$				± 0.9		± 2	
Nonlinearity ⁽⁴⁾	$T_{MIN} \leq T_A \leq T_{MAX}$, $-40^{\circ}\text{C} \leq T_J \leq 125^{\circ}\text{C}$	± 0.3		± 0.5	± 0.2		± 0.5	$^{\circ}\text{C}$
Sensor gain (average slope)	$T_{MIN} \leq T_A \leq T_{MAX}$, $-40^{\circ}\text{C} \leq T_J \leq 125^{\circ}\text{C}$	10	9.8		10		9.8	mV/°C
		10	10.2		10		10.2	
Load regulation ⁽⁵⁾ $0 \leq I_L \leq 1\text{ mA}$	$T_A = 25^{\circ}\text{C}$	± 0.4	± 2		± 0.4	± 2		mV/mA
	$T_{MIN} \leq T_A \leq T_{MAX}$, $-40^{\circ}\text{C} \leq T_J \leq 125^{\circ}\text{C}$	± 0.5		± 5	± 0.5		± 5	
Line regulation ⁽⁵⁾	$T_A = 25^{\circ}\text{C}$	± 0.01	± 0.1		± 0.01	± 0.1		mV/V
	$4\text{ V} \leq V_B \leq 30\text{ V}$, $-40^{\circ}\text{C} \leq T_J \leq 125^{\circ}\text{C}$	± 0.02		± 0.2	± 0.02		± 0.2	
Quiescent current ⁽⁶⁾	$V_B = 5\text{ V}$, 25°C	56	80		56	80		μA
	$V_B = 5\text{ V}$, $-40^{\circ}\text{C} \leq T_J \leq 125^{\circ}\text{C}$	105		158	91		138	
	$V_B = 30\text{ V}$, 25°C	56.2	82		56.2	82		
	$V_B = 30\text{ V}$, $-40^{\circ}\text{C} \leq T_J \leq 125^{\circ}\text{C}$	105.5		161	91.5		141	
Change of quiescent current ⁽⁶⁾	$4\text{ V} \leq V_B \leq 30\text{ V}$, 25°C	0.2	2		0.2	2		μA
	$4\text{ V} \leq V_B \leq 30\text{ V}$, $-40^{\circ}\text{C} \leq T_J \leq 125^{\circ}\text{C}$	0.5		3	0.5		3	
Temperature coefficient of quiescent current	$-40^{\circ}\text{C} \leq T_J \leq 125^{\circ}\text{C}$	0.39		0.7	0.39		0.7	$\mu\text{A}/^{\circ}\text{C}$
Minimum temperature for rate accuracy	In circuit of Figure 14 , $I_L = 0$	1.5		2	1.5		2	$^{\circ}\text{C}$
Long term stability	$T_J = T_{MAX}$, for 1000 hours	± 0.08			± 0.08			$^{\circ}\text{C}$

- (1) Tested Limits are ensured and 100% tested in production.
- (2) Design Limits are ensured (but not 100% production tested) over the indicated temperature and supply voltage ranges. These limits are not used to calculate outgoing quality levels.
- (3) Accuracy is defined as the error between the output voltage and 10 mV/°C times the case temperature of the device, at specified conditions of voltage, current, and temperature (expressed in °C).
- (4) Non-linearity is defined as the deviation of the output-voltage-versus-temperature curve from the best-fit straight line, over the rated temperature range of the device.
- (5) Regulation is measured at constant junction temperature, using pulse testing with a low duty cycle. Changes in output due to heating effects can be computed by multiplying the internal dissipation by the thermal resistance.
- (6) Quiescent current is defined in the circuit of [Figure 14](#).

ANEXO 13. HOJA DE DATOS DEL DS1307

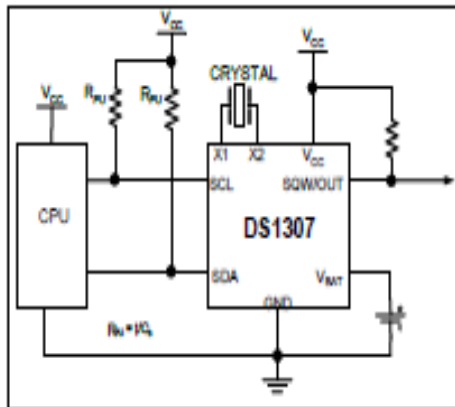


DS1307 64 x 8, Serial, I²C Real-Time Clock

GENERAL DESCRIPTION

The DS1307 serial real-time clock (RTC) is a low-power, full binary-coded decimal (BCD) clock/calendar plus 56 bytes of NV SRAM. Address and data are transferred serially through an I²C, bidirectional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with AM/PM indicator. The DS1307 has a built-in power-sense circuit that detects power failures and automatically switches to the backup supply. Timekeeping operation continues while the part operates from the backup supply.

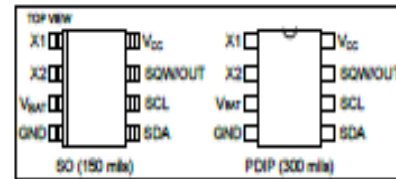
TYPICAL OPERATING CIRCUIT



BENEFITS AND FEATURES

- Completely Manages All Timekeeping Functions
 - Real-Time Clock Counts Seconds, Minutes, Hours, Date of the Month, Month, Day of the Week, and Year with Leap-Year Compensation Valid Up to 2100
 - 56-Byte, Battery-Backed, General-Purpose RAM with Unlimited Writes
 - Programmable Square-Wave Output Signal
- Simple Serial Port Interfaces to Most Microcontrollers
 - I²C Serial Interface
- Low Power Operation Extends Battery Backup Run Time
 - Consumes Less than 500nA in Battery-Backup Mode with Oscillator Running
 - Automatic Power-Fall Detect and Switch Circuitry
- 8-Pin DIP and 8-Pin SO Minimizes Required Space
- Optional Industrial Temperature Range: -40°C to +85°C Supports Operation in a Wide Range of Applications
- Underwriters Laboratories® (UL) Recognized

PIN CONFIGURATIONS



ORDERING INFORMATION

PART	TEMP RANGE	VOLTAGE (V)	PIN-PACKAGE	TOP MARK*
DS1307+	0°C to +70°C	5.0	8 PDIP (300 mils)	DS1307
DS1307N+	-40°C to +85°C	5.0	8 PDIP (300 mils)	DS1307N
DS1307Z+	0°C to +70°C	5.0	8 SO (150 mils)	DS1307
DS1307ZN+	-40°C to +85°C	5.0	8 SO (150 mils)	DS1307N
DS1307Z+T&R	0°C to +70°C	5.0	8 SO (150 mils) Tape and Reel	DS1307
DS1307ZN+T&R	-40°C to +85°C	5.0	8 SO (150 mils) Tape and Reel	DS1307N

+Denotes a lead-free/RoHS-compliant package.

A "" anywhere on the top mark indicates a lead-free package. An "N" anywhere on the top mark indicates an industrial temperature range device. Underwriters Laboratories, Inc. is a registered certification mark of Underwriters Laboratories, Inc.

OSCILLATOR CIRCUIT

The DS1307 uses an external 32.768kHz crystal. The oscillator circuit does not require any external resistors or capacitors to operate. Table 1 specifies several crystal parameters for the external crystal. Figure 1 shows a functional schematic of the oscillator circuit. If using a crystal with the specified characteristics, the startup time is usually less than one second.

CLOCK ACCURACY

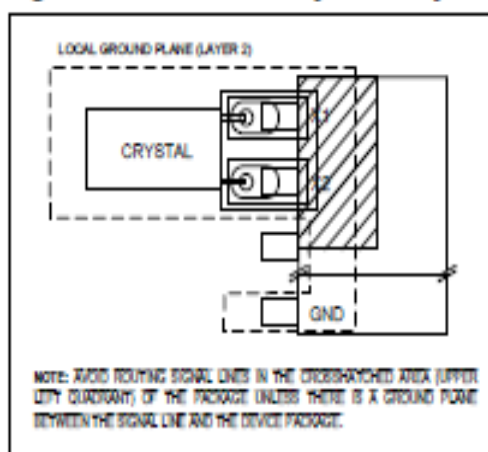
The accuracy of the clock is dependent upon the accuracy of the crystal and the accuracy of the match between the capacitive load of the oscillator circuit and the capacitive load for which the crystal was trimmed. Additional error will be added by crystal frequency drift caused by temperature shifts. External circuit noise coupled into the oscillator circuit may result in the clock running fast. Refer to Application Note 58: Crystal Considerations with Dallas Real-Time Clocks for detailed information.

Table 1. Crystal Specifications*

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS
Nominal Frequency	f_0		32.768		kHz
Series Resistance	ESR			45	k Ω
Load Capacitance	C_L		12.5		pF

*The crystal, traces, and crystal input pins should be isolated from RF generating signals. Refer to Application Note 58: Crystal Considerations for Dallas Real-Time Clocks for additional specifications.

Figure 2. Recommended Layout for Crystal



RTC AND RAM ADDRESS MAP

Table 2 shows the address map for the DS1307 RTC and RAM registers. The RTC registers are located in address locations 00h to 07h. The RAM registers are located in address locations 08h to 3Fh. During a multibyte access, when the address pointer reaches 3Fh, the end of RAM space, it wraps around to location 00h, the beginning of the clock space.

CLOCK AND CALENDAR

The time and calendar information is obtained by reading the appropriate register bytes. Table 2 shows the RTC registers. The time and calendar are set or initialized by writing the appropriate register bytes. The contents of the time and calendar registers are in the BCD format. The day-of-week register increments at midnight. Values that correspond to the day of week are user-defined but must be sequential (i.e., if 1 equals Sunday, then 2 equals Monday, and so on.) Illogical time and date entries result in undefined operation. Bit 7 of Register 0 is the clock halt (CH) bit. When this bit is set to 1, the oscillator is disabled. When cleared to 0, the oscillator is enabled. On first application of power to the device the time and date registers are typically reset to 01/01/00 01 00:00:00 (MM/DD/YY DOW HH:MM:SS). The CH bit in the seconds register will be set to a 1. The clock can be halted whenever the timekeeping functions are not required, which minimizes current (I_{BATTDR}).

The DS1307 can be run in either 12-hour or 24-hour mode. Bit 6 of the hours register is defined as the 12-hour or 24-hour mode-select bit. When high, the 12-hour mode is selected. In the 12-hour mode, bit 5 is the AM/PM bit with logic high being PM. In the 24-hour mode, bit 5 is the second 10-hour bit (20 to 23 hours). The hours value must be re-entered whenever the 12/24-hour mode bit is changed.

When reading or writing the time and date registers, secondary (user) buffers are used to prevent errors when the internal registers update. When reading the time and date registers, the user buffers are synchronized to the internal registers on any I²C START. The time information is read from these secondary registers while the clock continues to run. This eliminates the need to re-read the registers in case the internal registers update during a read. The divider chain is reset whenever the seconds register is written. Write transfers occur on the I²C acknowledge from the DS1307. Once the divider chain is reset, to avoid rollover issues, the remaining time and date registers must be written within one second.

Table 2. Timekeeper Registers

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	FUNCTION	RANGE
00h	CH	10 Seconds			Seconds				Seconds	00–59
01h	0	10 Minutes			Minutes				Minutes	00–59
02h	0	12	10 Hour	10 Hour	Hours			Hours	1–12 +AM/PM 00–23	
		24	PM/ AM							
03h	0	0	0	0	0	DAY		Day	01–07	
04h	0	0	10 Date		Date			Date	01–31	
05h	0	0	0	10 Month	Month			Month	01–12	
06h	10 Year			Year				Year	00–99	
07h	OUT	0	0	SQWE	0	0	RS1	RS0	Control	—
08h–3Fh								RAM 56 x 8		00h–FFh

0 = Always reads back as 0.

ANEXO 14. HOJA DE DATOS DEL PIR SENSOR



Web Site: www.parallax.com
Forums: forums.parallax.com
Sales: sales@parallax.com
Technical: support@parallax.com

Office: (916) 624-8333
Fax: (916) 624-8003
Sales: (888) 512-1024
Tech Support: (888) 997-4267

PIR Sensor (#555-28027)

General Description

The PIR (Passive Infra-Red) Sensor is a pyroelectric device that detects motion by measuring changes in the infrared levels emitted by surrounding objects. This motion can be detected by checking for a high signal on a single I/O pin.

Features

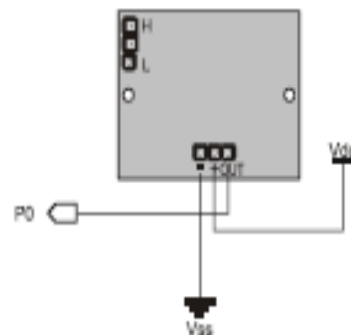
- Single bit output
- Small size makes it easy to conceal
- Compatible with all Parallax microcontrollers
- 3.3V & 5V operation with <100uA current draw

Application Ideas

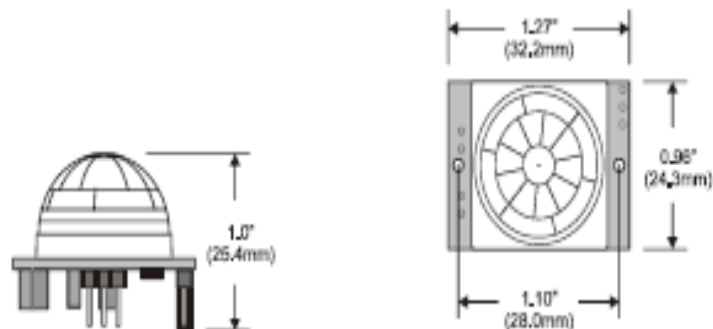
- Alarm Systems
- Halloween Props

Quick Start Circuit

Note: The sensor is active high when the jumper (shown in the upper left) is in either position.



Module Dimensions



Theory of Operation

Pyroelectric devices, such as the PIR sensor, have elements made of a crystalline material that generates an electric charge when exposed to infrared radiation. The changes in the amount of infrared striking the element change the voltages generated, which are measured by an on-board amplifier. The device contains a special filter called a Fresnel lens, which focuses the infrared signals onto the element. As the ambient infrared signals change rapidly, the on-board amplifier trips the output to indicate motion.

Pin Definitions and Ratings

Pin	Name	Function
-	GND	Connects to Ground or Vss
+	V+	Connects to Vdd (3.3V to 5V) @ ~100uA
OUT	Output	Connects to an I/O pin set to INPUT mode (or transistor/MOSFET)

Jumper Setting

Position	Mode	Description
H	Retrigger	Output remains HIGH when sensor is retriggered repeatedly. Output is LOW when idle (not triggered).
L	Normal	Output goes HIGH then LOW when triggered. Continuous motion results in repeated HIGH/LOW pulses. Output is LOW when idle.

Connecting and Testing

Connect the 3-pin header to your circuit so that the minus (-) pin connects to ground or Vss, the plus (+) pin connects to Vdd and the OUT pin connects to your microcontroller's I/O pin. One easy way to do this would be to use a standard servo/LCD extension cable, available separately from Parallax (#805-00002). This cable makes it easy to plug sensor into the servo headers on our Board Of Education or Professional Development Board. If you use the Board Of Education, be sure the servo voltage jumper (located between the 2 servo header blocks) is in the Vdd position, not Vin. If you do not have this jumper on your board you should manually connect to Vdd through the breadboard. You may also plug the sensor directly into the edge of the breadboard and connect the signals from there. Remember the position of the pins when you plug the sensor into the breadboard.

Calibration

The PIR Sensor requires a 'warm-up' time in order to function properly. This is due to the settling time involved in 'learning' its environment. This could be anywhere from 10-60 seconds. During this time there should be as little motion as possible in the sensors field of view.

Sensitivity

The PIR Sensor has a range of approximately 20 feet. This can vary with environmental conditions. The sensor is designed to adjust to slowly changing conditions that would happen normally as the day progresses and the environmental conditions change, but responds by making its output high when sudden changes occur, such as when there is motion.

Resources and Downloads

Check out the PIR Sensor product page for example programs and more:

http://www.parallax.com/detail.asp?product_id=555-28027

Source Code

BASIC Stamp® 1 Program

This program will display the current status of the output pin from the PIR Sensor connected to P0 by lighting an active high LED connected to P1 when motion is detected.

```
' -----  
' File..... PIR_Simple.bs1  
' Purpose... Show Output State Of PIR Sensor  
' Author.... Parallax, Inc.  
' E-mail.... support@parallax.com  
' Started... 12-14-2005  
' {$STAMP B81}  
' {$PBASIC 1.0}  
'  
' -----[ Program Description ]-----  
  
' This program displays the current state of the PIR Sensor connected to P0  
' by lighting an active high LED connected to P1 when motion is detected.  
  
' -----[ I/O Definitions ]-----  
  
SYMBOL      PIR =   PIN0          ' I/O Pin For PIR Sensor  
SYMBOL      LED =   PIN1          ' I/O Pin For LED  
  
' -----[ Initialization ]-----  
  
LET          DIRS = %00000010    ' Set Pin Directions  
  
' -----[ Program Code ]-----  
  
Main:  
  LET LED = PIR  
  GOTO Main
```

BASIC Stamp® 2 Program

This program will display the current status of the output pin from the PIR Sensor connected to P0 using the Debug Terminal.

```
' -----  
' File..... PIR_Simple.bs2  
' Purpose... Show Output State Of PIR Sensor  
' Author.... Parallax, Inc.  
' E-mail.... support@parallax.com  
' Started... 12-14-2005  
' {$STAMP B82}  
' {$PBASIC 2.5}  
'  
' -----[ Program Description ]-----  
  
' This program displays the current state of the PIR Sensor connected to P0  
' on the DEBUG screen.
```

ANEXO 15. PRÁCTICA DE LABORATORIO 1.1: CONFIGURACIÓN DE LOS MÓDULOS XBEE CON TOPOLOGÍA MESH.

Universidad nacional de Chimborazo
Facultad de ingeniería
Entrenador de Microcontroladores para Comunicaciones
Manipulación del Módulo Xbee



Práctica de Laboratorio 1.1: Configuración de los módulos Xbee con topología mesh.

Topología:

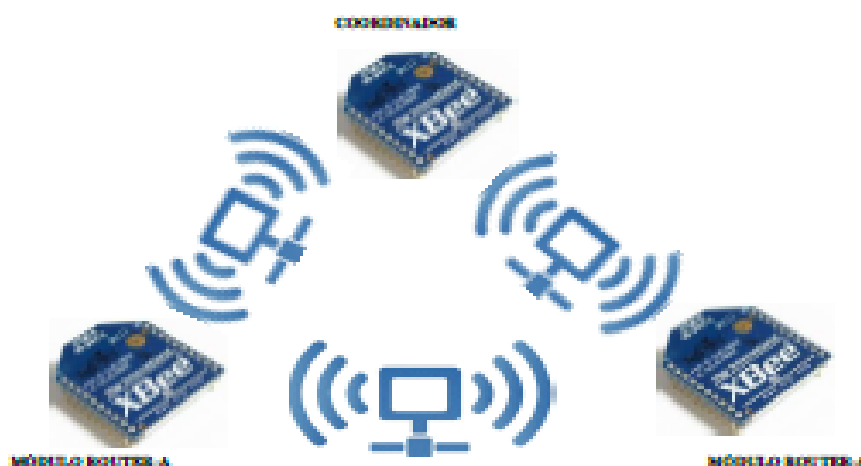


Figura 1.1: configuración en red mesh

Objetivos de Aprendizaje:

- Manejo del software X-CTU.
- Configuración del módulo Xbee con topología mesh.
- Realizar envío de mensajes.

Información básica/Situación.

El desarrollo de esta práctica explora conocimientos básicos necesarios para configurar los módulos Xbee Serie 1, una topología mesh coordinador (administrador de la red), Router-A (Sensores/Actuador), Router-B (Sensores/Actuadores).

Los módulos Xbee son soluciones integrales que brindan el servicio de una comunicación inalámbrica entre dispositivos. Estos módulos utilizan protocolos de red llamada IEEE 802.15.4 para crear redes como son: punto a punto, punto a multi punto y de tipo mesh. Son dispositivos que tienen baja latencia y una sincronización de comunicación muy aceptable, son de fácil manejo y rápida configuración para cualquier tipo de topología.



Modo comandos AT

Este modo permite ingresar comandos al módulo xbee, para configurar los distintos parámetros de acuerdo a las necesidades del usuario, es necesario utilizar el hyper terminal de Windows para tener conectividad entre el módulo Xbee y la PC, ver **Tabla 1**.

COMANDOS AT	DESCRIPCIÓN
+++	Entrar en modo de comandos AT
ATND	Verificar, módulos conectados a las Red

Tabla 1.1. Comandos AT

- **Descripción de los pines más importantes de Xbee.**

El Pin uno va conectado al sistema de alimentación requerida 3.3v, Pin 10 a GND, Pin dos transmisor, Pin tres receptor y Pin cinco reset.

- **Software X-CTU**

Software gratuito que se utiliza como interfaz para configurar el módulo Xbee a través de la PC, que proviene de la empresa Digi, entre sus partes elementales son:

- Configuraciones del PC.- Sirve para seleccionar el puerto serial por la cual se comunica el módulo Xbee con la PC.
- Opción terminal.- Pantalla que sirve para leer datos recibidos y enviados por el resto de módulos que están conectados.

Recursos Necesarios:

- 3 Cables USB-Serial
- Software X-CTU
- 3 PC.
- 3 entrenadores con módulos Xbee.

Desarrollo:

Parte 1: Instalar driver del Usb.

Paso 1: Instalar el driver del Usb para el reconocimiento del COM.

No todos los Sistemas operativos de Windows reconocen el cable serial USB por lo que es necesario instalar manualmente **Figura 1.2**.

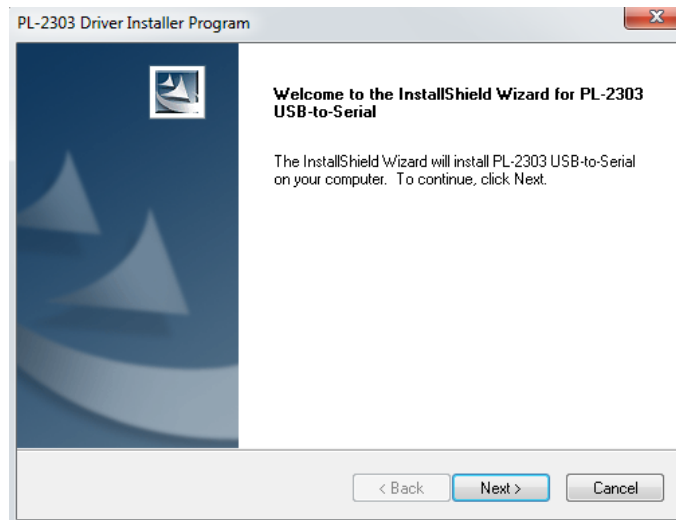


Figura 1.2:Instalación del driver-usb

Paso 2: Examinar el puerto serial COM.

Conectar el cable Usb a la PC y verificar el número de COM, **Figura 1.3**, que asigna la PC para la conexión del módulo Xbee

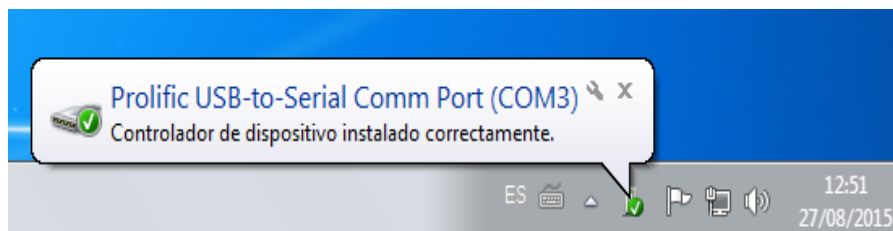


Figura 1.3: Reconocimiento del COM.

Parte 2: Instalación del software X-CTU en el PC.

Esta parte corresponde a la instalación del software X-CTU para la configuración de los módulos.

Paso 1: Instalar el software X-CTU.

a.- Es recomendable instalar la versión antigua del Software X-CTU, **Figura 1.4** para su fácil manejo.

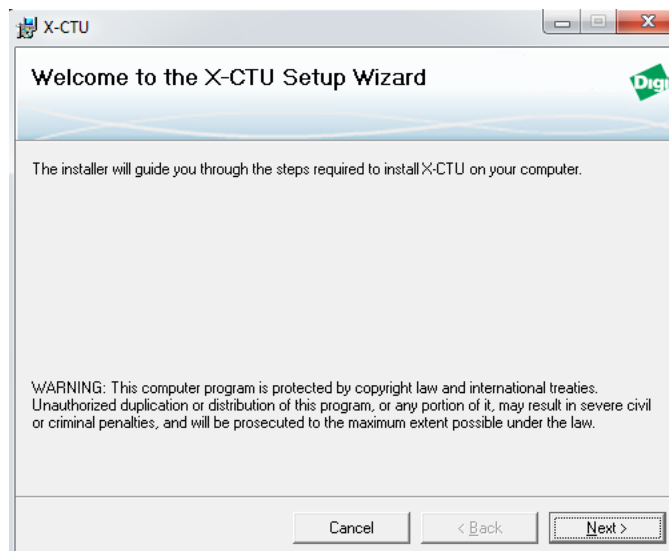


Figura 1.4: Ventana de instalación del X-CTU

b.- Examinar el programa que este bien instalado y listo para la conexión del módulo con la PC **Figura 1.5**

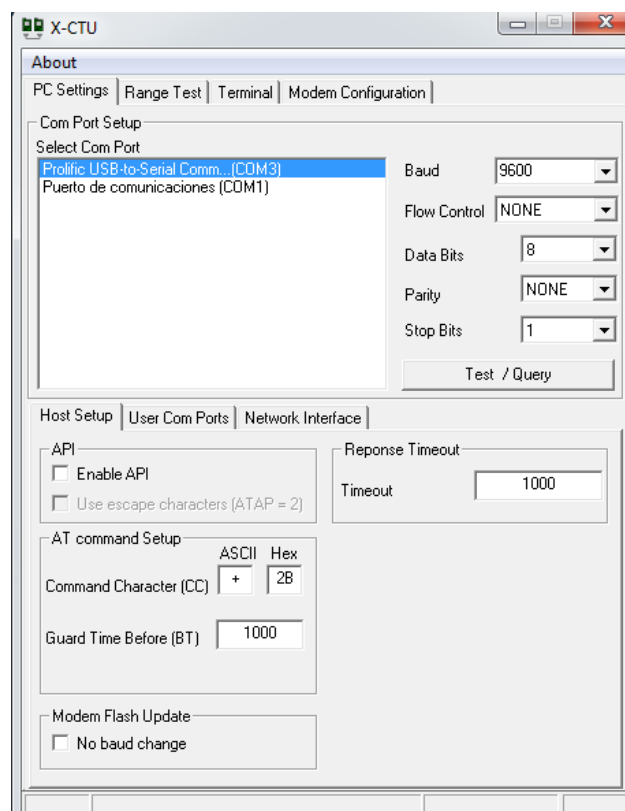


Figura 1.5: Verificación del X-CTU

Parte 3: Parte elemental de la practica

En este paso se realizar el desarrollo completo de la práctica una vez instalado todos los drivers necesarios.

Paso 1: Conexiones módulo Xbee con la PC.

El módulo Xbee se conecta con la PC a través de interfaz cable USB-Serial, cable transmisor de USB conectado al pin tres, cable receptor de USB conectado al pin dos y cable GND de USB conectado al GND del módulo Xbee como se muestra en la **Figura 1.6**:



Figura 1.6: Módulo encendido y conectado al pc

Paso 2. Testeo y borrado de los módulos.

Estos pasos se realiza para los tres módulos Xbee a configurar.

a.- Test del puerto COM

Abrir el software X-CTU, el mismo programa asigna puertos del cable USB-Serial. Como se muestra en la **Figura 1.7**:

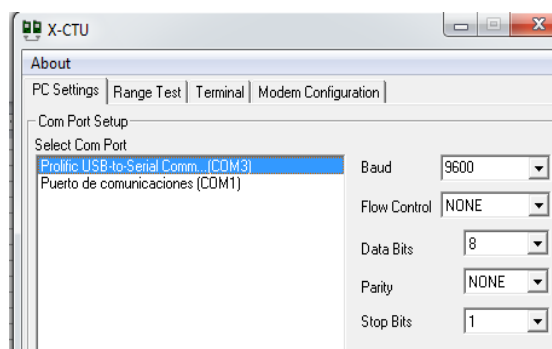


Figura 1.7: Asignación del COM

b.- Test del puerto COM

El botón **Test/Query** sirve para verificar la conexión del módulo con la Pc, saldrá el número de serie del módulo, de esa manera se asegura que el Xbee está listo para realizar la configuración, ver **Figura 1.8**.

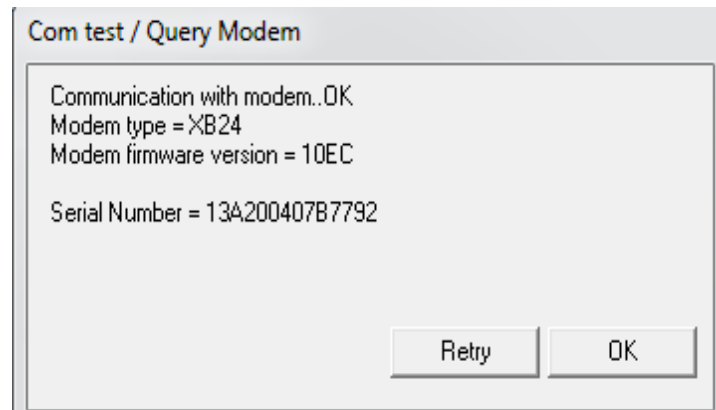


Figura 1.8: Asignación del COM

c.- Pestaña de Modem Configuration.

La pestaña Read sirve para ver la configuración por modo de fábrica en caso de ser nuevo o utilizando anteriormente **Figura 1.9**.

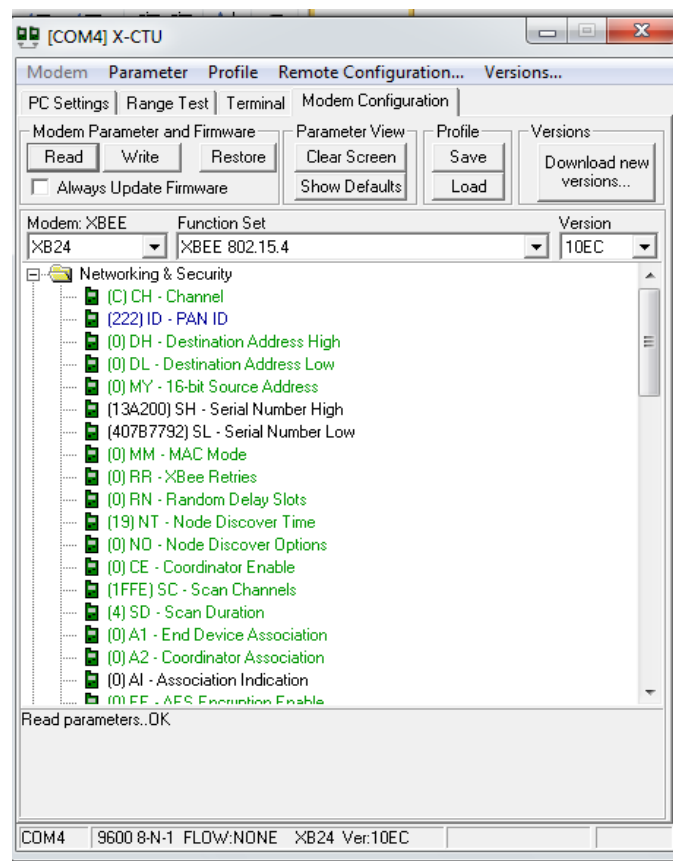


Figura 1.9: Lectura del módulo

d.- Borrador de la configuración.

Click en Clear Screen, esta pestaña borra la configuración existente en el Xbee, ver **Figura 1.10**.

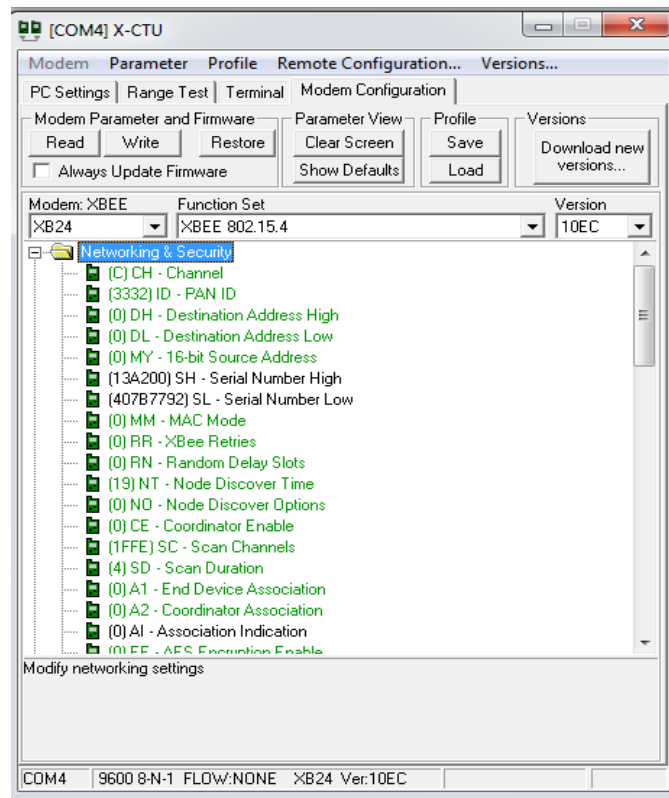


Figura 1.10: Borrado del módulo

e.- Restauración del módulo

Clic en restore para restaurar el módulo a modo de fábrica y quede listo para nueva configuración, ver **Figura 1.11**

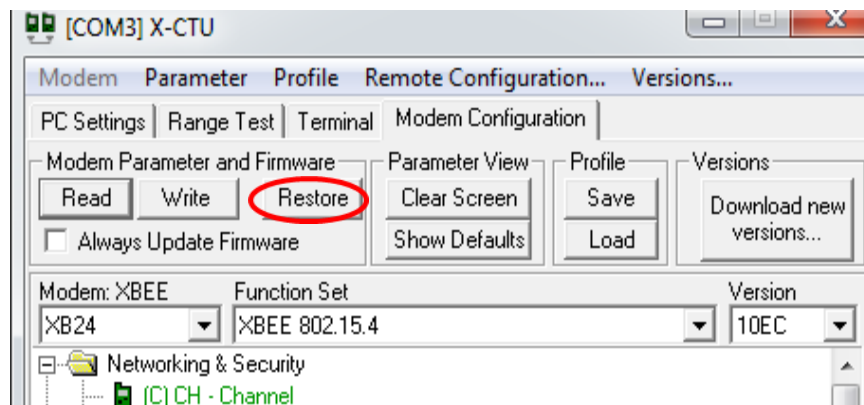


Figura 1.11: Restaura el módulo

Parte 4. Configuración del módulo Xbee coordinador

Se ingresa los valores de la **Tabla 1.2** para que módulo pueda trabajar como coordinador.

INDICADOR	NOMBRE	VALOR
NI	Nodo identificador	Coordinador
ID	PAN ID	17
MY	Dirección de red	10
SD	Dirección de escaneo	3
CH	Canal de operación	11
SC	Examina solo el canal elegido	40
DH	Dirección de destino en alto	0
DL	Dirección de destino en bajo	FFFF
SH	Número de serie en alto	13A200
SL	Número de serie en bajo	

Figura 1.2: Configuración del módulo Xbee coordinador

Los datos más relevaste que deben ser ingresados en el módulo son todos los datos de la Tabla 1.2. Para su correcto funcionamiento como se muestra en la **Figura 1.12.**

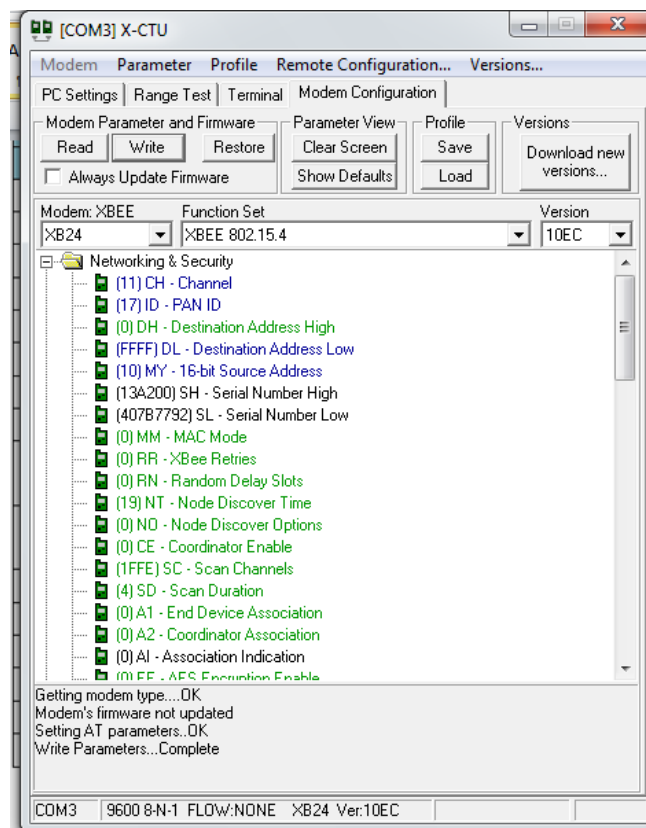


Figura 1.12: Grabación finalizada exitosamente

Una vez ingresado todos los valores clic en **Write** para grabar y saldrá un mensaje “Write Parameters...Complete OK” en la parte inferior del terminal X-CTU., como se muestra en la **Figura 1.13**.



Figura 1.13: Grabación finalizada exitosamente

Parte 5. Configuración del módulo Xbee como Router-A

A este dispositivo se configura como Router-A, para que tenga conectividad con el coordinador y el Router-B.

Paso 1. Ingreso de valores a configurar

Se ingresa los valores de la **Tabla 1.3** para que módulo pueda trabajar como Router-A.

INDICADOR	NOMBRE	VALOR
NI	Nodo identificador	Router-A
ID	PAN ID	17
MY	Dirección de red	10
SD	Dirección de escaneo	3
CH	Canal de operación	11
SC	Examina solo el canal elegido	40
DH	Dirección de destino en alto	0
DL	Dirección de destino en bajo	0
SH	Número de serie en alto	13A200
SL	Número de serie en bajo	409F2A73

Figura 1.3: Configuración del módulo Xbee Router-A

Los datos más relevaste que deben ser ingresados en el módulo son todos los datos de la Tabla 1.3. Para su correcto funcionamiento como se muestra en la **Figura 1.14**.

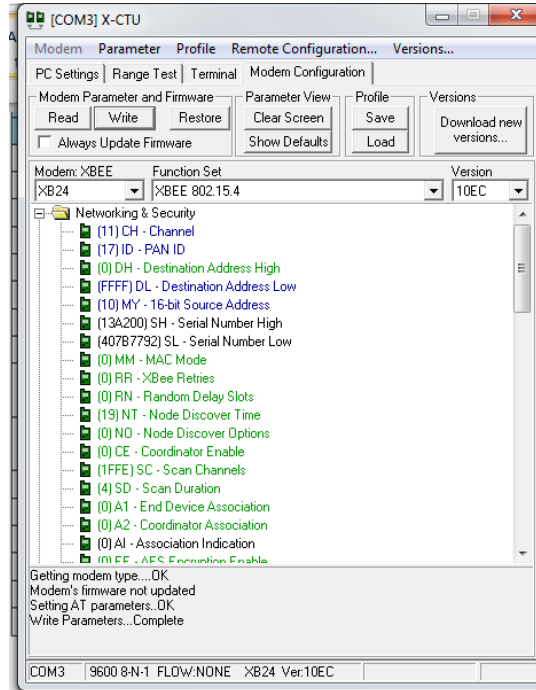


Figura 1.14: Grabación finalizada exitosamente

Una vez ingresado todos los valores clic en **Write** para grabar y saldrá un mensaje “Write Parameters...Complete OK” en la parte inferior del terminal X-CTU., como se muestra en la **Figura 1.15**.



Figura 1.15: Grabación finalizada exitosamente

Parte 6. Configuración del módulo Xbee como Router-B

A este módulo se configura como Router-B Utilizando los mismos pasos de la **PARTE 5**, excepto la tabla de valores de la configuración que es remplazada por la Tabla

INDICADOR	NOMBRE	VALOR
NI	Nodo identificador	Router-B

ID	PAN ID	17
MY	Dirección de red	10
SD	Dirección de escaneo	3
CH	Canal de operación	11
SC	Examina solo el canal elegido	40
DH	Dirección de destino en alto	0
DL	Dirección de destino en bajo	0
SH	Número de serie en alto	13A200
SL	Número de serie en bajo	406F2A83

Figura 1.16: Configuración del módulo Xbee Router-B

Parte 7. Funcionamiento

El funcionamiento de cada una de las placas se muestra en la **Figura 1.17**, dando como resultado el led de color naranja encendiendo cada vez que recibe datos de los módulos que están conectados a la red.



Figura 1.17. Led de aviso, recepción de datos

La **Figura 1.18**, muestra la ventana Terminal del software X-CTU que sirve para enviar comandos AT al módulo, como también escribir y leer mensajes enviados y recibidos por los módulos.

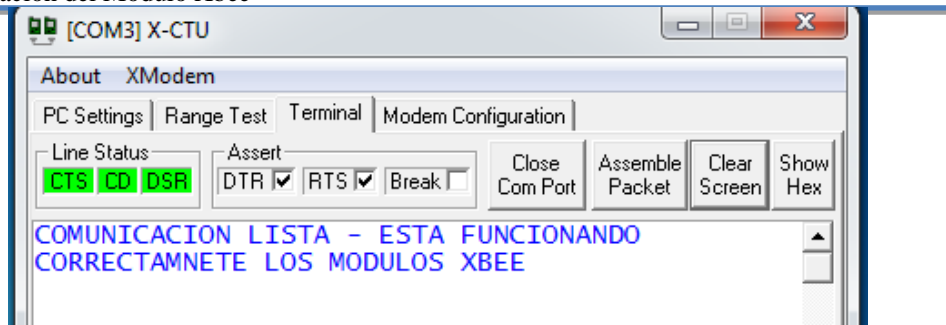


Figura 1.18: Módulo Xbee coordinador

La Figura 1.19, imprime los datos recibidos por el Router-A

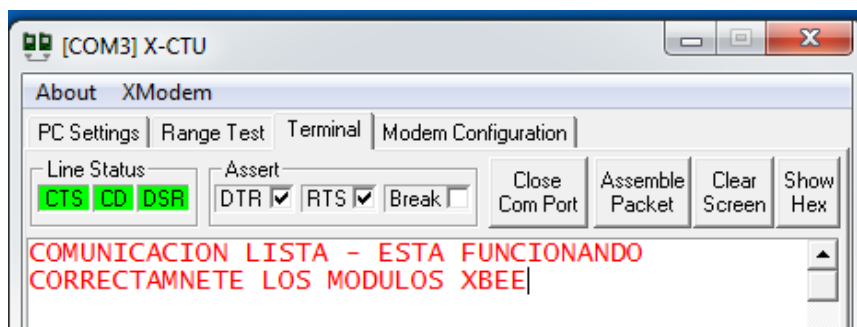


Figura 1.19: Recepción de datos

La Figura 1.20, imprime los datos recibidos por el Router-B

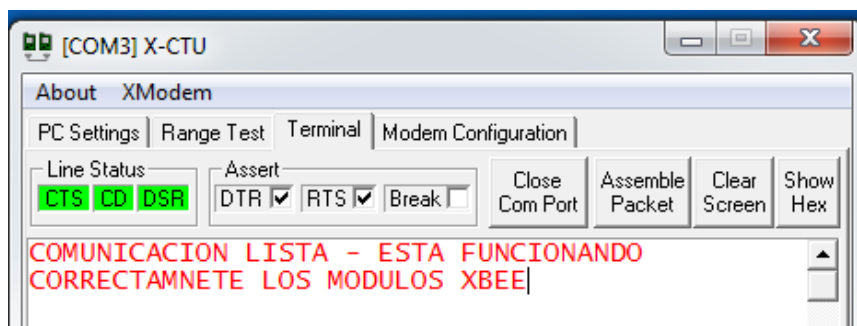
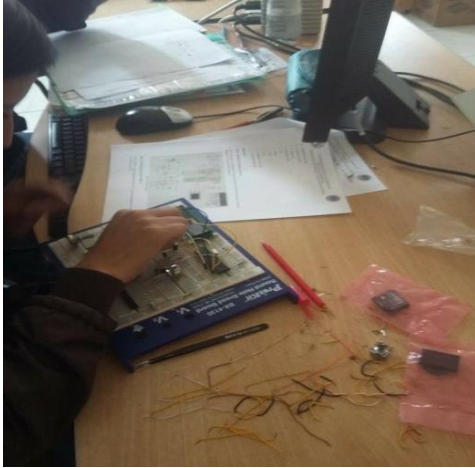


Figura 1.19: Recepción de datos

ANEXO 16. IMÁGENES OBTENIDAS EN LA COMPROBACIÓN DE HIPÓTESIS

➤ Pruebas realizadas sin el entrenador

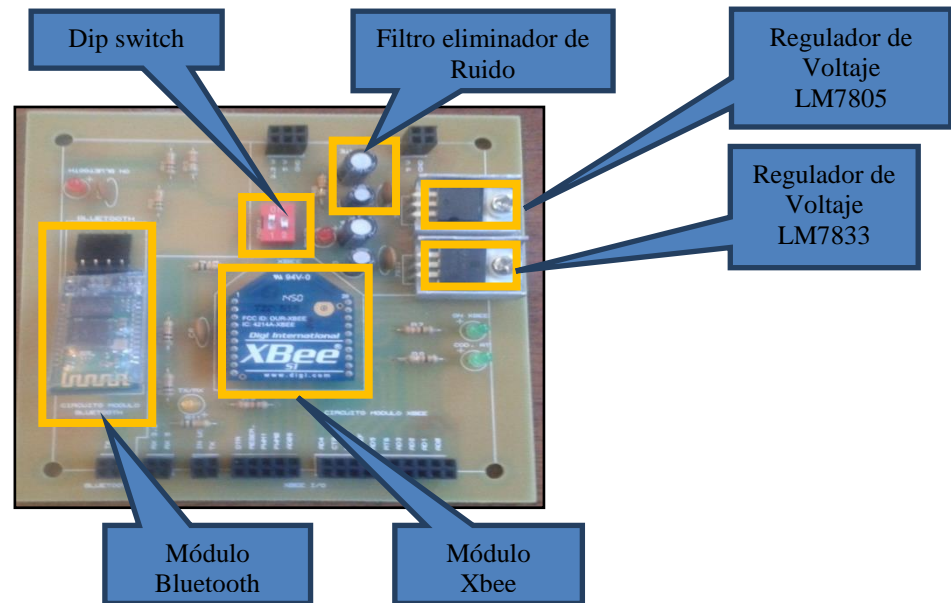


➤ Pruebas realizadas con el entrenador

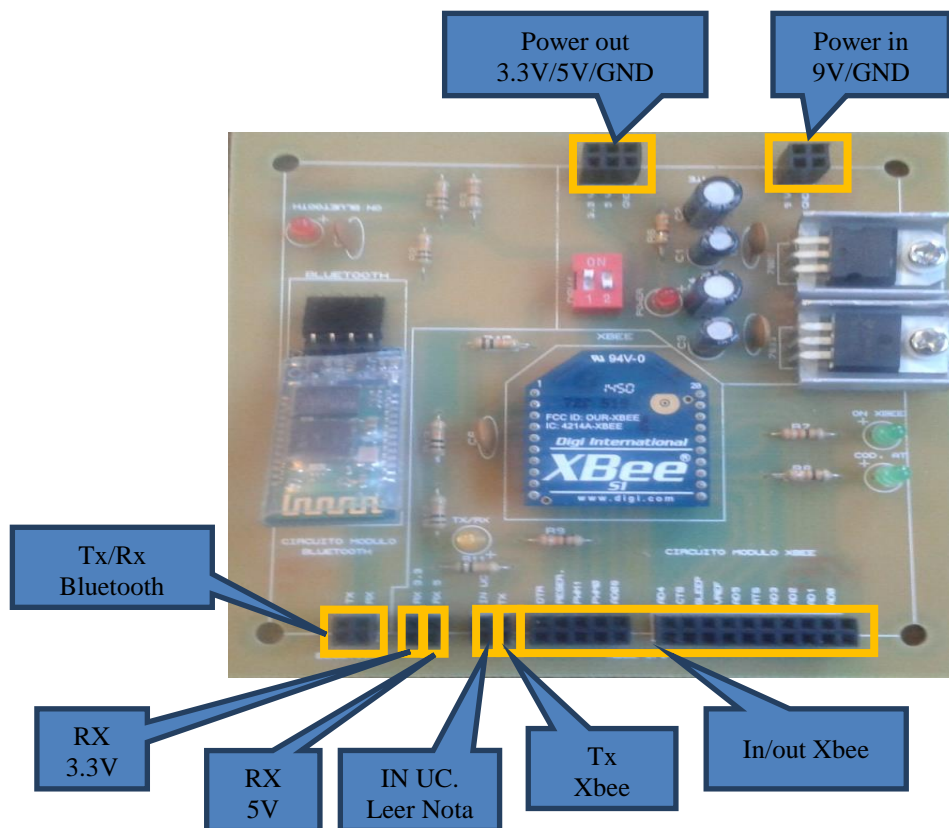


ANEXO 17. DESCRIPCIÓN DE LA PLACA DE COMUNICACIONES

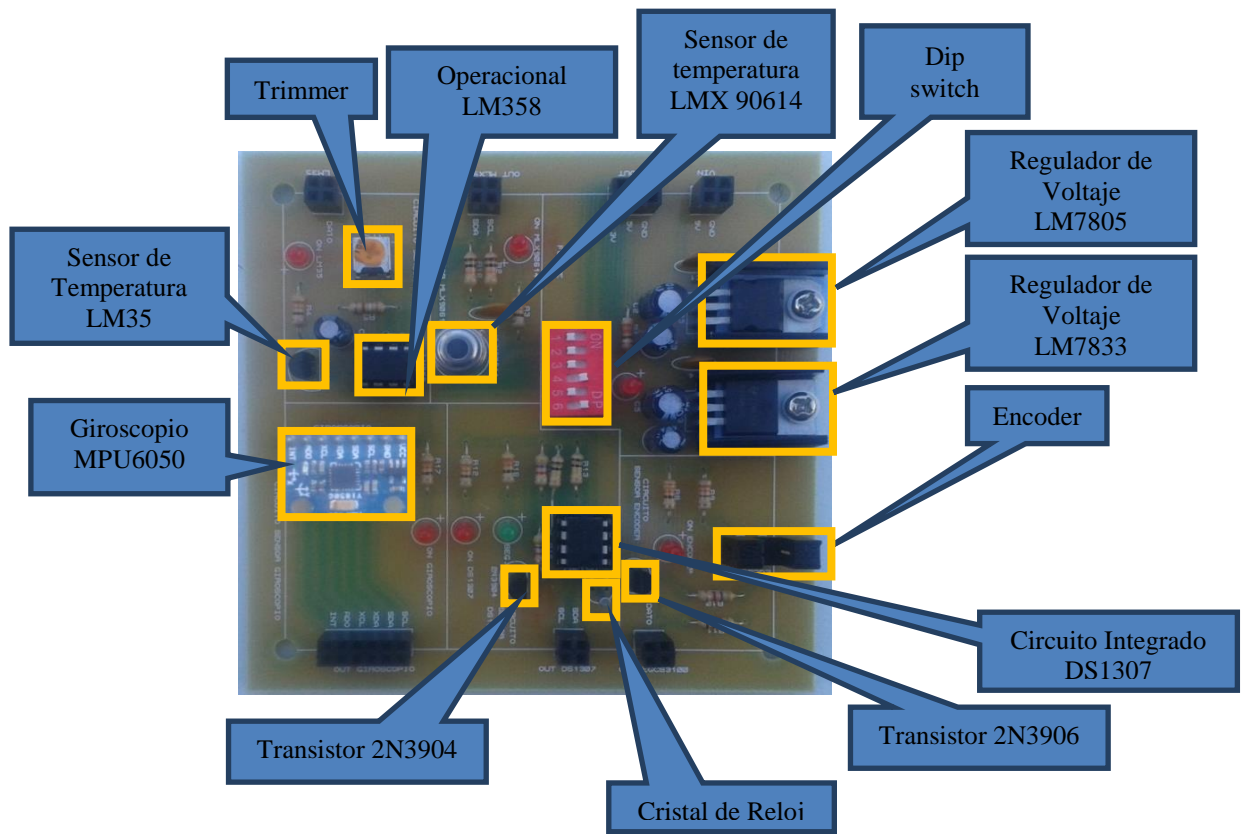
- Descripción de la placa de comunicaciones



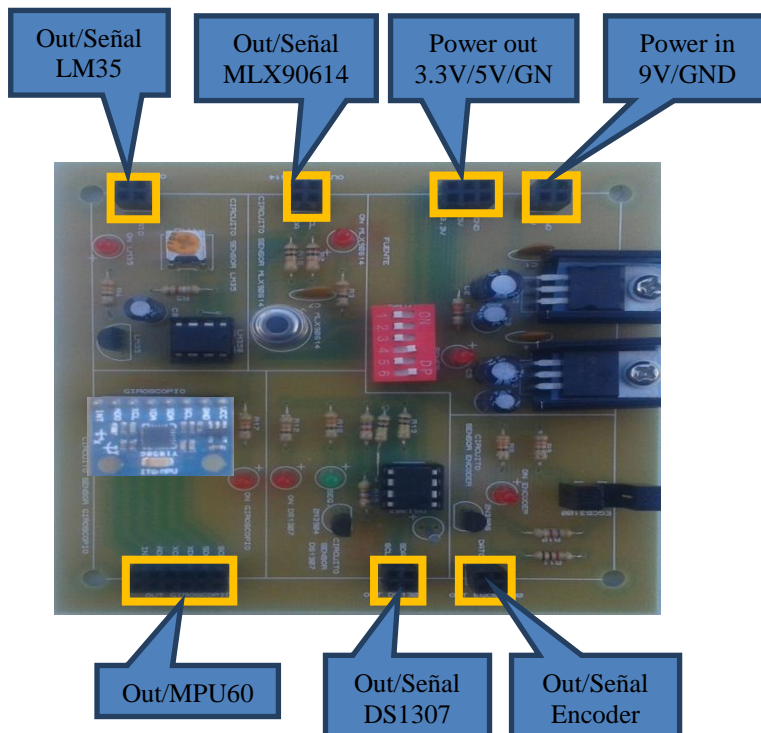
- Salidas y entradas de la placa de comunicaciones



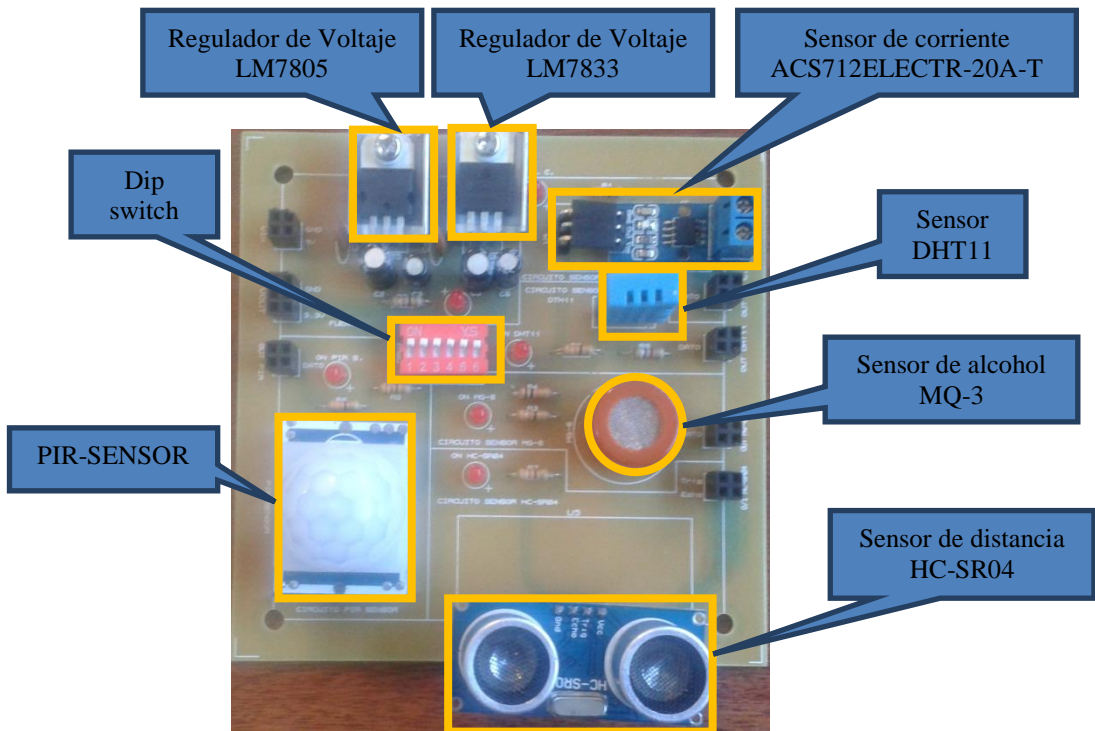
- **Descripción placa uno de sensores**



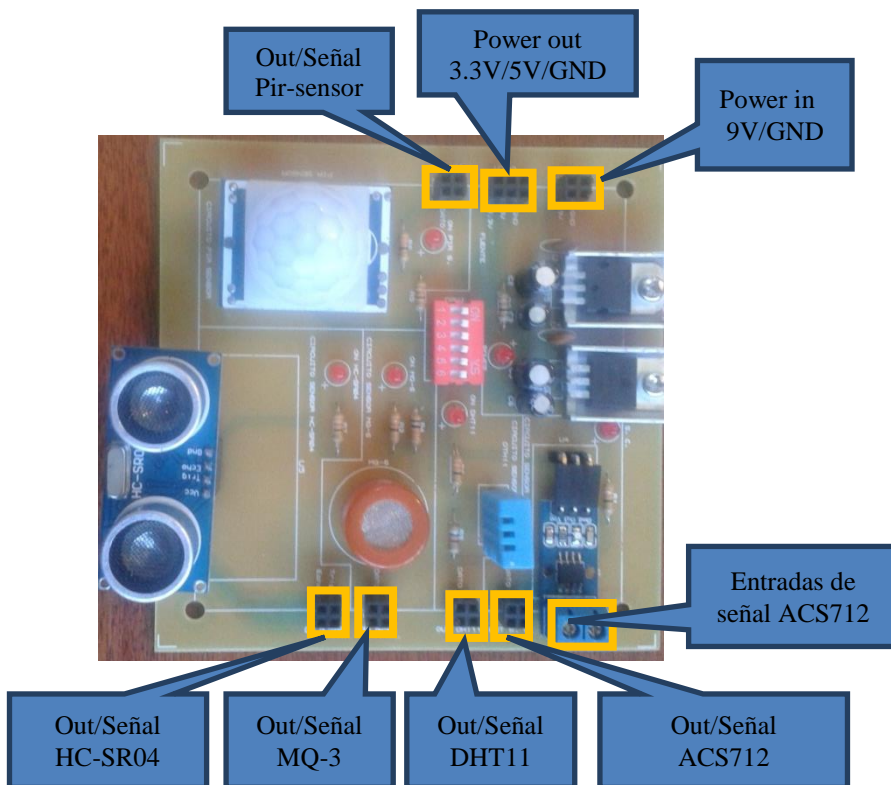
- **Salidas y entradas de la placa uno de sensores**



- **Descripción placa dos de sensores**



- **Salidas y entradas de la placa dos de sensores**



ANEXO 18. PRACTICA PARA LA COMPROBACION DE HIPOTESIS SIN ENTRENADOR

Universidad nacional de Chimborazo
Facultad de ingeniería
Entrenador de Microcontroladores para Comunicaciones
Manipulación del Módulo Xbee



Práctica de Laboratorio 1.5: lectura de temperatura en PC mediante la comunicación de módulos Xbee sin entrenador.

Topología:



Figura 5.1: Red Punto a Punto

Objetivos de Aprendizaje:

- Manejo del software X-CTU.
- Configuración del módulo Xbee, punto a punto.
- Realizar envío de mensajes.

Información básica/Situación.

Zigbee es un protocolo de comunicaciones inalámbrico basado en el estándar de comunicaciones para redes inalámbricas IEEE_802.15.4. Zigbee permite que dispositivos electrónicos de bajo consumo puedan realizar sus comunicaciones inalámbricas. Es especialmente útil para redes de sensores en entornos industriales, médicos y, sobre todo, domóticas

- Descripción de los pines más importantes de Xbee.

El Pin uno va conectado al sistema de alimentación requerida 3.3v, Pin 10 a GND, Pin dos transmisor, Pin tres receptor y Pin cinco a una resistencia de 10kΩ.



- **Software X-CTU**

Software gratuito que se utiliza como interfaz para configurar el módulo Xbee a través de la PC, que proviene de la empresa Digi.

Recursos Necesarios:

- 1 Cables USB-Serial
- Software X-CTU
- 1 PC
- 2 Módulos Xbee
- 1 16f877A
- 4 Capacitores de 100 uF
- 4 Capacitores de 10 uF
- 1 Capacitor de 22uF
- 2 LM7805
- 2 LM7833
- 1 104
- 4 Resistencias de 10K Ω
- 1 Resistencia de 3.9K Ω
- 3 Resistencias de 330 Ω
- 3 Led
- 1 LM35
- 1 LM358
- 1 Potenciómetro de 20 K Ω

Desarrollo:

Parte 1: Instalación del driver Usb

- No todos los Sistemas operativos de Windows reconocen el cable serial USB por lo que es necesario instalar manualmente.
- Conectar el cable Usb a la PC y verificar el número de COM, que asigna la PC para la conexión del módulo Xbee.

Parte 2: Instalación del software X-CTU en el PC.

Esta parte corresponde a la instalación del software X-CTU para la configuración de los módulos.

Paso 1: Instalar el software X-CTU.

- a.- Es recomendable instalar la versión antigua del Software X-CTU, **Figura 5.2**, para su fácil manejo.

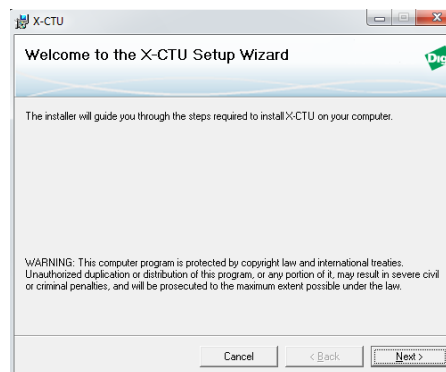


Figura 5.2: Ventana de instalación del X-CTU

- b.- Examinar el programa que este bien instalado y listo para la conexión del módulo con la PC **Figura 5.3**.

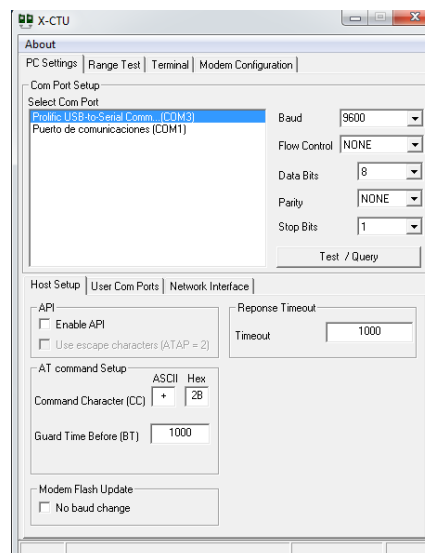


Figura 5.3: Verificación del X-CTU

Parte 3. Diseño del circuito en Proteus

Es una red Punto a Punto por lo tanto es necesario realizar dos circuitos; Transmisor y Receptor.

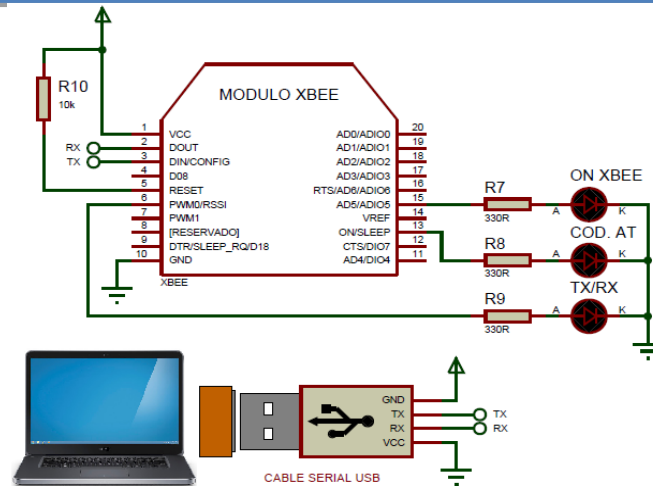


Figura 5.5: Circuito receptor

Paso 3. Armado de Circuito

Realizar conexiones en un ProtoBoard todos los elementos tanto para el transmisor como para el receptor ver **Figura 5.6**.

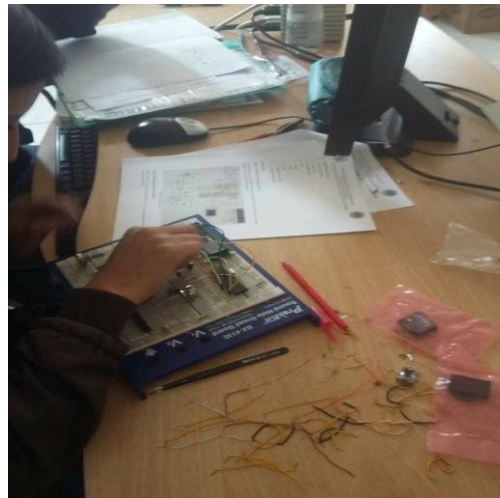


Figura 5.6: Armado en Protoboard

Paso 4. Programación del Microcontrolador

El presente programa está hecho en MickleC, software más conocido y de fácil manejo, pero también se puede utilizar otros lenguajes de programación que tengan librerías de comunicación.

```
char uart_rd;
```

```
sbit LCD_RS at RB0_bit;
```



```
sbit LCD_EN at RB1_bit;
sbit LCD_D4 at RB2_bit;
sbit LCD_D5 at RB3_bit;
sbit LCD_D6 at RB4_bit;
sbit LCD_D7 at RB5_bit;
sbit LCD_RS_Direction at TRISB0_bit;
sbit LCD_EN_Direction at TRISB1_bit;
sbit LCD_D4_Direction at TRISB2_bit;
sbit LCD_D5_Direction at TRISB3_bit;
sbit LCD_D6_Direction at TRISB4_bit;
sbit LCD_D7_Direction at TRISB5_bit;

unsigned int Radc = 0;           // Guarda los datos de la lectura del AN0
unsigned int vtemp = 0;         // Voltaje temporal tipo entero
char *tempC = "000.0";         // Apuntador, Voltaje final
char Text[15];                  // Cadena para conversion FloatToStr
char text1[] = "Temperatura:";
float temp1 = 0.0;              // Variable para procesos
int t,h;

void tiempo(){
Delay_ms(9000);
Lcd_Cmd(_LCD_CLEAR);
}

void tiempo1(){
delay_ms(500);
Lcd_Cmd(_LCD_CLEAR);           // Clear display
}

void LM35(){
if(vtemp < 10000){
tempC[0] = ' ';
tempC[1] = (vtemp/1000)%10 + 48;
tempC[2] = (vtemp/100)%10 + 48;
tempC[4] = (vtemp/10)%10 + 48;
Lcd_Out(2,5,tempC);
UART1_Write_Text(tempC);
UART1_Write(0x0D);           // ASCII - Retorno de Carro
UART1_Write(0x0A);           // ASCII - Avance de Linea
Delay_ms(20);
} else{
tempC[0] = (vtemp/10000)%10 + 48;
tempC[1] = (vtemp/1000)%10 + 48;
tempC[2] = (vtemp/100)%10 + 48;
tempC[4] = (vtemp/10)%10 + 48;
Lcd_Out(2,5,tempC);
UART1_Write_Text(tempC);
```



```
UART1_Write(0x0D);          // ASCII - Retorno de Carro
UART1_Write(0x0A);          // ASCII - Avance de Linea
Delay_ms(20);
}
}

void main() {
    UART1_Init(9600);         // Initialize UART module at 9600 bps
    Delay_ms(100);            // Wait for UART module to stabilize
    Lcd_Init();              // Initialize LCD
    Lcd_Cmd(_LCD_CLEAR);     // Clear display
    Lcd_Cmd(_LCD_CURSOR_OFF);
    TRISA = 0x0D;            // AN0, AN2, AN3 como entradas
    ADCON1.F4 = 1;          // VREF+ 1 = externo 0 = VDD
    ADCON1.F5 = 1;          // VREF- 1 = externo 0 = VSS
    ADC_Init();              // Iniciar módulo ADC

    do{
        Lcd_Out(1,3,text1);   // Muestra el texto en el LCD
        Lcd_Chr(2,10,223);    // Simbolo de grado
        Lcd_Chr(2,11,'C');    // Mostrar la C para centigrados
        Radc = ADC_Get_Sample(0); // Lectura del Canal AN0

        temp1 = (float)(Radc * 0.12); // Obtención del valor real de la conversión
        vtemp = (temp1*100);      // Para másinformación ir al artículo del blog
        FloatToStr(temp1,Text);   // Se convierte de float a char
        LM35();

        Delay_ms(400);          // Regula la velocidad en la cual se realizara
    } while(1);

}
```

Parte 4. Configuración de los módulos Xbee

Paso 1. Testeo y borrado de los módulos.

Estos pasos se realiza para los tres módulos Xbee a configurar.

a.- Test del puerto COM

Abrir el software X-CTU, el mismo programa asigna puertos del cable USB-Serial. Como se muestra en la **Figura 5.7:**

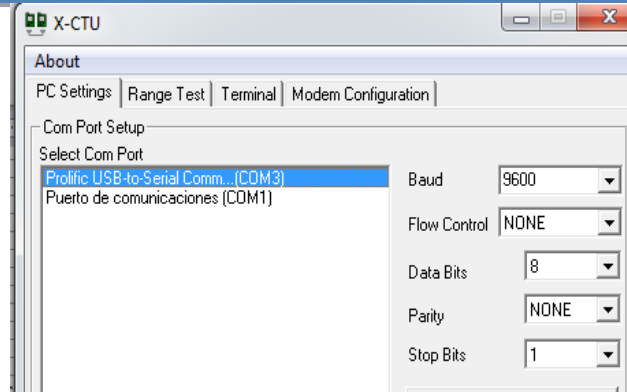


Figura 5.7: Asignación del COM

b.- Test del puerto COM

El botón **Test/Query** sirve para verificar la conexión del módulo con la Pc, saldrá el número de serie del módulo, de esa manera se asegura que el Xbee está listo para realizar la configuración, ver **Figura 5.8**.

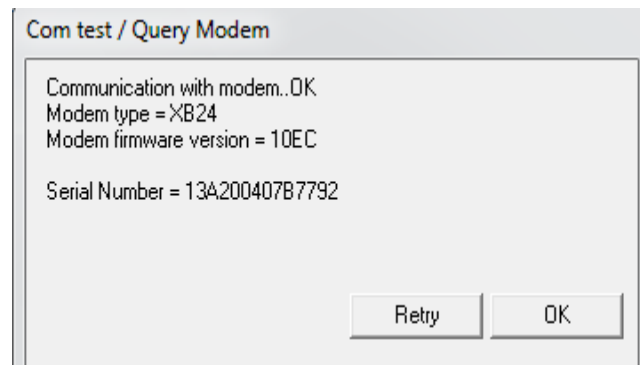


Figura 5.8: Asignación del COM

c.- Pestaña de Modem Configuration.

La pestaña Read sirve para ver la configuración por modo de fábrica en caso de ser nuevo o utilizando anteriormente **Figura 5.9**.

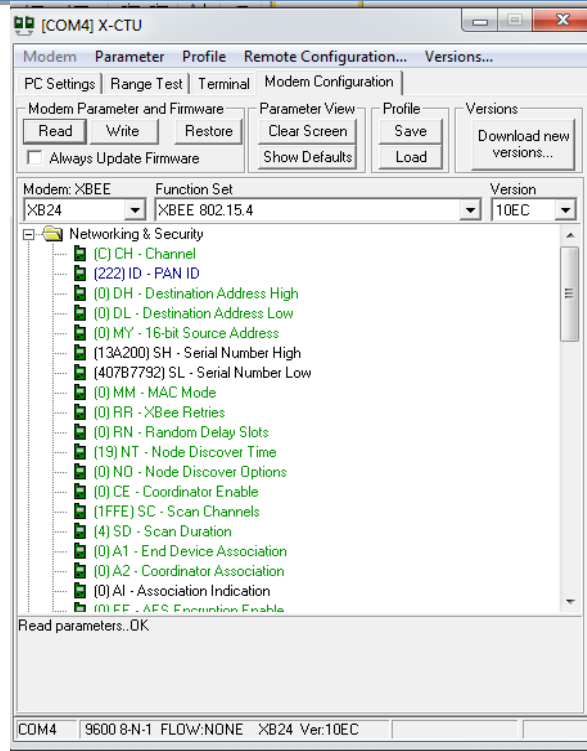


Figura 5.9: Lectura del módulo

d.- Borrador de la configuración.

Clip en Clear Screen, esta pestaña borra la configuración existente en el Xbee, ver Figura 5.10.

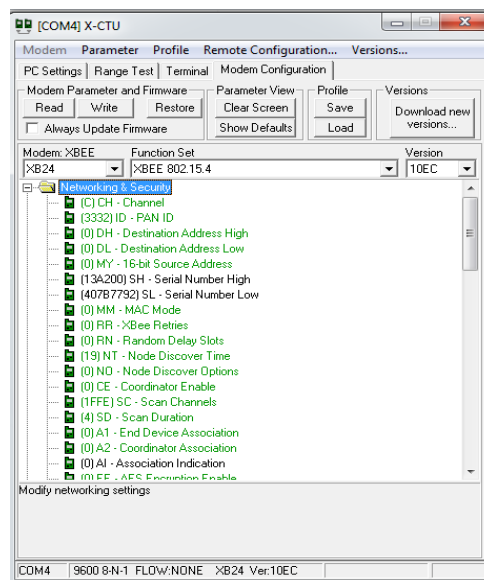


Figura 5.10: Borrado del módulo

e.- Restauración del módulo

Clic en restore para restaurar el módulo a modo de fábrica y quede listo para nueva configuración, ver **Figura 5.11**

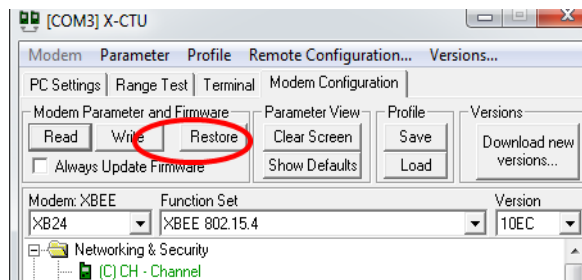


Figura 5.11: Restaura el módulo

Paso 2. Configuración del módulo Xbee transmisor

Se ingresa los valores de la **Tabla 5.1** como se muestra en la **Figura 5.12** para que módulo pueda trabajar como Transmisor.

INDICADOR	NOMBRE	VALOR
NI	Nodo identificador	Transmisor
ID	PAN ID	17
MY	Dirección de red	10

Figura 5.1: Configuración del módulo Xbee Trasmisor

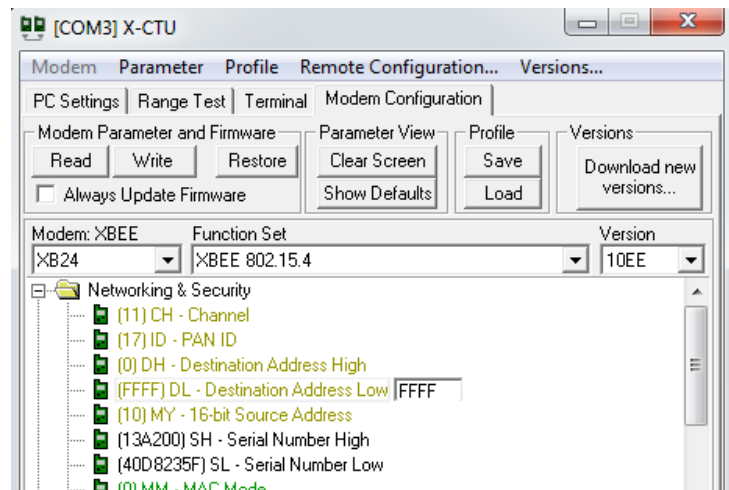


Figura 5.12: Ingreso de datos en el X-CTU

La **Figura 5.13** muestra el nombre asignado al módulo.

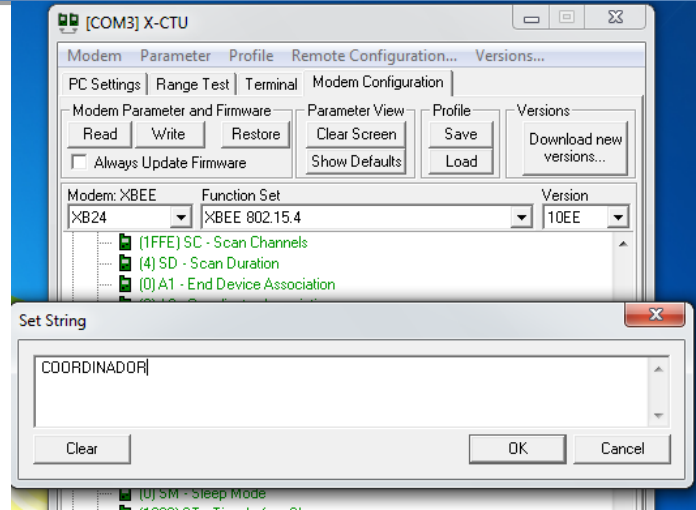


Figura 5.13: Nombre del módulo

Una vez ingresado los valores de la **Tabla 5.1**, clic en **Write** para grabar cambios en el módulo ver **Figura 5.14**.

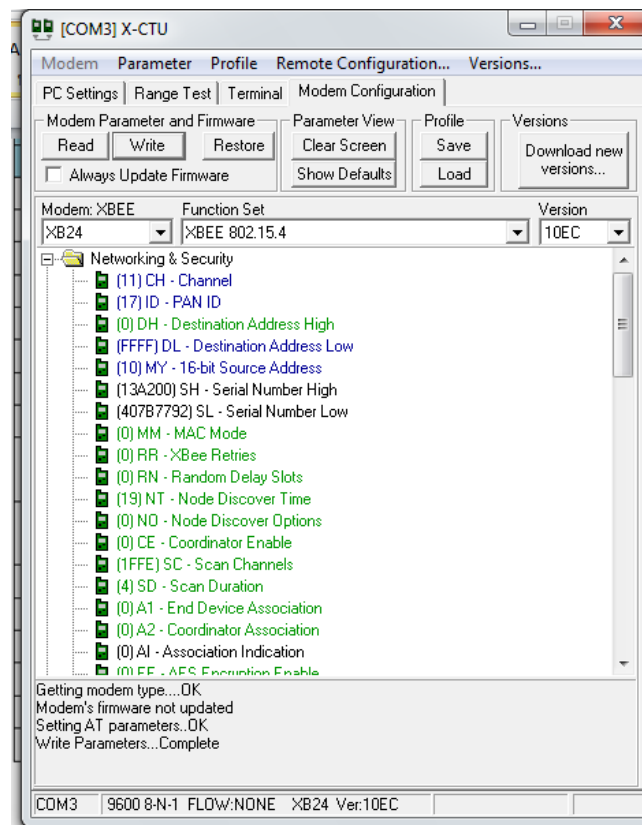


Figura 5.14: Grabación finalizada exitosamente

Paso 3. Configuración del módulo Xbee receptor

A este dispositivo se configura como Receptor, para que tenga conectividad con el Transmisor.

Paso 1. Ingreso de valores a configurar

Se ingresa los valores de la **Tabla 5.2** para que módulo pueda trabajar como Receptor.

INDICADOR	NOMBRE	VALOR
NI	Nodo identificador	Receptor
ID	PAN ID	17
MY	Dirección de red	10

Tabla 5.2: Configuración del módulo Xbee Receptor

Ingresar los datos de la **Tabla 5.2**, y clip en **Write** dando como resultado la **Figura 5.15**.

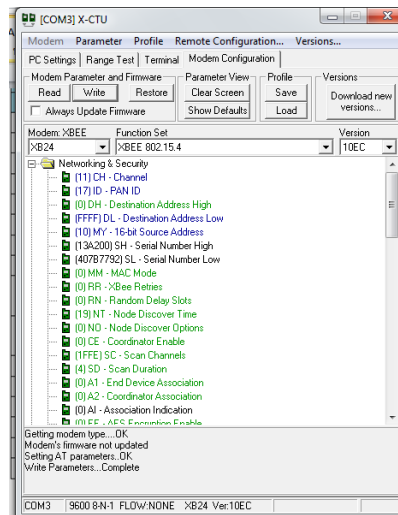


Figura 5.15: Grabación finalizada exitosamente

Parte 5. Funcionamiento

En el transmisor, el circuito sensor envía niveles de voltaje al Microcontrolador para que convierta de niveles de voltaje a sistema decimal que posteriormente será enviado en paquetes RF mediante los módulos Xbee al receptor, mientras que en circuito receptor recibe paquetes RF y muestra en el virtual terminal la temperatura censada vera **Figura 5.16**.

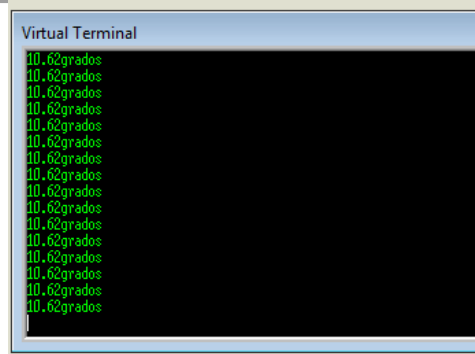


Figura 1.16. Recepción de datos.

ANEXO 19. PRACTICA PARA LA COMPROBACION DE HIPOTESIS CON ENTRENADOR

Universidad Nacional de Chimborazo
Entrenador de Microcontroladores para Comunicaciones
Manual de usuario



Práctica de Laboratorio 1.6: lectura de temperatura en PC mediante la comunicación de módulos Xbee con entrenador.

Topología:



Figura 6.1: Red Punto a Punto

Objetivos de Aprendizaje:

- Manejar el software X-CTU.
- Configurar el módulo Xbee.
- Transmitir datos del sensor a la PC.
- Utilizar comandos AT.
- Verificar la eficiencia de transmisión.

Información básica/Situación.

Es un conjunto de protocolos de alto nivel de comunicación inalámbrica para su utilización de radios digitales de bajo consumo que se basa en el estándar IEEE 802.14.4 de redes inalámbricas de área personal.

El estándar ZigBee amplía el estándar IEEE 802.14.4 aportando una capa de red y un entorno de aplicación que proporciona una sub capa de aplicaciones

Software X-CTU

Software gratuito que se utiliza como interfaz para configurar el módulo Xbee a través de la PC, que proviene de la empresa Digi.

Recursos Necesarios:

- 1 Cables USB-Serial
- Software X-CTU
- 1 PC

- 2 Entrenadores
- 1 16f877A

Desarrollo:

Parte 1: Instalación del driver Usb

- No todos los Sistemas operativos de Windows reconocen el cable serial USB por lo que es necesario instalar manualmente.
- Conectar el cable Usb a la PC y verificar el número de COM, que asigna la PC para la conexión del módulo Xbee.

Parte 2: Instalación del software X-CTU en el PC.

Esta parte corresponde a la instalación del software X-CTU para la configuración de los módulos.

Paso 1: Instalar el software X-CTU.

- a.- Es recomendable instalar la versión antigua del Software X-CTU, **Figura 6.2**, para su fácil manejo.

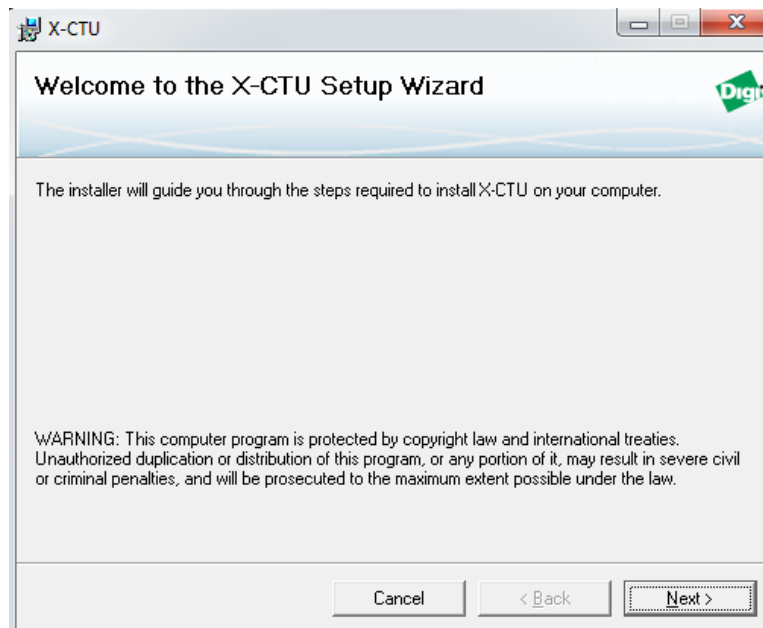


Figura 6.2: Ventana de instalación del X-CTU

- b.- Examinar el programa que este bien instalado y listo para la conexión del módulo con la PC **Figura 6.3**.

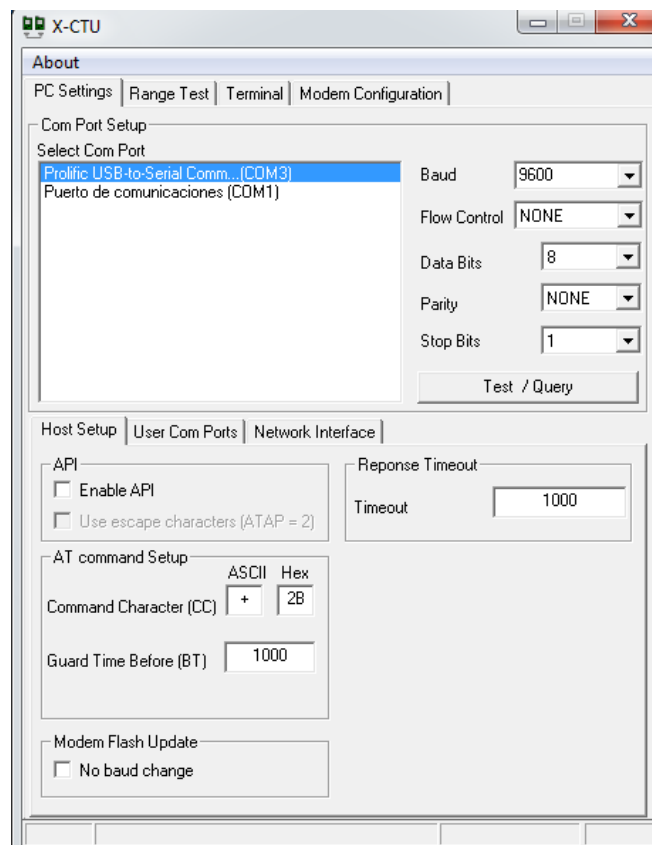


Figura 6.3: Verificación del X-CTU

Parte 3. Diseño del circuito en Proteus

Es una red Punto a Punto por lo tanto es necesario realizar dos circuitos; Transmisor y Receptor.

Paso 1: Circuito Transmisor

Se alimenta a la placa donde está el sensor LM35 se toma la salida Out y se conecta a 16F877A al puerto PORTA FO, se conecta con el módulo Xbee el transmisor y el receptor, es importante tener en cuenta que el microprocesador trabaja con 5V y el módulo Xbee trabaja con 3.3V por lo tanto se hace un Jumper la salida RX 5 y RX 3.3 como se muestra en la **Figura 6.4**:

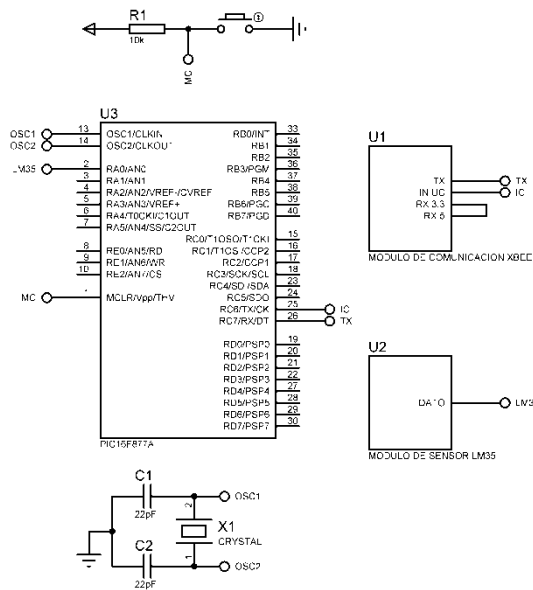


Figura 6.4: circuito transmisor

Paso 2: Circuito Receptor

La Figura 6.5, muestra el diagrama del Receptor, consta de un módulo Xbee. Cable serial USB y una PC.

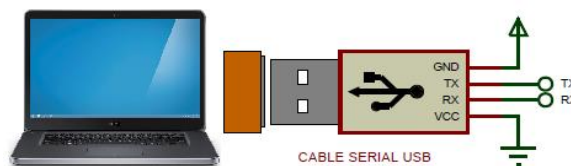
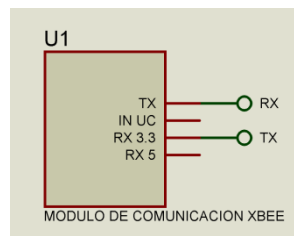


Figura 6.5: Circuito receptor

Paso 3. Armado de Circuito

Realizar conexiones con dos entrenadores, uno como transmisor y el otro como receptor ver Figura 6.6.



Figura 6.6: Armado con el entrenador

Paso 4. Programación del Microcontrolador

El presente programa está hecho en MicroC, software más conocido y de fácil manejo, pero también se puede utilizar otros lenguajes de programación que tengan librerías de comunicación.

```
char uart_rd;
sbit LCD_RS at RB0_bit;
sbit LCD_EN at RB1_bit;
sbit LCD_D4 at RB2_bit;
sbit LCD_D5 at RB3_bit;
sbit LCD_D6 at RB4_bit;
sbit LCD_D7 at RB5_bit;
sbit LCD_RS_Direction at TRISB0_bit;
sbit LCD_EN_Direction at TRISB1_bit;
sbit LCD_D4_Direction at TRISB2_bit;
sbit LCD_D5_Direction at TRISB3_bit;
sbit LCD_D6_Direction at TRISB4_bit;
sbit LCD_D7_Direction at TRISB5_bit;
unsigned int Radc = 0;           // Guarda los datos de la lectura del AN0
unsigned int vtemp = 0;         // Voltaje temporal tipo entero
char *tempC = "000.0";         // Apuntador, Voltaje final
char Text[15];                  // Cadena para conversion FloatToStr
char text1[] = "Temperatura:";
float temp1 = 0.0;              // Variable para procesos
int t,h;
void tiempo(){
Delay_ms(9000);
Lcd_Cmd(_LCD_CLEAR);}
void tiempo1(){
delay_ms(500);
Lcd_Cmd(_LCD_CLEAR);           // Clear display
```



```
}  
void LM35(){  
if(vtemp < 10000){  
    tempC[0] = ' ';  
    tempC[1] = (vtemp/1000)%10 + 48;  
    tempC[2] = (vtemp/100)%10 + 48;  
    tempC[4] = (vtemp/10)%10 + 48;  
    Lcd_Out(2,5,tempC);  
UART1_Write_Text(tempC);  
    UART1_Write(0x0D);          // ASCII - Retorno de Carro  
    UART1_Write(0x0A);          // ASCII - Avance de Línea  
Delay_ms(20);  
} else{  
    tempC[0] = (vtemp/10000)%10 + 48;  
    tempC[1] = (vtemp/1000)%10 + 48;  
    tempC[2] = (vtemp/100)%10 + 48;  
    tempC[4] = (vtemp/10)%10 + 48;  
    Lcd_Out(2,5,tempC);  
UART1_Write_Text(tempC);  
UART1_Write(0x0D);          // ASCII - Retorno de Carro  
UART1_Write(0x0A);          // ASCII - Avance de Linea  
Delay_ms(20);  
}  
}  
}  
void main() {  
    UART1_Init(9600);          // Initialize UART module at 9600 bps  
    Delay_ms(100);             // Wait for UART module to stabilize  
    Lcd_Init();                // Initialize LCD  
    Lcd_Cmd(_LCD_CLEAR);      // Clear display  
    Lcd_Cmd(_LCD_CURSOR_OFF);  
TRISA = 0x0D;                // AN0, AN2, AN3 como entradas  
ADCON1.F4 = 1;               // VREF+ 1 = externo 0 = VDD  
ADCON1.F5 = 1;               // VREF- 1 = externo 0 = VSS  
ADC_Init();                  // Iniciar módulo ADC  
do{  
    Lcd_Out(1,3,text1);        // Muestra el texto en el LCD  
    Lcd_Chr(2,10,223);         // Simbolo de grado  
    Lcd_Chr(2,11,'C');        // Mostrar la C para centígrados  
Radc = ADC_Get_Sample(0);    // Lectura del Canal AN0  
temp1 = (float)(Radc * 0.12); // Obtencion del valor real de la conversión  
vtemp = (temp1*100);         // Para mas informacion ir al articulo del blog  
FloatToStr(temp1,Text);      // Se convierte de float a char  
LM35();  
Delay_ms(400);               // Regula la velocidad en la cual se realizara  
} while(1);  
}
```

Parte 4. Configuración de los módulos Xbee

Paso 1. Testeo y borrado de los módulos.

Estos pasos se realiza para los tres módulos Xbee a configurar.

a.- Test del puerto COM

Abrir el software X-CTU, el mismo programa asigna puertos del cable USB-Serial. Como se muestra en la **Figura 6.7:**

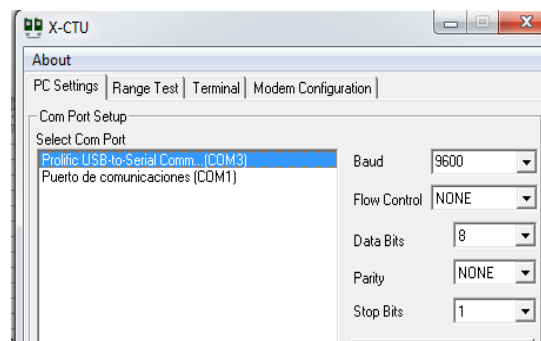


Figura 6.7: Asignación del COM

b.- Test del puerto COM

El botón **Test/Query** sirve para verificar la conexión del módulo con la Pc, saldrá el número de serie del módulo, de esa manera se asegura que el Xbee está listo para realizar la configuración, ver **Figura 6.8.**

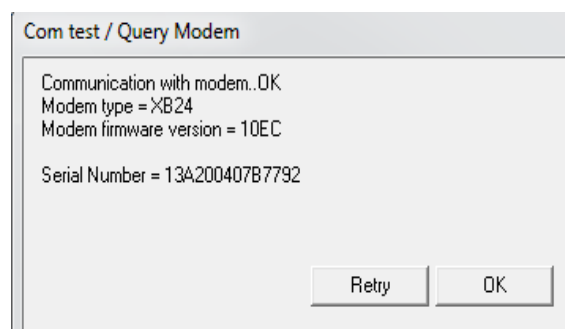


Figura 6.8: Asignación del COM

c.- Pestaña de Modem Configuration.

La pestaña Read sirve para ver la configuración por modo de fábrica en caso de ser nuevo o utilizando anteriormente **Figura 6.9.**

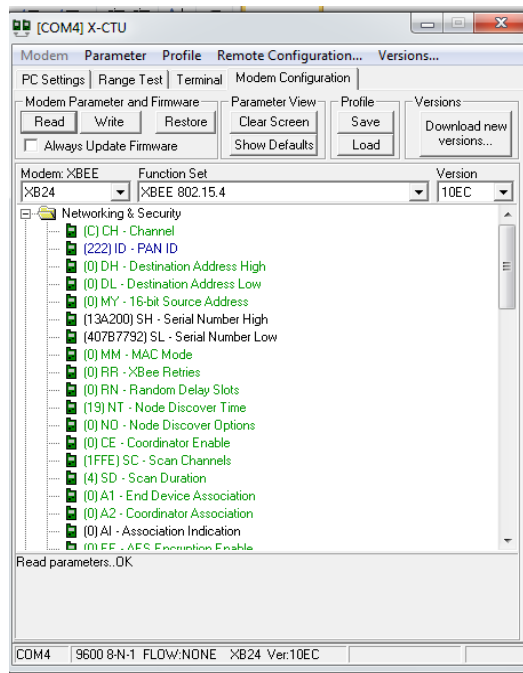


Figura 6.9: Lectura del módulo

d.- Borrador de la configuración.

Clip en Clear Screen, esta pestaña borra la configuración existente en el Xbee, ver **Figura 6.10**.

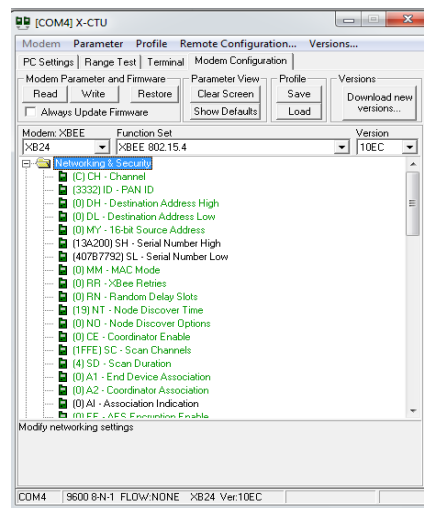


Figura 6.10: Borrado del módulo

e.- Restauración del módulo

Clic en restore para restaurar el módulo a modo de fábrica y quede listo para nueva configuración, ver **Figura 6.11**

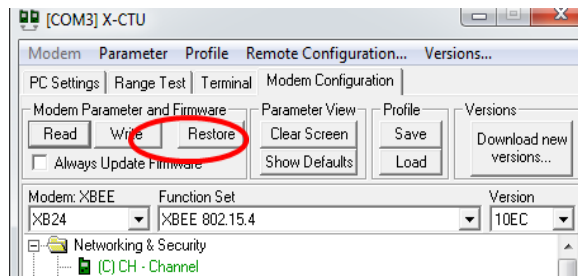


Figura 6.11: Restaura el módulo

Paso 2. Configuración del módulo Xbee transmisor

Se ingresa los valores de la **Tabla 6.1** como se muestra en la **Figura 6.12** para que módulo pueda trabajar como Transmisor.

INDICADOR	NOMBRE	VALOR
NI	Nodo identificador	Transmisor
ID	PAN ID	17
MY	Dirección de red	10

Figura 6.1: Configuración del módulo Xbee Trasmisor

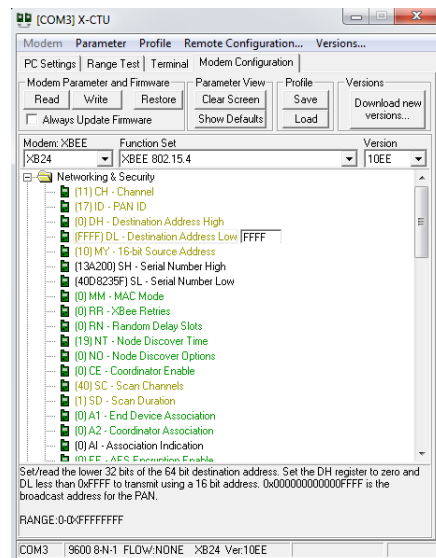


Figura 6.12: Ingreso de datos en el X-CTU

La **Figura 6.13** muestra el nombre asignado al módulo.

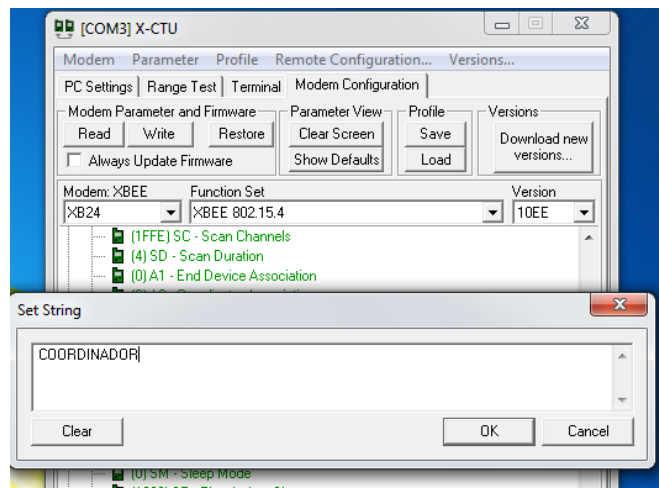


Figura 6.13: Nombre del módulo

Una vez ingresado los valores de la **Tabla 6.1**. Clic en **Write** para grabar cambios en el módulo ver **Figura 6.14**.

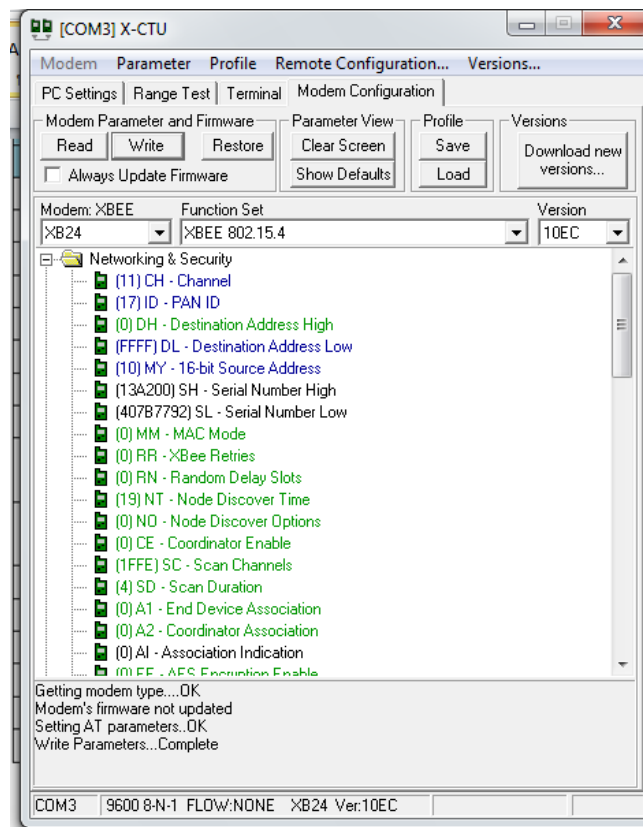


Figura 6.14: Grabación finalizada exitosamente

Paso 3. Configuración del módulo Xbee receptor

A este dispositivo se configura como Receptor, para que tenga conectividad con el Transmisor.

Paso 1. Ingreso de valores a configurar

Se ingresa los valores de la **Tabla 6.2** para que módulo pueda trabajar como Receptor.

INDICADOR	NOMBRE	VALOR
NI	Nodo identificador	Receptor
ID	PAN ID	17
MY	Dirección de red	10

Tabla 6.2: Configuración del módulo Xbee Receptor

Ingresar los datos de la **Tabla 6.2**, y clip en **Write** dando como resultado la **Figura 6.15**.

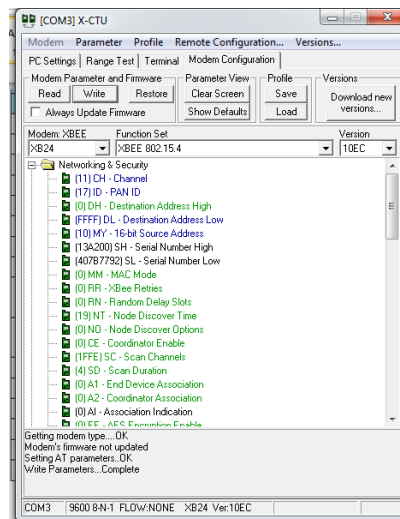


Figura 6.15: Grabación finalizada exitosamente

Parte 5. Funcionamiento

En el transmisor, el circuito sensor envía niveles de voltaje al Microcontrolador para que convierta de niveles de voltaje a sistema decimal que posteriormente será enviado en paquetes RF mediante los módulos Xbee al receptor, mientras que en circuito receptor recibe paquetes RF y muestra en el virtual terminal la temperatura censada vera **Figura 6.16**.

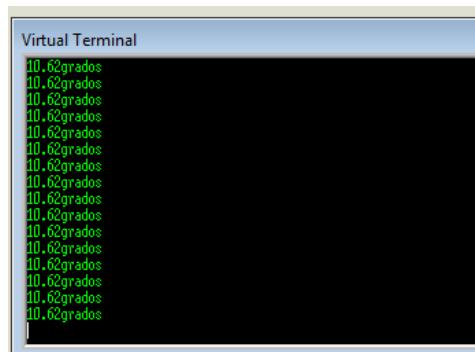


Figura 6.16. Recepción de datos