

# **ESCUELA POLITÉCNICA NACIONAL**

## **ESCUELA DE FORMACIÓN DE TECNÓLOGOS**

### **IMPLEMENTACIÓN DE UN PROTOTIPO DE UN CONTROL ELECTRÓNICO PARA ABRIR Y CERRAR AUTOMÁTICAMENTE LAS PUERTAS DE UN BUS ÚNICAMENTE EN SUS RESPECTIVAS PARADAS**

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN  
ELECTRÓNICA Y TELECOMUNICACIONES**

**VÍCTOR HUGO LAVERDE PAUCAR**  
e-mail: victorlaverdep@hotmail.com

**GUIDO VLADIMIR TAPIA ENCALADA**  
e-mail: guidotapia@outlook.com

**DIRECTOR: ING. PABLO WIGBERTO LÓPEZ MERINO MBA**  
e-mail: pablo.lopez@epn.edu.ec

**Quito, Julio 2014**

## DECLARACIÓN

Nosotros, Víctor Hugo Laverde Paucar y Guido Vladimir Tapia Encalada, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

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

-----  
**VÍCTOR HUGO LAVERDE P.**

-----  
**GUIDO VLADIMIR TAPIA E.**

## CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Víctor Hugo Laverde Paucar y Guido Vladimir Tapia Encalada, bajo mi supervisión.

-----  
**ING. PABLO LÓPEZ MBA**  
**DIRECTOR DEL PROYECTO**

## **AGRADECIMIENTO**

Este proyecto de titulación requirió de mucha paciencia tanto por nosotros por su elaboración como para su director de tesis que nos apoyo en mantener el tema hasta su finalización, gracias por su colaboración Ing. Pablo López.

VÍCTOR

## **AGRADECIMIENTO**

Todos los logros alcanzados en mi vida, incluyendo la realización de este trabajo van siempre dedicados a mis padres Narcisa y Medardo, de igual manera para mi hermana Jennifer, por dedicar todo su esfuerzo por ayudarme, para superarme y ser cada día mejor persona y un excelente profesional.

A todas las personas, amigos, familiares y profesores de la EPN que de igual manera me impulsan a seguir y nunca detenerme en el cumplimiento de mis sueños.

A todos ustedes, muchas gracias.

GUIDO

## **DEDICATORIA**

El proyecto de titulación la dedico a mi esposa y familia por el apoyo brindado en mi formación profesional. A mi madre y padre les agradezco por el esfuerzo realizado para que su hijo tenga una carrera profesional. A mi esposa le agradezco por el apoyo, dedicación y paciencia que ha tenido tanto en mi formación académica, como formación profesional, y finalmente con cariño se lo dedico a Dios por brindarme una grandiosa familia.

VÍCTOR

## **DEDICATORIA**

Dedico la elaboración de este trabajo al pilar fundamental de mi vida, mis padres, a mi hermana, ya que sin ellos no pudiera conseguir los logros paulatinos de mi vida.

GUIDO

## CONTENIDO

DECLARACIÓN .....	i
CERTIFICACIÓN .....	ii
AGRADECIMIENTO.....	iii
AGRADECIMIENTO.....	iv
DEDICATORIA.....	v
DEDICATORIA.....	vi
CONTENIDO.....	vii
ÍNDICE DE TABLAS .....	xi
ÍNDICE DE FIGURAS .....	xii
RESUMEN .....	xvi
PRESENTACIÓN .....	XVIII
CAPÍTULO 1 .....	1
MARCO TEÓRICO.....	1
1.1 FUNDAMENTOS DE RADIOFRECUENCIA.....	1
1.1.1 ESPECTRO ELECTROMAGNÉTICO .....	1
1.1.2 CLASIFICACIÓN DEL ESPECTRO ELECTROMAGNÉTICO .....	1
1.1.3 RADIOFRECUENCIA.....	3
1.1.4 BANDAS ISM .....	6



1.2 MÓDULOS DE RADIOFRECUENCIA .....	6
1.2.1 DEFINICIÓN DE MÓDULOS DE RADIOFRECUENCIA .....	6
1.2.2 FUNCIONAMIENTO DEI MÓDULO DE RF XBEE DEL FABRICANTE DIGI 7	
1.2.2.1 Modo Transmisor / Receptor.....	8
1.2.2.2 Modo de Bajo Consumo.....	8
1.2.2.3 Modo Transparente.....	8
1.2.2.4 Modo Comando.....	10
1.2.2.5 Modo operacional API.....	11
1.2.2.6 Modo IDLE .....	11
1.2.3 CARACTERÍSTICAS DEL MÓDULO DE RF DE 2 mW SERIES 2.5 CHIP-ANTENA A USARSE.....	12
1.3 MICRO-CONTROLADORES .....	13
1.3.1 CARACTERÍSTICAS DEL MICRO-CONTROLADOR PI18F452.....	13
1.3.2 CARACTERÍSTICAS DEL MICRO-CONTROLADOR PI16F870.....	20
1.4 LA ELECTROVÁLVULA.....	21
CAPITULO 2 .....	24
CONSTRUCCIÓN DEL SISTEMA DE CONTROL.....	24
2.1 FUNCIONAMIENTO DEL SISTEMA DE CONTROL DE PUERTAS .....	24
2.2 CONFORMACIÓN DEL sistema de control de puertas .....	25
2.3 CONSTRUCCIÓN DEL CIRCUITO TRANSMISOR DE RF Y CONTROL DE PUERTAS EN EL BUS.....	26

2.3.1 FUENTE DE ALIMENTACIÓN .....	26
2.3.2 RELOJ DE TIEMPO REAL .....	28
2.3.3 VISUALIZACIÓN MEDIANTE LCD.....	29
2.3.4 CONTROL DE ALIMENTACIÓN PARA LA ELECTROVÁLVULA .....	30
2.3.5 CONEXIÓN DEL RECEPTOR DE RF .....	31
2.3.6 CONEXIÓN DEL MÓDULO USB – FLASH.....	32
2.3.7 CONEXIÓN AL MICRO-CONTROLADOR PI18F452.....	33
2.4 CONSTRUCCIÓN DEL CIRCUITO TRANSMISOR DE RF EN PARADAS ...	38
2.4.1 CONEXIÓN AL MICRO-CONTROLADOR PI16F870.....	38
2.4.2 REGULADOR DE VOLTAJE DE 9 V A 5 V .....	39
2.4.3 REGULADOR DE VOLTAJE DE 5 V A 3.3 V .....	40
2.4.4 CONEXIÓN DEL RECEPTOR DE RF .....	40
2.5 CONSTRUCCIÓN DE UN CIRCUITO DE INTERFAZ SERIAL PARA PROGRAMACIÓN DE MÓDULOS DE RF .....	48
CAPÍTULO 3 .....	53
SOFTWARE .....	53
3.1 PROGRAMACIÓN DE MÓDULOS DE RF XBEE S2 COMO EMISOR Y RECEPTOR MEDIANTE EL PROGRAMA X-CTU.....	53
3.1.1 CONFIGURACIÓN MODO COORDINADOR.....	55
3.1.2 CONFIGURACIÓN MODO ROUTER.....	59
3.2 PROGRAMACIÓN DEL MICRO-CONTROLADOR DE ACUERDO AL SISTEMA DE CONSTRUCCIÓN .....	60

3.2.1 PROGRAMACIÓN DEL PI18F452 PARA EL BUS.....	60
3.2.2 PROGRAMACIÓN DEL PI16F870 PARA CADA PARADA.....	66
3.3 BASE DE DATOS PARA EL ALMACENAMIENTO DE INFORMACIÓN OBTENIDA DEL BUS.....	67
3.4 PRUEBAS.....	78
CAPITULO 4 .....	80
CONCLUSIONES Y RECOMENDACIONES .....	80
4.1 CONCLUSIONES .....	80
4.2 RECOMENDACIONES.....	82
REFERENCIAS BIBLIOGRÁFICAS .....	83
ANEXOS .....	85
ANEXO No. 1 .....	86
PROGRAMACIÓN DEL PI18F452.....	86
ANEXO No. 2.....	106
PROGRAMACIÓN DEL PI16F870.....	106
ANEXO No. 3.....	108
HOJAS DE DATOS.....	108

## ÍNDICE DE TABLAS

Tabla 1. 1 Clasificación del espectro electromagnético.....	3
Tabla 1. 2 Clasificación de las bandas de radiofrecuencia.....	5
Tabla 1. 3 Características principales del PI18F452 .....	14
Tabla 1. 4 Pines del puerto A .....	15
Tabla 1. 5 Pines del puerto E .....	15
Tabla 1. 6 Pines del puerto C .....	16
Tabla 1. 7 Pines del puerto B .....	17
Tabla 1. 8 Divisor del Timer 0.....	18
Tabla 1. 9 Características del micro-controlador PI16F870 .....	20
Tabla 3. 1 Configuraciones adicionales del módulo Xbee .....	59
Tabla 3. 2 Cuadro comparativo de tiempos de apertura y cierre de la puerta .....	79

## ÍNDICE DE FIGURAS

Figura 1. 1 Canales dentro de la frecuencia 2.4 GHz.....	6
Figura 1. 2 Módulos Tx y Rx de RF del Xbee.....	7
Figura 1. 3 Modos de funcionamiento del módulo Xbee .....	7
Figura 1. 4 Modo bajo consumo .....	9
Figura 1. 5 Diseño interno del módulo de RF .....	9
Figura 1. 6 Formato del comando AT .....	11
Figura 1. 7 Modo comando usando software del fabricante .....	11
Figura 1. 8 Módulo de RF Xbee 2mW series 2.5 chip-antena.....	13
Figura 1. 9 Pines del PI18F452 .....	14
Figura 1. 10 Funcionamiento del módulo USART .....	19
Figura 1. 11 Pines del PI16F870 .....	21
Figura 1.12 Electroválvula Bi-Estable.....	22
Figura 1. 13 Funcionamiento de la Electroválvula Bi-Estable .....	23
Figura 2. 1 Circuito de control de puertas en un bus.....	25
Figura 2. 2 Regulador de voltaje de 24 V a 5 V .....	27
Figura 2. 3 Regulador de voltaje de 5 V a 3.3 V .....	27
Figura 2. 4 Esquema de conexión del reloj de tiempo real DS1307 .....	29
Figura 2. 5 Display LCD 16x4.....	30

Figura 2. 6 Circuito de alimentación de la electroválvula .....	31
Figura 2. 7 Conexión del receptor de RF .....	32
Figura 2. 8 Módulo USB-FLASH .....	33
Figura 2. 9 Conexión al micro-controlador PI18F452 .....	35
Figura 2. 10 Esquema electrónico del control de puertas del bus .....	36
Figura 2. 11 Diagrama del circuito realizado en ares del control de puertas del bus .....	37
Figura 2. 12 Circuito impreso del control de puertas del bus.....	37
Figura 2. 13 Circuito impreso del opto-acoplador.....	38
Figura 2. 14 Conexión al micro-controlador PI16F870 .....	39
Figura 2. 15 Regulador de voltaje de 24 V a 5 V.....	40
Figura 2. 16 Regulador de voltaje de 5 V a 3.3 V .....	40
Figura 2. 17 Conexión del receptor de RF .....	41
Figura 2. 18 Circuito transmisor de cada parada.....	42
Figura 2. 19 Esquema electrónico del transmisor en las paradas .....	43
Figura 2. 20 Diagrama del circuito realizado en ares del transmisor en las paradas .....	44
Figura 2. 21 Circuito impreso del transmisor en las paradas.....	44
Figura 2. 22 Circuito del sistema de control de puertas de bus.....	45
Figura 2. 23 Circuito del opto-acoplador .....	45
Figura 2. 24 Módulo USB-FLASH .....	46

Figura 2. 25 Circuito Transmisor en las paradas.....	46
Figura 2. 26 Ensamblaje del bus prototipo .....	47
Figura 2. 27 Ensamblaje del Bus prototipo y el sistema de control de bus .....	47
Figura 2. 28 Placa de programación de módulo Xbee del fabricante .....	48
Figura 2. 29 Circuito del programador Xbee .....	50
Figura 2. 30 Diagrama del circuito realizado en ares del programador del módulo Xbee.....	51
Figura 2. 31 Circuito programador del módulo Xbee.....	51
Figura 2. 32 Circuito impreso del programador del módulo Xbee .....	52
Figura 3. 1 Configuración de puertos COM.....	54
Figura 3. 2 Configuración estándar del programa X-CTU .....	54
Figura 3. 3 Validación de comunicación del módulo y el ordenador.....	55
Figura 3. 4 Firmware del módulo coordinador .....	56
Figura 3. 5 Actualización de firmware .....	57
Figura 3. 6 Grabación de firmware del módulo coordinador.....	58
Figura 3. 7 Configuración modo router .....	60
Figura 3. 8 Configuraciones básicas del proyecto en Mikro-C .....	61
Figura 3. 9 Programación del reloj de tiempo real DS1307 .....	63
Figura 3. 10 Programación del sistema de control de puertas .....	65
Figura 3. 11 Programación del transmisor de parada .....	67

Figura 3. 12 Selección de ubicación para la creación de nuevo proyecto .....	68
Figura 3. 13 Inicialización mediante tabla vacía .....	69
Figura 3. 14 Ingreso de los nombres de los campos.....	70
Figura 3. 15 Enumeración para los datos creados .....	70
Figura 3. 16 Asistente para crear una consulta .....	71
Figura 3. 17 Selección de datos a importar .....	71
Figura 3. 18 Consulta finalizada .....	72
Figura 3. 19 Ubicación de datos a importar en tabla creada .....	73
Figura 3. 20 Guardar especificación para importación .....	73
Figura 3. 21 Asistente para importación de texto .....	74
Figura 3. 22 Resultado de la importación .....	75
Figura 3. 23 Asistente de formulario.....	76
Figura 3. 24 Plantilla del macro .....	77
Figura 3. 25 Formulario culminado.....	77



## RESUMEN

La implementación de este sistema nació de la necesidad de una movilidad pública segura que no tenemos los ecuatorianos y principalmente las personas que vivimos en Quito, controlar los tiempos y llevar un orden en las rutas de los autobuses permitirán optimizar el tiempo del usuario y además movilizarse de forma segura, para el desarrollo de nuestro proyecto inicialmente en el capítulo 1 tenemos los fundamentos teóricos de las herramientas utilizadas para un óptimo y mejor desempeño del mismo, como son el estudio del espectro electromagnético, radiofrecuencia y sus respectivas clasificaciones, veremos lo que son módulos de RF y las características principales del módulo XBEE que es el utilizado por nosotros, cómo nuestro proyecto está desarrollado en la banda libre de los 2,4 GHz, veremos lo que son las bandas ISM y las utilidades que se las pueden dar; respecto a los micro-controladores que utilizamos son el PI18F452 y el PI16F870, cada uno de ellos con características especiales para la aplicación requerida por el proyecto.

A continuación en el capítulo 2 veremos la construcción de nuestro sistema de control, uno de ellos es el circuito de interfaz serial para la programación del módulo de RF, el XBEE es aquel que permite dicha programación por las características de transmisión que posee; respecto a la construcción del circuito de control de puertas tenemos que anunciar que la fuente de alimentación es de 24 V DC, existente en el bus y que además es la que permitirá abrir y cerrar la puerta, posteriormente a esto tenemos 2 etapas de regularización de voltaje, una de 5 V, partiendo de la fuente de 24 V y otra de 3,3 V partiendo de la fuente de 5 V ya que varios dispositivos que utilizamos funcionan con estos voltajes; el reloj de tiempo real que utilizamos es el DS1307 que es necesario para controlar tiempos y valores dados por el control ejercido, así como contar con una pantalla LCD para visualización. La conexión del receptor de RF permitirá recibir los datos emitidos por los transmisores ubicados en las paradas, además contamos con una conexión USB-flash que permitirá extraer los datos resultados del control del sistema.

En el capítulo 3 veremos la programación de los módulos de RF XBEE S2 (series 2) como emisor y receptor mediante el programa X-CTU, dicho programa es realizado por los fabricantes del módulo utilizado, este permite una interfaz gráfica amigable con el usuario, ayudando a una configuración sencilla de manipular. Las configuraciones de modo coordinador y modo router que posee la programación las veremos en el desarrollo del proyecto. La programación utilizada para los micro-controladores PI18F452 y PI16F870 del bus y la parada respectivamente fue realizada bajo el programa Mikro-C Basic, el cual nos permitirá realizar las configuraciones necesarias para el funcionamiento correcto del control, como por ejemplo capturar la hora de apertura y cierre de la puerta del bus, la función para grabar la hora en el DS1307, la visualización de hora y fecha en el LCD, activación y desactivación del timer, entre otras.

Por último encontraremos el capítulo 4 que es de conclusiones y recomendaciones, en el cual destacaremos hechos relevantes en el desarrollo de nuestro proyecto.

## PRESENTACIÓN

Dentro del ámbito ecuatoriano no existe un sistema de control electrónico que permita abrir y cerrar las puertas de un bus urbano no articulado automáticamente en sus respectivas paradas, lo cual provoca problemas sociales como son de movilidad y/o seguridad debido a la irresponsabilidad de peatones y choferes que no respetan las paradas preestablecidas.

El sistema de control electrónico se encargará tanto de abrir y cerrar las puertas de un bus no articulado dentro de sus respectivas paradas mediante el envío de una señal a través de un dispositivo de radiofrecuencia que accionará una electróvalvula, como también se encargará de almacenar en una memoria extraíble la información acerca de los intervalos de tiempo entre cada parada por donde pasa el bus, y así poder realizar a futuro un mejor sistema que controle la salida y llegada de los buses.

# **CAPÍTULO 1**

## **MARCO TEÓRICO**

### **1.1 FUNDAMENTOS DE RADIOFRECUENCIA**

#### **1.1.1 ESPECTRO ELECTROMAGNÉTICO**

Una onda electromagnética es la que viaja a través del espacio transportando energía y cantidad de movimiento de una fuente y a medida que se propaga esta onda permitirá proporcionar energía a los cuerpos que se encuentren en su camino es decir un receptor.

Señalado esto podremos indicar que el espectro electromagnético es el conjunto de ondas electromagnéticas que son continuas, es decir que no existen separaciones entre una forma de radiación y otra, además está constituido por intervalos de frecuencias correspondientes a las ondas largas de radio, banda de radio, ondas cortas de radio, infrarrojo, ultravioleta, rayos x, fotones cósmicos y rayos gamma.

#### **1.1.2 CLASIFICACIÓN DEL ESPECTRO ELECTROMAGNÉTICO**

El espectro de las ondas electromagnéticas es inmenso, puesto que cubren una muy amplia gama de longitudes de onda o de frecuencias, en él se encuentran clasificadas todas las ondas electromagnéticas naturales como los colores del arco iris y la luz solar, también las artificiales como son las generadas por transmisores para televisión y radio.

Dentro de la clasificación general del espectro electromagnético tenemos las que veremos a continuación:

- Ondas de radiofrecuencia.- Son ondas que se emplean para aplicaciones de telecomunicaciones como telefonía celular, televisión, etc.

- Microondas.- Son ondas que se generan mediante dispositivos electrónicos que tienen aplicaciones para enlaces de radiofrecuencia para uso comercial o militar, también tienen aplicaciones en el calentamiento de alimentos como son los hornos microondas.
- Radiación Infrarroja.- Está dividida en tres regiones que son la cercana, media y lejana. Son usadas en controles remotos, sensores electrónicos, tienen aplicaciones en la medicina, astronomía y en la industria; además todos los cuerpos irradian este tipo de ondas que pueden verse a través de visores térmicos ya que la radiación infrarroja está asociada con la temperatura.
- Región visible.- Son las ondas electromagnéticas en este caso luminosas que ocupan una pequeña porción del espectro y que pueden ser percibidos por el ojo humano y se la conoce como luz blanca y está compuesta por 7 colores que son desde el rojo, anaranjado, amarillo, verde, azul, índigo hasta el violeta.
- Rayos ultravioleta.- Los rayos ultravioletas (UV) son producidos por cuerpos muy calientes como el sol ya que son el componente principal de la radiación solar.
- Rayos X.- Los rayos X se producen al bombardear sobre algún objeto electrones a gran velocidad, al chocar sobre una placa metálica la radiación del frenado produce estos rayos, los fotones de los rayos X son de gran energía por lo que son peligrosos para los organismos vivos, aun así son utilizados en medicina ya que los huesos absorben más radiación que los tejidos blandos.
- Rayos Gamma y Rayos Cósmicos.- Son altamente radioactivos ya que se los obtienen al desintegrar sustancias radioactivas en procesos nucleares y además es componente de la radiación cósmica, bajo una prolongada radiación incidente sobre un cuerpo vivo ocasiona reacciones cancerígenas, pese a que también es usado para eliminar estos síntomas en medicina, como para esterilización de materiales, tiene interés para la astrofísica y al igual que las radiaciones ópticas aparecen en la naturaleza

pero se diferencian de la luz no solo por la longitud de onda sino por su producción.<sup>1</sup>

Tipo de onda	Longitud	Frecuencia
	de onda	
Rayos gamma	< 10 pm	> 30.0 EHz
Rayos X	< 10 nm	> 30.0 PHz
Ultravioleta Extremo	< 200 nm	> 1.5 PHz
Ultravioleta Cercano	< 380 nm	> 789 THz
Luz Visible	< 780 nm	> 384 THz
Infrarrojo Cercano	< 2.5 um	> 120 THz
Infrarrojo Medio	< 50 um	> 6.00 THz
Infrarrojo Lejano	< 1 mm	> 300 GHz
Microondas	< 30 cm	> 1.0 GHz
Ultra Alta Frecuencia Radio	< 1 m	> 300 MHz
Muy Alta Frecuencia Radio	< 10 m	> 30 MHz
Onda corta Radio	< 180 m	> 1.7 MHz
Onda Media(AM) Radio	< 650 m	> 650 kHz
Onda Larga Radio	< 10 km	> 30 kHz
Muy Baja Frecuencia Radio	> 10 km	< 30 kHz

**Tabla 1. 1 Clasificación del espectro electromagnético<sup>2</sup>**

### 1.1.3 RADIOFRECUENCIA

La Radiofrecuencia (RF) o Espectro de Radiofrecuencia es una porción del Espectro Electromagnético en el que se puede generar ondas electromagnéticas aplicando corriente alterna a un dipolo. Dicha porción del espectro electromagnético es la menos energética y se encuentra desde los 3 KHz para frecuencias extremadamente bajas y aproximadamente hasta los 300 GHz para frecuencias extremadamente altas y cuya longitud de onda varía respectivamente entre los 100 km a 0,1 mm.

<sup>1</sup> García, 2011

<sup>2</sup> Frenzel, Mayo, 2013

En la actualidad sabemos que el intervalo de frecuencias pertenecientes al espectro electromagnético es inmenso. De esto tenemos que la longitud de onda ( $\lambda$ ) de la radiación electromagnética se relaciona con su frecuencia ( $f$ ) mediante la siguiente ecuación general:

$$C = f\lambda = 3 \times 10^8 \text{ m/s} \quad (1.1)$$

Donde consideraremos a la longitud de onda ( $\lambda$ ) como la distancia que existe entre los picos máximos de una serie de onda y la frecuencia ( $f$ ) como el número de veces en las que se repitiera en un periodo de tiempo, C es la velocidad de la luz teórica.

A continuación se indica la clasificación de las bandas de radiofrecuencia y una breve descripción de las aplicaciones en las que son usadas.

- Frecuencias muy bajas (Very Low Frequencies, VLF).- Su difusión es mediante onda de tierra, y es generalmente utilizada para fines militares y del gobierno.
- Frecuencias bajas (Low Frequencies, LF).- Generalmente conocida como onda larga, su difusión es mediante onda de tierra con atenuación débil, se utiliza generalmente para conexiones de radio de gran escala, ayuda a la navegación marítima y aérea.
- Frecuencias medias (Medium Frequencies, MF).- Su difusión es por onda de tierra con una absorción elevada a lo largo del día y difusión ionosférica durante la noche. El uso más común es en radiodifusión sobre todo en amplitud modulada AM, aunque encontramos otras aplicaciones sobre todo marítimas con los radio faros.
- Frecuencias altas (High Frequencies, HF).- Se las conoce como Onda Corta y su difusión es principalmente ionosférica con fuertes cambios estacionales durante cualquier hora del día o de la noche, prácticamente en este rango de frecuencias se encuentran la mayoría de los servicios de telecomunicaciones, sus aplicaciones son para emisoras fijas, conexiones marítimas móviles, emisoras de radiodifusión, radioaficionados, bandas aeronáuticas móviles, emisoras terrestres móviles, estaciones de frecuencia estándar, entre otras.

- Frecuencias muy altas (Very High Frequencies, VHF).- Es usada para conexiones de radio a corta distancia, en televisión, en radiodifusión de frecuencia modulada FM, comunicaciones marítimas y aeronáuticas, y ciertas bandas para radioaficionados.
- Frecuencias ultra altas (Ultra High Frequencies, UHF).- Su difusión es únicamente directa. Entre las principales aplicaciones tenemos los enlaces de radio, ayuda a la navegación aérea, para los radares, televisión, comunicaciones militares y de tierra, telefonía celular, etc.
- Frecuencias súper altas (Super High Frequencies, SHF).- Su principal aplicación es para radio-enlaces terrestres y en las comunicaciones satelitales, pero también es utilizada para aplicaciones militares con cierto tipo de radares.
- Frecuencias extremadamente altas (Extremely High Frequencies, EHF).- No existen dispositivos capaces de difundir este tipo de señales hasta la actualidad.<sup>3</sup>

NOMBRE	FRECUENCIA INFERIOR	FRECUENCIA SUPERIOR	LONGITUD DE ONDA
Muy Baja Frecuencia VLF	3 KHZ	30 KHZ	100 KM - 10 KM
Baja Frecuencia LF	30 KHZ	300 KHZ	10 KM - 1 KM
Frecuencia Media MF	300 KHZ	3000 KHZ	1 KM - 100 M
Altas Frecuencias HF	3 MHZ	30 MHZ	100 M - 10 M
Muy Altas Frecuencias VHF	30 MHZ	300 MHZ	10 M - 1 M
Ultra Alta Frecuencia UHF	300 MHZ	3000 MHZ	1 M - 100 CM
Súper Alta Frecuencia SHF	3 GHZ	30 GHZ	100 CM - 1 CM
Extra Alta Frecuencia EHF	30 GHZ	300 GHZ	1 CM - 1 MM
S/N	300 GHZ	3000 GHZ	1 MM - 0.1 MM

**Tabla 1. 2 Clasificación de las bandas de radiofrecuencia<sup>4</sup>**

<sup>3</sup> Yahoo, 2011

<sup>4</sup> ECURED, 2014



### 1.1.4 BANDAS ISM

Las bandas de radiofrecuencia electromagnética ISM (Industrial, Scientific and Medical) para aplicaciones médicas, científicas e industriales son bandas para uso no comercial en dichas áreas de trabajo, aunque actualmente también se utilizan para conexiones inalámbricas utilizados por dispositivos móviles como teléfonos, computadores y dispositivos que trabajan en los estándares IEEE<sup>5</sup> 802.15.1 (WPAN<sup>6</sup> y dispositivos Bluetooth), IEEE 802.11 (WLAN<sup>7</sup>), IEEE 802.15.4, entre otros.

Dependiendo del lugar geográfico, la banda de 2,4 GHz utiliza de 7 a 14 canales, donde se tiene una frecuencia central, una inferior y una superior con una separación de 5 MHz, para evitar la interferencia entre canales adyacentes se realiza una separación de 5 canales por lo que tradicionalmente se utilizan los 1, 6 y 11 o los 1, 9 y 13, según el estándar.

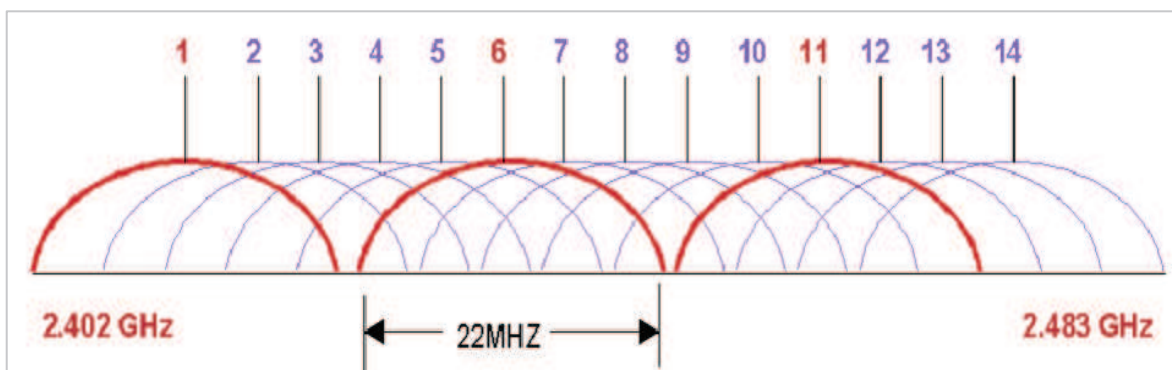


Figura 1. 1 Canales dentro de la frecuencia 2.4 GHz<sup>8</sup>

## 1.2 MÓDULOS DE RADIOFRECUENCIA

### 1.2.1 DEFINICIÓN DE MÓDULOS DE RADIOFRECUENCIA

Se considera a los módulos de radiofrecuencia RF como dispositivos que permiten enviar y recibir información análoga o digital a través de ondas de radiofrecuencia.

<sup>5</sup> IEEE: Instituto de Ingeniería Eléctrica y Electrónica/ estandarización mediante protocolos

<sup>6</sup> WPAN: Red inalámbrica de área personal

<sup>7</sup> WLAN: Red de área local inalámbrica

<sup>8</sup> SIMONS, 2013

Los módulos de radiofrecuencia están conformados por un microchip que controla la información recibida o enviada, un transmisor, un receptor y las antenas.

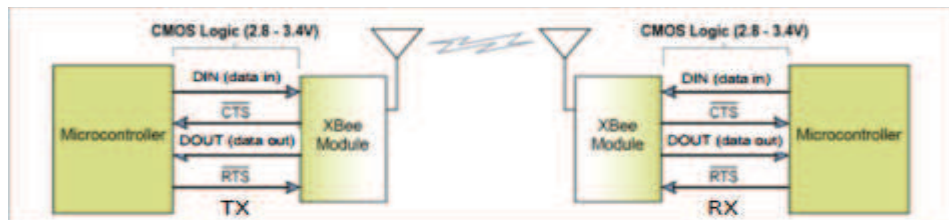


Figura 1. 2 Módulos Tx y Rx de RF del Xbee<sup>9</sup>

### 1.2.2 FUNCIONAMIENTO DEL MÓDULO DE RF XBEE DEL FABRICANTE DIGI

Los módulos de RF XBEE<sup>10</sup> son dispositivos de comunicación inalámbrica para el envío de información analógica o digital a través de ondas en la frecuencia de 2.4 GHz. Los módulos de radio frecuencia XBEE trabajan bajo el estándar IEEE 802.15.4 que definen que los módulos de radiofrecuencia forman su propia red PAN<sup>11</sup> sean punto a punto o punto a multipunto. Generalmente los módulos de radiofrecuencia XBEE poseen 7 tipos de funcionamiento como se indica en la Figura 1. 3, donde el módulo XBEE se encontrará en el modo IDLE<sup>12</sup> para poder ingresar en los siguientes modos de funcionamiento.



Figura 1. 3 Modos de funcionamiento del módulo Xbee

<sup>9</sup> Digi International, 2008

<sup>10</sup> XBEE: Nombre comercial de Digi Internacional para una familia de módulos de radio.

<sup>11</sup> PAN: Red de área personal

<sup>12</sup> IDLE: Traducción: Detenido, Parado.

### **1.2.2.1 Modo Transmisor / Receptor**

Se encuentra en estos modos cuando el módulo recibe algún paquete de radiofrecuencia a través de la antena o cuando se envía información directamente al buffer<sup>13</sup> del pin 3 que luego será transmitido, consideramos también que el módulo puede retener información hasta que esta sea solicitada por el destinatario.

### **1.2.2.2 Modo de Bajo Consumo**

Este modo permitirá al módulo de radiofrecuencia disminuir el consumo de energía, lo que implica un ahorro en la fuente de alimentación, al realizar esta función, el módulo de radiofrecuencia se apagará por un determinado ciclo de tiempo mientras está tratando de escuchar al canal de radiofrecuencia en espera de que se detecte otro módulo de radiofrecuencia; cuando pase esto, se mantendrá activo hasta que se pierda la comunicación y nuevamente se apague por un tiempo determinado. El tiempo que permanecerá apagado dependerá de la configuración realizada en el firmware<sup>14</sup> del módulo.

### **1.2.2.3 Modo Transparente**

Este modo está destinado para las conexiones de punto a punto o de multipunto a punto realizados con los módulos XBEE donde todo lo que ingresa por el pin 3 es guardado en el buffer de entrada y luego transmitido, además todo lo que ingresa como paquete de RF, es guardado en el buffer de salida y luego enviado por el pin 2 al microcontrolador.

Hay que considerar que si el buffer de entrada o salida se llena, no permitirá la recepción o transmisión de información hasta que se envíe la información correspondiente.

---

<sup>13</sup> Buffer: Espacio de memoria, dedicado a evitar quedarse sin datos durante una transferencia.

<sup>14</sup> Firmware: Es el encargado de sincronizar el hardware con el software para el correcto funcionamiento de un sistema por lo general electrónico.

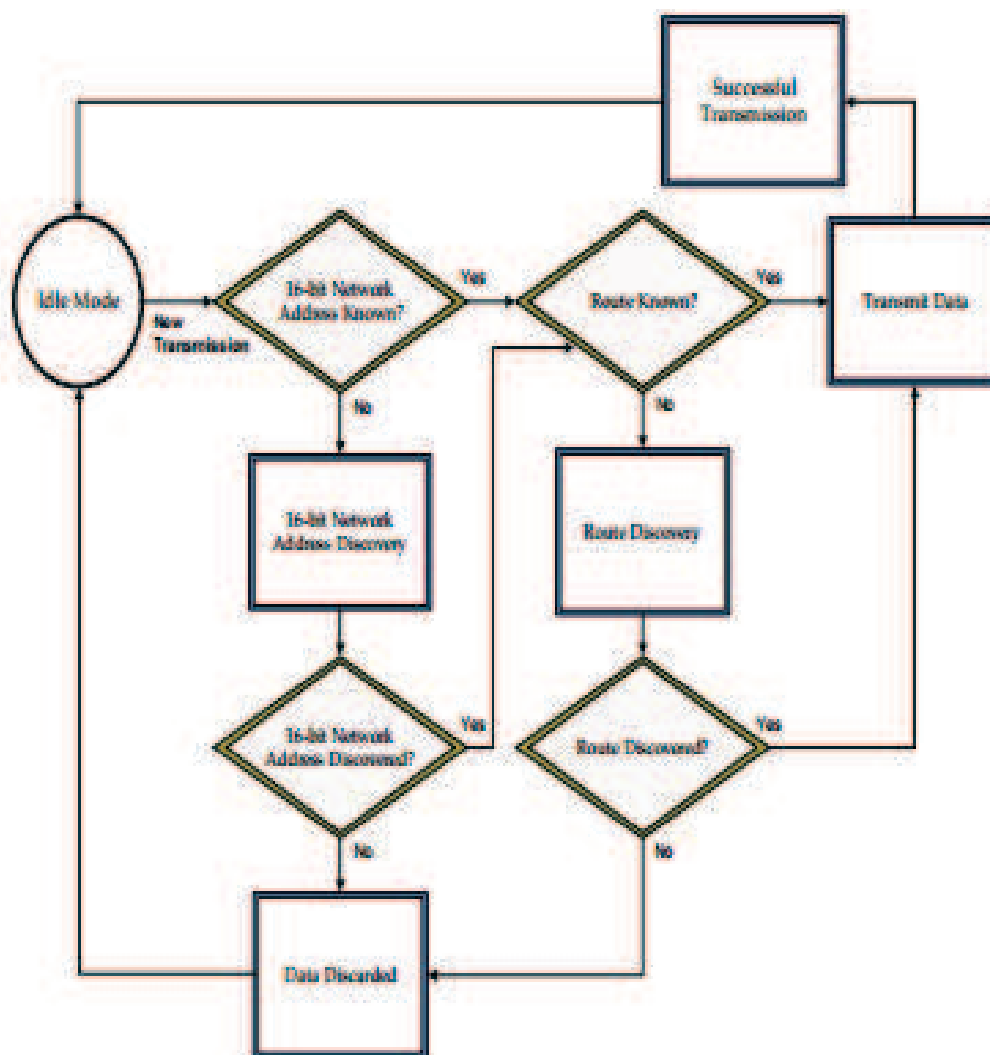


Figura 1. 4 Modo bajo consumo<sup>15</sup>

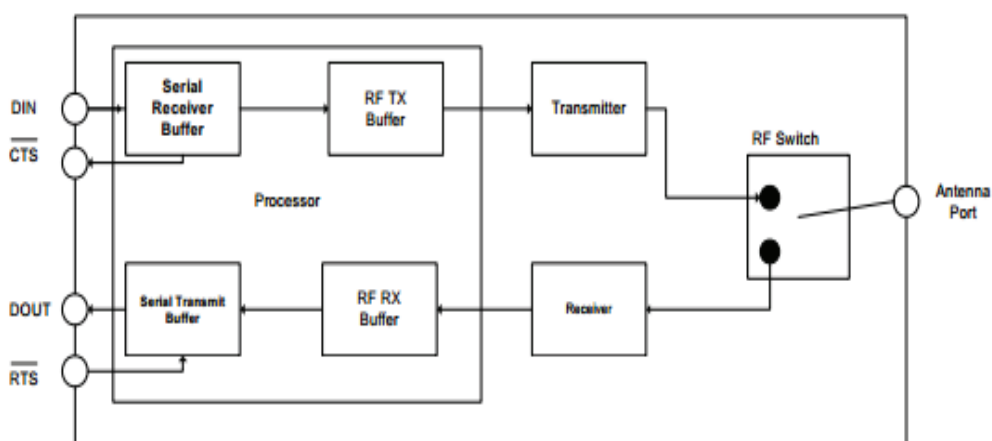


Figura 1. 5 Diseño interno del módulo de RF<sup>16</sup>

<sup>15</sup> Digi International, 2008

<sup>16</sup> Digi International, 2008

#### 1.2.2.4 Modo Comando

En el modo comando se permite el ingreso al firmware para realizar la visualización o escritura de configuraciones mediante el uso de comandos propios del fabricante. Entre los principales comandos se tienen los siguientes con su respectiva función:

+++ Permite el ingreso al firmware mediante comando, para validar el ingreso el módulo devolverá un ok, confirmando la comunicación e ingreso.

ATMY Permitirá visualizar la dirección MY configurada en el módulo en un valor hexadecimal, en el caso de no estar configurada el valor que retorna es 0.

ATMY 2520 El comando ATMY acompañado de un valor hexadecimal permitirá la escritura del valor hexadecimal como dirección.

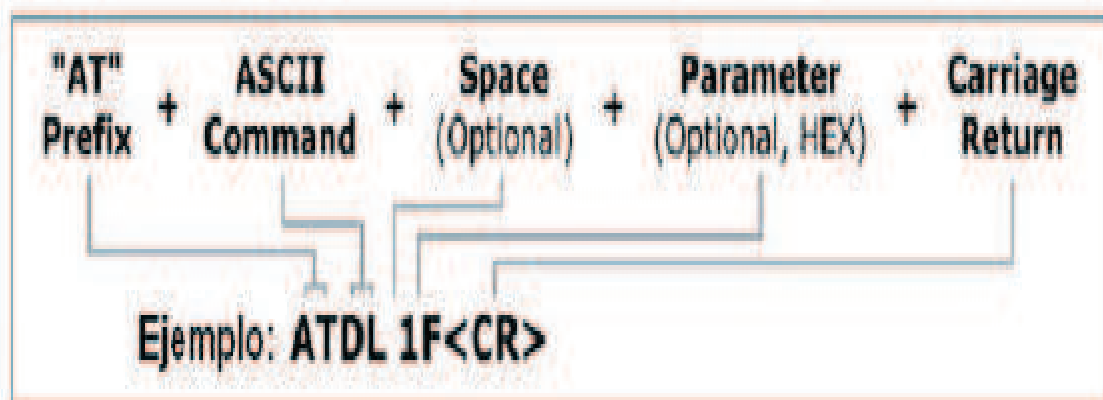
ATDL Permite visualizar la dirección en bajo del módulo.

ATDH Permite visualizar la dirección en alto del módulo.

ATND Este comando permitirá visualizar la validación de los módulos con que tienen una comunicación después de ser configurados.

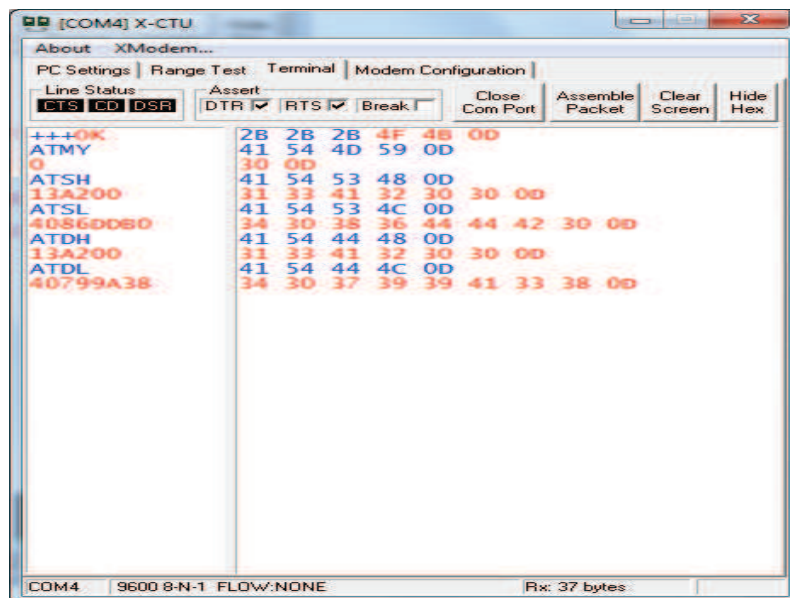
ATWR Permite guardar los cambios realizados en la memoria no volátil del módulo.

ATCN Permite salir del modo comando.



**Figura 1. 6 Formato del comando AT<sup>17</sup>**

Una vez indicado el formato y algunos de los comandos AT se podrá visualizar en la Figura 1. 7 un ejemplo de los comandos usando el software del fabricante.



**Figura 1. 7 Modo comando usando software del fabricante**

### 1.2.2.5 Modo operacional API

El modo API permitirá crear frames<sup>18</sup> con cabeceras que aseguran la transmisión de datos de un módulo a otro como si fuera el protocolo TCP<sup>19</sup>. Entre las opciones que permite la API, se tienen:

- Transmitir información a múltiples destinatarios.
- Recibir estado de éxito/falla de cada paquete de RF transmitido.
- Identificar la dirección de origen de cada paquete recibido.

### 1.2.2.6 Modo IDLE

Cuando el módulo no está en ningún modo de los que se indicó anteriormente, se encuentra en modo IDLE, es decir, si no está transmitiendo, recibiendo,

<sup>17</sup> Digi International, 2008

<sup>18</sup> Frame: es una unidad de envío de datos pertenecientes a la capa de enlace del modelo referencia OSI.

<sup>19</sup> TCP: Protocolo de Control de Transmisión

ahorrando energía ni en el modo de comandos, entonces se dice que se encuentra en un estado al que se le llama IDLE que significa detenido.

### **1.2.3 CARACTERÍSTICAS DEL MÓDULO DE RF DE 2 mW SERIES 2.5 CHIP-ANTENA A USARSE**

Las características del módulo de radiofrecuencia usado para la implementación de un sistema de control de puertas de un bus:

- Como se ha indicado en el inicio de este capítulo los módulos XBEE trabajan en las bandas ISM es decir en una frecuencia de 2.4 GHz por lo que no se requiere tener permisos de entidades gubernamentales para el funcionamiento en este rango.
- Los módulos de estas series se basan en el protocolo Zigbee<sup>20</sup> que es semejante al protocolo 802.11.4 es decir que no solo permitirán crear redes punto a punto y punto a multipunto sino que permiten crear redes mesh<sup>21</sup> y frames con cabeceras de destino y origen es decir una mejor protección de los datos enviados.
- El módulo XBEE posee una antena tipo chip que puede trabajar hasta con 2 mW de potencia de transmisión, o a su vez puede ser disminuida a través de programación.
- El tipo de modulación que realiza el transmisor es una modulación 16 QAM<sup>22</sup> y una codificación de línea NRZ<sup>23</sup> para poder optimizar los niveles de potencia usados.
- Este módulo tiene un alcance de 40 metros en un ambiente cerrado, y puede alcanzar hasta 120 metros en un ambiente abierto.
- Poseen 6 convertidores análogo-digitales, 8 entradas digitales y 2 pines correspondientes para la transmisión Tx y recepción Rx de datos.
- Permiten el acceso al firmware del módulo XBEE, mediante la comunicación a

---

<sup>20</sup> Zigbee: Conjunto de protocolos de alto nivel destinados a la utilización con radiodifusión digital

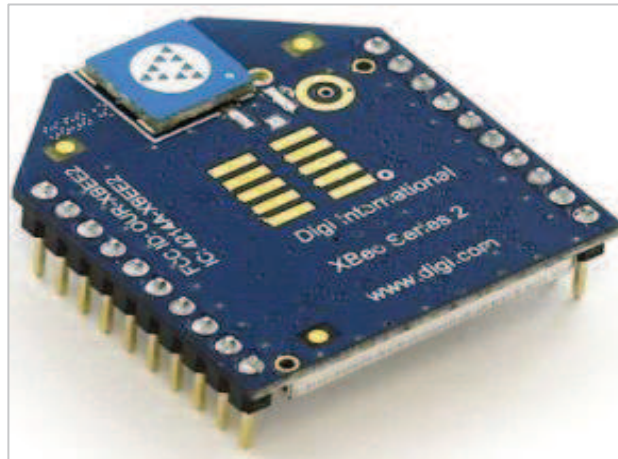
<sup>21</sup> Mesh: Es una red inalámbrica en malla, que combina dos topologías distintas de redes inalámbricas

<sup>22</sup> QAM: Modulación de amplitud en cuadratura, modula la señal portadora en amplitud y fase

<sup>23</sup> NRZ: No Retorno a Cero, es un código de línea que utiliza un nivel de tensión diferente para cada uno de los dígitos binarios

través de un dispositivo intermedio y un software proporcionado por el fabricante.

- Tienen una velocidad de comunicación entre los 1200 baudios<sup>24</sup> hasta 115200 baudios pasando por todos los valores intermedios conocidos<sup>25</sup>.



**Figura 1. 8 Módulo de RF Xbee 2mW series 2.5 chip-antena**

## 1.3 MICRO-CONTROLADORES

### 1.3.1 CARACTERÍSTICAS DEL MICRO-CONTROLADOR PI18F452

Hemos escogido el PI18F452 por las funciones y cualidades que nos brinda para la realización de nuestro proyecto, como por ejemplo soportar la comunicación serial, tener alrededor de 5 puertos de comunicación disponibles para cubrir las necesidades dadas por nuestro proyecto, cuenta con una mayor memoria de programa de 32 Kbytes y 1.5 Kbytes para memoria de datos, posee 40 pines en su totalidad, un set de instrucciones de 75 palabras, 4 timers (reloj), 2 módulos generadores de señales; pero nos enfocaremos en dar y mencionar las características principales del micro-controlador enfocando siempre a lo que

<sup>24</sup> Baudio: Unidad de medida en telecomunicaciones, representa en número de símbolos por segundo en un medio de transmisión

<sup>25</sup> Junior Figueroa Olmedo, 2013



requerimos, además debemos dar a conocer que el programa empujado sobre el controlador va a ocupar toda su memoria.

Frecuencia de operación	DC - 40MHz
Memoria de programa	32K
Memoria de datos	1.5K
Memoria de dtos EEPROM	256 bytes
Fuentes de interrupción	18
Puertos de I/O	5
Timers	4
Módulos PWM	2
Comunicación serial	MSSP, USART
Comunicación en paralelo	1 PSP
Módulo ADC	8 canales de ingreso
Set de instrucciones	75 instrucciones
Encapsulado	40 pines

Tabla 1. 3 Características principales del PIC18F452<sup>26</sup>

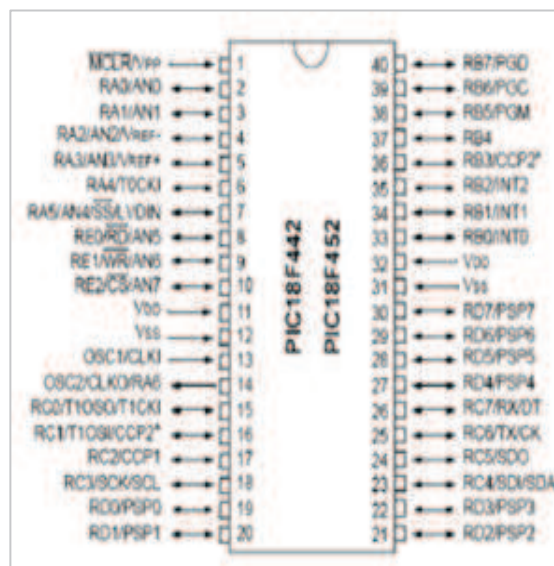


Figura 1. 9 Pines del PIC18F452<sup>27</sup>

<sup>26</sup> Salesiana, 2009

<sup>27</sup> Microchip, Data Sheet, microcontrolador 16F870, 2008

El pin 1 (MCLR/Vpp), es el master clear, que es un pin de ingreso que resetea el micro-controlador, además también funciona como pin de voltaje al actuar de ese modo.

La descripción de los pines del puerto A se detalla en la Tabla 1.4.

Pin No.	Nombre de Pin	Descripción
2	RA0	E/S digital 0
	AN0	Ingreso analógico 0
3	RA1	E/S digital 1
	AN1	Ingreso analógico 1
4	RA2	E/S digital 2
	AN2	Ingreso analógico 2
	VREF-	Ingreso del voltaje de referencia A/D
5	RA3	E/S digital 3
	AN3	Ingreso analógico 3
	VREF+	Ingreso del voltaje de referencia A/D
6	RA4	E/S digital 4
	T0CKI	Señal de ingreso externa para el timer 0
7	RA5	E/S digital 5
	AN4	Ingreso analógico 4
	SS	Selección de integrado SPI esclavo
	LVDIN	Detección de nivel bajo de voltaje

**Tabla 1. 4 Pines del puerto A<sup>28</sup>**

La descripción de los pines del puerto E se detalla en la Tabla 1. 5.

Pin No.	Nombre de Pin	Descripción
8	RE0	E/S digital 0
	RD	Señal de lectura para el puerto paralelo
9	RE1	E/S digital 1
	WR	Señal de escritura para el puerto paralelo
10	RE2	E/S digital 2
	CS	Señal de selección del integrado

**Tabla 1. 5 Pines del puerto E<sup>26</sup>**

<sup>28</sup> Salesiana, 2009

En los pines 11 y 32 (Vdd), encontramos la alimentación positiva del integrado, nivel lógico alto (5V dc).

En los pines 12 y 31 (Vss), debemos conectar la señal de alimentación de nivel lógico bajo (0V dc).

Los pines 13 y 14 (OSC1 y OSC2), corresponden a las señales del reloj externa, además el pin 14 puede ser utilizado como sexto bit de E/S (Entrada-salida) del puerto A.

La descripción de los pines del puerto C se detalla en la Tabla 1. 6.

Pin No.	Nombre de Pin	Descripción
15	RC0	E/S digital 0
	T1OSO	Salida de oscilador del timer1
	T1CKI	Ingreso de la señal de reloj externa
16	RC1	E/S digital 1
	T1OSI	Ingreso señal oscilador timer 1
	CCP2	Compare input/output 2, salida PWM 2
17	RC2	E/S digital 2
	CCP1	Compare input/output 1, salida PWM 1
18	RC3	E/S digital 3
	SCK	Señal de reloj en modo SPI
	SCL	Señal de reloj en modo I2C
23	RC4	E/S digital 4
	SDI	Dato de ingreso en modo SPI
	SDA	Dato de ingreso/salida en modo I2C
24	RC5	E/S digital 5
	SDO	Dato de salida en modo SPI
25	RC6	E/S digital 6
	TX	Transmisor en modo USART Asincrono
	CK	Señal de reloj en modo USART
26	RC7	E/S digital 7
	RX	Receptor en modo USART Asincrono
	DT	Datos en modo USART Asincrono

**Tabla 1. 6 Pines del puerto C<sup>29</sup>**

<sup>29</sup> Salesiana, 2009

Los pines del puerto D, sirven como función alterna, de respaldo, este puerto junto con sus pines puede ser configurado bidireccionalmente como todos los demás, puede ser un puerto esclavo junto con sus 8 bits.

La descripción de los pines del puerto B se detalla en la Tabla 1. 7.

Pin No.	Nombre de Pin	Descripción
33	RB0	E/S digital 0
	INT0	Interrupción externa 0
34	RB1	E/S digital 1
	INT1	Interrupción externa 1
35	RB2	E/S digital 2
	INT2	Interrupción externa 2
36	RB3	E/S digital 3
	CCP2	Compara- Ingreso/salida 2 PWM2
37	RB4	E/S digital 4
38	RB5	E/S digital 5
	PGM	Habilitador programación ICSP
39	RB6	E/S digital 6
	PGC	Clock de programación ICSP
40	RB7	E/S digital 7
	PGD	Dato de la programación ICSP

**Tabla 1. 7 Pines del puerto B<sup>30</sup>**

A continuación veremos el funcionamiento de algunos componentes relevantes para el manejo de nuestro proyecto:

Los módulos del TIMER0, TIMER1, TIMER2 y TIMER3 acceden directamente a los respectivos puertos a través del bus de datos, la memoria de datos EEPROM<sup>31</sup>, y finalmente tenemos el módulo del convertor análogo a digital.

El Timer0 puede funcionar como un contador de tiempo o como un contador simplemente. El modo de temporizador se selecciona borrando el bit T0CS, además el módulo Timer0 se incrementará cada ciclo de instrucción. Si el registro

<sup>30</sup> Salesiana, 2009

<sup>31</sup> EEPROM: Es un tipo de memoria ROM que puede ser programada, borrada y reprogramada

TMR0L está activo, el incremento se deshabilitará por los siguientes dos ciclos de instrucción.

El usuario puede evitar esto escribiendo un valor ajustado al registro TMR0L, el modo contador se selecciona ajustando el bit T0CS y el Timer0 se incrementará, ya sea en cada flanco ascendente o descendente del pin RA4/T0CKI

En el caso que la interrupción Timer0 se lleve en un tiempo determinado se deberá configurar como se ha indicado el TMR0L con la siguiente ecuación descrita:

$$Interrupcion = 4 \frac{1}{F} x(256 - ValorTMR0L)xPrescaler \quad (1.2)$$

Donde F consideraremos como la frecuencia de cristal oscilador, el Prescaler el valor del divisor de la frecuencia como se indica en la Tabla 1. 8 y finalmente el valor del TMR0L que puede ser un valor que asignemos o un valor que deseamos despejar según el tiempo de interrupción.

PS2	PS1	PS0	Divisor de TMRO	Divisor de WDT
0	0	0	1:2	1:1
0	0	1	1:4	1:2
0	1	0	1:8	1:4
0	1	1	1:16	1:8
1	0	0	1:32	1:16
1	0	1	1:64	1:32
1	1	0	1:128	1:64
1	1	1	1:256	1:128

**Tabla 1. 8 Divisor del Timer 0<sup>32</sup>**

El USART es un módulo que funciona como entrada y salida es decir establece una comunicación serial sincrónico o asincrónico. El USART puede configurarse

<sup>32</sup> Microship, Data Sheet, microcontrolador 16F870, 2008

como un sistema asíncrono full-dúplex<sup>33</sup> que puede comunicarse con dispositivos periféricos, como por ejemplo los ordenadores personales, o puede ser configurado como un sistema síncrono half-duplex<sup>34</sup> que se puede comunicar con dispositivos periféricos, tales como A/D<sup>35</sup> o D/A<sup>36</sup> de circuitos integrados, series de memorias EEPROM, etc.

A continuación se presenta un diagrama de bloques donde se visualiza el funcionamiento del módulo USART del micro-controlador:

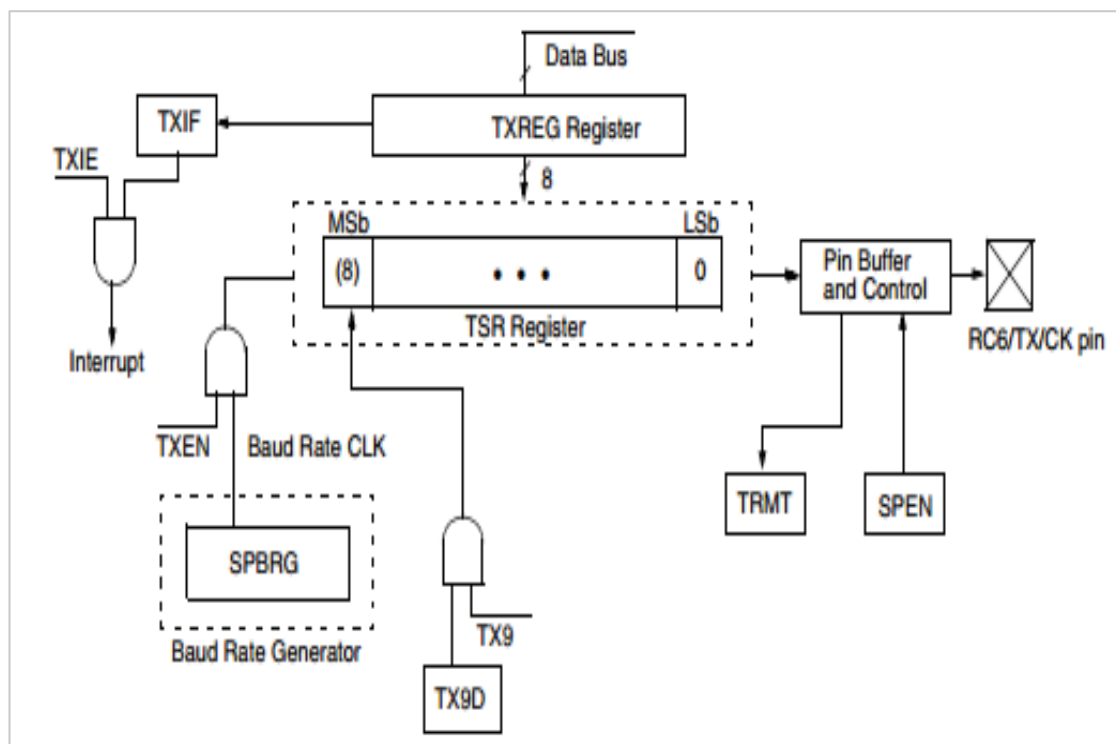


Figura 1. 10 Funcionamiento del módulo USART<sup>37</sup>

El I2C es un protocolo de comunicación que permite la transmisión de datos de forma serial, entre sus principales características tenemos que la comunicación es controlada por una señal de reloj común es decir es un bus de comunicación síncrono, el bus está formado por 2 hilos, el primero el SDA (Serial Data Line) para los datos y el SCL (Serial Clock Line), aunque también se ocupa una tercera línea que sirve de referencia, posee una velocidad de transmisión de 100 Kbits/s

<sup>33</sup> Full-duplex: Cuando los datos circulan en ambas direcciones a la vez

<sup>34</sup> Half-duplex: Cuando los datos circulan en una sola dirección por vez

<sup>35</sup> A/D: Análogo/Digital

<sup>36</sup> D/A: Digital/Análogo

<sup>37</sup> Microship, Data Sheet, microcontrolador 16F870, 2008

en estándar, 400 Kbits/s en rápida y una velocidad de 3,4 Mbits/s en ultra rápida, cada dispositivo del bus tiene una dirección única, la distancia y el número de dispositivos es limitado por la capacidad del bus, por lo general de 2 o 3 metros, el protocolo de acceso al bus del I2C es de tipo maestro-esclavo.

### 1.3.2 CARACTERÍSTICAS DEL MICRO-CONTROLADOR PI16F870

Debido a que el controlador de cada parada únicamente tiene que enviar una señal inalámbricamente se ha escogido al micro-controlador PI16F870 que tiene los parámetros básicos para realizar la función de transmisión.

Entre las principales características de este micro-controlador tenemos las siguientes:

Key Features PICmicro™ Mid-Range MCU Family Reference Manual (DS33023)	PIC16F870
Operating Frequency	DC - 20 MHz
RESETS (and Delays)	POR, BOR (PWRT, OST)
FLASH Program Memory (14-bit words)	2K
Data Memory (bytes)	128
EEPROM Data Memory	64
Interrupts	10
I/O Ports	Ports A,B,C
Timers	3
Capture/Compare/PWM modules	1
Serial Communications	USART
Parallel Communications	—
10-bit Analog-to-Digital Module	5 input channels
Instruction Set	35 Instructions

**Tabla 1. 9 Características del micro-controlador PI16F870<sup>38</sup>**

<sup>38</sup> Microship, Data Sheet, microcontrolador 16F870, 2008

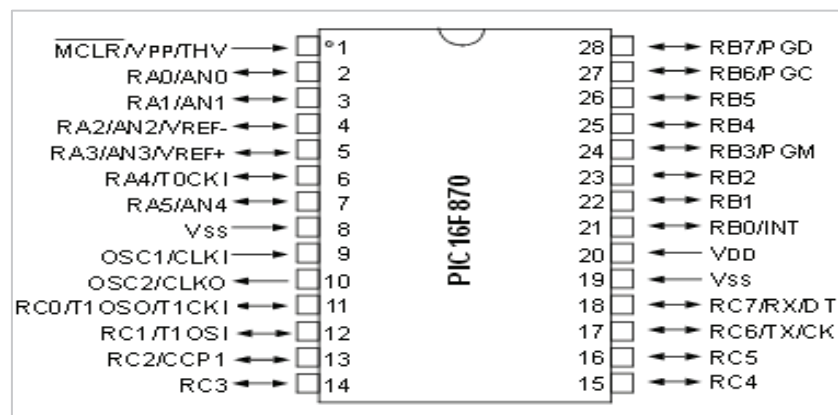


Figura 1. 11 Pines del PIC16F870

## 1.4 LA ELECTROVÁLVULA

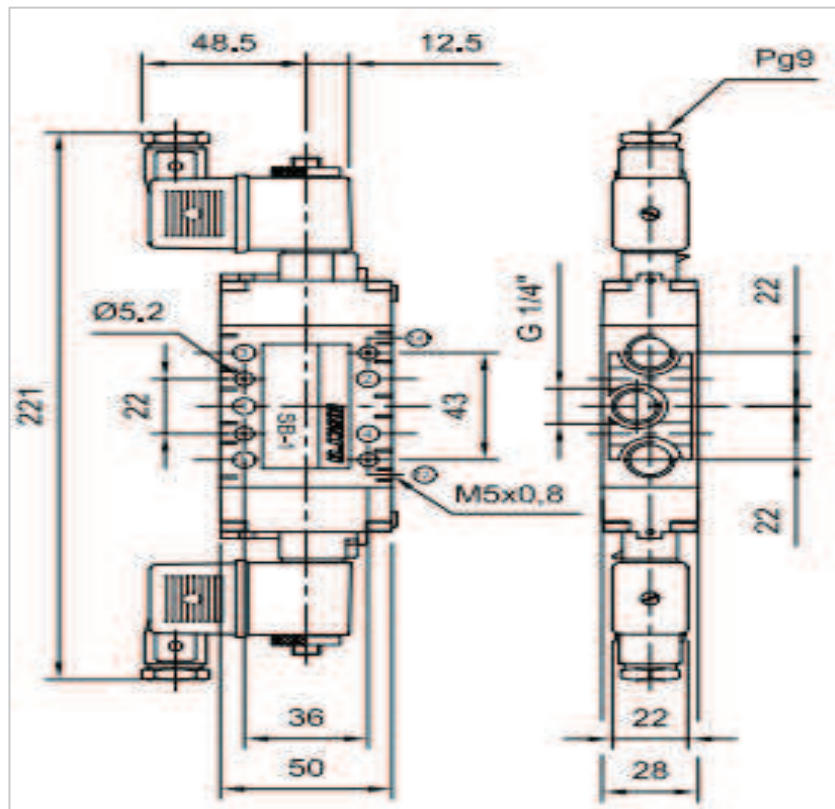
La electroválvula es un dispositivo diseñado para controlar el flujo de un fluido a través de un conducto como puede ser una tubería, no se debe confundir la electroválvula con válvulas motorizadas, que son aquellas en las que un motor acciona el cuerpo de la válvula<sup>39</sup>.

La electroválvula que ocupamos en nuestro proyecto es la “ELECTROVÁLVULA BI-ESTABLE”, cuya actuación es eléctrica con actuador manual biestable, es decir que se mueve en dos direcciones. También es posible construir electroválvulas bi-estables que usan un solenoide para abrir la válvula y otro para cerrar o bien un solo solenoide que abre con un impulso y cierra con el siguiente.

La temperatura de trabajo oscila entre los -5 grados hasta los 50 grados centígrados, el fluido es aire comprimido. En la Figura 1.12 se visualiza el esquema de la electroválvula.

<sup>39</sup> Villa, 2014

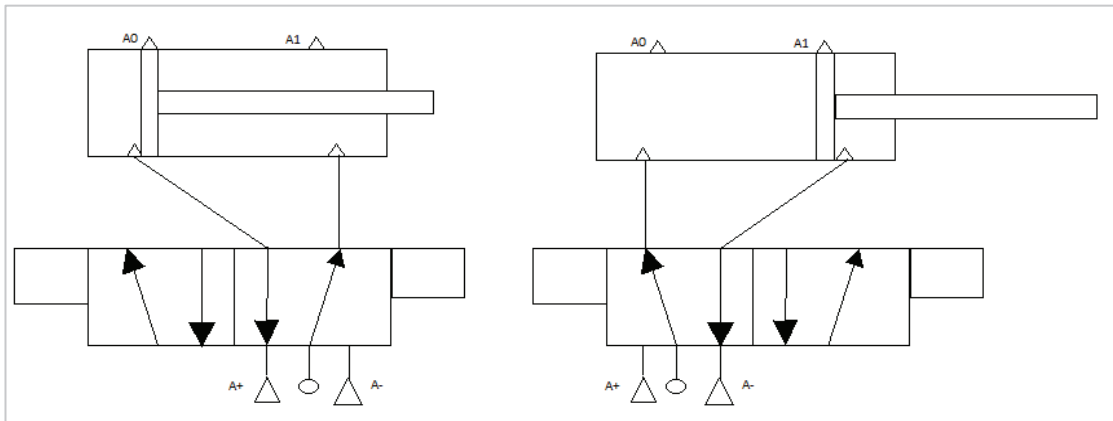




**Figura 1.12 Electroválvula Bi-Estable**

El funcionamiento como vemos en la Figura 1. 13, consiste en accionar la entrada de aire en la cámara de nuestro cilindro de doble efecto, para esto necesitamos impulsar nuestra electroválvula de la manera correcta, aplicando un voltaje de 24 voltios al solenoide y un compresor de aire apropiado, de tal manera que se puede simular el funcionamiento de un autobus real.

El gráfico de la izquierda muestra el cierre de la puerta ya que el aire que ingresa se encuentra en la cámara diestra, y el gráfico de la derecha indica la apertura de la puerta ya que el aire ingresa en la parte izquierda de la cámara.



**Figura 1. 13 Funcionamiento de la Electroválvula Bi-Estable**

## **CAPITULO 2**

### **CONSTRUCCIÓN DEL SISTEMA DE CONTROL**

#### **2.1 FUNCIONAMIENTO DEL SISTEMA DE CONTROL DE PUERTAS**

El presente capítulo se iniciará con una descripción del funcionamiento del sistema de control de puertas para un bus. Inicialmente se tendrá el circuito de control de puertas ubicado en el bus por lo que la alimentación de voltaje de todo el circuito se lo tomará de la batería del bus, la misma que tendrá que ser regulada para poder polarizar los diferentes componentes con su respectivo voltaje.

En el momento que los circuitos se encuentren polarizados, el micro-controlador se inicializará con los parámetros necesarios para su funcionamiento y validará si tiene un valor previamente almacenado en su memoria EEPROM para saber si tiene que leer o grabar la fecha y hora, en el caso de que se tenga que escribir la hora y fecha el micro-controlador enviará la señal con los parámetros a almacenarse en el circuito del reloj de tiempo real DS1307 para su posterior lectura y finalizará el proceso grabando un dato en la memoria EEPROM del micro-controlador lo que permitirá que al reiniciarse el circuito no se vuelva a grabar la hora y únicamente permita visualizar la fecha y hora que se visualiza en la pantalla LCD, la misma que tendrá que estar ubicada en el tablero del bus.

Después de inicializarse el circuito del control de bus, el módulo de radiofrecuencia empezará a buscar alguna señal que haya sido enviada por otro módulo de radiofrecuencia que pertenezca a la misma red de módulos. Por tal motivo cuando el bus se encuentre dentro del rango de transmisión de un circuito colocado previamente en cada parada, se recibirá una señal que será procesada por el módulo de radiofrecuencia y posteriormente enviada al micro-controlador, la misma que será usada para identificar la parada en que se encuentra el bus durante su recorrido. Después de validar la señal, el micro-controlador enviará un

nivel lógico al circuito que permitirá controlar la electroválvula que se encuentra ubicada en cada puerta, la misma que será usada para abrir físicamente las puertas del bus y posteriormente el micro-controlador capturará la hora en que se realiza esta acción.

Una vez que el bus parta de la parada y salga del rango de transmisión, el micro-controlador enviará una señal lógica que permitirá que la electroválvula cierre las puertas del bus y posteriormente el micro-controlador almacenará la hora en la que ocurre el evento.

Finalmente el bus toma la información obtenida al momento de pasar por cada parada y enviará la información a través de una conexión serial hacia el módulo USB-flash, el mismo que procesará la información y permitirá grabarla en un archivo TXT creado previamente en una memoria extraíble.

## 2.2 CONFORMACIÓN DEL SISTEMA DE CONTROL DE PUERTAS

A continuación presentamos un diagrama de bloques que refleja los elementos usados para el circuito de control de bus.

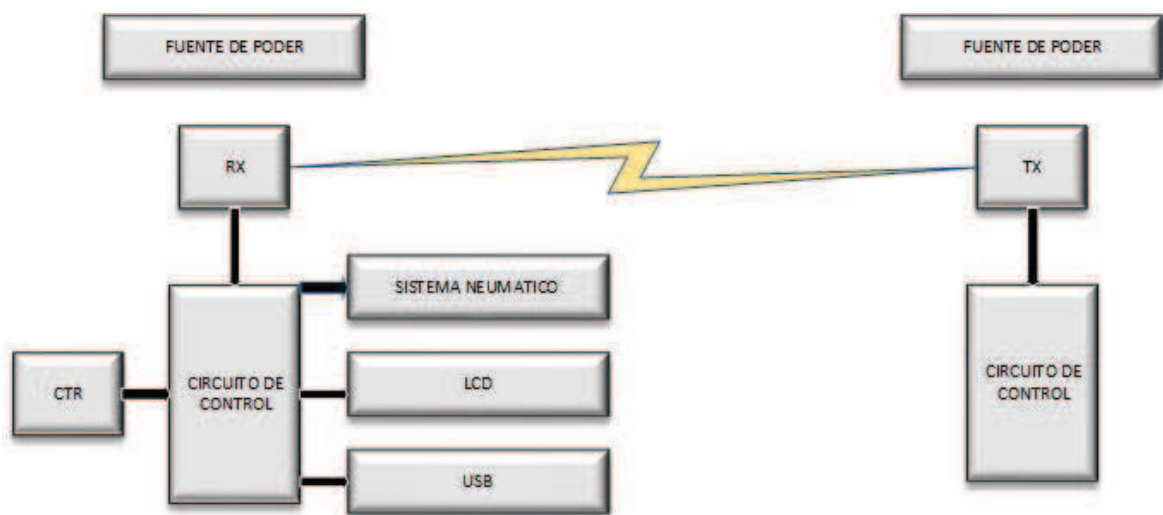


Figura 2. 1 Diagrama del sistema de control de puertas

## 2.3 CONSTRUCCIÓN DEL CIRCUITO TRANSMISOR DE RF Y CONTROL DE PUERTAS EN EL BUS

### 2.3.1 FUENTE DE ALIMENTACIÓN

La fuente de alimentación permitirá polarizar los componentes formados por nuestro sistema de control de puertas según el voltaje necesario.

En la Figura 2. 29 se visualizará el esquema de la alimentación completa del circuito donde se considera una alimentación de 24 V que es el voltaje utilizado en las baterías de un bus. Debido a que se tiene varias etapas que funcionan con un valor determinado de voltaje se ha establecido 3 etapas de alimentación 24, 5 y 3.3 V, según el dispositivo a manejar.

La primera etapa de alimentación es de 24 V que alimentará específicamente al funcionamiento de la electroválvula a usar.

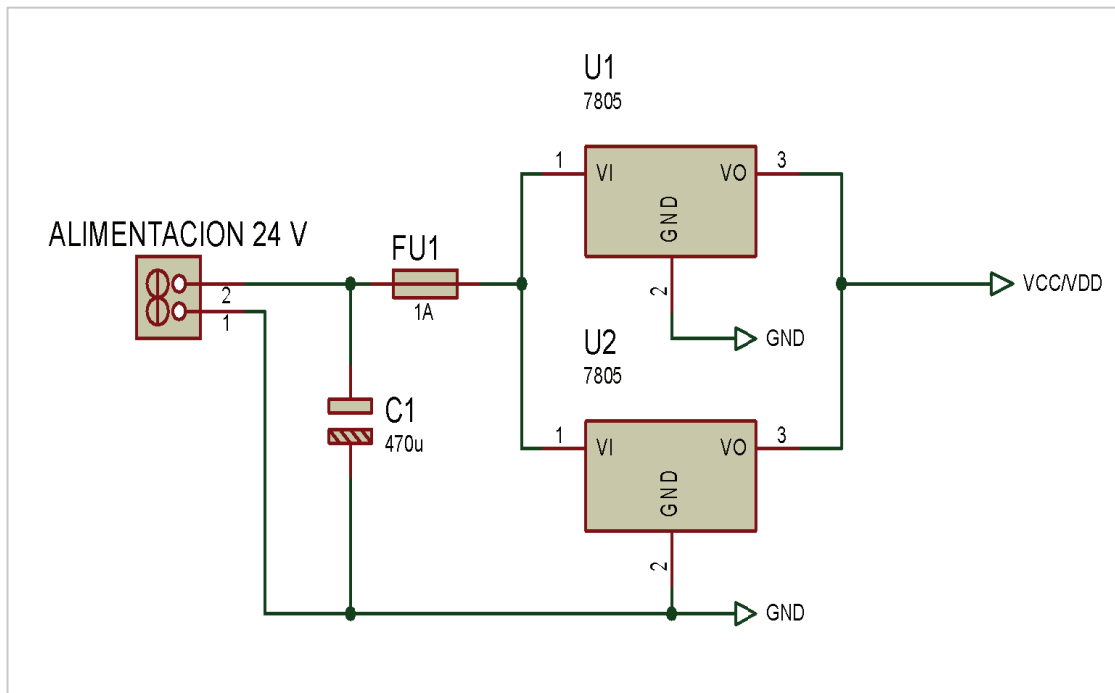
En la Figura 2. 2 se visualiza la segunda etapa de alimentación de 24 V a 5 V mediante el uso en paralelo de dos circuitos LM7805 que regulariza el voltaje a 5 V mediante el uso del capacitadores C1 (470uF) que mantendrá un voltaje de entrada constante. Hay que aclarar que el uso de los dos reguladores en paralelo se debe a que en la realización de los circuitos al conectar varios dispositivos que funcionan con diferentes cantidades de corriente DC no permitían su correcto funcionamiento por tal motivo se escogió este modelo de conexión que aumenta la corriente a unos niveles que no afecte a los circuitos.

Debido al voltaje a regular y la temperatura de operación del integrado (78°C) se deberá utilizar disipadores de temperatura<sup>40</sup>.

Una vez obtenido el voltaje de referencia de 5 V, se lo utilizará como fuente de alimentación tanto para el micro-controlador, pantalla LCD, reloj de tiempo real, como para la tarjeta de grabación de USB-FLASH.

---

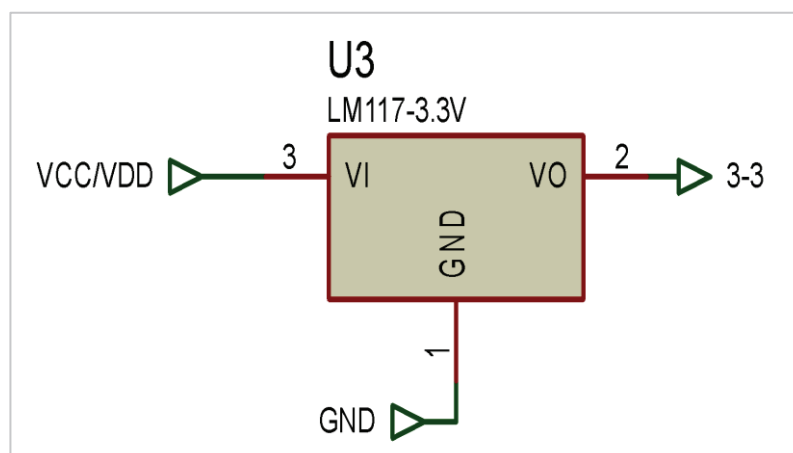
<sup>40</sup> Instruments, Data Sheet, regulador de voltaje LM7805, 2014



**Figura 2. 2 Regulador de voltaje de 24 V a 5 V**

En la Figura 2. 3 se visualiza la tercera etapa de regularización de la fuente de alimentación de 5 V a 3.3 V mediante la utilización del LM117T, debido a que ya se tiene un voltaje estable de 5 V por parte del LM7805 se ha establecido un esquema de conexión básico del LM117T propuesto por el fabricante<sup>41</sup>.

Obtenido el voltaje de alimentación de 3.3 V se procederá con la alimentación de los módulos de radiofrecuencia XBEE.



**Figura 2. 3 Regulador de voltaje de 5 V a 3.3 V**

<sup>41</sup> Instruments, Data Sheet, regulador de voltaje LM117T, 2014

### 2.3.2 RELOJ DE TIEMPO REAL

El DS1307 es un reloj de tiempo real con batería de respaldo que permite a nuestro proyecto con micro-controlador realizar un seguimiento del tiempo, aún si no se reprograma o si se pierde el voltaje de alimentación.

Mediante la programación del micro-controlador, este permitirá que el DS1307 proporcione información de segundo, minuto, hora, día, fecha, mes y año. El reloj funciona tanto en el formato de 24 horas o de 12 horas con indicador AM / PM.

Cuando el bus corta su suministro eléctrico el circuito del control de bus también detendrá su funcionamiento a excepción del circuito DS1307 que internamente está conformado por un circuito integrado que detecta fallas de energía y cambia automáticamente a la alimentación de reserva.

Como vemos en la Figura 2. 4, los pines X1(1) y X2(2) son las, entrada y salida respectivamente del oscilador, por lo general colocamos un oscilador de 32,768 KHz, el pin VBAT (3) es la entrada de la alimentación de reserva, que puede ser una batería de litio de 3 V, en caso de no tener batería de respaldo este pin debe ser conectado a tierra. El pin SDA (5) es la entrada/salida de datos para la interfaz serial I2C, el pin SCL (6) es la entrada del reloj para la interfaz I2C y se utiliza para sincronizar el movimiento de datos en la interfaz de serie, por último tenemos el pin SOUT (7) que es el conductor de onda cuadrada, requiere de conexión con alguna fuente externa que suele ser de unos 5,5 V junto con una resistencia que nos dará una señal de onda cuadrada de 1, 4, 8 y 12 KHz según lo requerido<sup>42</sup>.

---

<sup>42</sup> Dallas, 2014

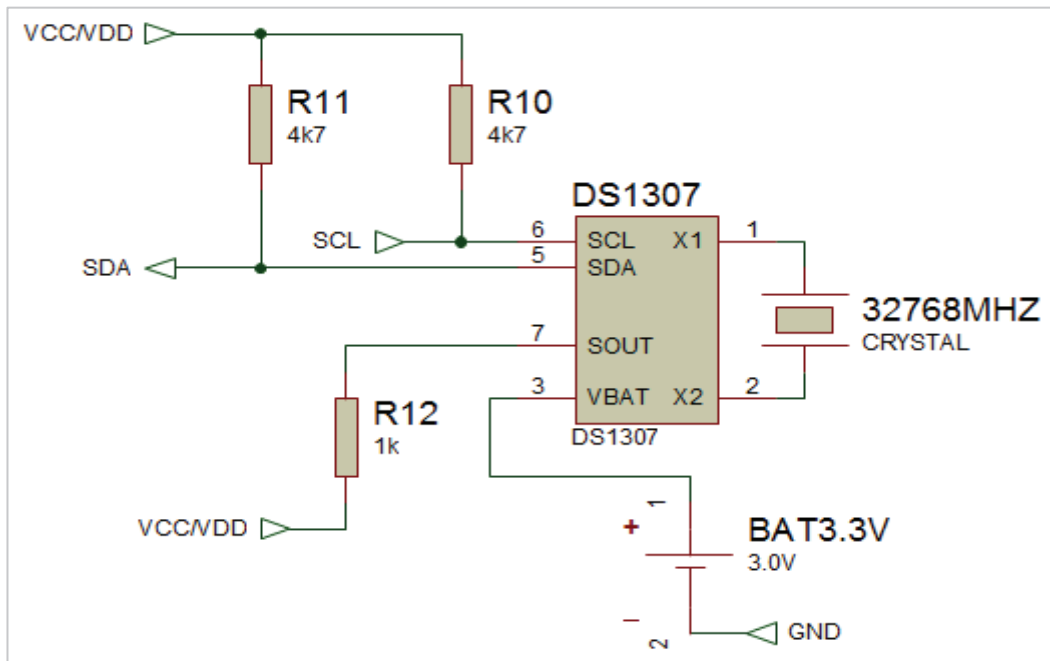


Figura 2. 4 Esquema de conexión del reloj de tiempo real DS1307

### 2.3.3 VISUALIZACIÓN MEDIANTE LCD

El display LCD 16x4 es un visualizador pasivos que mediante pantallas planas y delgadas conformadas por cierto número de pixeles monocromos que permitirán realizar la visualización de la fecha y hora enviados desde el micro-controlador, como también permitirá visualizar puntualmente cuando se realice la apertura y cierre de las puertas cuando se reciba la señal del transmisor de cada parada<sup>43</sup>.

Como vemos en la Figura 2. 5 este LCD es alimentado mediante los pines 2 y 15 con 5 V, los pines 1 y 16 (A y K) son enviados a tierra generalmente, el pin 3 nos sirve para ajustar el contraste de la pantalla, todo esto dependiendo de la configuración que el usuario le quiera dar a su LCD pero en este caso se lo ha conectado a través de una resistencia de 1K $\Omega$  a tierra para mantener un contraste constante; los pines D0, D1, D2, D3, D4, D5, D6 y D7 para nuestro caso son de recepción de información enviada por nuestro micro-controlador PI18F452 simulando un bus de datos paralelo para que posteriormente se pueda visualizar la información en la pantalla.

<sup>43</sup> Hernández, 2012



Por último tenemos a los pines RS que es el de selección de señal de registro y el pin E, que habilita el paso o no de la información, que de igual manera están conectados al micro-controlador utilizado.

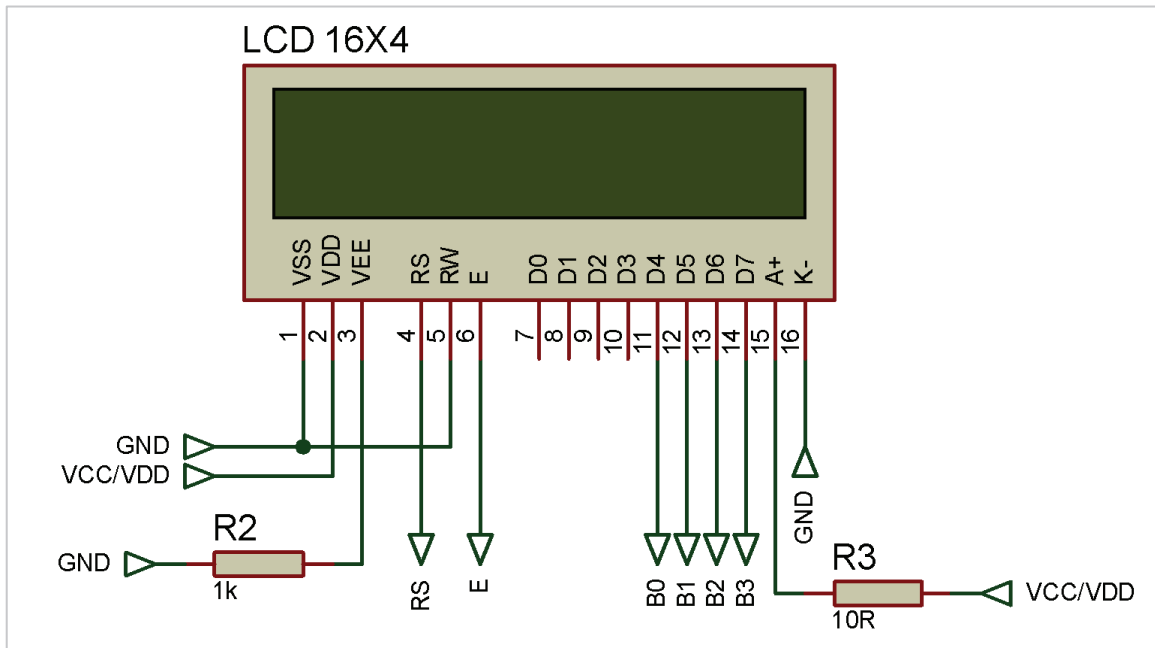


Figura 2. 5 Display LCD 16x4

### 2.3.4 CONTROL DE ALIMENTACIÓN PARA LA ELECTROVÁLVULA

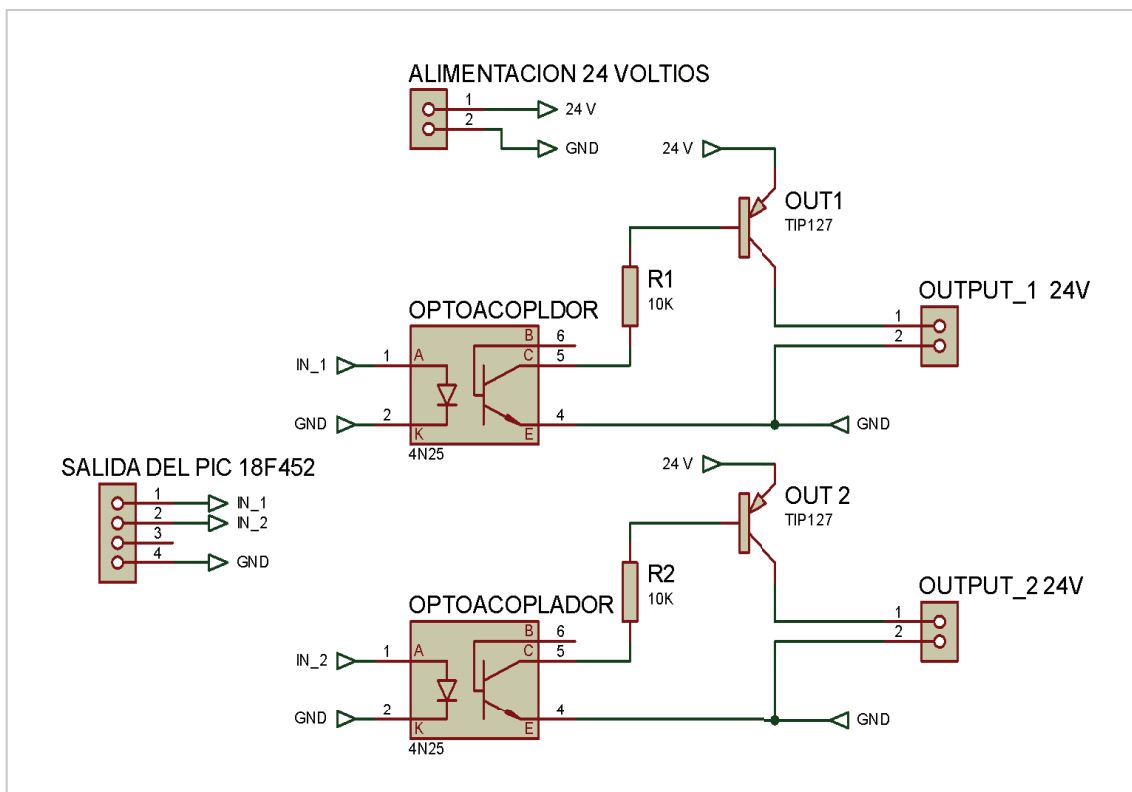
Como se ha indicado en capítulos anteriores la electroválvula funciona con 24 V, por lo que el presente circuito nos permitirá controlar la alimentación de la electroválvula mediante un 1 y 0 lógico del micro-controlador y tomando en cuenta que nos ayudará también como circuito de protección entre la alimentación de 24 voltios y el micro-controlador.

En la Figura 2. 6 se visualiza la conexión entre el micro-controlador y el optoacoplador 4N25 en los pines RD4 – 1 y RD5 - 1, donde si la señal enviada desde el micro-controlador es de 1 lógico provocará que internamente en el optoacoplador el led se encienda, activando el fototransistor NPN<sup>44</sup> y haciéndole que entre en saturación, cuya base activará la carga y se complete el circuito de

<sup>44</sup> NPN: Componente semiconductor con 3 terminales, base, emisor y colector, formado por un cristal que contiene una región positiva entre 2 negativas

polarización del transistor TIL127, mientras que al enviar un 0 lógico al optoacoplador, el led no se encenderá, no activará al fototransistor y por ende no entrará en saturación por lo que la base no podrá activar la carga y no se polarizará el transistor TIL127.

Cuando se encuentra polarizado el circuito del transistor TIL127 se tendrá 24 V en el solenoide A de la electroválvula el mismo que abrirá las puertas del bus; y en el caso de que el circuito de polarización se encuentre en estado abierto, el voltaje en el solenoide A será de 0 V; pero se mantendrá abierto las puertas del bus hasta que no se active el solenoide B, por tal motivo se tiene un segundo circuito de polarización del TIL127 que se activará según las órdenes del microcontrolador en el momento adecuado.



**Figura 2. 6 Circuito de alimentación de la electroválvula**

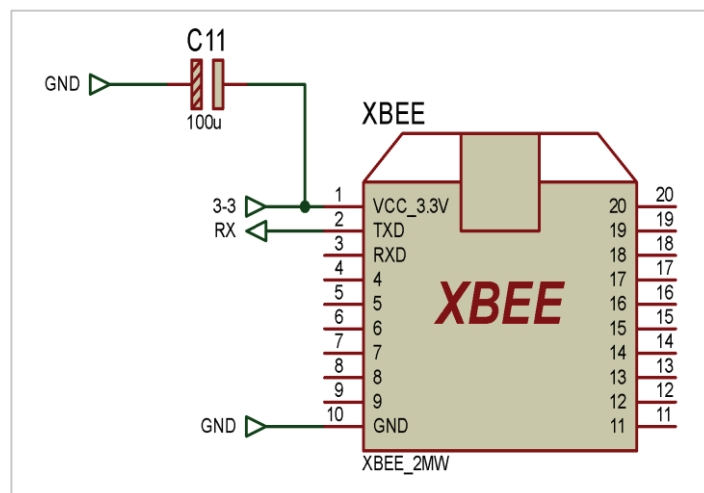
### 2.3.5 CONEXIÓN DEL RECEPTOR DE RF

En la Figura 2. 7 se visualiza la conexión del receptor de radiofrecuencia XBEE que permitirá recibir los datos del transmisor ubicado en las paradas, es decir que

el módulo de RF se encontrará escuchando el canal de RF en espera de algún dato, el mismo que lo enviará al micro-controlador.

La alimentación del módulo de radiofrecuencia será de 3.3 V obtenido del regulador LM117T del pin 3 hacia el pin 1 del módulo de radiofrecuencia a través de un capacitor electrolítico de 1 uF que servirá de protección para el circuito manteniendo un voltaje constante en el caso de una disminución momentánea de la alimentación.

Cuando el transmisor de las paradas envía una señal en este caso la letra A el módulo de radiofrecuencia recibirá la señal y lo enviará al registro de Rx donde se decodificará la señal y se enviará la información al buffer de entrada pero como previamente se realizó la configuración del módulo de radiofrecuencia en modo transparente la información será enviada mediante una conexión half-duplex desde el pin 3 del módulo hacia el micro-controlador pin RX, lo que previamente permitirá la apertura de las puertas hasta que el módulo XBEE no reciba la señal del transmisor.



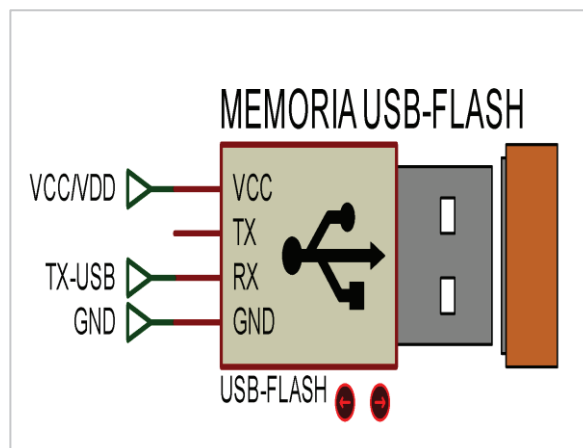
**Figura 2. 7 Conexión del receptor de RF**

### 2.3.6 CONEXIÓN DEL MÓDULO USB – FLASH

El módulo USB-FLASH nos permitirá almacenar información en una memoria extraíble como son las memorias USB.

Cuando el bus llega o parte de una parada el micro-controlador se encargará de enviar la información de la parada, hora y fecha a través de una comunicación serial desde el pin Tx del micro-controlador hacia el PIN 1 del módulo USB-Flash que se encargará de recibir y almacenar la información en un archivo TXT previamente creada en una memoria extraíble.

Como se observa en la Figura 2. 8 el módulo USB-Flash consta de 4 pines de conexión donde el Pin 1 y Pin 4 corresponderá a la alimentación de 5 V y GND que necesita tanto para la alimentación del módulo como también para la alimentación de la memoria extraíble. Los Pines Rx y Tx corresponderán a los puertos de recepción y envío de información serial que se realizará hacia el micro-controlador, debido a que no es necesario tener un eco de información del comportamiento del módulo USB-Flash por tal motivo únicamente se conectará al puerto Rx y no al Tx.



**Figura 2. 8 Módulo USB-FLASH**

### 2.3.7 CONEXIÓN AL MICRO-CONTROLADOR PI18F452

El micro-controlador es la cabeza del circuito que permitirá manejar la información recibida y manejar los dispositivos conectados a este previo una programación realizada en el micro-controlador.

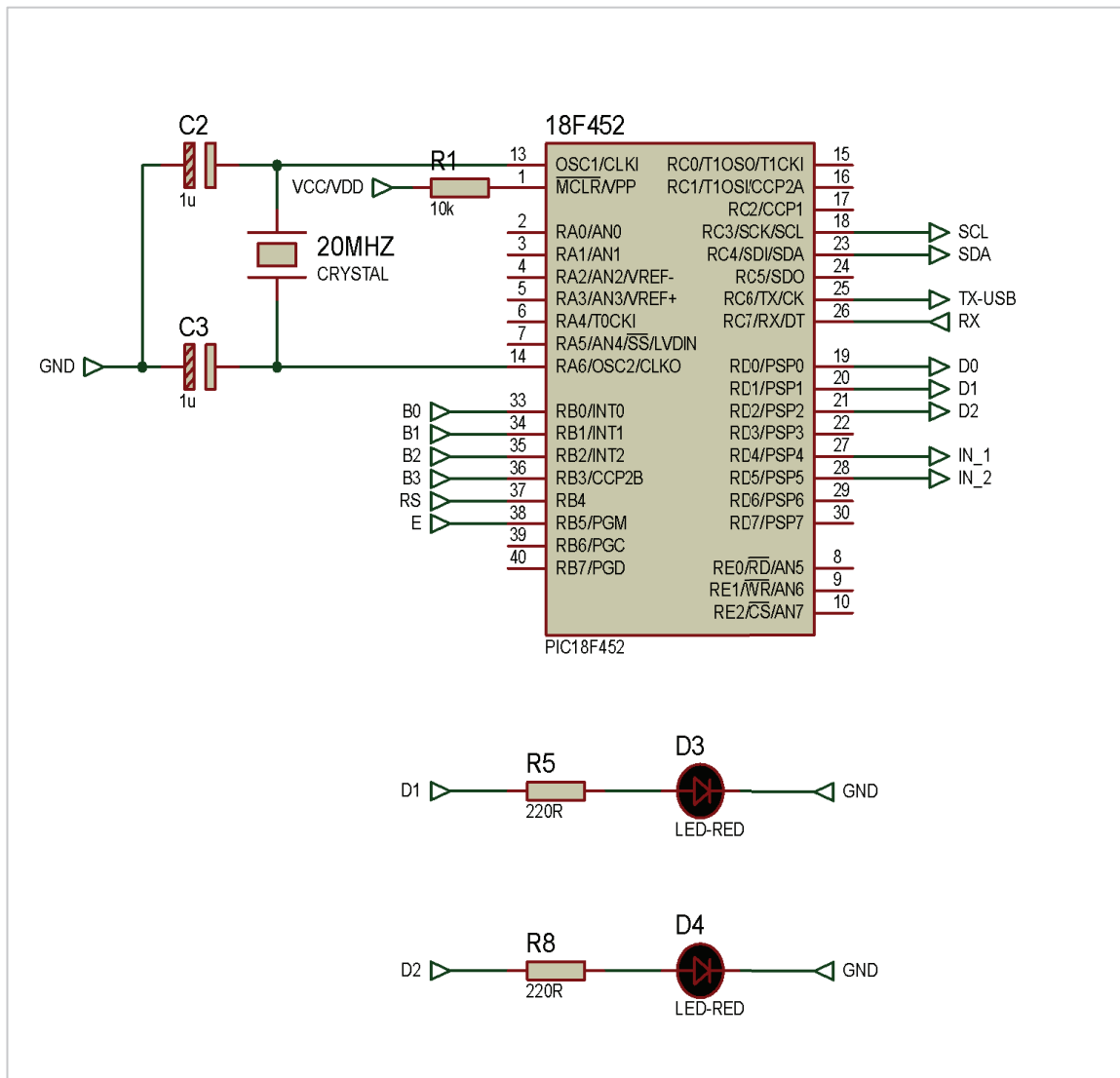
Como podemos ver en la Figura 2. 9, la fuente de alimentación ingresa por el pin 1 MCLR/Vpp, a través de una resistencia de  $10K\Omega$ , colocamos un cristal oscilador de frecuencia de 20 MHz entre los pines 13 OSC1 y 14 RG6/OSC2, direccionamos dichos pines a tierra pasando por un capacitor de 1uF cada uno.

Hay que aclarar que se escoge el cristal oscilador con un valor de 20 MHz, debido a que se tiene una cantidad de 500 líneas de programación por lo que se requiere que cada instrucción tome un ciclo de 200 ns, lo que permitirá que el micro-controlador escuche el canal de RF en un tiempo determinado en un tiempo menor, también hay que aclarar que en la programación del micro-controlador se inicializó la comunicación USART con una velocidad de transmisión en 9600 baudios por lo que, al escoger un cristal de 20 MHz, nos permitirá disminuir el error de generar la velocidad de transmisión de 9600 baudios.

Para la conexión de la pantalla LCD se han utilizado los pines 33 RB0, 34 RB1, 35 RB2, 36 RB3, 37 RB4 y 38 RB5 en el micro-controlador y los pines B0, B1, B2, B3, RS y E respectivamente en el LCD. El pin 18 SCL que es de salida del controlador, es la entrada del reloj de tiempo real para la comunicación a través de la interfaz I2C, el pin 23 SDA es de entrada/salida para el interfaz I2C de comunicación con el reloj de tiempo real de la misma manera.

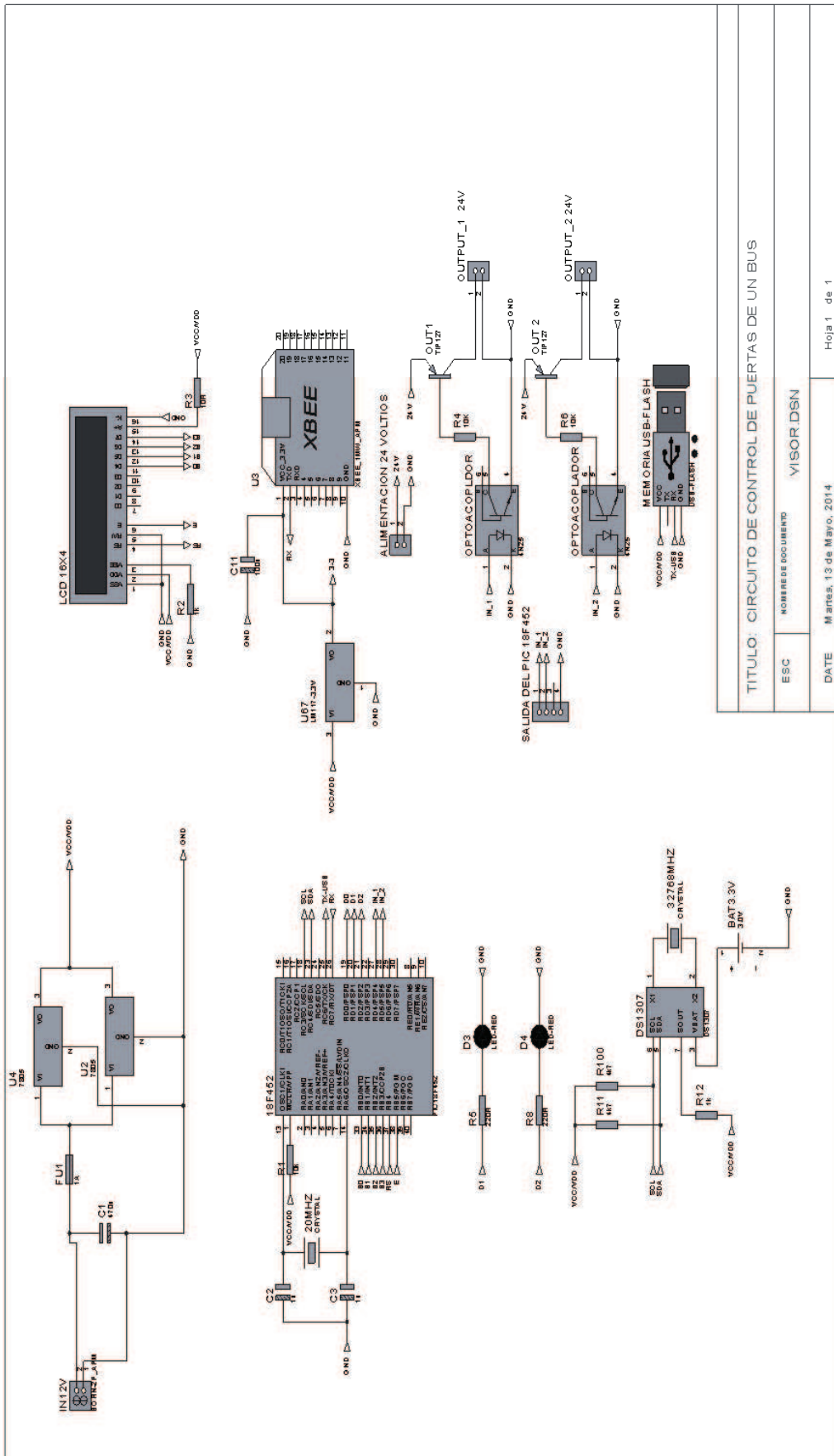
El pin 25 TX/RC6 está direccionado hacia el circuito de la flash-USB para la extracción de los datos, Rx es el pin utilizado para la recepción de información emitida por nuestro módulo de RF XBEE.

Los pines 19 RD0 y 20 RD1 se encuentran conectados a tierra pasando por resistencias de 220  $\Omega$  y leds de señalización que muestran el funcionamiento correcto mediante indicadores luminosos; por último tenemos los pines 27 RD4 y 28 RD5 direccionados hacia la entrada de los opto-acopladores que manejará la electroválvula y que posteriormente se dedicará a la apertura y cierre de la puerta según lo planificado y programado durante todo el desarrollo del proyecto.



**Figura 2. 9 Conexión al micro-controlador PI18F452**

En las Figuras 2.10, 2.11, 2.12 y 2.13 se muestran el esquema electrónico, diagrama del circuito realizado en ares y el circuito impreso del control de puertas del bus como el circuito impreso del opto-acoplador.



TITULO: CIRCUITO DE CONTROL DE PUERTAS DE UN BUS	
ESC	NOMBRE DE DOCUMENTO VISOR.DSN
DATE	Martes, 13 de Mayo, 2014
Hoja 1 de 1	

Figura 2. 10 Esquema electrónico del control de puertas del bus

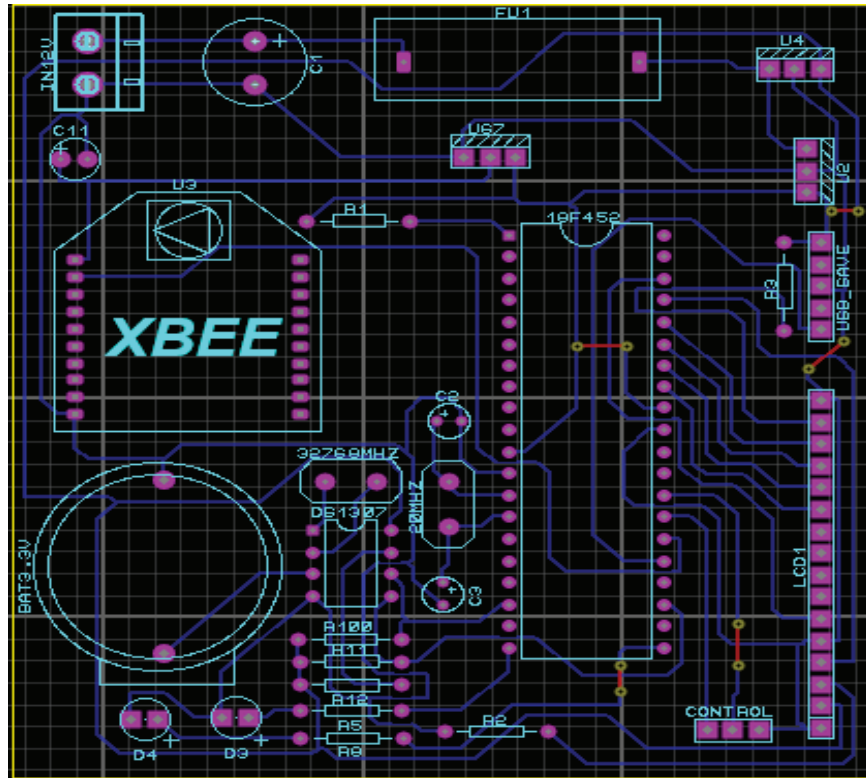


Figura 2. 11 Diagrama del circuito realizado en ares del control de puertaa del bus

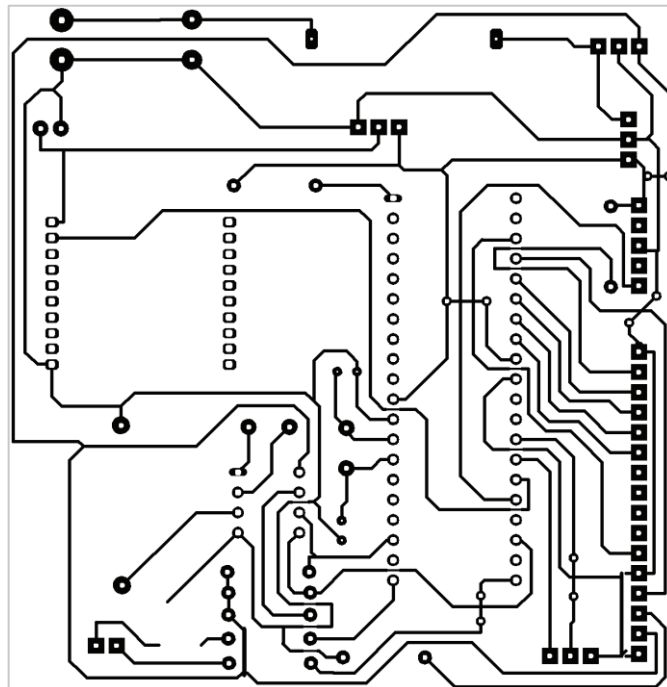
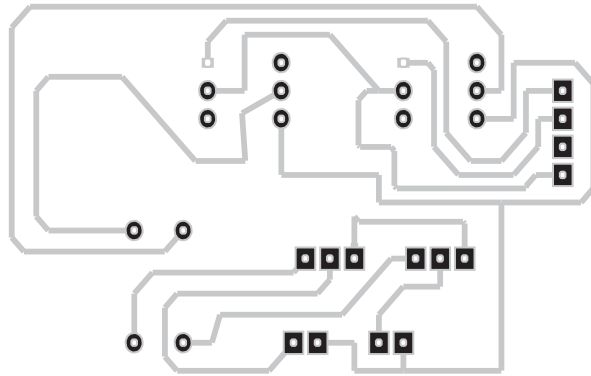


Figura 2. 12 Circuito impreso del control de puertaa del bus





**Figura 2. 13 Circuito impreso del opto-acoplador**

## **2.4 CONSTRUCCIÓN DEL CIRCUITO TRANSMISOR DE RF EN PARADAS**

A continuación detallaremos los elementos utilizados en la construcción de nuestro circuito transmisor de RF en las paradas, el mismo que está diseñado con el propósito de enviar una señal constante para que el control del bus reconozca si se encuentra en una parada y permita la apertura de las puertas.

### **2.4.1 CONEXIÓN AL MICRO-CONTROLADOR PI16F870**

El propósito del micro-controlador PI16F870 es de generar la señal que será enviada al transmisor de RF para su transmisión.

Entre los pines 9 y 10 (osc1 y osc2), colocamos un cristal oscilador de frecuencia de 20 MHz, además direccionamos cada uno de estos pines a tierra pasando por un capacitor de 1 uF, la fuente de alimentación ingresa por el pin 1 (MCLR, Vpp) con la resistencia adecuada para el ingreso de voltaje requerido, a los pines 21 RBO y 22 RB1 los utilizamos como señalización de funcionamiento, dirigidos a tierra pasando por resistencias adecuadas y leds para visualización, el pin 17 RC6 sirve para transmitir datos de sincronización que puede ser para la señal de reloj, por último, el pin 18 RC7 es la entrada de la transmisión de datos otorgados por el módulo de radiofrecuencia Xbee.

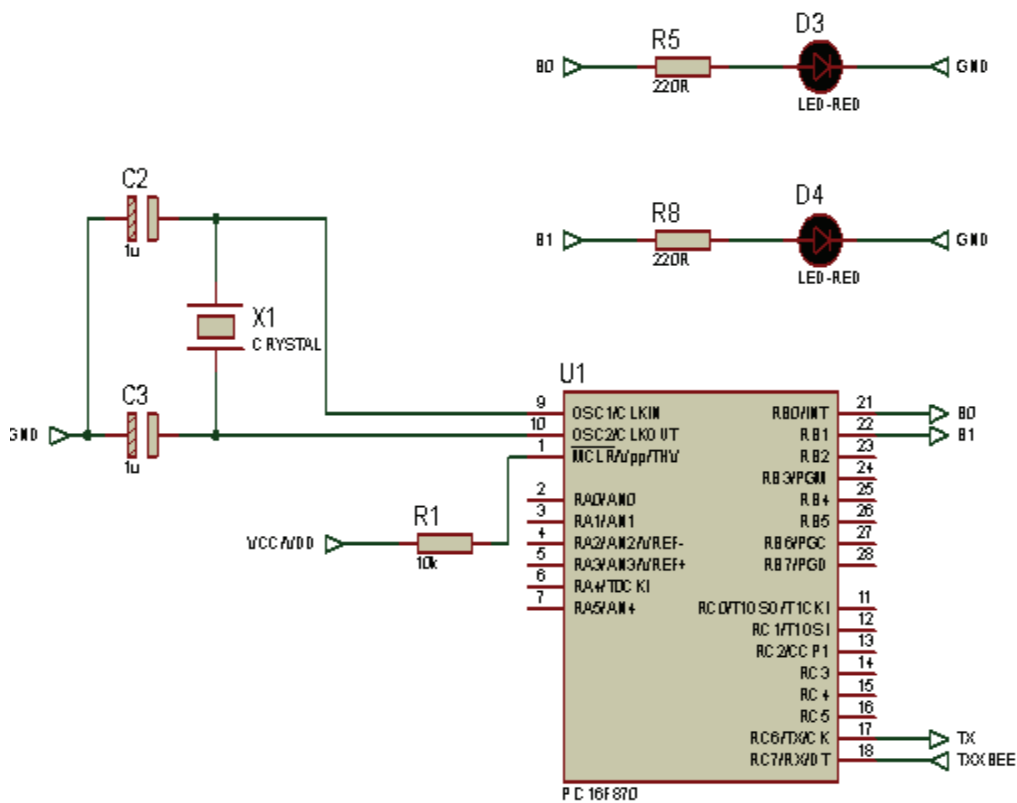
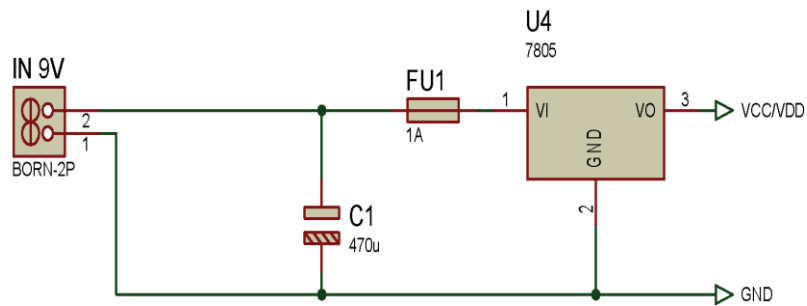


Figura 2. 14 Conexión al micro-controlador PIC16F870

#### 2.4.2 REGULADOR DE VOLTAJE DE 9 V A 5 V

Como se ha indicado en capítulos anteriores es necesario establecer un voltaje para los elementos que conforman nuestro circuito.

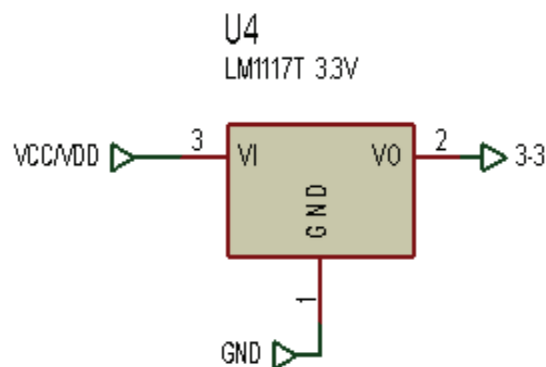
Respecto al voltaje que ingresa en el micro-controlador antes mencionado, es de 5V, este después de haber pasado por un proceso de regulación de voltaje cuya descripción se encuentra a continuación; inicialmente tenemos una fuente de alimentación de 9 V, la cual ingresa a un regulador de voltaje LM7805 por el pin 1 V1, pasando por un fusible de 1 A para resguardar el circuito, para posteriormente entregar un voltaje de 5 V por el pin 3 V0 requerido por nuestro controlador, el pin 2 GND del regulador de voltaje lo dirigimos a tierra junto con la salida del capacitor conectado entre la fuente de alimentación de 9 V.



**Figura 2. 15 Regulador de voltaje de 9 V a 5 V**

### 2.4.3 REGULADOR DE VOLTAJE DE 5 V A 3.3 V

Luego del proceso de regularización anteriormente mencionado pasamos a otro, en donde regulamos el voltaje a 3,3 V que es el requerido por el módulo de radiofrecuencia Xbee, para obtener dicho voltaje utilizamos el regulador LM117T, en donde el pin 3 V1 es de entrada de voltaje de 5 V otorgado por el anterior proceso, el pin 2 V0 es el de salida de voltaje requerido, 3,3 V y el pin 1 GND va direccionado a tierra como hemos visto anteriormente.

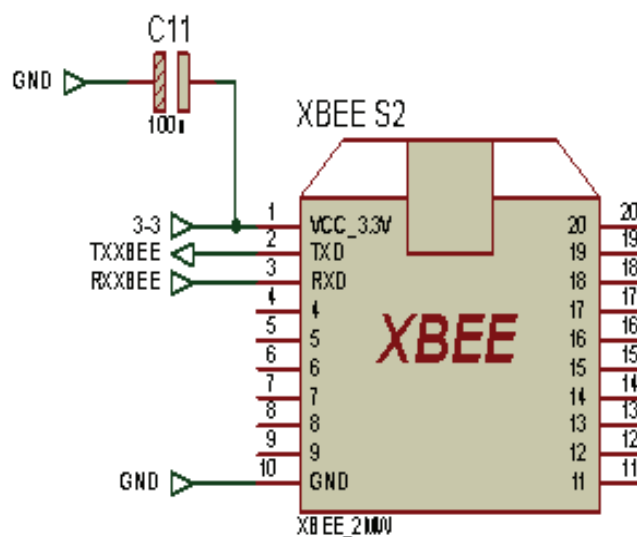


**Figura 2. 16 Regulador de voltaje de 5 V a 3.3 V**

### 2.4.4 CONEXIÓN DEL RECEPTOR DE RF

En este caso el módulo Xbee nos permitirá transmitir la información en la frecuencia de 2.4 GHz a los diferentes módulos de RF que se encuentren dentro de su rango.

Las configuraciones de los pines del módulo de RF Xbee, serán por el pin 1 Vcc, tenemos el ingreso de nuestra fuente de alimentación de 3,3 V otorgada por los 2 procesos de regulación anteriores, de igual manera conectamos un capacitor direccionado a tierra para mejorar la entrada de dicho voltaje hacia el módulo; el pin 2 TXD permite la transmisión de datos hacia el micro-controlador, el pin 3 RXD es el que recibe la información entregada por el micro-controlador, pasando por un proceso de filtrado con un diodo y una resistencia en paralelo, por último tenemos el pin 10 GND que obviamente va conectado a tierra.



**Figura 2. 17 Conexión del receptor de RF**

A continuación tenemos la Figura 2. 18, que es el circuito completo del transmisor RF en las paradas, como vemos está compuesto principalmente por nuestro micro-controlador PI16F870 y nuestro módulo de RF Xbee, junto con los componentes anteriormente mencionados necesarios para su funcionamiento.

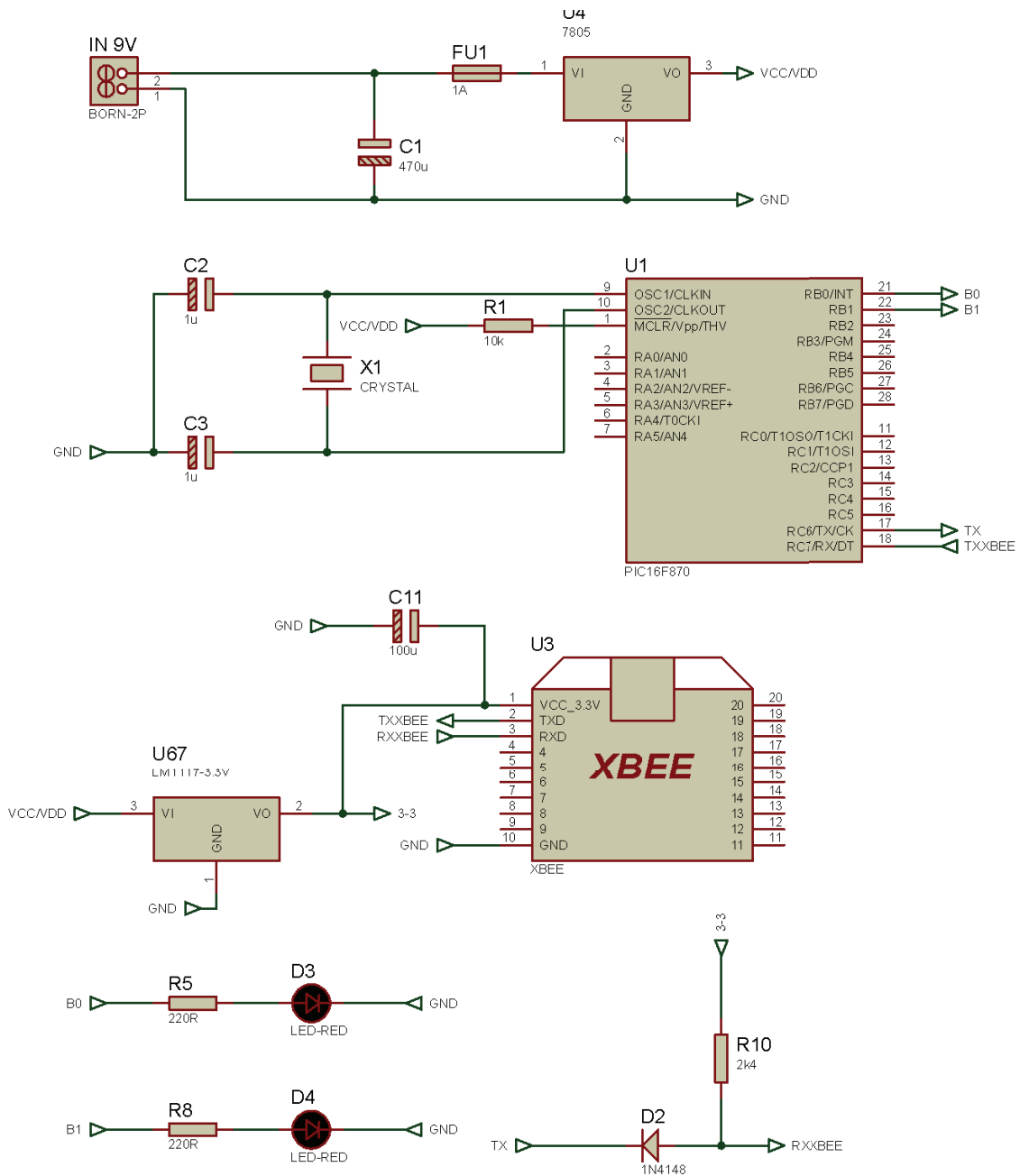


Figura 2. 18 Circuito transmisor de cada parada

En las Figuras 2.19, 2.20 y 2.21 se muestran el esquema electrónico, diagrama del circuito realizado en ares y el circuito impreso del transmisor en las paradas.

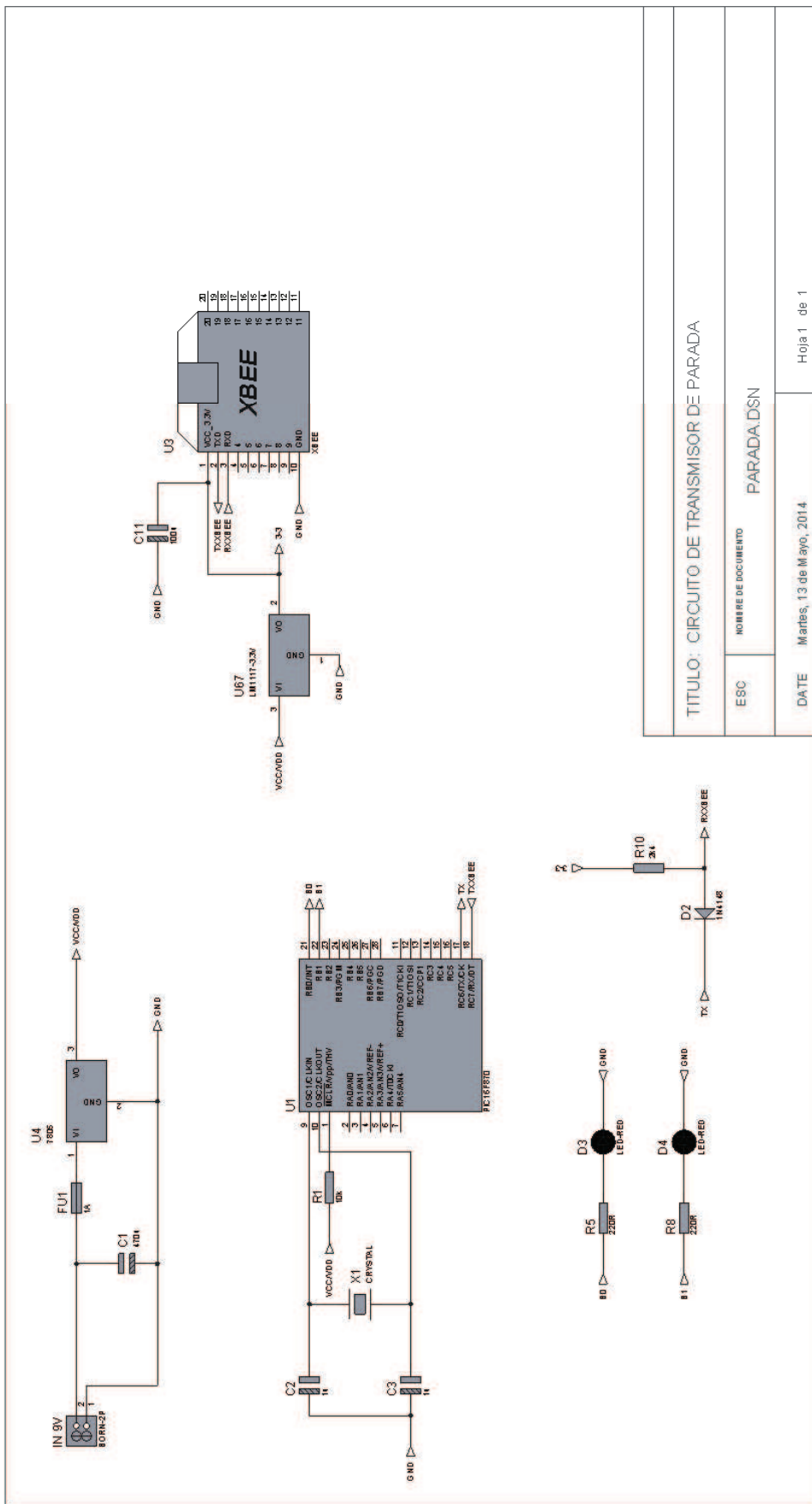


Figura 2. 19 Esquema electrónico del transmisor en las paradas

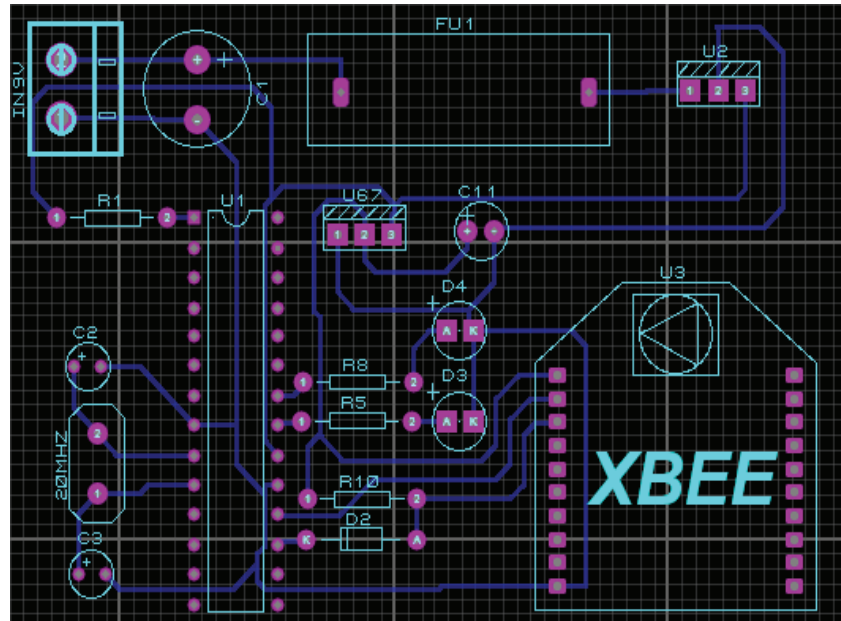


Figura 2. 20 Diagrama del circuito realizado en ares del transmisor en las paradas

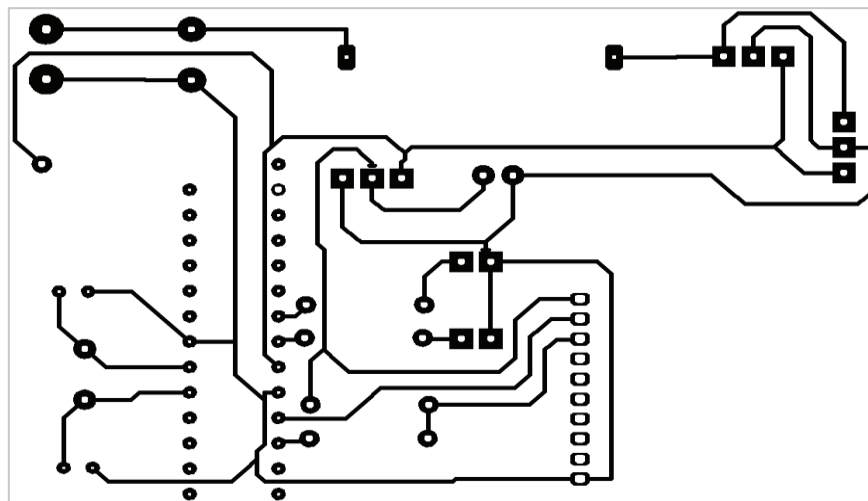


Figura 2. 21 Circuito impreso del transmisor en las paradas

En las siguientes figuras se presentan las fotografías que hacen referencia a los circuitos ya realizados tanto del control del bus como el circuito de cada parada y su implementación en el prototipo de bus.



Figura 2. 22 Circuito del sistema de control de puertas de bus



Figura 2. 23 Circuito del opto-acoplador



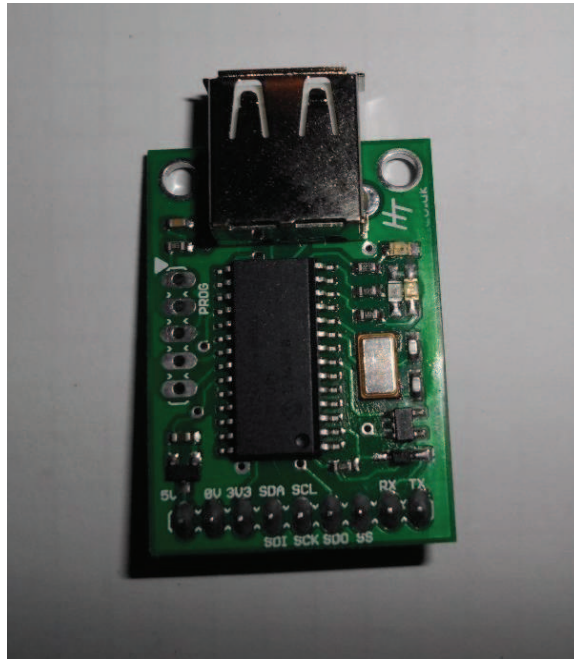


Figura 2. 24 Módulo USB-FLASH

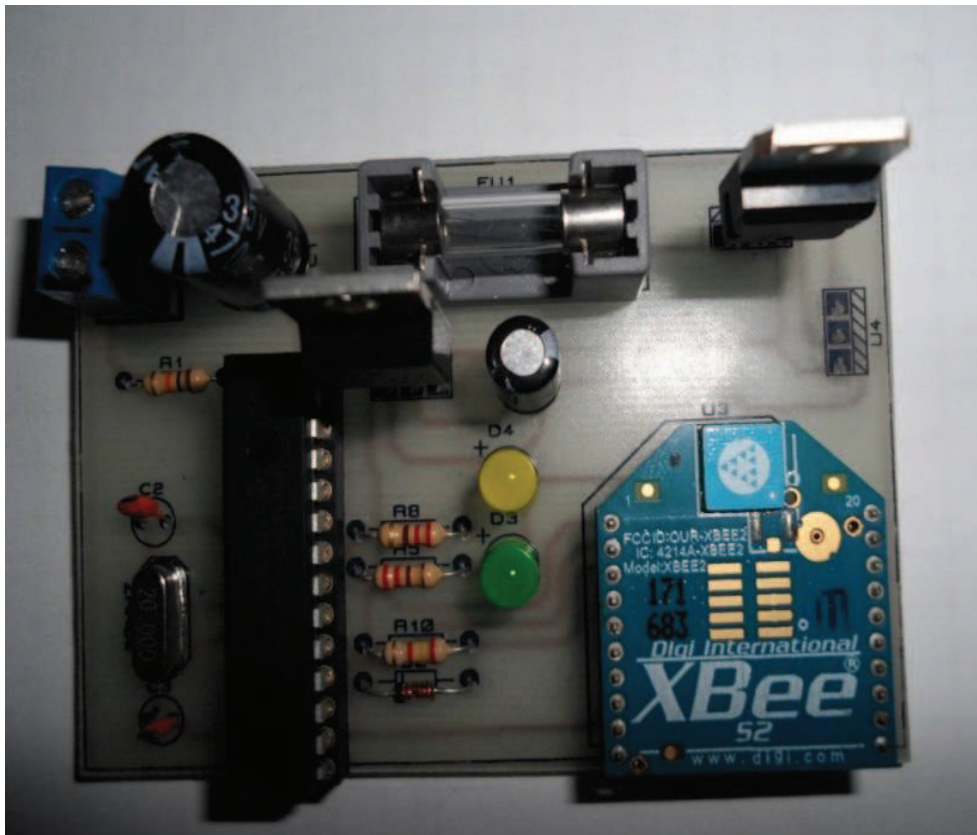


Figura 2. 25 Circuito Transmisor en las paradas



**Figura 2. 26 Ensamblaje del bus prototipo**



**Figura 2. 27 Ensamblaje del Bus prototipo y el sistema de control de bus**

## 2.5 CONSTRUCCIÓN DE UN CIRCUITO DE INTERFAZ SERIAL PARA PROGRAMACIÓN DE MÓDULOS DE RF

Para el desarrollo del sistema de transmisión del sistema de control de puertas del prototipo del bus, lo realizamos a través de los módulos de radiofrecuencias Xbee, como se ha indicado en capítulos anteriores los módulos Xbee poseen una memoria que permite la programación de las características de transmisión.

Para ingresar a estas características es necesario establecer una comunicación serial entre un ordenador y el módulo de transmisión, para lo cual el fabricante DIGI ha proporcionado una placa para establecer la comunicación serial mediante un puerto USB<sup>45</sup> o RS232<sup>46</sup>.



**Figura 2. 28 Placa de programación de módulo Xbee del fabricante**

Para tener un mejor entendimiento de la comunicación serial como el funcionamiento del módulo se procedió con la construcción de una placa de programación con el uso del circuito integrado MAX232.

---

<sup>45</sup> USB: Bus Universal Serial, es una interfaz que provee un estándar de bus serie para conectar dispositivos.

<sup>46</sup> RS232: Es una interfaz que designa una norma para el intercambio de una serie de datos binarios entre el equipo terminal y el equipo de comunicación.

En la Figura 2. 29 se muestra el esquema electrónico de la placa de programación donde el voltaje de alimentación del circuito es de 5 V que ingresa al MAX232 y al regulador de voltaje LM117T.

Como se ha indicado anteriormente el módulo Xbee trabaja con niveles de voltaje de 3.3 V tanto para la transmisión y recepción de señales como el voltaje de alimentación, es por esta razón el uso de un regulador de voltaje.

El LM117T permitirá regular el voltaje de entrada de 5 V a 3.3 V mediante la conexión del capacitor C1 (100) en la entrada de voltaje y C2 (100) en el voltaje de salida, en este caso la colocación del capacitor C1 es por recomendación del fabricante para mantener constante el voltaje de entrada cuando la fuente de voltaje se encuentra alejada de la entrada del regulador, mientras que el capacitor C2 es fundamental en la salida del regulador para mantener la estabilidad del regulador desde que recibe la alimentación de la fuente. En el caso del regulador LM117T se deberá utilizar capacitores con una mínima capacitancia de  $10\mu\text{F}$  y el ESR (Equivalent Series Resistance) debe oscilar entre  $0.3\Omega$  y  $22\Omega$ .

MAX232 es un controlador/receptor dual (Half y Full Duplex) que incluye un generador de tensión capacitiva formado por los capacitores C3,C4,C5,C6 para suministrar los niveles de tensión de una fuente en este caso 3.3 V.

Cuando la computadora envía información a través del cable RS232, como se ha indicado serán cambiados a nivel de 3.3 V para poder leer y escribir en los módulos de radiofrecuencia.

Se visualizará en el led rojo cuando está energizada la placa de grabación, el led verde y amarillo cuando se tiene comunicación al módulo de radiofrecuencia y este se encuentra en estado ON (activo) y no en sleep (stand-by).

Finalmente cuando dos módulos se encuentren polarizados y previamente configurados se visualizará que el led azul se prende indicando que se ha establecido comunicación entre dos módulos y se está enviando y recibiendo información.

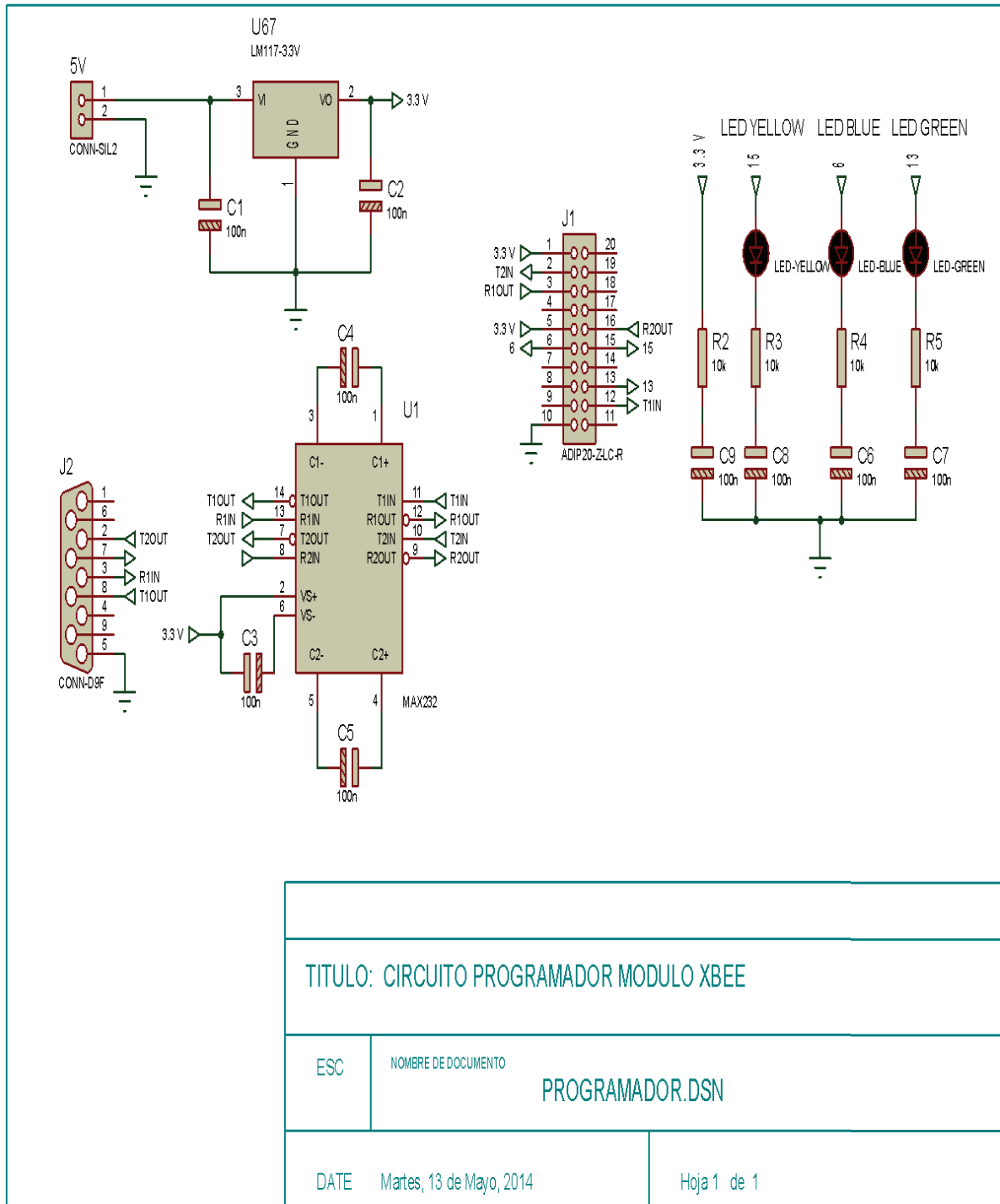


Figura 2. 29 Circuito del programador Xbee

En las Figuras 2.30, 2.31 y 2.32, se muestran el diagrama del circuito realizado en ares, foto y circuito impreso del programador del módulo XBEE.

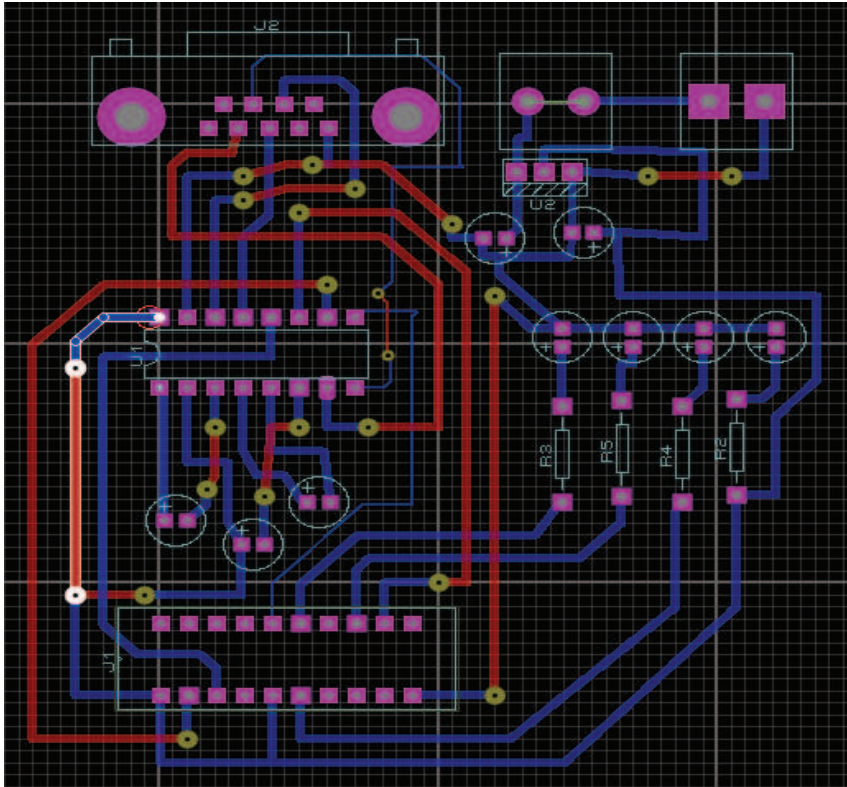


Figura 2. 30 Diagrama del circuito realizado en ares del programador del módulo Xbee

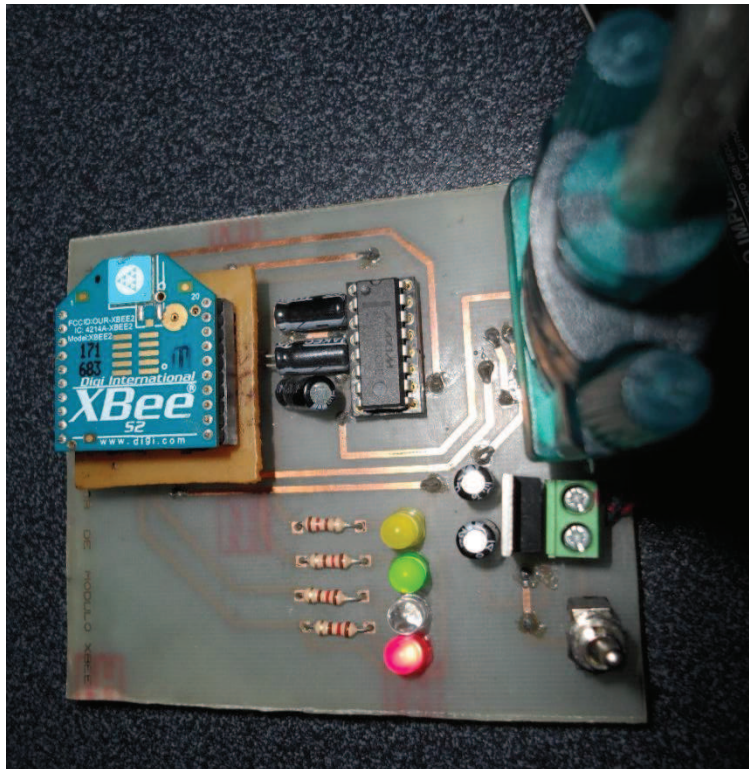


Figura 2. 31 Circuito programador del módulo Xbee

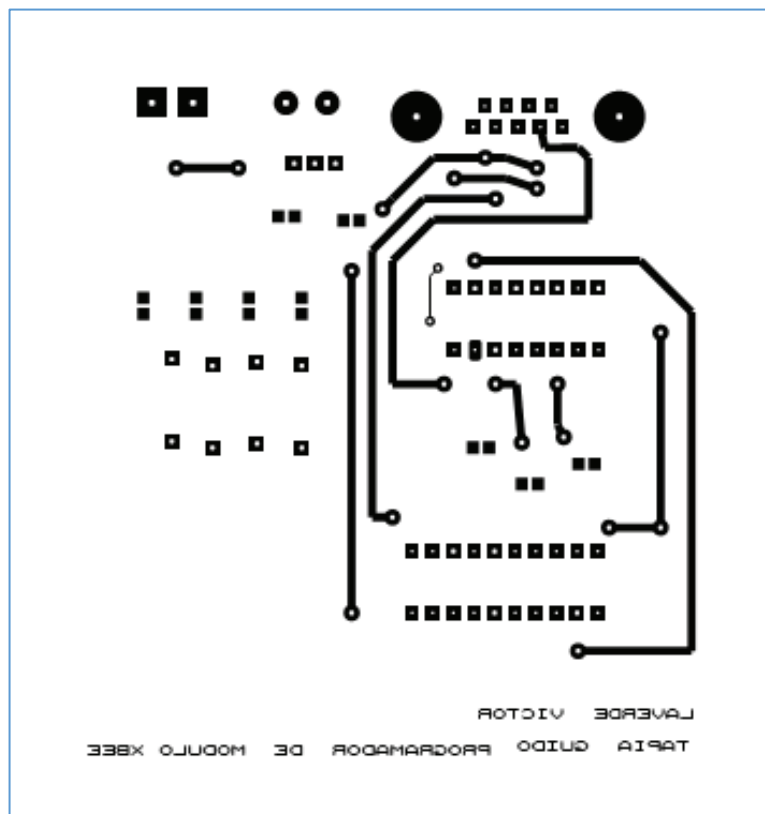
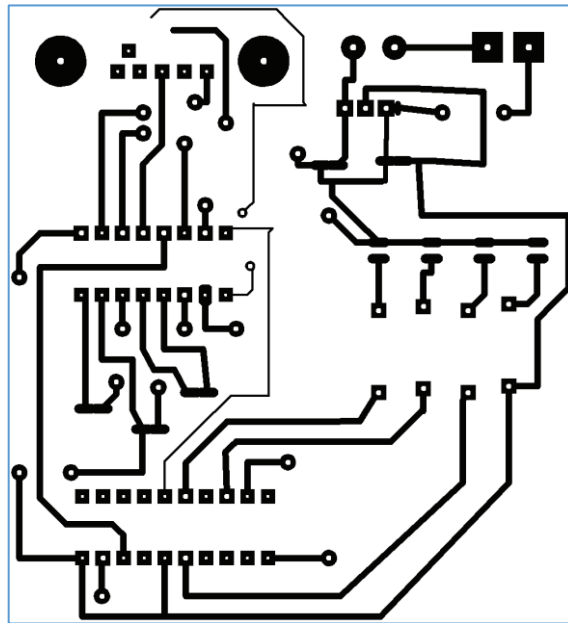


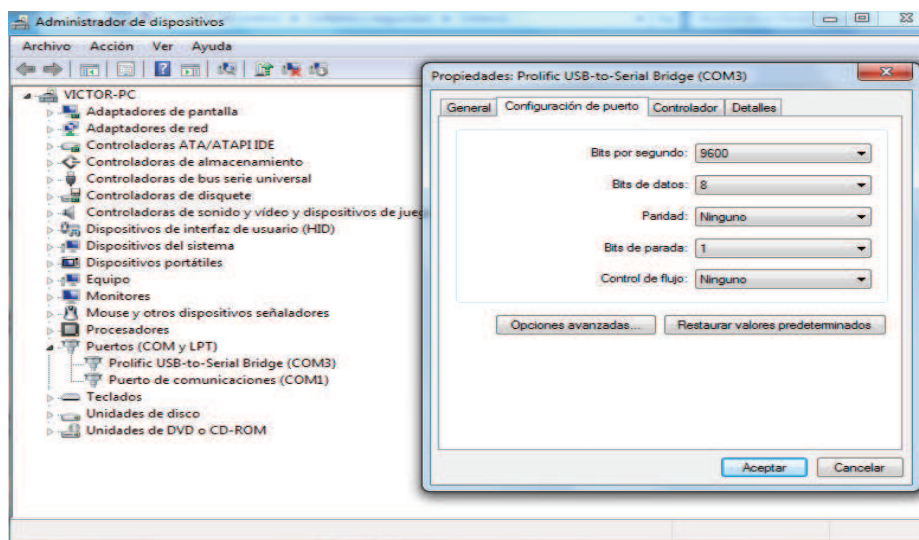
Figura 2. 32 Circuito impreso del programador del módulo Xbee

## CAPÍTULO 3

### SOFTWARE

#### 3.1 PROGRAMACIÓN DE MÓDULOS DE RF XBEE S2 COMO EMISOR Y RECEPTOR MEDIANTE EL PROGRAMA X-CTU

Para realizar la programación de los módulos de RF, el fabricante DIGI ha proporcionado un programa con interfaz gráfico que permitirá la programación de las características de transmisión/recepción, consumo de energía e identificación, dicho programa se llama X-CTU<sup>47</sup>. Para iniciar la configuración se debe realizar otra configuración previa tanto en el programa X-CTU como en el sistema operativo de Windows 7. En el caso del sistema operativo se deberá ingresar al administrador de dispositivos (Inicio-panel de control-administrador de dispositivos) y en la opción de puertos COM como se indica en la Figura 3. 1 se deberá ubicar el cable RS232 conectado e ingresar a sus propiedades, una vez ingresado se deberá validar que la comunicación se encuentre en 9600 baudios, el número de bits de datos en 8, el tipo de paridad en ninguno, el bit de parada 1 y el control de flujo ninguno.

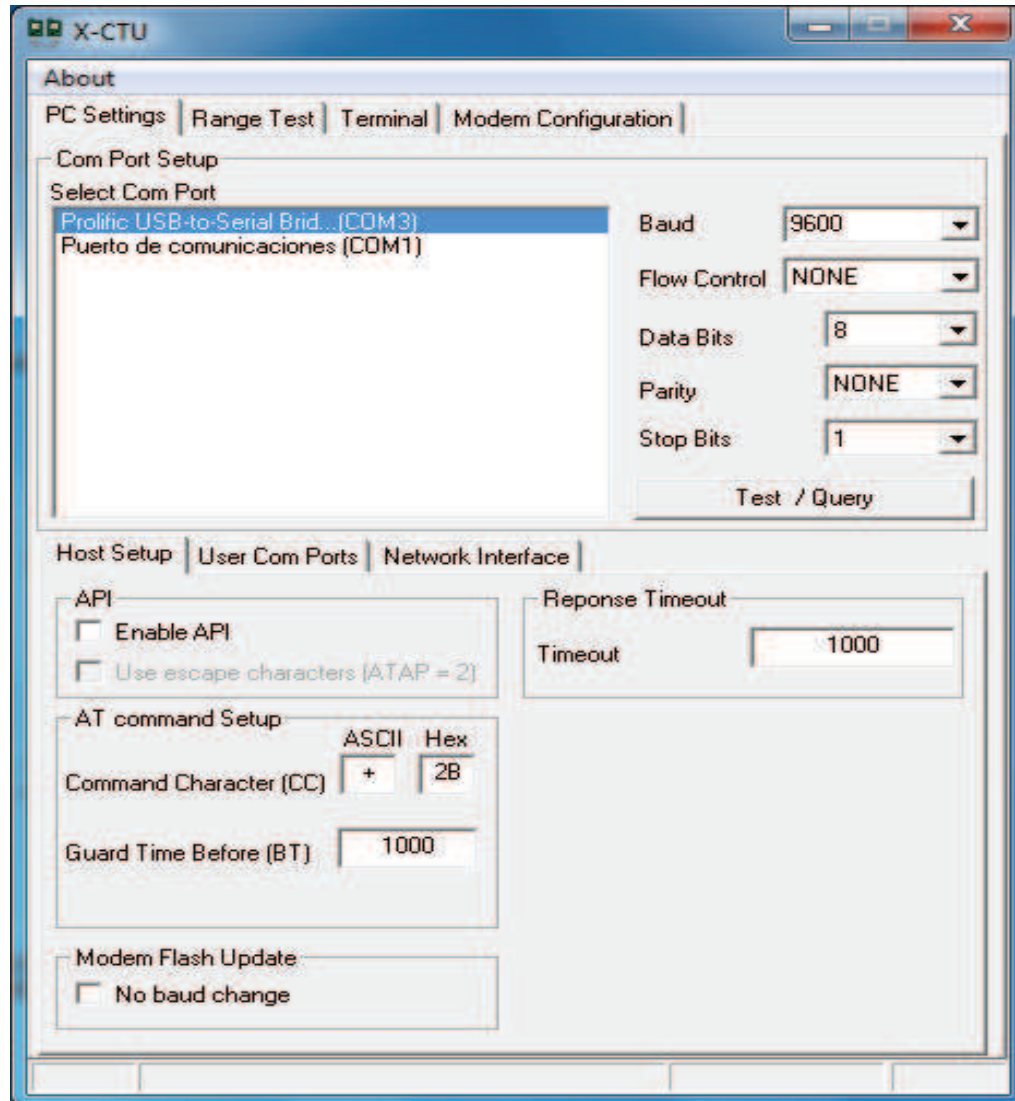


<sup>47</sup> XCTU, 2014



**Figura 3. 1 Configuración de puertos COM**

En el caso del programa X-CTU una vez iniciado se deberá seleccionar el puerto COM 3 donde se encuentra conectado el cable serial-USB a la placa de programación; y validar que se encuentre los parámetros indicados anteriormente en la opción del conector COM como se indica en la Figura 3. 2.



**Figura 3. 2 Configuración estándar del programa X-CTU**

Se deberá conectar el módulo Xbee a la placa de programación y este al cable serial RS232 que se encuentra conectado al computador, una vez obtenido esta conexión se deberá presionar el botón de *TEST* para validar que se tiene una comunicación exitosa entre el módulo Xbee y el computador como se indica en la Figura 3. 3, posteriormente se visualiza la confirmación de comunicación y el

número único de la serial de cada módulo, como también la versión del firmware que tiene el módulo.

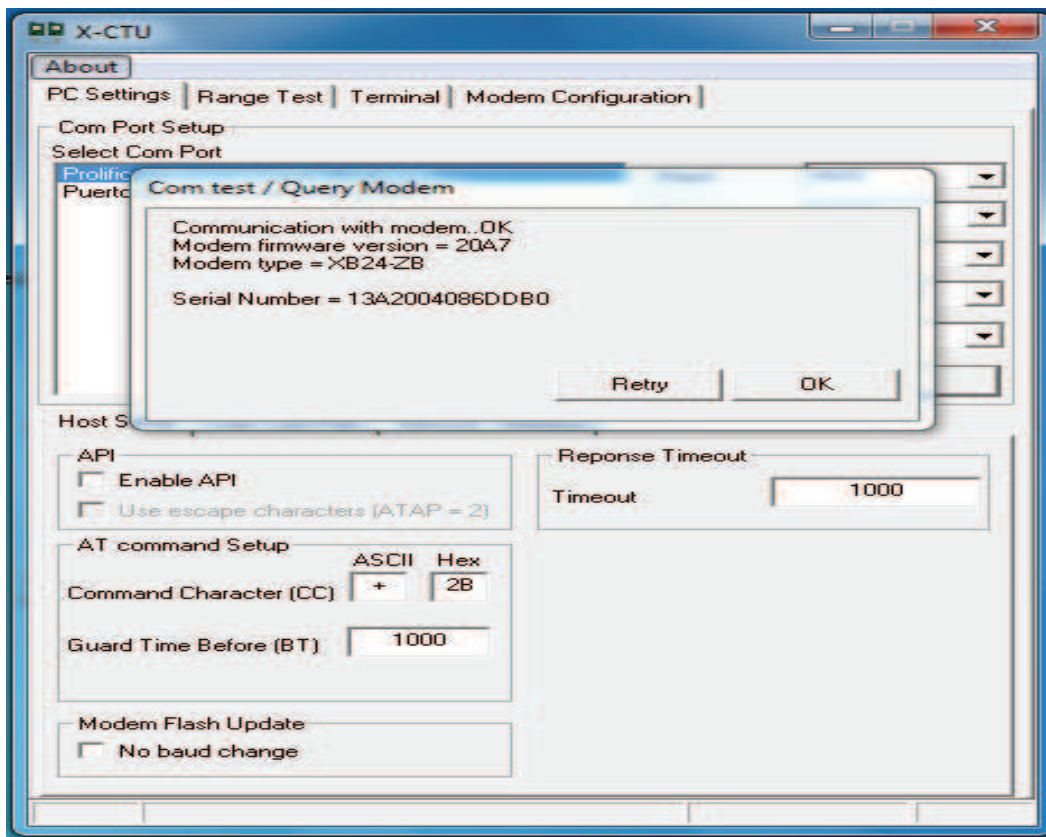


Figura 3. 3 Validación de comunicación del módulo y el ordenador

Según lo indicado en capítulos anteriores se deberá establecer una red PAN por lo que se procederá con la configuración de un módulo como coordinador y otro como router en modo Punto-Multipunto con el propósito de que el módulo coordinador pueda detectar los N módulos en las paradas.

### 3.1.1 CONFIGURACIÓN MODO COORDINADOR

El modo coordinador tiene la única función de formar una red. Es el responsable de establecer el canal de comunicaciones y del PAN ID para toda la red. Una vez establecidos estos parámetros, el coordinador puede formar una red, permitiendo unirse a él a dispositivos Routers y End Points.

Para proceder con las configuraciones básicas del módulo coordinador hay que ingresar en la pestaña *Modem Configuration*<sup>48</sup> donde se tendrá los parámetros para leer, escribir y actualizar el firmware del módulo como se indica en la Figura 3. 4.

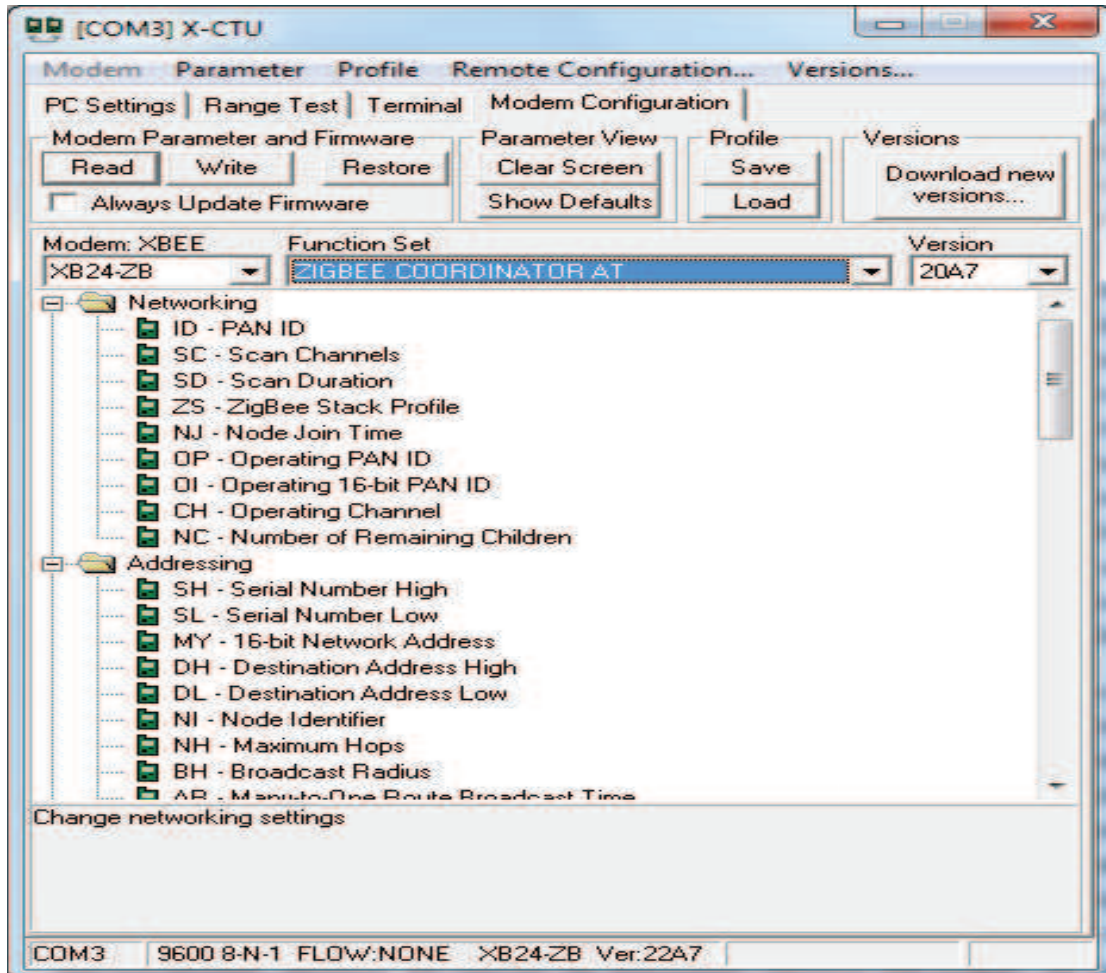


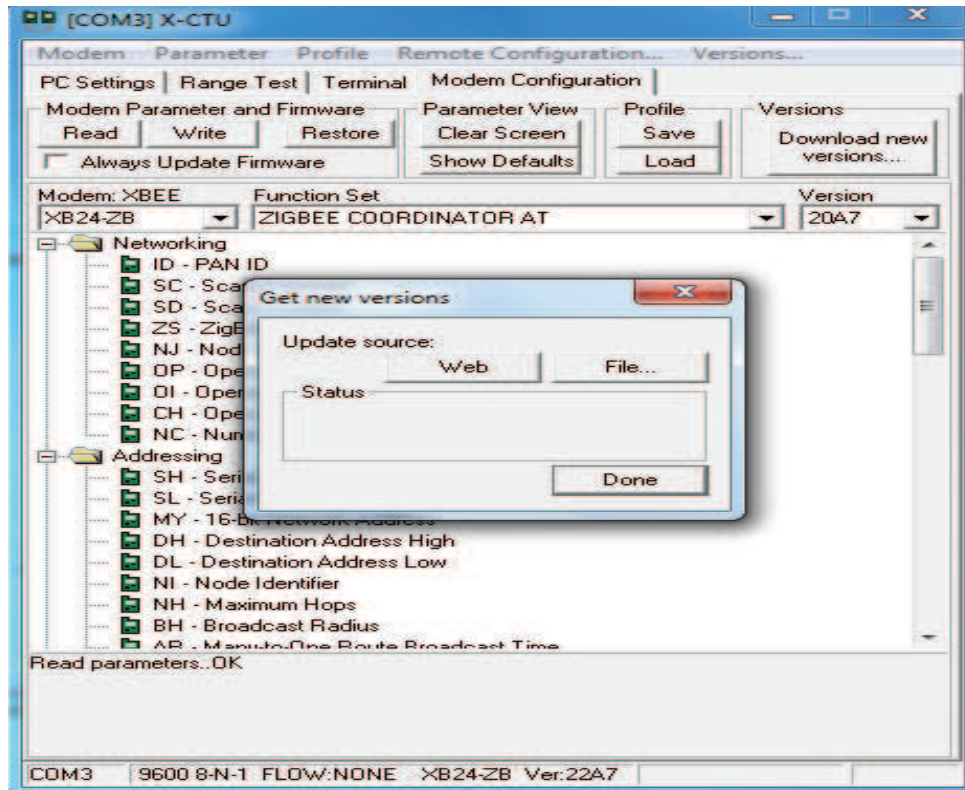
Figura 3. 4 Firmware del módulo coordinador

Se deberá seleccionar la opción *Read* para leer las configuraciones que viene de fábrica, después se deberá seleccionar la opción *Function Set*<sup>49</sup> donde se visualizará los diferentes tipos de funcionamientos del módulo (coordinador, *router*, *end-point*) y según el modo de comando a usar (AT, API) se seleccionará Coordinador AT, después se deberá validar que la versión de los firmwares sea la más actualizada caso contrario se deberá dar clic en la opción *Download New*

<sup>48</sup>Modem Configuration: Configuración del módulo

<sup>49</sup> FUNCTION SET: Conjunto de funciones

Version<sup>50</sup> y se descargarán los firmwares actualizados de la versión 20A7 como se indica en la Figura 3. 5.



**Figura 3. 5 Actualización de firmware**

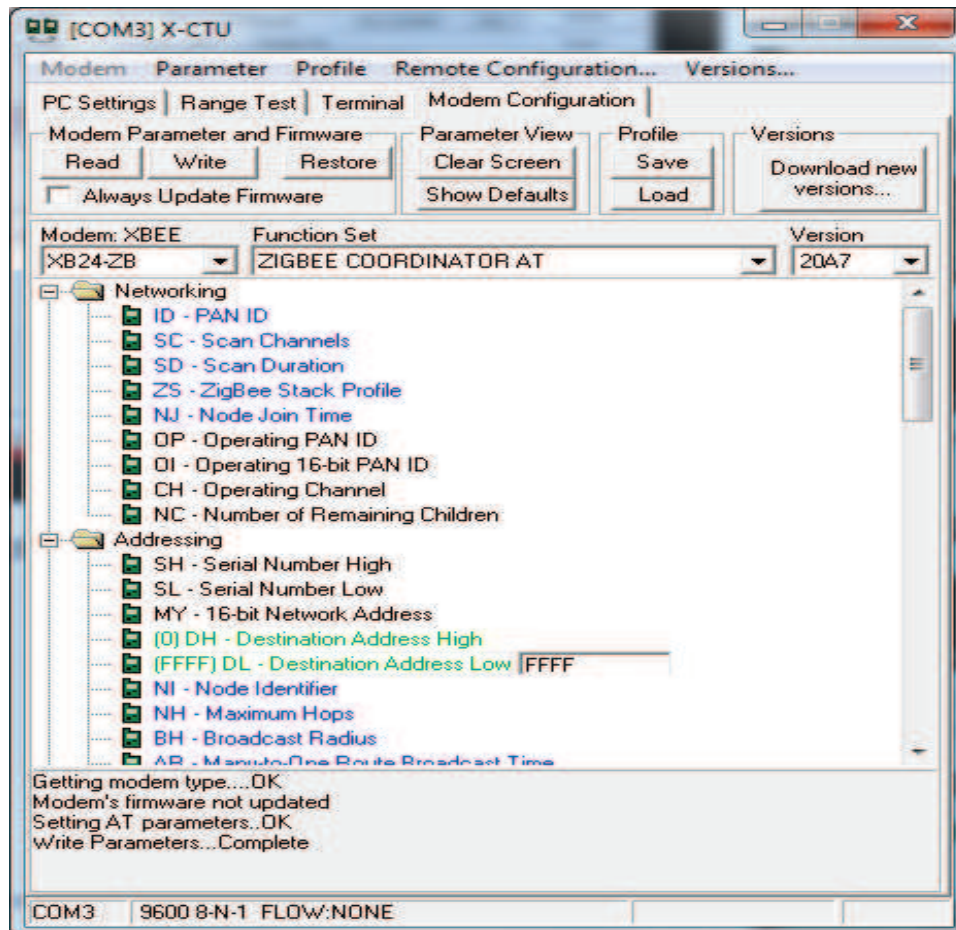
Una vez realizado los pasos anteriormente descritos se procederá a dar clic en la opción *Write* para proceder a cambiar el firmware del módulo.

Según los parámetros a configurar se tiene que la primera opción importante es el identificador de red (ATID) que previamente ha sido seleccionado y es 2520, cabe recalcar que el fabricante permite seleccionar 65000 direcciones posibles para el identificador de la red.

El siguiente parámetro a configurarse son las direcciones que tomarán cada módulo para poder identificarse en la red PAN con un diagrama de punto-multipunto, para ello seleccionaremos las opciones de DH y DL y se ingresará los valores hexadecimales 0 y FFFF respectivamente, lo que permitirá tener una dirección MY 0 y que los demás módulos routers puedan detectar al módulo coordinador.

<sup>50</sup> DOWNLOAD NEW VERSION: Descargar nueva versión

Finalmente para guardar los cambios del módulo coordinador se va a la opción de *write* y se grabarán los cambios en el firmware, obteniendo una confirmación de lo realizado en la parte inferior con un OK si es exitoso o FAILE si es fallido.



**Figura 3. 6 Grabación de firmware del módulo coordinador**

Adicionalmente se anexa la Tabla 3. 1 donde se visualiza las configuraciones adicionales que se han realizado para tener un mejor resultado en el ambiente en el que se realizó la implementación.

CONFIGURACIONES ADICIONALES DEL MÓDULO XBEE		
COMANDO AT	VALOR CONFIGURADO	OBSERVACIÓN
SC SCAN CHANNEL	FFFF	Valor de fábrica no se requiere cambios
SD SCAN DURATION	3	Valor de fábrica no se requiere cambios
NI NODE IDENTIFIER	COORDINADOR BUS	Establece el nombre del modulo

PL POWER LEVEL	0 LOWEST	Configura la potencia de TX a la más baja
PM POWER MODE	0 DISABLED	valor de fábrica no se requiere aumentar sensibilidad de antena a 2 db

**Tabla 3. 1 Configuraciones adicionales del módulo Xbee**

### 3.1.2 CONFIGURACIÓN MODO ROUTER

El módulo configurado como modo Router, crea y mantiene información sobre la red para determinar la mejor ruta para enviar un paquete de información. Lógicamente un router debe unirse a una red Zigbee antes de poder actuar como router retransmitiendo paquetes de otros Routers o de End points.

La configuración del módulo Xbee como router se lo realizará cambiando su firmware de fábrica por el firmware de Router AT en *Function Set*, cabe aclarar que ya no será necesaria la actualización del firmware debido a que ya se realizó en la configuración del anterior módulo. Para que el módulo pueda enviar información al coordinador tiene que estar dentro de la misma red por lo que se tiene que configurar el ATID con el valor de 2520. Las direcciones de ATDL y ATDH que se deben configurar es 0 para que el módulo pueda enviar a los módulos configurados en los buses.

Finalmente se dará clic en la opción de *write* y se tendrá la configuración del módulo en modo router pero se tendrá que validar la asignación de una dirección MY mediante el comando ATND que se podrá ingresar en la pestaña terminal del programa X-CTU.

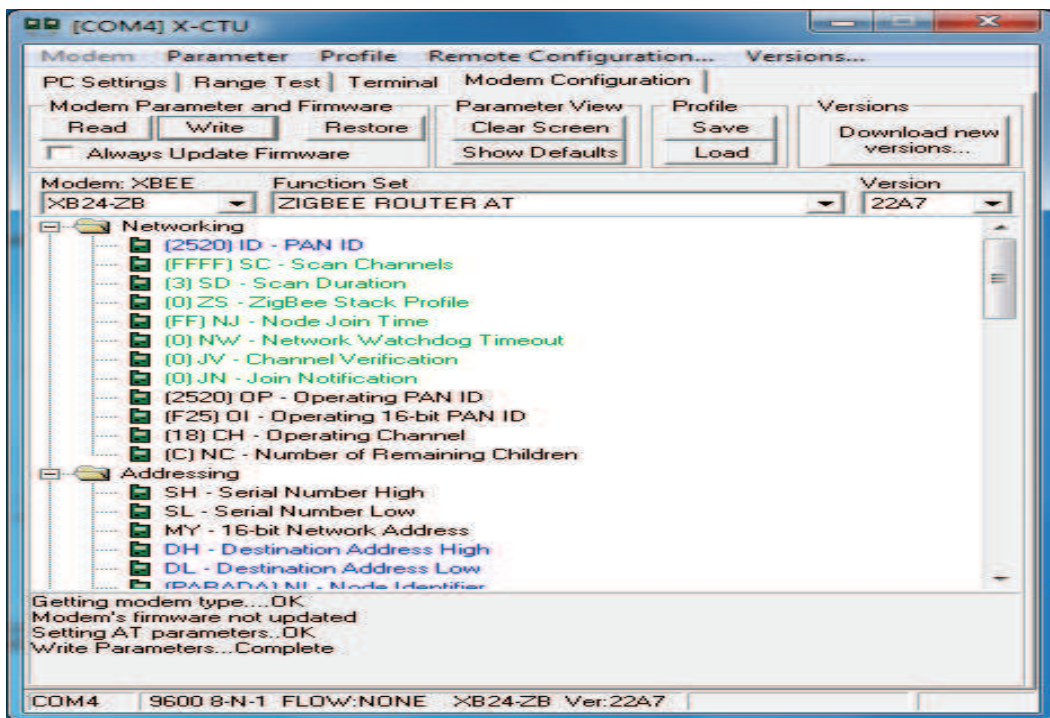


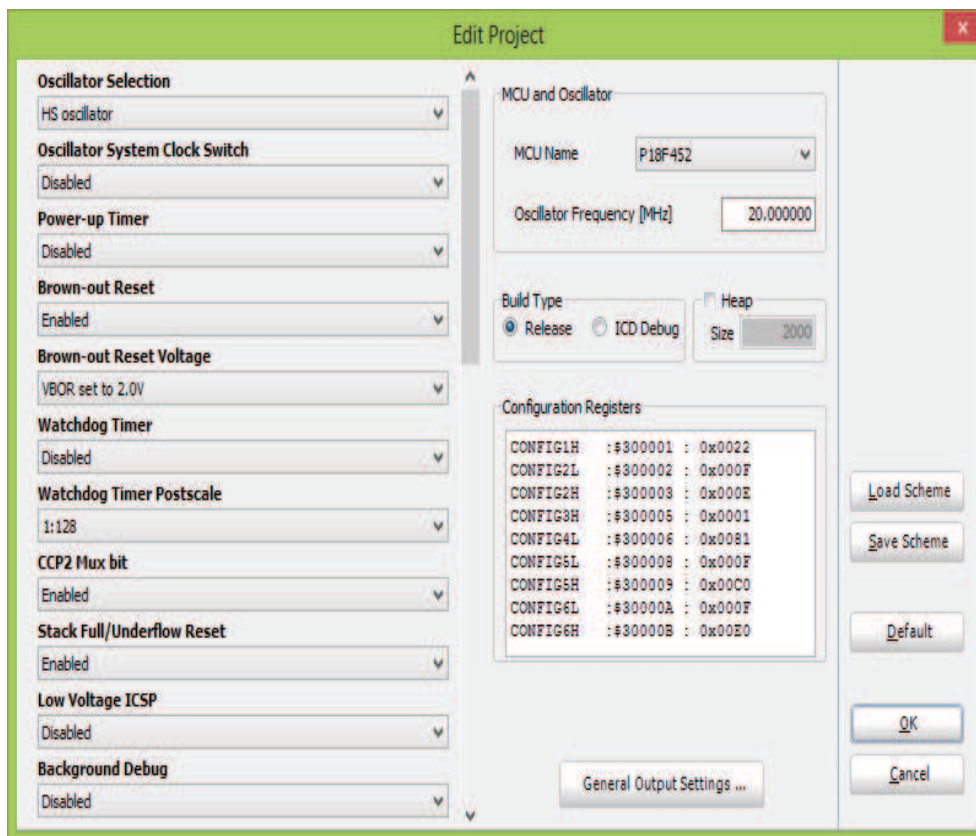
Figura 3. 7 Configuración modo router

## 3.2 PROGRAMACIÓN DEL MICRO-CONTROLADOR DE ACUERDO AL SISTEMA DE CONSTRUCCIÓN

### 3.2.1 PROGRAMACIÓN DEL PI18F452 PARA EL BUS

La programación del micro-controlador PI18F452 se lo ha realizado con el programa Mikro-C Basic. El motivo de que hayamos usado este programa es debido que es el usuario puede familiarizarse rápidamente con el tipo de lengua de programación que utiliza debido que es semejante al lenguaje en C, y dispone de diferentes librerías las mismas que cumplen con las funciones que necesitamos configurar, adicionalmente el fabricante pone a disposición de varios ejemplos de programación los mismos que son de gran utilidad en el presente proyecto.

En la Figura 3. 8 se muestra la ventana del programa Mikro-C donde se realizarán las configuraciones básicas como es el cristal oscilador a usarse posterior a la creación del proyecto.



**Figura 3. 8 Configuraciones básicas del proyecto en Mikro-C**

Para la realización del programa a usarse se debió realizar previamente un diagrama que nos dé una visión de cómo crear el programa que controle la apertura de las puertas únicamente en sus paradas, por tal motivo, en primera instancia se visualizará los diagramas de flujo realizados para validar el funcionamiento del programa.

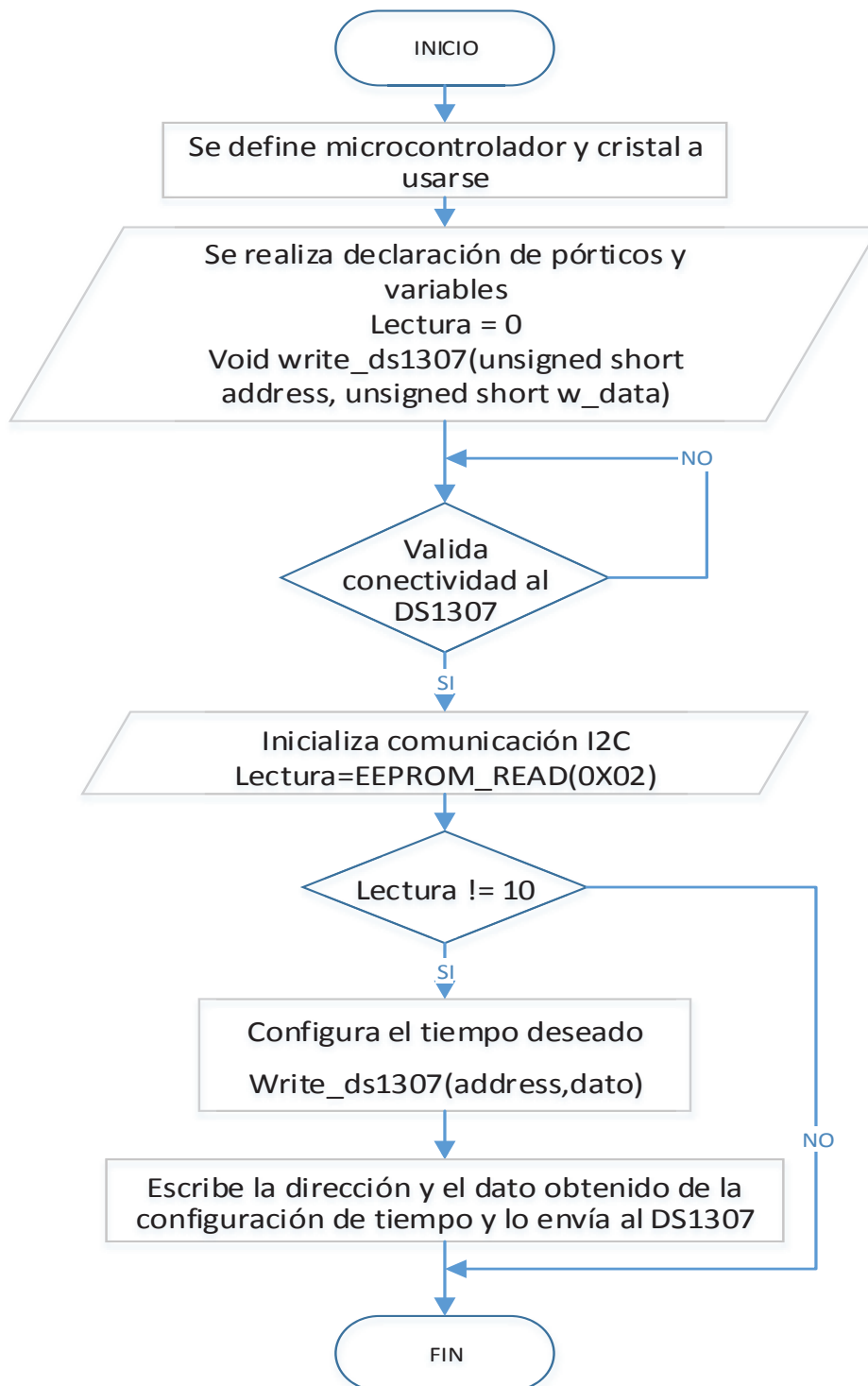
En la Figura 3. 9, se verifica el proceso de configuración del reloj DS1307 para poder grabar la hora y fecha deseadas, también se puede visualizar o capturar por evento, así como también se podrá visualizar a partir de un dato almacenado en la memoria EEPROM,

A continuación se detalla una explicación del diagrama:

1. Se realiza la apertura de programa indicando el tipo de micro-controlador y cristal a usarse en la creación del proyecto.



2. Se declaran las variables a usarse tanto para la lectura como para la escritura de datos en el reloj DS1307, también se declara la variable lectura que identificará si se desea configurar la hora y fecha en el DS1307
3. Se válida la conectividad al DS1307, si este no es exitoso entrará en un flujo hasta validar conectividad y en el caso de ser exitoso se continuará con los demás parámetros, cabe recalcar que esta parte del programa se encontrará después de la declaración de variables.
4. Se inicializa la comunicación I2C que es el protocolo de comunicación para el DS1307 y se indica que el valor de lectura es igual al valor almacenado en la dirección 0x02 de la memoria EEPROM.
5. Se valida si el valor de lectura es cualquier valor diferente de 10 lo que permitirá grabar los datos previamente ingresados en el micro-controlador que corresponderían a la fecha y hora en el DS1307, caso contrario si es valor de lectura es 10, el programa continuará con las instrucciones en la programación como es la visualización de la fecha y hora o hasta que llegue al final del programa.



**Figura 3. 9 Programaci<sup>o</sup>n del reloj de tiempo real DS1307**

En la Figura 3. 10 se indicar<sup>á</sup> la programaci<sup>o</sup>n realizado para el sistema de control de puertas, mediante la utilizaci<sup>o</sup>n de la librer<sup>í</sup>a ya anteriormente indicada I2C y tambi<sup>é</sup>n las librer<sup>í</sup>as USART (UART1), LCD\_INIT.

Hay que tomar en cuenta que el diagrama de flujo que se presenta a continuación es una extensión del diagrama anterior.

A continuación se detalla una explicación del diagrama:

1. Se declaran las variables a usar durante todo el programa.
2. Se inicializa las librerías a usar comunicación I2C, LCD y comunicación serial USART
3. Se inicia el cuerpo del programa
4. Mediante la librería de USART se escucha el canal de RF, en el caso que no se reciba ningún dato se captura la fecha y hora y lo almacena en las variables time [] y date [] para posteriormente ser visualizado en la pantalla LCD. Para el caso que se reciba algún dato del proceso de escuchar el canal de RF se almacenará la información recibida.
5. El siguiente proceso es de validar si la información recibida es correcta, en el caso de que no sea un dato válido, nuevamente se regresará al proceso de escuchar al canal de RF, pero en el caso de que la información recibida es válida se continuará con el proceso para abrir puertas.
6. Una vez que sea válida la información recibida, se captura la hora en que ocurre el evento y se almacena en una variable. Inmediatamente se envía un 1 lógico a través de los pines D para abrir las puertas del bus, después se visualizará el evento de apertura en la pantalla LCD.
7. El siguiente proceso es de validar si se encuentra en 1 la bandera, cerrar si no es así y el programa entrará en un flujo hasta que se deje de recibir información, en el caso de que la bandera se encuentre en 1 significa que el bus se ha empezado a movilizar y el programa continuará con su proceso.
8. Una vez que el bus se retira de la parada, el bus manda un 1 lógico en el pino D lo que permitirá cerrar las puertas del bus y posteriormente capturar la hora en la que ocurre el evento para ser almacenado en una variable.
9. Finalmente se grabará en un archivo TXT la hora capturada en cada evento al momento de abrir y cerrar las puertas mediante la librería USART.

10. Se culmina el programa.

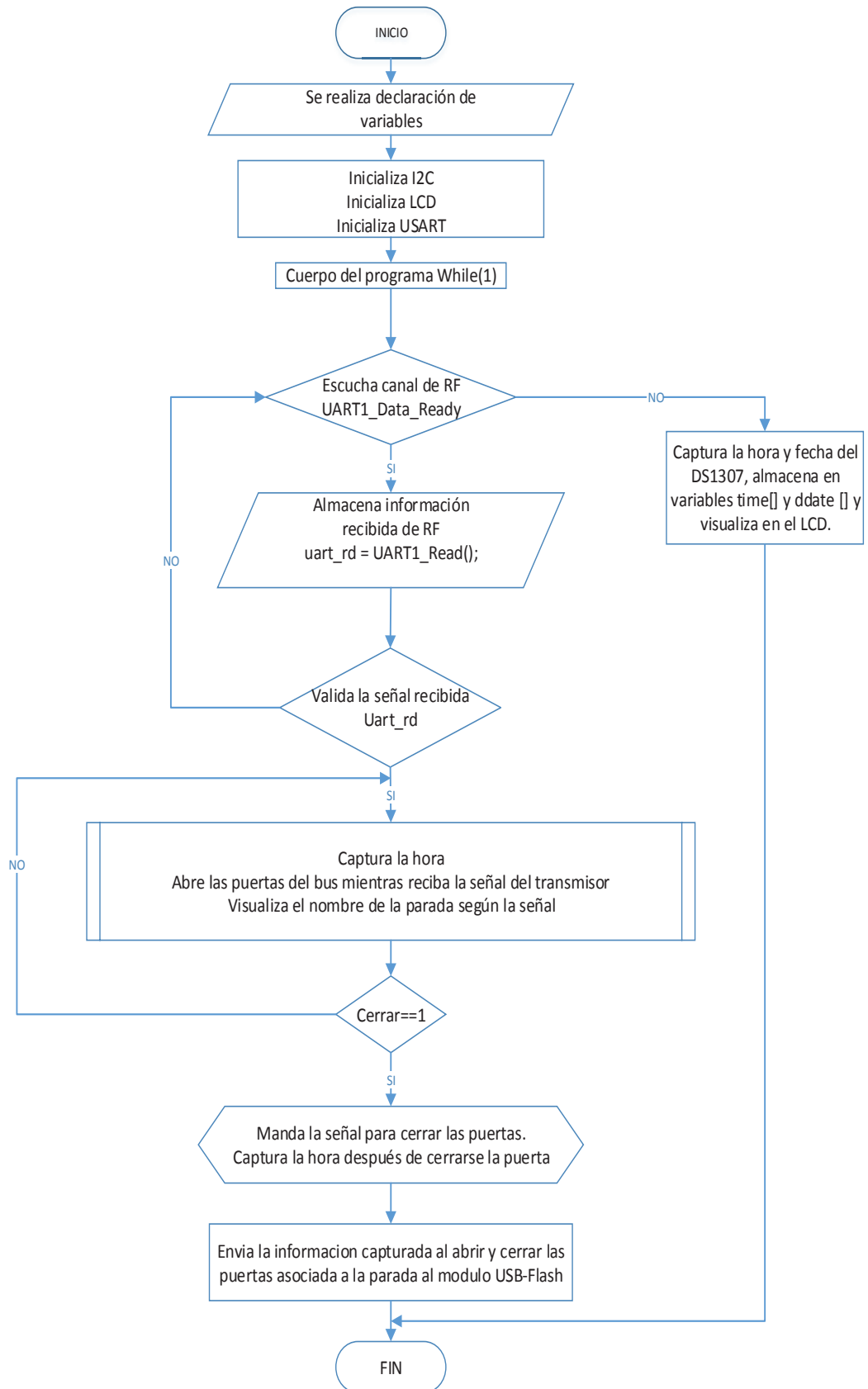


Figura 3. 10 Programación del sistema de control de puertas

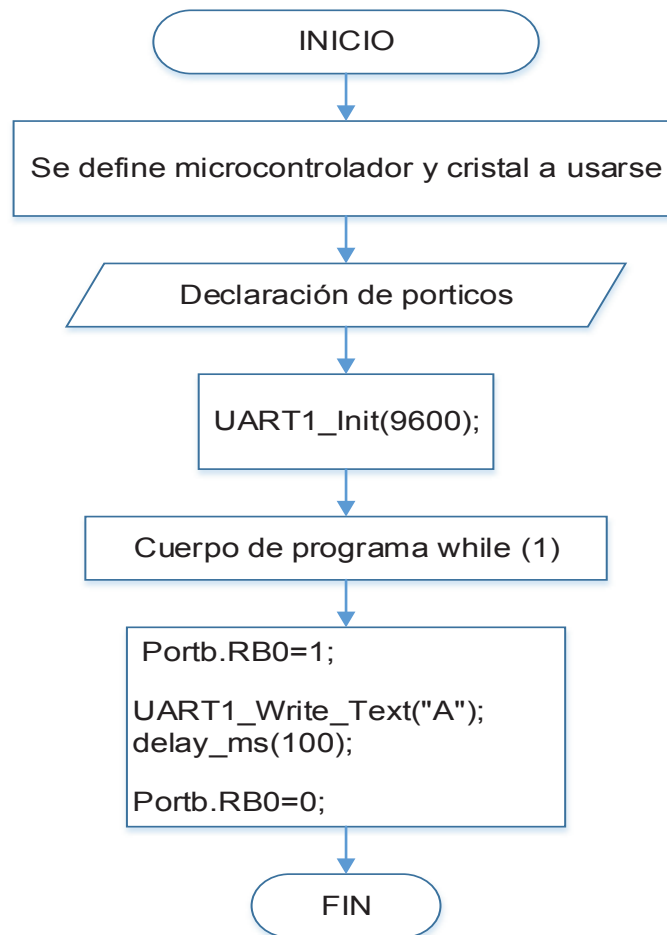
Las partes del programa realizado, las podremos observar en el ANEXO No. 1.

### **3.2.2 PROGRAMACIÓN DEL PI16F870 PARA CADA PARADA**

De igual manera las partes de la programación realizada para el micro-controlador PI16F870 la tenemos en el ANEXO No. 2, ocupando el mismo software Mikro-C Basic.

A continuación se describe el funcionamiento del programa:

1. Se define el micro-controlador y cristal a usarse
2. Se declaran los pódicos de salida.
3. Se inicializa la librería USART
4. Se inicia el cuerpo de programa
5. Se da 1 lógico al port B para iluminar un led, se envía la señal A mediante la comunicación serial. Se da un 0 lógico al port B para apagar un led.
6. Se culmina el programa



**Figura 3. 11 Programación del transmisor de parada**

### **3.3 BASE DE DATOS PARA EL ALMACENAMIENTO DE INFORMACIÓN OBTENIDA DEL BUS**

Para la finalización de este proyecto se ha propuesto crear una base de datos para importar y almacenar los datos recibidos del bus, los mismos que pueden ser manipulados según el administrador de la base de datos como el usuario final que manejará la base de datos.

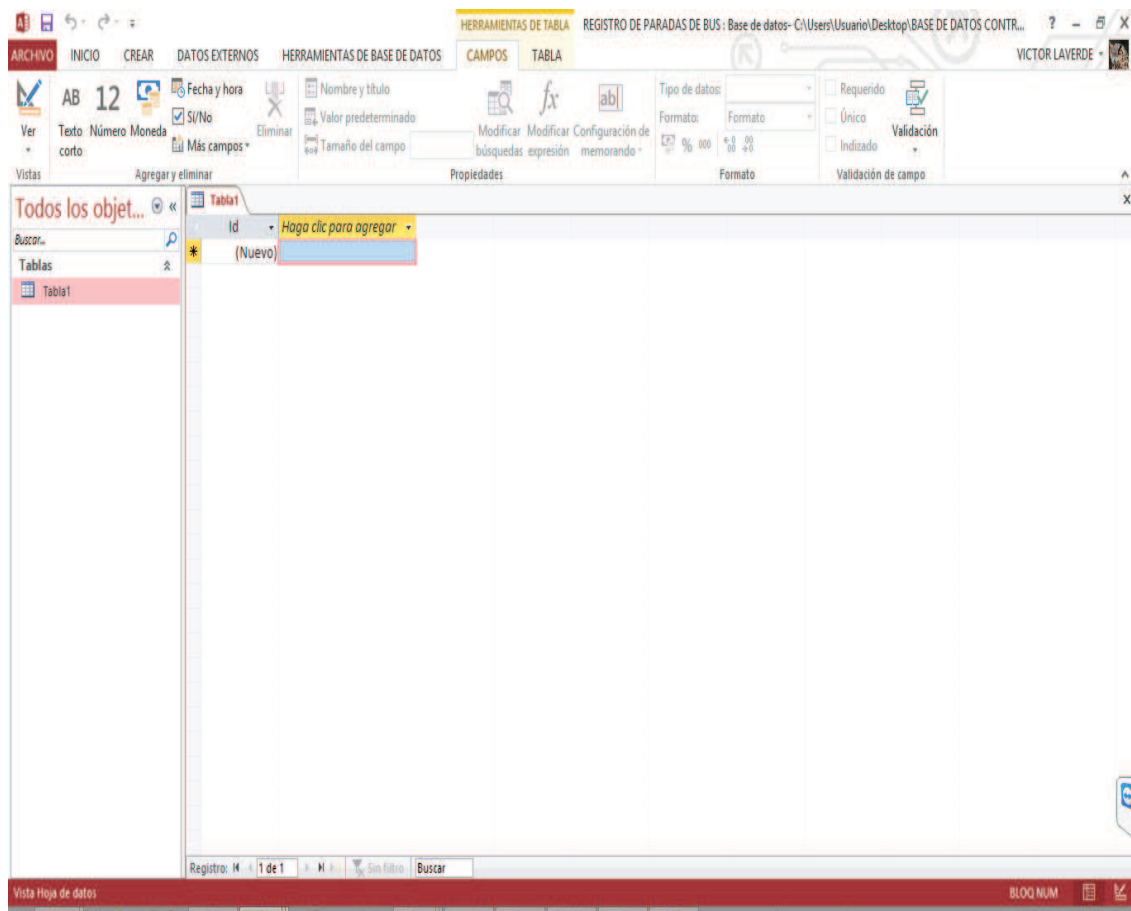
A continuación se explicará una serie de pasos que permitirán crear una base de datos, lo más importante es importar la información proporcionada por el microcontrolador.

En primer lugar es necesario crear un proyecto nuevo para iniciar la base de datos como se indica en la Figura 3. 12, donde se selecciona la ubicación donde se creará el archivo.



**Figura 3. 12 Selección de ubicación para la creación de nuevo proyecto**

En la Figura 3. 13, se podrá visualizar un proyecto ya creado donde se inicia con una tabla vacía y a partir de esto se ingresarán los datos que requerimos almacenar en nuestra tabla para realizar esto damos clic sobre la tabla y damos clic derecho para obtener la función de visualizar edición.



**Figura 3. 13 Inicialización mediante tabla vacía**

Al abrir la edición podremos ingresar los nombres de los campos a tener en la tabla y el tipo de dato que va almacenar en cada campo, debido a que nuestro trabajo consiste en almacenar especialmente la hora y fecha recibida del micro-controlador escogeremos el dato de fecha y lo especificaremos como fecha corta dentro de los parámetros de cada dato.



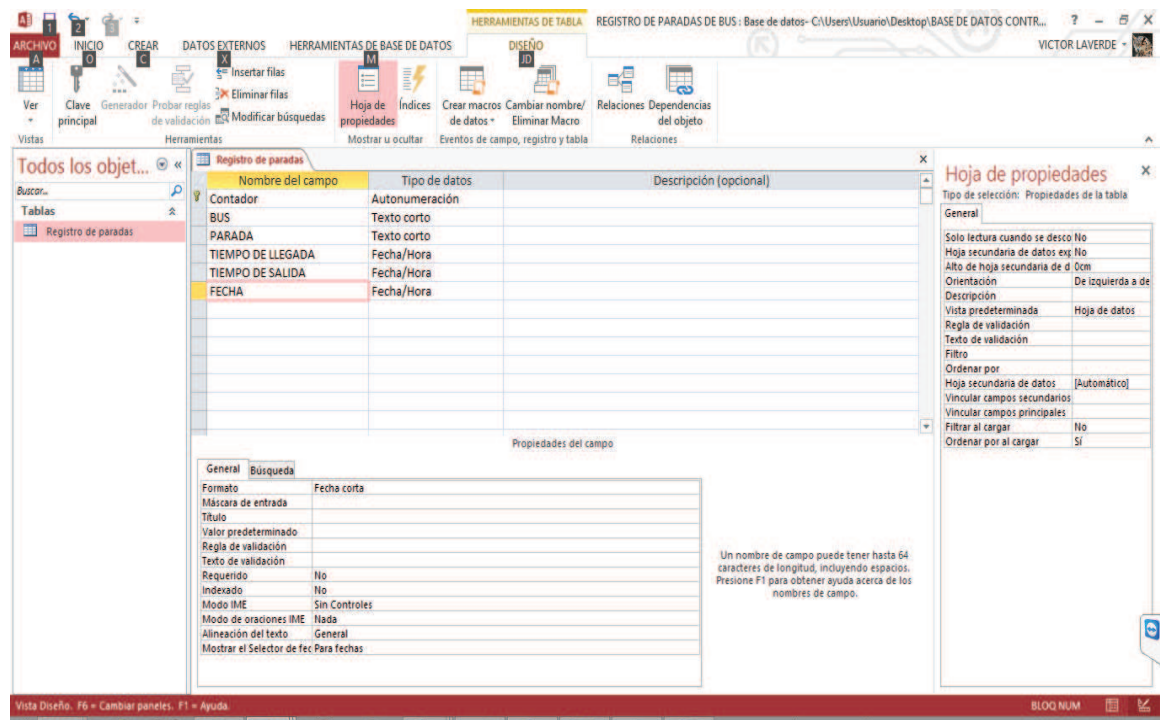


Figura 3. 14 Ingreso de los nombres de los campos

En la Figura 3. 15, se podrá visualizar una tabla ya creada donde se va a tener un campo que enumere cada dato que se cree.

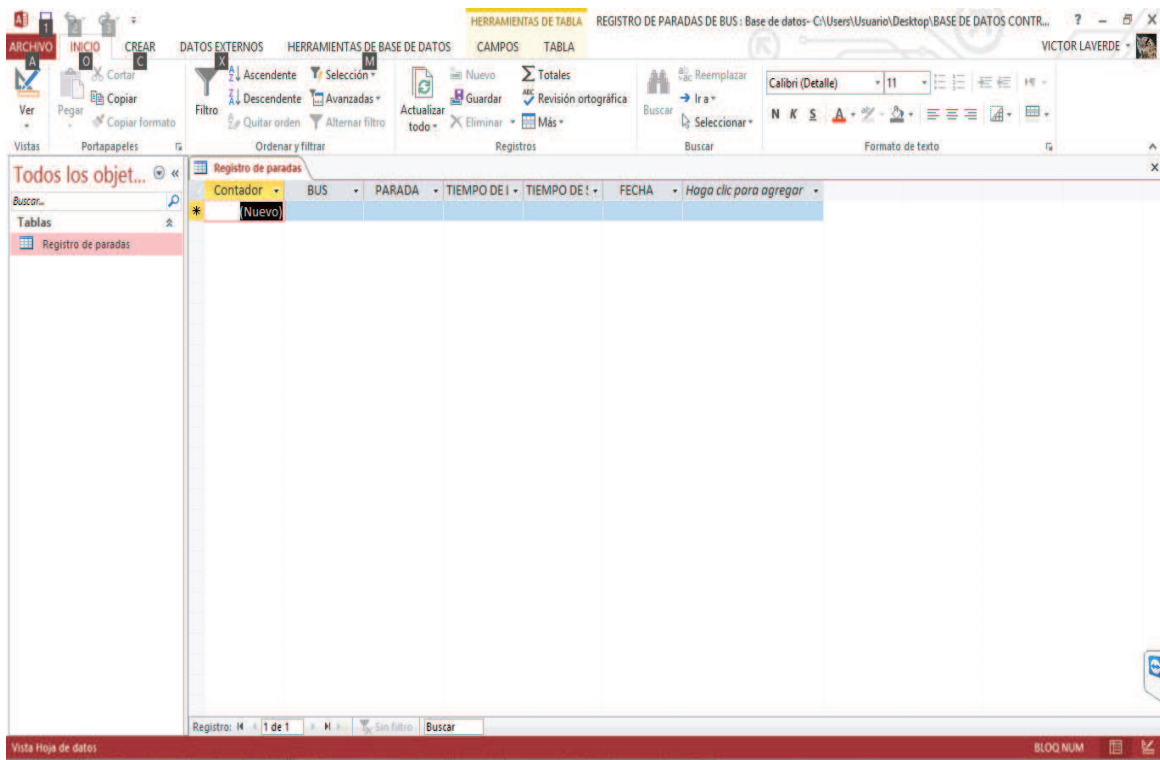


Figura 3. 15 Enumeración para los datos creados

El siguiente paso para la creación de una base de datos es realizar una consulta, para lo cual utilizaremos un asistente de creación de consulta que lo podremos ubicar en la opción de CREAR. En la Figura 3. 16, se podrá visualizar el asistente para crear la consulta, en este caso se seleccionará una consulta sencilla y como se visualiza en la Figura 3. 17 se procede a seleccionar los datos que se desea importar de nuestra tabla.

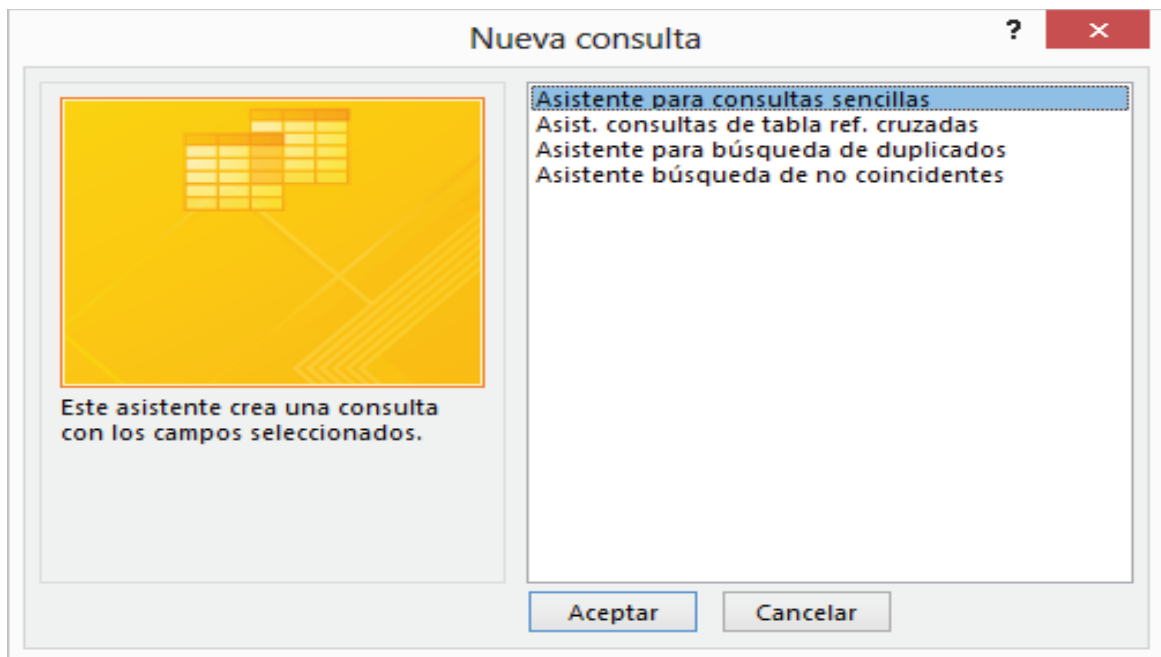


Figura 3. 16 Asistente para crear una consulta

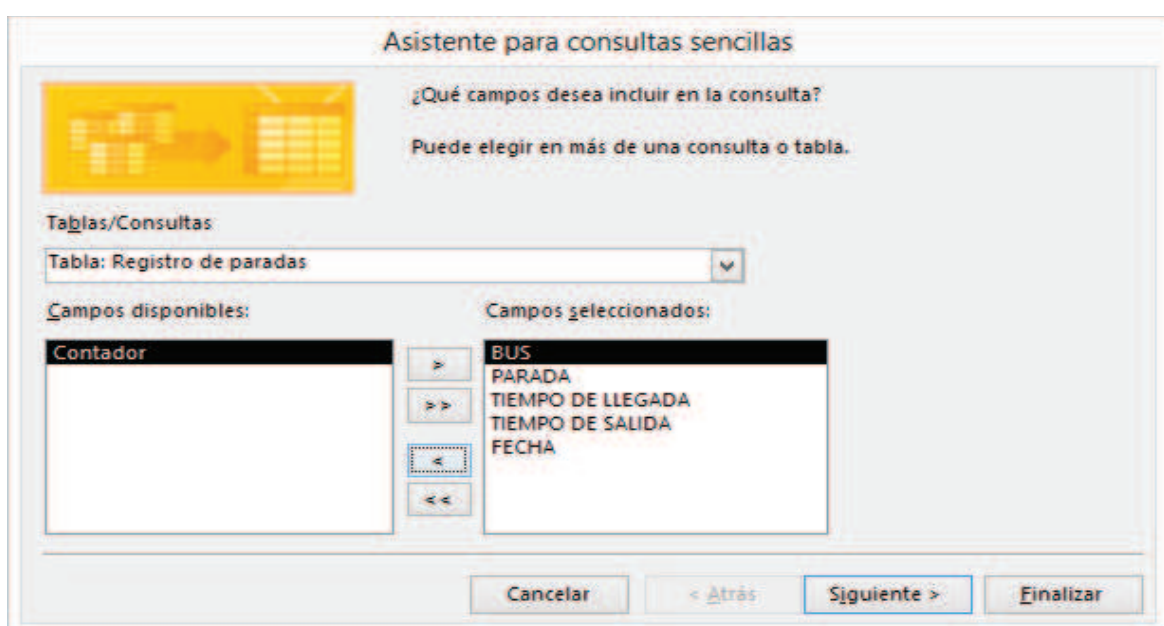
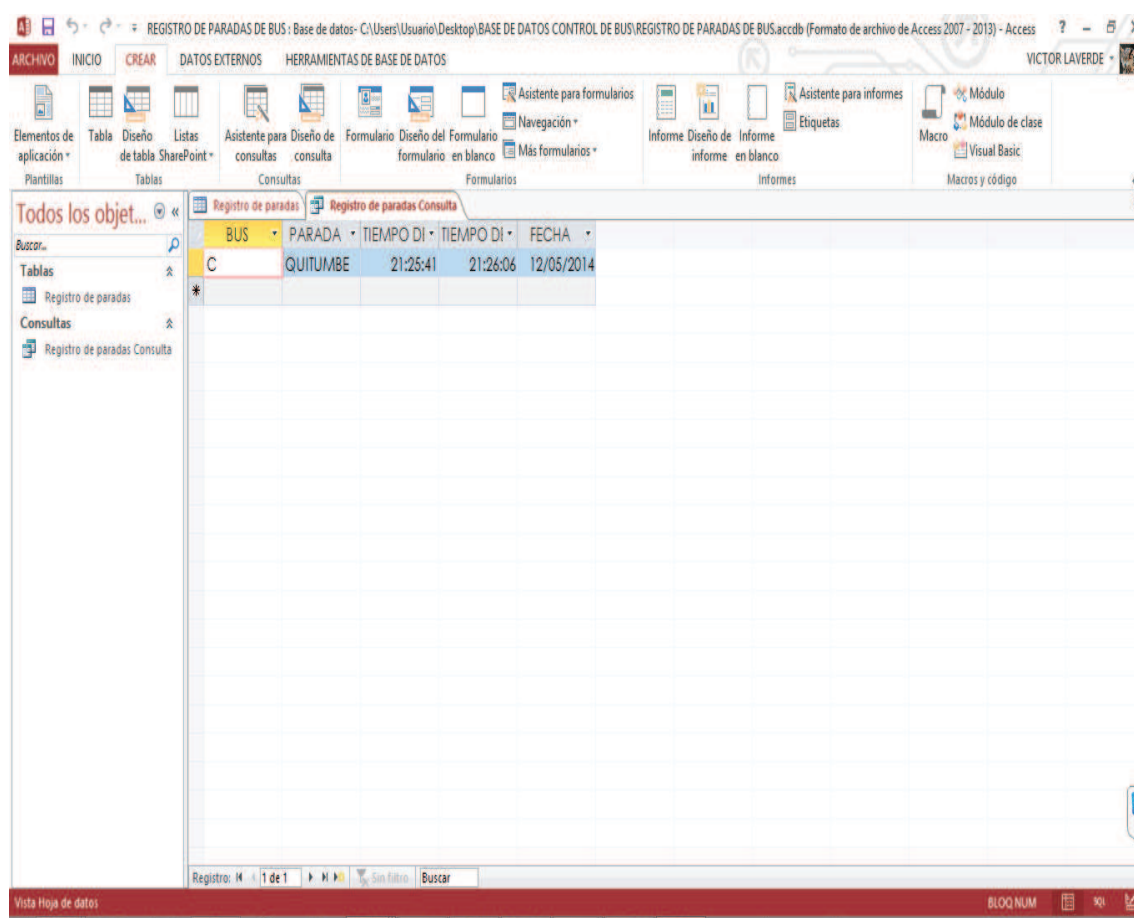


Figura 3. 17 Selección de datos a importar

En la Figura 3. 18, se visualizará la consulta ya finalizada y además se visualizará los datos seleccionados de nuestra tabla original.

El siguiente paso a configurar es la de importar los datos de nuestro archivo donde se almacena los datos del micro-controlador, para realizar esto seleccionaremos la opción de DATOS EXTERNOS y seleccionaremos el asistente para importación de archivo TXT.



**Figura 3. 18 Consulta finalizada**

En el asistente de importación de archivo deberemos ubicar la dirección donde se encuentra almacenado nuestro archivo PIC.TXT y lo seleccionaremos, después se deberá especificar donde los datos a importar se deberán ubicar en la tabla creada como se indica en la Figura 3. 19.

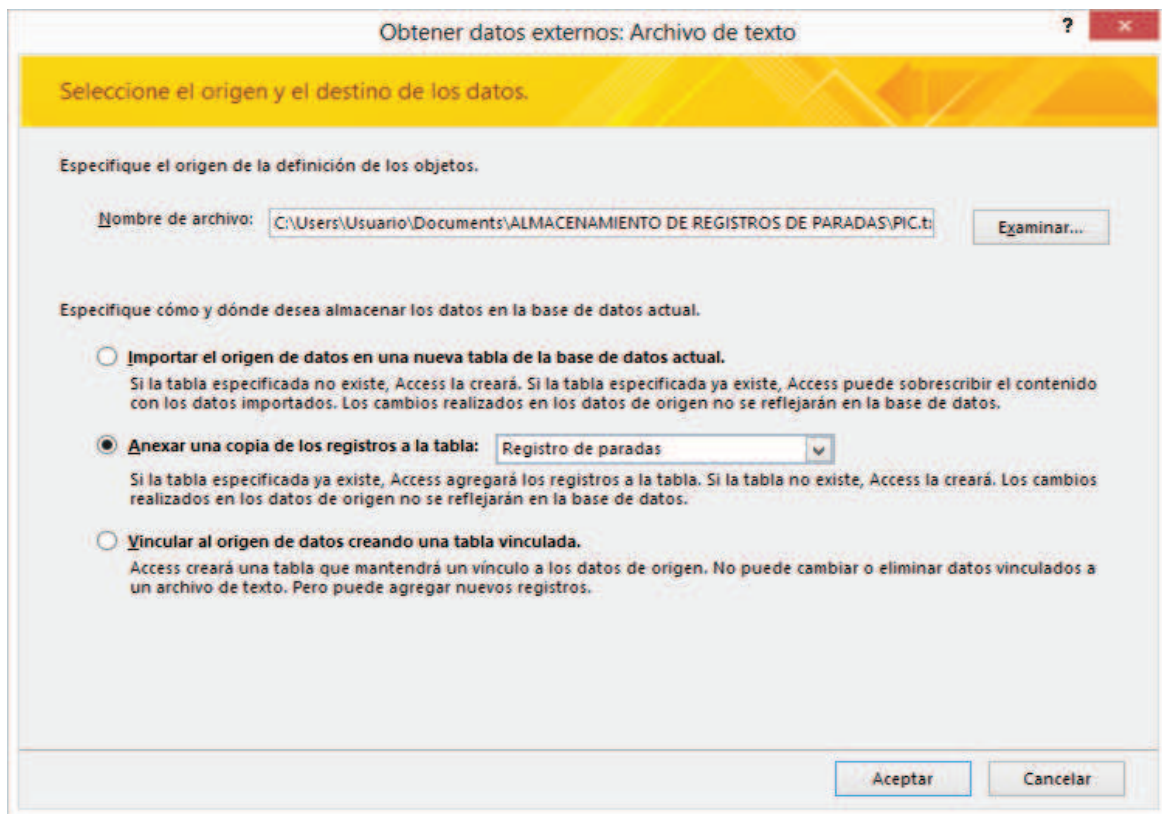


Figura 3. 19 Ubicación de datos a importar en tabla creada

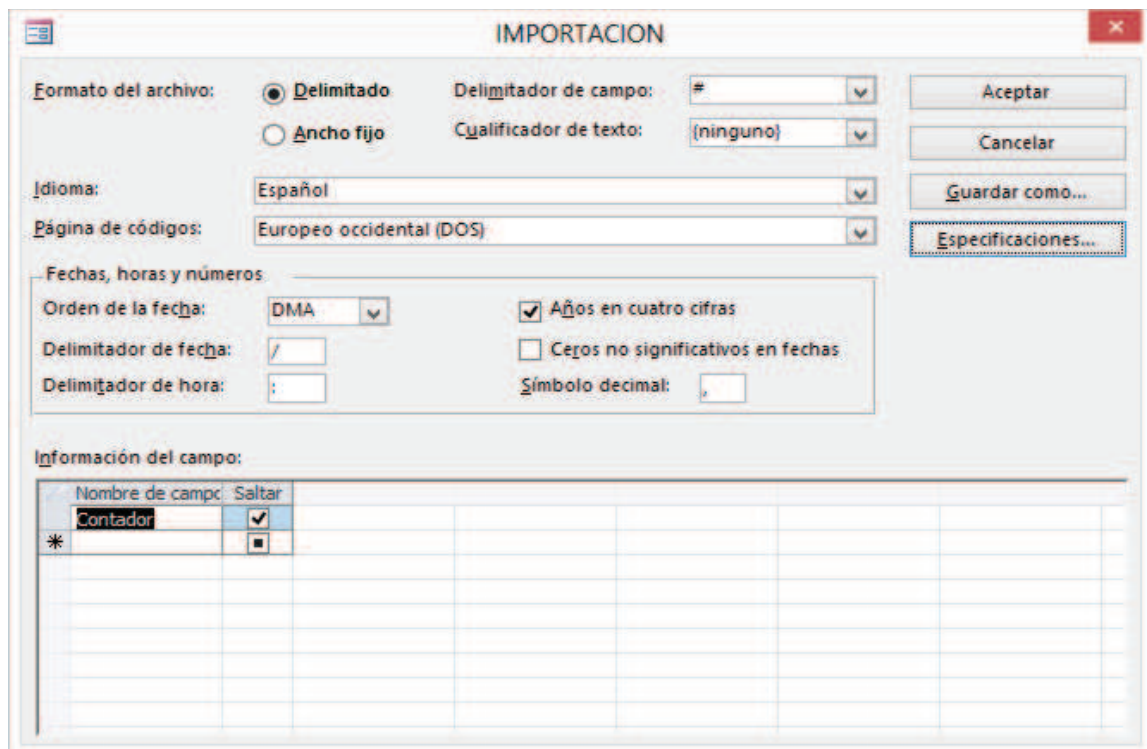


Figura 3. 20 Guardar especificación para importación

En la Figura 3. 21, se podrá visualizar el archivo TXT importado y la especificación que se creó.

¿Qué delimitador separa los campos? Seleccione el delimitador apropiado y compruebe cómo se ve afectado el texto en la vista previa que aparece abajo.

Elija el delimitador que separa los campos:

Tabulación     Punto y coma     Coma     Espacio     Otros:

Primera fila contiene nombres de campos    Cualificador de texto: [ninguno] v

BUS	PARADA	TIEMPO DE LLEGADA	TIEMPO DE SALIDA	FECHA
C1	QUITUMBE	21:25:41	21:26:06	12/05/2014

Ayanzado...    Cancelar    < Atrás    Siguiete >    Finalizar

**Figura 3. 21 Asistente para importación de texto**

Una vez finalizada la importación de datos se podrá visualizar en la Figura 3. 23 el resultado de la importación, como se puede observar al usar un delimitador se puede asignar cada valor a un campo.

Se recomienda que la primera fila de nuestro archivo TXT tenga los mismos valores que la de nuestra base de datos, es decir que se tenga los mismos campos de BUS, PARADA, TIEMPO DE LLEGADA, TIEMPO DE SALIDA.

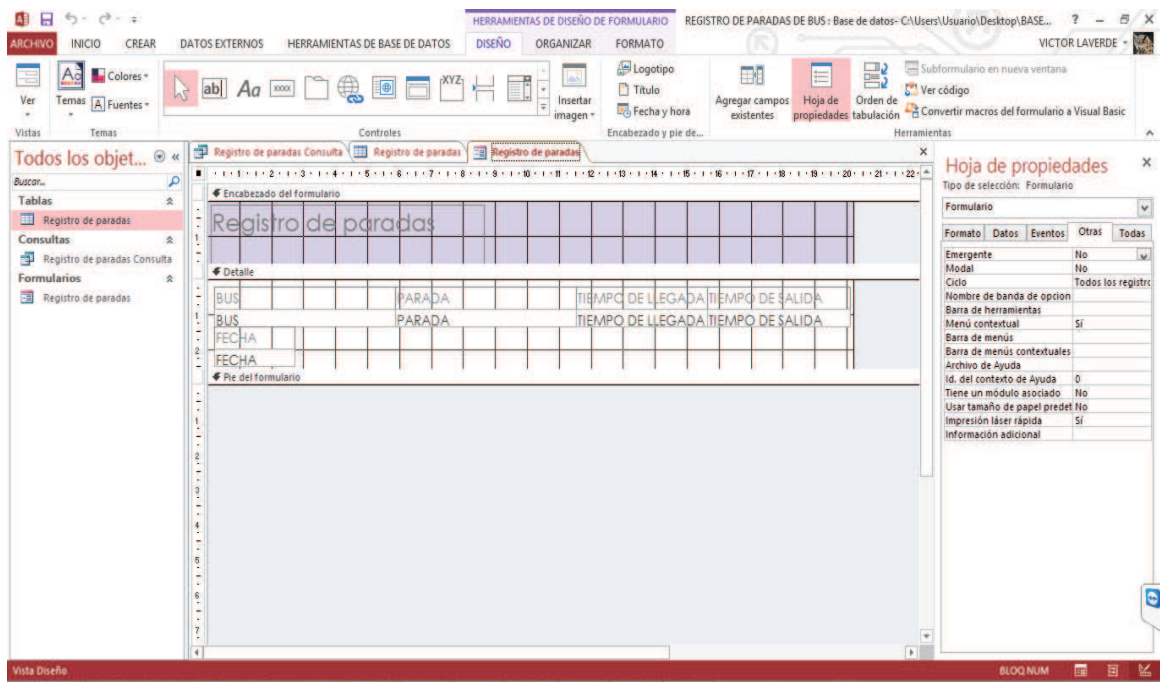
The screenshot shows the Microsoft Access interface with the 'Registro de paradas' table open. The table contains the following data:

Contador	BUS	PARADA	TIEMPO DI	TIEMPO DI	FECHA
1	C	QUITUMBE	21:25:41	21:26:06	12/05/2014
2	C1	QUITUMBE	21:25:41	21:26:06	12/05/2014
3					
*	(Nuevo)				

**Figura 3. 22 Resultado de la importación**

Para finalizar la creación de nuestra base de datos de importación y almacenamiento de datos desde nuestro micro-controlador es creando un formulario, donde se tendrá una plantilla de visualización, en el mismo que se podrá ingresar más datos en la base como realizar funciones específicas a través de macros.

Para realizar el formulario se deberá utilizar el asistente de formulario donde se especifica la tabla de origen, y los campos que se desee visualizar en el formulario.



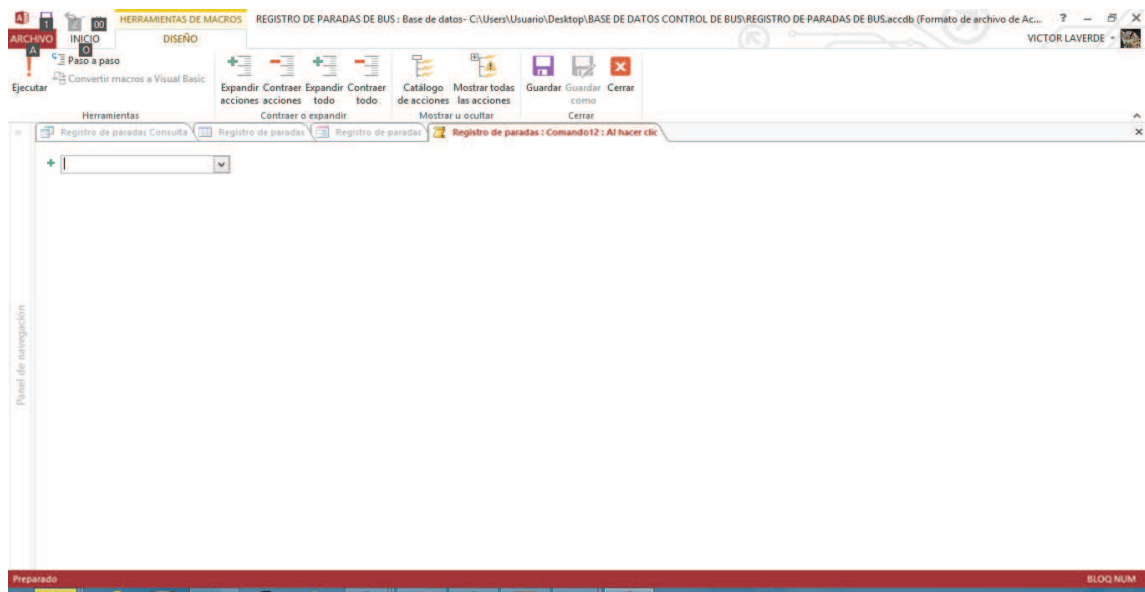
**Figura 3. 23 Asistente de formulario**

Para culminar la creación del formulario se procederá con la creación de un ícono que permite automatizar la importación del archivo TXT para lo cual se creará en primer lugar un ícono de botón obtenido en la pestaña de diseño de la barra de herramientas y en segundo lugar se deberá desplegar la opción de hoja de propiedades.

Ubicado sobre el botón creado se deberá seleccionar la opción de eventos de la hoja de propiedades, donde especificaremos la función a realizar cuando se dé un clic sobre este. Para la creación de la función a utilizar, en el botón se da varias opciones como es por medio de programación de Visual Basic o por macros.

En este se escogió por macros que ya que tiene una interfaz más amigable y permite seleccionar funciones ya preparadas.

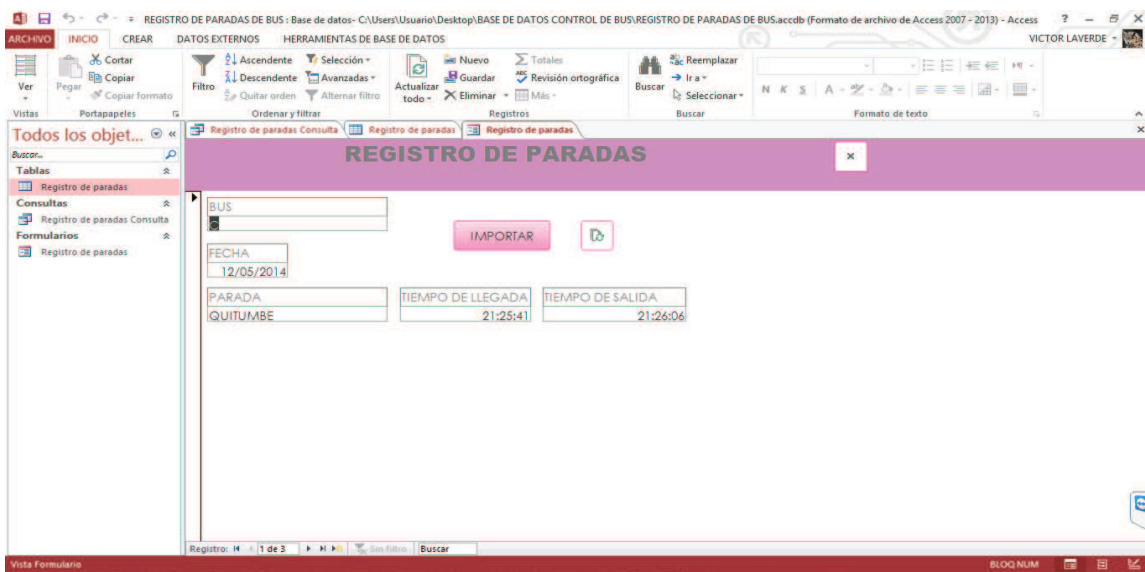
En la Figura 3. 24, se visualiza la plantilla del macro donde agregaremos la función de importar datos de texto, en este se especificará el nombre del archivo a importar, el nombre de la tabla donde queremos que se importe y seleccionaremos el especificador, como se indicó el especificador me da los parámetros del delimitador de datos.



**Figura 3. 24 Plantilla del macro**

En la Figura 3. 25, se podrá visualizar el formulario culminado donde se observa los datos tomados de la tabla de origen; y también se observa el botón que nos permitirá importar directamente los archivos TXT.

Como se ha indicado, solo se crea una base de datos básica para almacenar los datos obtenidos de nuestro sistema de control de puertas. Pero hay que recalcar que en esta aplicación se podrá realizar modificaciones según las necesidades del usuario como son asociar a otras tablas, o crear funciones que nos permitan obtener los tiempos entre cada parada, etc.



**Figura 3. 25 Formulario culminado**



### 3.4 PRUEBAS

En este punto del presente trabajo se realizarán pruebas para determinar tanto las ventajas y desventajas de nuestro sistema de control de puertas del bus como también proceder a realizar las correcciones necesarias.

En la Tabla 3. 2 se presenta un cuadro comparativo, donde se pudo obtener la distancia mínima en la que se tiene una óptima comunicación usando una potencia de salida de 1mW y 2mW.

En las pruebas realizadas se consideró la ubicación de circuito del sistema de control de puertas en la parte de adelante del bus y la parada con el transmisor de la señal ubicada desde 15, 12, 9, 5 y 1 metros de distancia, teniendo como resultado un tiempo máximo de apertura de la puerta desde que recibe la señal y un tiempo máximo de cierre de la puerta desde que deja de recibir la señal.

Como se puede observar a distancias mayores de 9 metros no se puede comunicar los módulos de radiofrecuencia incluso pasado un tiempo de 60 segundos cuando se tiene una potencia de 1 mW, pero en el caso de 2 mW y realizando cambios en la sensibilidad de la recepción del bus se tiene comunicación entre los 2 módulos incluso hasta más de 30 metros.

En el caso de distancias menores a 9 metros y con una potencia de 1 mW se puede establecer una buena comunicación entre los módulos de radiofrecuencia con lo que el bus podrá abrir las puertas y cerrarlas en un promedio de 5 segundos aproximadamente por lo que se considera como una prueba exitosa debido a que se considera como parámetros la longitud del bus, la velocidad del bus y el tiempo de apertura de las puertas.

Finalmente se comprueba el tiempo en demorarse al abrir la puerta del bus en una distancia de 1 metro donde se obtuvo un tiempo promedio de 1 segundo, por lo tanto se toma como una prueba exitosa porque se considera que el bus está a 1 metro de distancia de la parada cuando este se encuentre detenido.

P. TX	Distancia (m)	Tiempo de apertura	Tiempo de cerrado	T1 (S)	T2 (S)	SI	NO	Observaciones
1 mW	1	21:43:52	21:44:09	1,13	1	X		Se valida una rápida apertura y cierre
1 mW	5	21:44:37	21:44:44	4,84	3,93	X		Se valida un promedio de 4 s de apertura y cierre
1 mW	9	21:46:27	21:46:46	5,72	5,94	x		Se tiene recepción a partir de 9 metros
1 mW	12	N/A	N/A	60	60		x	No se tiene conectividad
1 mW	15	N/A	N/A	60	60		x	No se tiene conectividad
2 mW	1	21:50:01	21:52:05	1,09	1,03	X		Se valida una rápida apertura y cierre
2 mW	5	21:54:23	21:55:46	1,13	1,06	X		Se valida una rápida apertura y cierre
2 mW	9	21:56:38	21:57:12	3,08	2,91	X		Se valida una rápida apertura y cierre
2 mW	12	21:57:23	21:57:58	3,57	3,5	X		Se valida una rápida apertura y cierre
2 mW	15	22:01:01	22:01:43	4,09	4	X		Se valida una rápida apertura y cierre

**Tabla 3. 2 Cuadro comparativo de tiempos de apertura y cierre de la puerta**

## CAPITULO 4

### CONCLUSIONES Y RECOMENDACIONES

#### 4.1 CONCLUSIONES

Este capítulo cierra con el proyecto de titulación incorporando las reflexiones de los análisis realizados durante el presente trabajo, así mismo, las conclusiones referentes a la elaboración de dicho proyecto.

- Implementamos un prototipo de control electrónico para abrir y cerrar automáticamente las puertas de un bus únicamente en sus respectivas paradas, mediante el uso de micro-controladores, módulos de radiofrecuencia, dispositivos electro-mecánicos, opto-acopladores, entre otros.
- Investigamos que tipo de dispositivo electrónico puede transmitir información en la banda libre de radiofrecuencia de 2,4 GHz para enviar y recibir información de la llegada de un bus a su respectiva parada, entonces el haber escogido el módulo de radiofrecuencia XBEE nos permitió un excelente manejo y transportación de datos, ya que dicho transporte de datos es confiable entre dispositivos remotos, las características que este módulo nos ofrece es un pro muy importante; bajo costo para implementaciones en gran escala, bajo consumo de energía entre otras.
- Utilizamos un trasmisor-receptor para enviar y recibir información desde un módulo de radiofrecuencia ubicado en un bus hacia sus respectivas paradas, elementos como el reloj de tiempo real, el display LCD, o el módulo USB-FLASH, fueron primordiales en la construcción del circuito para su correcto funcionamiento y posterior visualización al usuario.

- Investigamos los controladores que poseen los puertos de transmisión para radiofrecuencia y conexión USB para así permitir la transferencia de datos desde una memoria flash hacia un ordenador. El haber seleccionado los PI18F452 y PI16F870 en la construcción de nuestro circuito del bus y las paradas fue una decisión acertada, ya que por las cualidades y funciones que estos nos ofrecen, permiten la conexión e interacción con nuestro módulo de RF a través de la comunicación serial; destacamos también otras cualidades como poseer una mayor memoria de programa de 32 K y 1,5 K para memoria de datos, así como contar con 5 puertos para cubrir todas las necesidades de nuestro proyecto.
- Implementamos un circuito actuador que se encarga del sistema de apertura y cerradura de puertas de un bus mediante el control de un micro-controlador. Al utilizar una electroválvula con un cilindro de doble efecto, nos permitió establecer un ambiente real ya que físicamente los buses trabajan con estos cilindros de aire y podremos observar los tiempos de respuesta al aplicar un voltaje de 24 V, además debido a que la electroválvula trabaja con este voltaje también nos permitió validar un mejor manejo de la regulación de voltaje tomando como referencia una fuente de voltaje de un bus.
- Realizamos varios programas que sirvieron para la configuración de los controladores, del módulo de RF y base de datos necesarios para el funcionamiento global de nuestro proyecto. Los programas utilizados fueron el XCTU, el Mikro-C y el Microsoft Access, y fueron los justos y necesarios para lograr el funcionamiento del sistema, el primero por ser de hecho propiedad del fabricante, el otro por ser una aplicación a la cual estamos familiarizados, ya que la utilizamos durante nuestra formación académica y la última por ser sencilla para administrar la información obtenida de nuestro sistema.

## 4.2 RECOMENDACIONES

Finalmente las recomendaciones que se darán a continuación servirán para ampliar el campo aplicativo de nuestro proyecto, así como errores que deben evitarse:

- El proyecto que hemos desarrollado permitirá múltiples aplicaciones afines, como por ejemplo la posibilidad de conectarse mediante dispositivos multimedia, móviles como teléfonos celulares, tablets, ordenadores, entre otros, y que mediante el soporte de múltiples plataformas que existen hoy en día se podrá acceder a la información otorgada por nuestro sistema.
- Al momento de configurar los módulos de radiofrecuencia XBEE dentro de nuestro sistema, no debemos olvidar que uno siempre tiene que ser el coordinador, ya que este permite sincronizar al resto de módulos dentro de nuestra red, esto se lo realiza al programarlo con nuestro software XCTU, como vemos esta es una red, por lo que una implementación posterior puede ser a niveles más grandes, se puede crear posteriormente aplicaciones para android, blackberry, mac o windows para facilitar el acceso al control de nuestro sistema en red a través de internet.
- Como estamos trabajando con radiofrecuencia, y especialmente con los módulos XBEE, debemos tomar en cuenta que puede ser susceptible a interferencias o compartición de canal con otras señales emitidas bajo el mismo canal, en la misma banda de los 2,4 GHz, por ser libre, como pueden ser el bluetooth o el wifi, para ello se recomienda el uso de la versión 2 del módulo de radiofrecuencia que se encargará de validar un canal libre de interferencia.

## REFERENCIAS BIBLIOGRÁFICAS

- Dallas, s. (2014). *Data Sheet, reloj de tiempo real ds1307*. Obtenido de <http://www.dalsemi.com>
- Digi International, I. (2 de 11 de 2008). *Digi international, Inc*. Obtenido de <https://www.aparkfun.com/datasheets/Wireless/Zigbee-2,5-Manual.pdf>
- ECURED. (2014). *Ecured, Conocimientos con todos y para todos*. Obtenido de [http://www.ecured.cu/index.php/Bandas\\_de\\_radiofrecuencia](http://www.ecured.cu/index.php/Bandas_de_radiofrecuencia)
- Frenzel, L. (mayo, 2003). *Sistemas electrónicos de comunicaciones*. México, DF: Alfaomega.
- García, Á. F. (2011). *Física con ordenador*. Obtenido de <http://www.sc.ehu.es/sbweb/fisica/cuantica/negro/espectro/espectro.htm>
- Hernández, I. J. (enero/abril de 2012). *SCIELO, Ingeniería electrónica, automática y comunicaciones*. Obtenido de [http://www.scielo.sld.cu./scielo.php?pid=S1815-5928201200010000&script=sci\\_arttext](http://www.scielo.sld.cu./scielo.php?pid=S1815-5928201200010000&script=sci_arttext)
- Instruments, T. (enero de 2014). *Data Sheet, regulador de voltaje LM117T*. Obtenido de <http://www.ti.com.cn/lit/ds/symlink/lm117.pdf>
- Instruments, T. (2014). *Data Sheet, regulador de voltaje LM7805*. Obtenido de <http://www.sparfun.com/datasheets/Components/LM7805.pdf>
- Junior Figueroa Olmedo, E. Q. (2013). *Scribd*. Obtenido de <http://www.scribd.com/doc/58980339/Teoria-y-Programacion-Modulos-XBEE>
- Microchip. (2008). *Data Sheet, microcontrolador 16F870*. Obtenido de <http://www.microchip.com>

Microchip. (13 de mayo de 2008). *Data Sheet, microcontrolador 18f452*. Obtenido de <http://www.microchip.com>

Olmedo, J. F. (s.f.). *Scribd*. Obtenido de es

Salesiana, U. P. (2009). *Universidad Politecnica Salesiana*. Obtenido de [http://dspace.uos.edu.ec/bitstream/123456789/150/4/CAPITULO III.pdf](http://dspace.uos.edu.ec/bitstream/123456789/150/4/CAPITULO%20III.pdf)

Simons, E. (9 de octubre de 2012). *Ingeniería de microondas*. Obtenido de <http://microondasism.blogspot.com/2012/10/bandaism-lasbandas-ism-industrial.html>

Villa, J. (2014). *Scribd*. Obtenido de <http://es.scribd.com/doc/28247774/Electrovalvula-Una-Electrovalvula-Es-Un-Dispositivo-Disenado>

XCTU. (2014). *Digi, Your M2M Expert*. Obtenido de <http://www.digi.com>

Yahoo, r. (2011). *Yahoo, respuestas*. Obtenido de <http://espanol.answers.yahoo.com/question/index?qid=20101213162551AA7rdf8>

## **ANEXOS**



**ANEXO No. 1**

**PROGRAMACIÓN DEL PI18F452**

```

//ESCUELA POLITÉCNICA NACIONAL
//ESCUELA DE FORMACIÓN DE TECNÓLOGOS//
//PROYECTO DE TITULACIÓN//
//PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN
ELECTRÓNICA Y TELECOMUNICACIONES
//VICTOR LAVERDE
//GUIDO TAPIA//
//=====CONEXIONES PARA LCD=====
sbit LCD_RS at RB4_bit;
sbit LCD_EN at RB5_bit;
sbit LCD_D4 at RB0_bit;
sbit LCD_D5 at RB1_bit;
sbit LCD_D6 at RB2_bit;
sbit LCD_D7 at RB3_bit;
sbit LCD_RS_Direction at TRISB4_bit;
sbit LCD_EN_Direction at TRISB5_bit;
sbit LCD_D4_Direction at TRISB0_bit;
sbit LCD_D5_Direction at TRISB1_bit;
sbit LCD_D6_Direction at TRISB2_bit;
sbit LCD_D7_Direction at TRISB3_bit;
//=====DECLARACIÓN DE VARIABLES=====
char uart_rd;
char lcd=0;
char hora=0;
unsigned char dato[100];
int conteo1=0,segundos=10,cerrar=0;
short int control=0;

unsigned short read_ds1307(unsigned short address );
void write_ds1307(unsigned short address,unsigned short w_data);

```

```
unsigned short sec;
unsigned short minute;
unsigned short hour;
unsigned short day;
unsigned short date;
unsigned short month;
unsigned short year;
unsigned short dat;
```

```
char time[9];
char timei[9];
char ddate[11];
```

```
unsigned char txt[10];
unsigned char BCD2UpperCh(unsigned char bcd);
unsigned char BCD2LowerCh(unsigned char bcd);
```

```
short int lectura=0;
short int para=0;
```

```
//=====INTERRUPCIÓN POR TIMER 0=====
```

```
void interrupt(){
if (INTCON.T0IF=1) //ENTRA INTERRUPCIÓN A TIMER 0
{
    conteo1=conteo1+1;
    if (conteo1>=10){
        segundos=segundos+1;
        conteo1=0;
    }
    if (segundos>=100)
```

```

{
  cerrar=1;
  PORTD.RD1=1;
  delay_ms(400);
  PORTD.RD1=0;
  segundos=0;
  conteo1=0;
}
TMR0H=0;
TMR0L=177;
INTCON.T0IF=0; //ENCERA EL TIMER 0
}//=====FIN DE INTERRUPCIÓN TIMER0=====
}
//=====FUNCIÓN PARA GRABAR LA HORA EN DS1307=====
void Config_time(){
//Punto para grabación de fecha y hora
  if (lectura!=10){
    write_ds1307(0,0x80); //Reinicia los segundos a 0, y detiene el oscilador
    write_ds1307(1,0x17); //Escribe los minutos en decimales
    write_ds1307(2,0x21); //Escribe la hora
    write_ds1307(3,0x07); //Escribe el día de la semana
    write_ds1307(4,0x15); // Escribe el día en el mes
    write_ds1307(5,0x03); // Escribe el mes
    write_ds1307(6,0x14); // Escribe los dos últimos dígitos del año (2014)
    write_ds1307(7,0x10); //SQWE output at 1 Hz
    write_ds1307(0,0x00); //Reinicia los segundos a 0, e inicia el oscilador
  }
}
//===== SOLO REJOJ =====
unsigned short read_ds1307(unsigned short address)
{

```

```

I2C1_Start();
I2C1_Wr(0xd0); //address 0x68 followed by direction bit (0 for write, 1 for read)
0x68 followed by 0 --> 0xD0
I2C1_Wr(address);
I2C1_Repeated_Start();
I2C1_Wr(0xd1); //0x68 followed by 1 --> 0xD1
dat=I2C1_Rd(0);
I2C1_Stop();
return(dat);
}

```

```

//=====TRANSFORMA TIEMPO A BCD=====

```

```

unsigned char BCD2UpperCh(unsigned char bcd){
return ((bcd >> 4) + '0');
}

```

```

unsigned char BCD2LowerCh(unsigned char bcd){
return ((bcd & 0x0F) + '0');
}

```

```

void write_ds1307(unsigned short address,unsigned short w_data){
EEPROM_Write(0x02,10);
I2C1_Start(); // issue I2C start signal
I2C1_Wr(0xD0); // send byte via I2C (device address + W)
I2C1_Wr(address); // send byte (address of DS1307 location)
I2C1_Wr(w_data); // send data (data to be written)
I2C1_Stop(); // issue I2C stop signal
}

```

```

//=====FUNCIÓN PARA VISUALIZAR FECHA Y HORA=====

```

```

//Visualiza la fecha y hora una vez que se configuro la fecha en el DS1307

```

```

void Display_time(){

```

```

if (hora==0){          // Ingresa a la función de lectura, y visualización
//Únicamente si la bandera hora se encuentra en 0
    sec=read_ds1307(0); // Lee segundos y almacena en variable sec
    minute=read_ds1307(1); // Lee minutos y almacena en variable minute
    hour=read_ds1307(2); // Lee la hora y almacena en variable hour
    day=read_ds1307(3); // Lee el día de la semana y almacena en variable day
    date=read_ds1307(4); // Lee el día al mes y almacena en variable date
    month=read_ds1307(5); // Lee el mes y almacena en variable month
    year=read_ds1307(6); // Lee los dos dígitos del año y lo almacena en variable
//Transforma a BCD el tiempo obtenido del DS1307 y almacena en variable
// time[]
time[0] = BCD2UpperCh(hour);
time[1] = BCD2LowerCh(hour);
time[2] = ':';
time[3] = BCD2UpperCh(minute);
time[4] = BCD2LowerCh(minute);
time[5] = ':';
time[6] = BCD2UpperCh(sec);
time[7] = BCD2LowerCh(sec);
time[8] = '\0';

// Transforma a BCD la fecha obtenida del DS1307 y almacena en variable
ddate[]
ddate[0] = BCD2UpperCh(date);
ddate[1] = BCD2LowerCh(date);
ddate[2] = '/';
ddate[3] = BCD2UpperCh(month);
ddate[4] = BCD2LowerCh(month);
ddate[5] = '/';
ddate[6] = '2';

```

```

ddate[7] = '0';
ddate[8] = BCD2UpperCh(year);
ddate[9] = BCD2LowerCh(year);
ddate[10] = '\0';
Delay_ms(50);
//Muestra la hora y la fecha en el LCD
Lcd_Out(3,-3,time);
Lcd_Out(4,-3,ddate);
}}

```

==FUNCIÓN PARA CAPTURAR LA HORA AL ABRIR LA PUERTA DEL BUS==

// La función permitirá capturar la fecha y hora al momento de abrir la puerta en  
//cada parada y almacenarlo en las variables deseadas, y posteriormente  
//almacenarlos en la memoria extraíble USB. Se tiene que tomar en cuenta que la  
//puerta únicamente se abrirá cuando se reciba la señal del transmisor de las  
//paradas

```

void leehora(){
    sec=read_ds1307(0);
    minute=read_ds1307(1);
    hour=read_ds1307(2);
    day=read_ds1307(3);
    date=read_ds1307(4);
    month=read_ds1307(5);
    year=read_ds1307(6);
    time[0] = BCD2UpperCh(hour);
    time[1] = BCD2LowerCh(hour);
    time[2] = ':';
    time[3] = BCD2UpperCh(minute);
    time[4] = BCD2LowerCh(minute);
    time[5] = ':';
    time[6] = BCD2UpperCh(sec);
}

```

```

time[7] = BCD2LowerCh(sec);
time[8] = '\0';

ddate[0] = BCD2UpperCh(date);
ddate[1] = BCD2LowerCh(date);
ddate[2] = '/';
ddate[3] = BCD2UpperCh(month);
ddate[4] = BCD2LowerCh(month);
ddate[5] = '/';
ddate[6] = '2';
ddate[7] = '0';
ddate[8] = BCD2UpperCh(year);
ddate[9] = BCD2LowerCh(year);
ddate[10] = '\0';
Delay_ms(50);
}

```

//FUNCIÓN PARA CAPTURAR LA HORA AL CERRARSE LA PUERTA DEL BUS

// La función permitirá capturar la fecha y hora al momento de cerrar la puerta en  
//cada parada y almacenarlo en las variables deseadas, y posteriormente  
//almacenarlos en la memoria extraíble USB. Se tiene que tomar en cuenta que la  
//puerta únicamente se cerrara la puerta del bus cuando se deje de recibir la señal  
//del transmisor de las paradas

```

void leehoraa(){
    sec=read_ds1307(0);
    minute=read_ds1307(1);
    hour=read_ds1307(2);
    day=read_ds1307(3);
    date=read_ds1307(4);
    month=read_ds1307(5);
}

```



```

year=read_ds1307(6);

timei[0] = BCD2UpperCh(hour);
timei[1] = BCD2LowerCh(hour);
timei[2] = ':';
timei[3] = BCD2UpperCh(minute);
timei[4] = BCD2LowerCh(minute);
timei[5] = ':';
timei[6] = BCD2UpperCh(sec);
timei[7] = BCD2LowerCh(sec);
timei[8] = '\0';

ddate[0] = BCD2UpperCh(date);
ddate[1] = BCD2LowerCh(date);
ddate[2] = '/';
ddate[3] = BCD2UpperCh(month);
ddate[4] = BCD2LowerCh(month);
ddate[5] = '/';
ddate[6] = '2';
ddate[7] = '0';
ddate[8] = BCD2UpperCh(year);
ddate[9] = BCD2LowerCh(year);
ddate[10] = '\0';
Delay_ms(50);
}

//=====INICIALIZACIÓN DE PROGRAMA=====
void main(){
    I2C1_Init(100000);           //Inicializa la comunicación I2C en 100KHz
    lectura=EEPROM_Read(0x02); //Almacena la información previamente

```

```

//almacenada en la memoria EEPROM del PIC
//para configuración del tiempo
Config_time();           //Llamada de la función Config_time para
//configuración del tiempo en el DS1307
//=====CONFIGURACION DE PORTICOS=====
TRISD=0;
PORTD=0;
PORTD.RD4=0;
PORTD.RD5=1;
PORTD.RD=1;
delay_ms(500);
PORTD.RD2=1;
delay_ms(500);
TRISB=0;
//=====CONFIGURACIÓN DE LCD=====
Lcd_Init();             // Inicializa el LCD
Lcd_Cmd(_LCD_CLEAR);   // Limpia el Display
Lcd_Cmd(_LCD_CURSOR_OFF); // Quita el cursor del display
Lcd_Out(1,1,"ESCUELA");
Lcd_Out(2,1,"POLITECNICA");
Lcd_Out(3,-3,"NACIONAL");
Delay_ms(1000);
Lcd_Out(1,1,"ESCUELA DE");
Lcd_Out(2,1,"FORMACION DE");
Lcd_Out(3,-3,"TECNOLOGOS");
Delay_ms(1000);
Lcd_cmd(_lcd_clear);
Lcd_Out(1,1,"POLIBUS # 01");
Lcd_out(2,1,"CIRCUITO C1");
//=====INICIALIZA EL PUERTO SERIAL UART1=====

```

```
    UART1_Init(9600);    //Inicializa las funciones de la comunicación serial
//UART1 con una velocidad de 9600 baudios
```

```
// =====CUERPO DEL PROGRAMA=====
```

```
while(1){
```

```
    Display_time();    //Llama a la función Display_time para leer la fecha y //hora
del DS1307 y poder visualizarlo en el LCD
```

```
//=====ESCUCHANDO EL CANAL DE RF=====
```

```
    if (UART1_Data_Ready()){    //El PIC se mantiene escuchando constantemente
//el canal de recepción, y continuara con el flujo
```

```
//del programa solo si recibe información del //receptor RF
```

```
        uart_rd = UART1_Read(); //Si hay un dato recibido, almacena la información
//en variable.
```

```
        hora=1;                //Detiene la visualización de la fecha y hora
```

```
//=====APERTURA DE LA PUERTA DEL BUS=====
```

```
//En el caso de no ser la señal que esta previamente configurada en el PIC, //NO
permitirá la apertura de las puertas y el PIC volverá a escuchar el canal de RF
```

```
    if (uart_rd==65){ //Si la señal recibida del receptor es 0X0A continuara
//con el flujo del programa
```

```
    if (para==0) //y si la bandera para=0
```

```
        leehoraa(); //Procederá a capturar la fecha y hora antes de apertura
//de la puerta.
```

```
        para=1; //activa la bandera para, e informa el nombre de la parada
//que va a ser almacenada en la memoria extraíble USB
```

```
portd.RD5=0; //Manda un 0 lógico por el pörtico RD5 y detiene el flujo de
//aire del solenoide A de la electroválvula.
```

```
portd.RD4=1; //Manda un 1 lógico por el pörtico RD4 y abre el flujo del
//aire del solenoide B de la electroválvula.
```

```
//Muestra en el LCD el nombre de la parada y visualiza la apertura de la orden de
apertura de puerta
```

```
Lcd_out(3,-3,"QUITUMBE ");
```

```
Lcd_out(4,-3,"ABRE PUERTAS ");
```

```
//===== ACTIVACIÓN DE TIMER 0 =====
```

```
//Se inicia las configuraciones de las interrupciones en este caso usando el Timer
0 para posteriormente llamar a la función de interrupción
```

```
INTCON = 0b11100000; //Activa las interrupciones
```

```
TMR0H=0;
```

```
TMR0L=177; //Se asigna el valor a TMR0L
```

```
INTCON.TMR0IE=1; //Activa la interrupción por Timer 0
```

```
INTCON.T0IF=0;
```

```
T0CON=0b11000111;
```

```
conteo1=0;
```

```
segundos=0;
```

```
//===== FIN ACTIVACIÓN =====
```

```
}
```

```
//=====PARADAS=====
```

```
// Se replica parte del programa que permitirá abrir la puerta según las señales de
//las respectivas paradas
```

// También se replicara la función que permitirá mantener la puerta abierta del bus  
//hasta que se deje de recibir la señal del transmisor de parada

```
if (uart_rd==66){      // Parada Quillallacta
    if (para==0)
        leehoraa();
    para=2;
    Portd.RD5=0;      //Manda un 0 lógico por el pörtico RD5 y detiene el flujo de
//aire del solenoide A de la electroválvula.
    Portd.RD4=1;      //Manda un 1 lógico por el pörtico RD4 y abre el flujo del
//aire del solenoide B de la electroválvula.
    lcd_out(3,-3,"QUILLALLACTA  ");
    Lcd_out(4,-3,"ABRE PUERTAS  ");
    //UART1_Write_Text(time);
    //===== ACTIVAR TIMER =====
    INTCON = 0b11100000;
    TMR0H=0;
    TMR0L=177;
    INTCON.TMR0IE=1;
    INTCON.T0IF=0;
    T0CON=0b11000111;
    conteo1=0;
    segundos=0;
    //===== FIN ACTIVACIÓN =====
}
```

```
if (uart_rd==67){      // Parada Otoyá
    if (para==0)
        leehoraa();
    para=3;
    Portd.RD5=0;      //Manda un 0 lógico por el pörtico RD5 y detiene el flujo de
```

```

        //aire del solenoide A de la electroválvula.
Portd.RD4=1;    //Manda un 1 lógico por el pörtico RD4 y abre el flujo del
                //aire del solenoide B de la electroválvula.

Lcd_out(3,-3,"OTOYA    ");
Lcd_out(4,-3,"ABRE PUERTAS  ");
//UART1_Write_Text(time);
//===== ACTIVAR TIMER =====
INTCON = 0b11100000;
TMR0H=0;
TMR0L=177;
INTCON.TMR0IE=1;
INTCON.T0IF=0;
T0CON=0b11000111;
conteo1=0;
segundos=0;
//===== FIN ACTIVACIÓN =====
}

if (uart_rd==68){    // Parada El Capulí
    if (para==0)
        leehoraa();
    para=4;
    Portd.RD5=0;    //Manda un 0 lógico por el pörtico RD5 y detiene el flujo de
                    //aire del solenoide A de la electroválvula.
    Portd.RD4=1;    //Manda un 1 lógico por el pörtico RD4 y abre el flujo del aire
                    //del solenoide B de la electroválvula.

    Lcd_out(3,-3,"EL CAPULI    ");
    Lcd_out(4,-3,"ABRE PUERTAS  ");

```

```

//UART1_Write_Text(time);
//===== ACTIVAR TIMER =====
INTCON = 0b11100000;
TMR0H=0;
TMR0L=177;
INTCON.TMR0IE=1;
INTCON.T0IF=0;
T0CON=0b11000111;
conteo1=0;
segundos=0;
//===== FIN ACTIVACIÓN =====
}

```

```

if (uart_rd==69){ // Parada Pacarillacta
    if (para==0)
        leehoraa();
    para=5;
}

```

Portd.RD5=0; //Manda un 0 lógico por el pórtilo RD5 y detiene el flujo de aire  
//del solenoide A de la electroválvula.

Portd.RD4=1; //Manda un 1 lógico por el pórtilo RD4 y abre el flujo del aire  
//del solenoide B de la electroválvula.

Lcd\_out(3,-3,"PACARILLACTA ");

Lcd\_out(4,-3,"ABRE PUERTAS ");

//UART1\_Write\_Text(time);

//===== ACTIVAR TIMER =====

INTCON = 0b11100000;

TMR0H=0;

TMR0L=177;

INTCON.TMR0IE=1;

```

INTCON.T0IF=0;
T0CON=0b11000111;
conteo1=0;
segundos=0;
//===== FIN ACTIVACIÓN =====
}
if (uart_rd==70){    // Parada Pte. Guajalo
    if (para==0)
        leehoraa();
    para=6;
    Portd.RD5=0; //Manda un 0 lógico por el pörtico RD5 y detiene el flujo de aire
//del solenoide A de la electroválvula.
    Portd.RD4=1; //Manda un 1 lógico por el pörtico RD4 y abre el flujo del aire
        //del solenoide B de la electroválvula.
    Lcd_out(3,-3,"PTE. GUAJALO  ");
    Lcd_out(4,-3,"ABRE PUERTAS  ");
    //UART1_Write_Text(time);
    //===== ACTIVAR TIMER =====
    INTCN = 0b11100000;
    TMR0H=0;
    TMR0L=177;
    INTCN.TMR0IE=1;
    INTCN.T0IF=0;
    T0CON=0b11000111;
    conteo1=0;
    segundos=0;
    //===== FIN ACTIVACIÓN =====
}

if (uart_rd==71){    //Parada San Cristobal

```



```
if (para==0)
    leehoraa();
para=7;
```

```
Portd.RD5=0; //Manda un 0 lógico por el pórtico RD5 y detiene el flujo de aire
//del solenoide A de la electroválvula.
```

```
Portd.RD4=1; //Manda un 1 lógico por el pórtico RD4 y abre el flujo del aire
//del solenoide B de la electroválvula.
```

```
Lcd_out(3,-3,"SAN CRISTOBAL ");
```

```
Lcd_out(4,-3,"ABRE PUERTAS ");
```

```
//UART1_Write_Text(time);
```

```
//===== ACTIVAR TIMER =====
```

```
INTCON = 0b11100000;
```

```
TMR0H=0;
```

```
TMR0L=177;
```

```
INTCON.TMR0IE=1;
```

```
INTCON.T0IF=0;
```

```
T0CON=0b11000111;
```

```
conteo1=0;
```

```
segundos=0;
```

```
//===== FIN ACTIVACIÓN =====
```

```
}
```

```
if (uart_rd==72){ // Parada Ayapamba
```

```
if (para==0)
```

```
    leehoraa();
```

```
para=8;
```

```
Portd.RD5=0; //Manda un 0 lógico por el pórtico RD5 y detiene el flujo de aire
//del solenoide A de la electroválvula.
```

```
Portd.RD4=1; //Manda un 1 lógico por el pörtico RD4 y abre el flujo del aire
           //del solenoide B de la electroválvula.
```

```
Lcd_out(3,-3,"AYAPAMBA  ");
```

```
Lcd_out(4,-3,"ABRE PUERTAS  ");
```

```
//UART1_Write_Text(time);
```

```
//===== ACTIVAR TIMER =====
```

```
INTCON = 0b11100000;
```

```
TMR0H=0;
```

```
TMR0L=177;
```

```
INTCON.TMR0IE=1;
```

```
INTCON.T0IF=0;
```

```
T0CON=0b11000111;
```

```
conteo1=0;
```

```
segundos=0;
```

```
//===== FIN ACTIVACIÓN =====
```

```
} }
```

```
//=====ACTIVACIÓN DEL CIERRE DE PUERTAS=====
```

```
if (cerrar==1){ //Si se encuentra en 1 la bandera obtenido de la interrupción del
                //Timer 0 continuara con el flujo del programa
```

```
Lcd_out(4,-3," CIERRA PUERTAS ");
```

```
Portd.RD4=0; //Manda un 0 lógico por el pörtico RD5 y detiene el flujo de aire
           //del solenoide B de la electroválvula.
```

```
Portd.RD5=1; //Manda un 1 lógico por el pörtico RD4 y abre el flujo del aire
//del solenoide A de la electroválvula.
```

```
hora=0; //Encera la bandera hora y se retorna la visualización de la
        //fecha y hora
```

```
cerrar=0; //Encera la bandera cerrar
```

```
TMR0H=0;
```

```

TMR0L=177;
INTCON.TMR0IE=1;
INTCON.T0IF=0;
intcon=0;

leehora();          //Captura la fecha y hora cuando se cierra la puerta del bus
//===== CREACIÓN DEL ARCHIVO PIC =====
//Debido que el microcontrolador se comunica con comunicación serial al módulo
//USB-Flash se ha utilizado las librerías de la comunicación USART que se
//dispone en el programa MIKROC como en la hoja del fabricante.

UART1_Write_Text("APPEND PIC.txt");    //Llama al archivo PIC.txt previamente
    //almacenado en la memoria extraíble
    //USB para proceder con la escritura
UART1_Write(13);        //Envía un Enter y se almacena en el archivo PIC.txt
Delay_ms(1000);
UART1_Write(13);        //Envía un Enter y se almacena en el archivo PIC.txt

//Escribe el nombre de la parada en el archivo PIC.txt según el valor de la
//bandera para que identifica a cada parada mediante el comando UART1_WRITE

if (para==1)
    UART1_Write_Text(" QUITUMBE");
if (para==2)
    UART1_Write_Text(" QUILLALLACTA");
if (para==3)
    UART1_Write_Text(" OTOYA");
if (para==4)
    UART1_Write_Text(" CAPULI");
if (para==5)

```

```

    UART1_Write_Text(" PACARILLACTA");
if (para==6)
    UART1_Write_Text(" PTE. DE GUAJALO");
if (para==7)
    UART1_Write_Text(" SAN CRISTOBAL");
if (para==8)
    UART1_Write_Text(" AYAPAMBA");

//=====GRABACIÓN DE FECHA Y HORA=====
//Envía la fecha y hora capturada en las variables timei, time, ddate cuando se
//realizó la apertura y cierre de las puertas al archivo PIC.txt
    UART1_Write_Text("///");
    UART1_Write_Text(timei);
    UART1_Write_Text("///");
    UART1_Write_Text(time);
    UART1_Write_Text("///");
    UART1_Write_Text(ddate);
    delay_ms(200);

    UART1_Write(26);      //Envía un CTRL Z
    delay_ms(200);

    para=0;              //Encera la bandera 0
    Lcd_Out(1,1,"POLIBUS # 01");
    Lcd_out(2,1,"CIRCUITO C1");
    Lcd_out(3,-3,"      ");
    Lcd_out(4,-3,"      ");
    Lcd=1;

}}}
//=====FIN DE PROGRAMA=====

```

**ANEXO No. 2**

**PROGRAMACIÓN DEL PI16F870**

```
//ESCUELA POLITÉCNICA NACIONAL
//ESCUELA DE FORMACIÓN DE TECNÓLOGOS//
//PROYECTO DE TITULACIÓN//
//PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN
ELECTRÓNICA Y TELECOMUNICACIONES
//VÍCTOR LAVERDE
//GUIDO TAPIA//
void main() {
UART1_Init(9600);      //Inicialización de comunicación serial
delay_ms(1000);
trisb=0;              //Configuración de registro TRISB como salida
portb=0;              //Configuración de PORTB
Portb.RB0=1;
while(1){
Portb.RB0=1;         //Manda 1 lógico y enciende led
UART1_Write_Text("A"); //Envía la letra A como señal al transmisor, este varia
                        //según la parada.
delay_ms(100);
Portb.RB0=0;         //Manda 0 lógico y apaga led
delay_ms(100);}}
```

**ANEXO No. 3**

**HOJAS DE DATOS**



# PIC18FXX2

## 28/40-pin High Performance, Enhanced FLASH Microcontrollers with 10-Bit A/D

### High Performance RISC CPU:

- C compiler optimized architecture/Instruction set
  - Source code compatible with the PIC16 and PIC17 instruction sets
- Linear program memory addressing to 32 Kbytes
- Linear data memory addressing to 1.5 Kbytes

Device	On-Chip Program Memory		On-Chip RAM (bytes)	Data EEPROM (bytes)
	FLASH (bytes)	# Single Word Instructions		
PIC18F242	16K	8192	156	256
PIC18F252	32K	16384	156	256
PIC18F442	16K	8192	166	256
PIC18F452	32K	16384	166	256

- Up to 10 MIPS operation:
  - DC - 40 MHz osc./clock input
  - 4 MHz - 10 MHz osc./clock input with PLL active
- 16-bit wide instructions, 8-bit wide data path
- Priority levels for interrupts
- 8 x 8 Single Cycle Hardware Multiplier

### Peripheral Features:

- High current sink/source 25 mA/25 mA
- Three external interrupt pins
- Timer0 module: 8-bit/16-bit timer/counter with 8-bit programmable prescaler
- Timer1 module: 16-bit timer/counter
- Timer2 module: 8-bit timer/counter with 8-bit period register (time-base for PWM)
- Timer3 module: 16-bit timer/counter
- Secondary oscillator clock option - Timer1/Timer3
- Two Capture/Compare/PWM (CCP) modules. CCP pins that can be configured as:
  - Capture input: capture is 16-bit, max. resolution 6.25 ns (T<sub>cy</sub>/16)
  - Compare is 16-bit, max. resolution 100 ns (T<sub>cy</sub>)
  - PWM output: PWM resolution is 1- to 10-bit, max. PWM freq. @: 8-bit resolution = 156 kHz, 10-bit resolution = 39 kHz
- Master Synchronous Serial Port (MSSP) module. Two modes of operation:
  - 3-wire SPI™ (supports all 4 SPI modes)
  - I<sup>2</sup>C™ Master and Slave mode

### Peripheral Features (Continued):

- Addressable USART module:
  - Supports RS-485 and RS-232
- Parallel Slave Port (PSP) module

### Analog Features:

- Compatible 10-bit Analog-to-Digital Converter module (A/D) with:
  - Fast sampling rate
  - Conversion available during SLEEP
  - Linearity < 1 LSB
- Programmable Low Voltage Detection (PLVD)
  - Supports Interrupt on-Low Voltage Detection
- Programmable Brown-out Reset (BOR)

### Special Microcontroller Features:

- 100,000 erase/write cycle Enhanced FLASH program memory typical
- 1,000,000 erase/write cycle Data EEPROM memory
- FLASH/Data EEPROM Retention > 40 years
- Self-reprogrammable under software control
- Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own On-Chip RC Oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options including:
  - 4X Phase Lock Loop (of primary oscillator)
  - Secondary Oscillator (32 kHz) clock input
- Single supply 5V In-Circuit Serial Programming™ (ICSP™) via two pins
- In-Circuit Debug (ICD) via two pins

### CMOS Technology:

- Low power, high speed FLASH/EEPROM technology
- Fully static design
- Wide operating voltage range (2.0V to 5.5V)
- Industrial and Extended temperature ranges
- Low power consumption:
  - < 1.5 mA typical @ 5V, 4 MHz
  - 25 µA typical @ 3V, 32 kHz
  - < 0.2 µA typical standby current

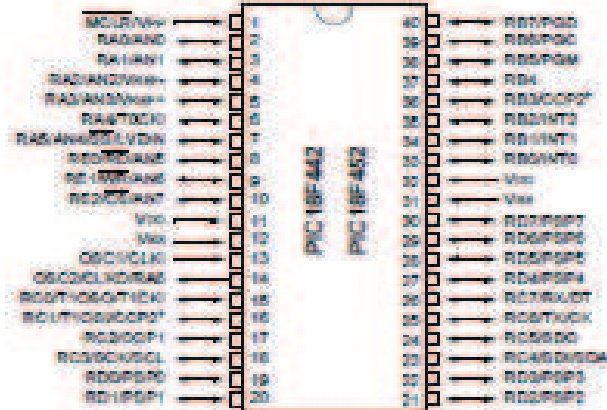




# PIC18FXX2

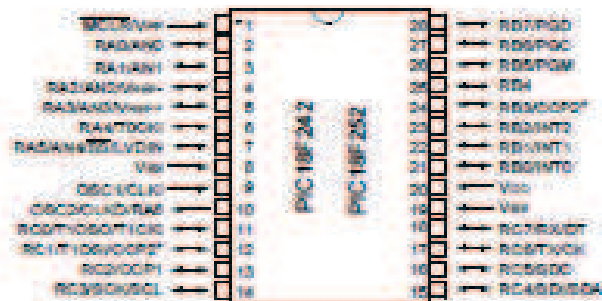
## Pin Diagrams (Cont'd)

### DIP



Note: Pin compatible with 40-pin PIC18CXX devices.

### DIP, SOIC



\* RB3 is the alternate pin for the OC0 pin multiplexing.



# PIC16F870/871

## 28/40-Pin, 8-Bit CMOS FLASH Microcontrollers

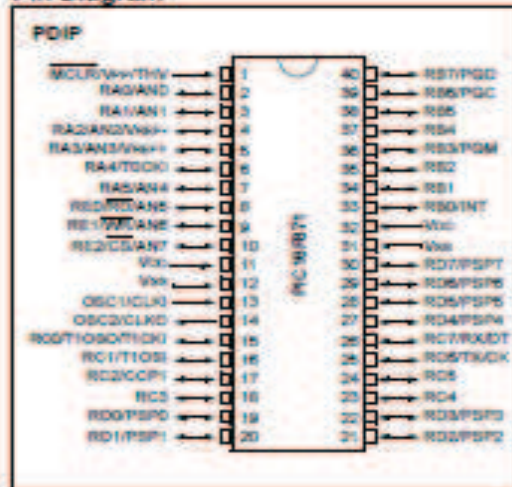
### Devices Included in this Data Sheet:

- PIC16F870
- PIC16F871

### Microcontroller Core Features:

- High performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two-cycle
- Operating speed: DC - 20 MHz clock input  
DC - 200 ns instruction cycle
- 2K x 14 words of FLASH Program Memory  
128 x 8 bytes of Data Memory (RAM)  
64 x 8 bytes of EEPROM Data Memory
- Pinout compatible to the PIC16CXXX 28 and 40-pin devices
- Interrupt capability (up to 11 sources)
- Eight level deep hardware stack
- Direct, Indirect and Relative Addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low power, high speed CMOS FLASH/EEPROM technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP™) via two pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial and Industrial temperature ranges
- Low power consumption:
  - < 1.5 mA typical @ 5V, 4 MHz
  - 20 µA typical @ 3V, 32 kHz
  - < 1 µA typical standby current

### Pin Diagram

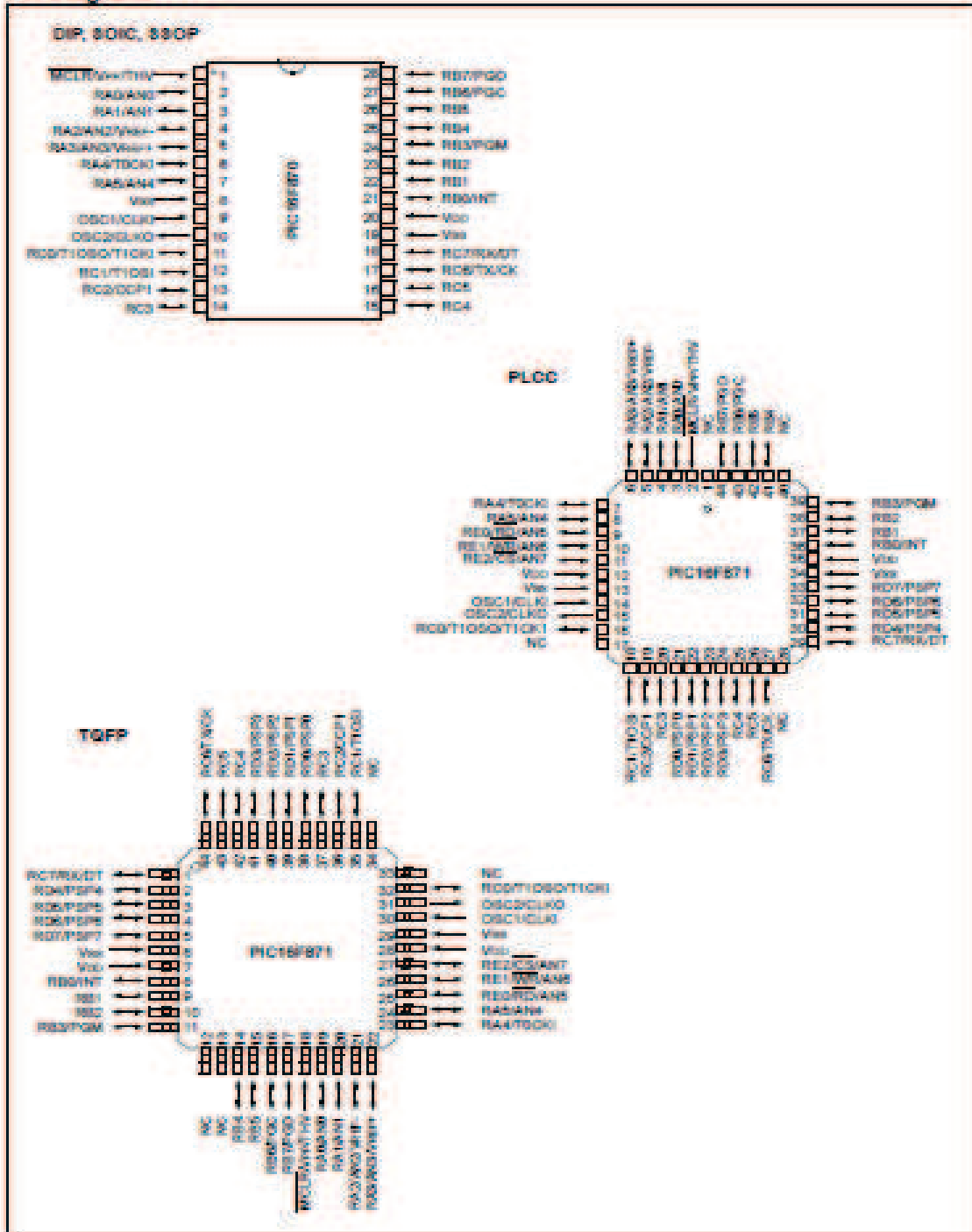


### Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during SLEEP via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- One Capture, Compare, PWM module
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) 8-bits wide, with external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

# PIC16F870/871

## Pin Diagrams



# PIC16F870/871

Key Features PICmicro™ Mid-Range MCU Family Reference Manual (DS33023)	PIC16F870	PIC16F871
Operating Frequency	DC - 20 MHz	DC - 20 MHz
RESETS (and Delays)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
FLASH Program Memory (14-bit words)	2K	2K
Data Memory (bytes)	128	128
EEPROM Data Memory	64	64
Interrupts	10	11
I/O Ports	Ports A,B,C	Ports A,B,C,D,E
Timers	3	3
Capture/Compare/PWM modules	1	1
Serial Communications	USART	USART
Parallel Communications	—	PSP
10-bit Analog-to-Digital Module	5 input channels	8 input channels
Instruction Set	35 Instructions	35 Instructions

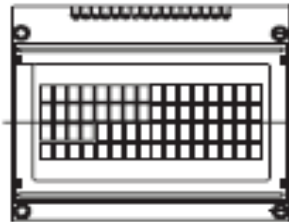
# HOJA DE DATOS LCD



LCD-016M004B

Vishay

## 16 x 4 Character LCD



### FEATURES

- 5 x 8 dots includes cursor
- Built - in controller (KS 0066 or Equivalent)
- + 5V power supply (Also available for + 3V)
- 1/16 duty cycle
- B/L to be driven by pin 1, pin 2, or pin 15, pin 16 or A and K
- N.V. optional for + 3V power supply

MECHANICAL DATA		
ITEM	STANDARD VALUE	UNIT
Module Dimension	70.6 x 60.0	mm
Viewing Area	60.0 x 32.6	mm
Mounting Hole	65.6 x 50.0	mm
Character Size	2.95 x 4.75	mm

ABSOLUTE MAXIMUM RATING					
ITEM	SYMBOL	STANDARD VALUE			UNIT
		MIN.	TYP.	MAX.	
Power Supply	VDD-VSS	- 0.3	-	7.0	V
Input Voltage	VI	- 0.3	-	VDD	V

NOTE: VSS = 0 Volt, VDD = 5.0 Volt

ELECTRICAL SPECIFICATIONS						
ITEM	SYMBOL	CONDITION	STANDARD VALUE			UNIT
			MIN.	TYP.	MAX.	
Input Voltage	VDD	VDD = +5V	4.7	5.0	5.3	V
		VDD = +3V	2.7	3.0	3.3	V
Supply Current	IDD	VDD = +5V	-	1.65	-	mA
Recommended LC Driving Voltage for Normal Temp. Version Module	VDD - V0	- 20 °C	5.0	5.1	5.7	V
		0 °C	4.6	4.8	5.2	
		25 °C	4.1	4.5	4.7	
		50 °C	3.9	4.2	4.5	
EL Power Supply Current	IEL	Vol = 110VAC; 400Hz	-	-	5.0	mA

DISPLAY CHARACTER ADDRESS CODE:																
Display Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DD RAM Address	00	01														0F
DD RAM Address	40	41														4F
DD RAM Address	10	11														1F
DD RAM Address	50	51														5F

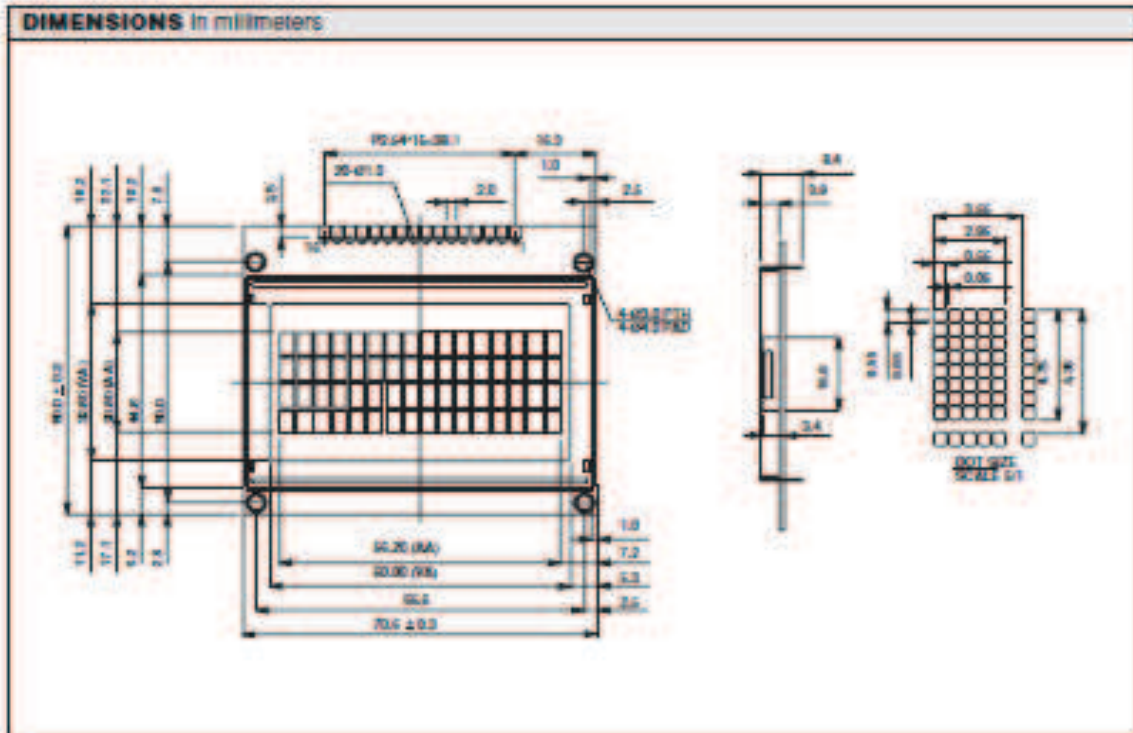
# LCD-016M004B



Vishay

16 x 4 Character LCD

PIN NUMBER	SYMBOL	FUNCTION
1	Vss	GND
2	Vdd	+3V or +5V
3	Vc	Contrast Adjustment
4	RS	H/L Register Select Signal
5	R/W	H/L Read/Write Signal
6	E	H → L Enable Signal
7	DB0	H/L Data Bus Line
8	DB1	H/L Data Bus Line
9	DB2	H/L Data Bus Line
10	DB3	H/L Data Bus Line
11	DB4	H/L Data Bus Line
12	DB5	H/L Data Bus Line
13	DB6	H/L Data Bus Line
14	DB7	H/L Data Bus Line
15	A/Vss	+4.2V for LED (RA = 0Ω)/Negative Voltage Output
16	K	Power Supply for BL (0V)





**DS1307**

**64 x 8, Serial, I<sup>2</sup>C Real-Time Clock**

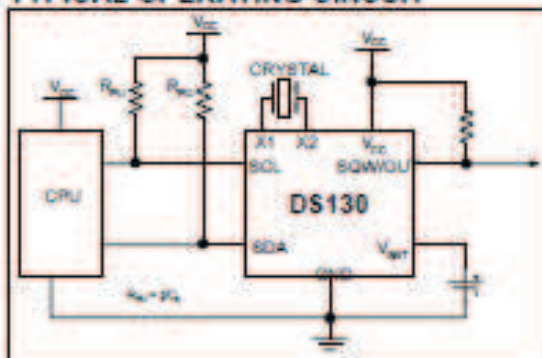
**GENERAL DESCRIPTION**

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 I<sup>2</sup>C, 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.

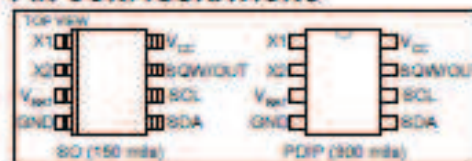
**FEATURES**

- Real-Time Clock (RTC) 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
- I<sup>2</sup>C Serial Interface
- Programmable Square-Wave Output Signal
- Automatic Power-Fail Detect and Switch Circuitry
- Consumes Less than 500nA in Battery-Backup Mode with Oscillator Running
- Optional Industrial Temperature Range: -40°C to +85°C
- Available in 8-Pin Plastic DIP or SO
- Underwriters Laboratories (UL) Recognized

**TYPICAL OPERATING CIRCUIT**



**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/Pb-free 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.



## PIN DESCRIPTION

PIN	NAME	FUNCTION
1	X1	Connections for Standard 32.768kHz Quartz Crystal. The internal oscillator circuitry is designed for operation with a crystal having a specified load capacitance ( $C_L$ ) of 12.5pF. X1 is the input to the oscillator and can optionally be connected to an external 32.768kHz oscillator. The output of the internal oscillator, X2, is floated if an external oscillator is connected to X1.
2	X2	Note: For more information on crystal selection and crystal layout considerations, refer to Application Note 58: Crystal Considerations with Dallas Real-Time Clocks.
3	V <sub>BAT</sub>	Backup Supply Input for Any Standard 3V Lithium Cell or Other Energy Source. Battery voltage must be held between the minimum and maximum limits for proper operation. Diodes in series between the battery and the V <sub>BAT</sub> pin may prevent proper operation. If a backup supply is not required, V <sub>BAT</sub> must be grounded. The nominal power-fail trip point (V <sub>PF</sub> ) voltage at which access to the RTC and user RAM is denied is set by the internal circuitry as 1.25 x V <sub>BAT</sub> nominal. A lithium battery with 45mAh or greater will back up the DS1307 for more than 10 years in the absence of power at +25°C.  UL recognized to ensure against reverse charging current when used with a lithium battery. Go to: <a href="http://www.maxim-ic.com/qainfo/ul">www.maxim-ic.com/qainfo/ul</a> .
4	GND	Ground
5	SDA	Serial Data Input/Output. SDA is the data input/output for the I <sup>2</sup> C serial interface. The SDA pin is open drain and requires an external pullup resistor. The pullup voltage can be up to 5.5V regardless of the voltage on V <sub>CC</sub> .
6	SCL	Serial Clock Input. SCL is the clock input for the I <sup>2</sup> C interface and is used to synchronize data movement on the serial interface. The pullup voltage can be up to 5.5V regardless of the voltage on V <sub>CC</sub> .
7	SQW/OUT	Square Wave/Output Driver. When enabled, the SQWE bit set to 1, the SQW/OUT pin outputs one of four square-wave frequencies (1Hz, 4kHz, 8kHz, 32kHz). The SQW/OUT pin is open drain and requires an external pullup resistor. SQW/OUT operates with either V <sub>CC</sub> or V <sub>BAT</sub> applied. The pullup voltage can be up to 5.5V regardless of the voltage on V <sub>CC</sub> . If not used, this pin can be left floating.
8	V <sub>CC</sub>	Primary Power Supply. When voltage is applied within normal limits, the device is fully accessible and data can be written and read. When a backup supply is connected to the device and V <sub>CC</sub> is below V <sub>TR</sub> , read and writes are inhibited. However, the timekeeping function continues unaffected by the lower input voltage.

## DETAILED DESCRIPTION

The DS1307 is a low-power clock/calendar with 56 bytes of battery-backed SRAM. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The DS1307 operates as a slave device on the I<sup>2</sup>C bus. Access is obtained by implementing a START condition and providing a device identification code followed by a register address. Subsequent registers can be accessed sequentially until a STOP condition is executed. When V<sub>CC</sub> falls below 1.25 x V<sub>BAT</sub>, the device terminates an access in progress and resets the device address counter. Inputs to the device will not be recognized at this time to prevent erroneous data from being written to the device from an out-of-tolerance system. When V<sub>CC</sub> falls below V<sub>BAT</sub>, the device switches into a low-current battery-backup mode. Upon power-up, the device switches from battery to V<sub>CC</sub> when V<sub>CC</sub> is greater than V<sub>BAT</sub> + 0.2V and recognizes inputs when V<sub>CC</sub> is greater than 1.25 x V<sub>BAT</sub>. The block diagram in Figure 1 shows the main elements of the serial RTC.



# LM1117/LM1117I

## 800mA Low-Dropout Linear Regulator

### General Description

The LM1117 is a series of low dropout voltage regulators with a dropout of 1.2V at 800mA of load current. It has the same pin-out as National Semiconductor's industry standard LM317. The LM1117 is available in an adjustable version, which can set the output voltage from 1.25V to 13.8V with only two external resistors. In addition, it is also available in five fixed voltages, 1.8V, 2.5V, 2.85V, 3.3V, and 5V. The LM1117 offers current limiting and thermal shutdown. Its circuit includes a zero-trimmed bandgap reference to assure output voltage accuracy to within  $\pm 1\%$ . The LM1117 series is available in LLP, TO-263, SOT-223, TO-220, and TO-252 D-PAK packages. A minimum of 10 $\mu$ F tantalum capacitor is required at the output to improve the transient response and stability.

### Features

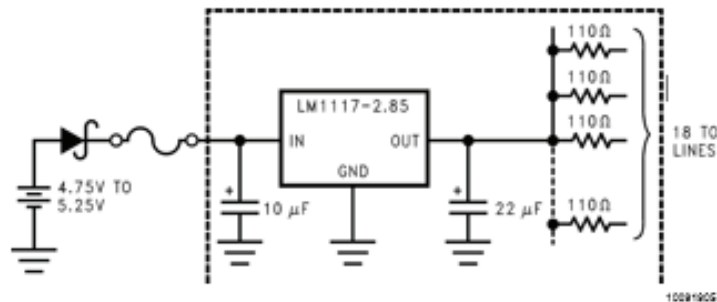
- Available in 1.8V, 2.5V, 2.85V, 3.3V, 5V, and Adjustable Versions
- Space Saving SOT-223 and LLP Packages
- Current Limiting and Thermal Protection
- Output Current 800mA
- Line Regulation 0.2% (Max)
- Load Regulation 0.4% (Max)
- Temperature Range
  - LM1117 0°C to 125°C
  - LM1117I -40°C to 125°C

### Applications

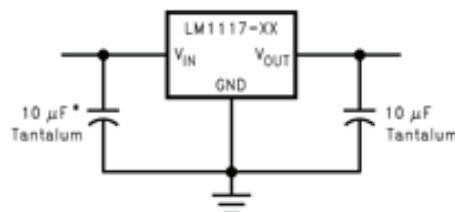
- 2.85V Model for SCSI-2 Active Termination
- Post Regulator for Switching DC/DC Converter
- High Efficiency Linear Regulators
- Battery Charger
- Battery Powered Instrumentation

### Typical Application

Active Terminator for SCSI-2 Bus



### Fixed Output Regulator



\* Required if the regulator is located far from the power source  
100919026

**Ordering Information**

Package	Temperature Range	Part Number	Packaging Marking	Transport Media	NSC Drawing			
3-lead SOT-223	0°C to +125°C	LM1117MPX-ADJ	N03A	Tape and Reel	MP04A			
		LM1117MPX-1.8	N12A	Tape and Reel				
		LM1117MPX-2.5	N13A	Tape and Reel				
		LM1117MPX-2.85	N04A	Tape and Reel				
		LM1117MPX-3.3	N05A	Tape and Reel				
	-40°C to +125°C	LM1117IMPX-ADJ	N03B	Tape and Reel				
		LM1117IMPX-3.3	N05B	Tape and Reel				
		LM1117IMPX-5.0	N06B	Tape and Reel				
		3-lead TO-220	0°C to +125°C	LM1117T-ADJ		LM1117T-ADJ	Reels	T03B
				LM1117T-1.8		LM1117T-1.8	Reels	
LM1117T-2.5	LM1117T-2.5			Reels				
LM1117T-2.85	LM1117T-2.85			Reels				
LM1117T-3.3	LM1117T-3.3			Reels				
3-lead TO-252	0°C to +125°C	LM1117DTX-ADJ	LM1117DT-ADJ	Tape and Reel	T003B			
		LM1117DTX-1.8	LM1117DT-1.8	Tape and Reel				
		LM1117DTX-2.5	LM1117DT-2.5	Tape and Reel				
		LM1117DTX-2.85	LM1117DT-2.85	Tape and Reel				
		LM1117DTX-3.3	LM1117DT-3.3	Tape and Reel				
	-40°C to +125°C	LM1117IDTX-ADJ	LM1117IDT-ADJ	Tape and Reel				
		LM1117IDTX-3.3	LM1117IDT-3.3	Tape and Reel				
		LM1117IDTX-5.0	LM1117IDT-5.0	Tape and Reel				
		8-lead LLP	0°C to +125°C	LM1117LDX-ADJ		1117ADJ	Tape and Reel	LDC08A
				LM1117LDX-1.8		1117-18	Tape and Reel	
LM1117LDX-2.5	1117-25			Tape and Reel				
LM1117LDX-2.85	1117-28			Tape and Reel				
LM1117LDX-3.3	1117-33			Tape and Reel				
-40°C to 125°C	LM1117ILDY-ADJ		1117IAD	Tape and Reel				
	LM1117ILDY-3.3		1117I33	Tape and Reel				
	LM1117ILDY-5.0		1117I50	Tape and Reel				
	TO-263		0°C to +125°C	LM1117SX-ADJ	LM1117SADJ	Tape and Reel	TS3B	
				LM1117SX-2.85	LM1117S2.85	Tape and Reel		
LM1117SX-3.3		LM1117S3.3		Tape and Reel				
LM1117SX-5.0		LM1117S5.0		Tape and Reel				

**Absolute Maximum Ratings** (Note 1)

If Military/Aerospace specified devices are required, contact the National Semiconductor Sales Office/ for availability and specifications.

Maximum Input Voltage ( $V_{IN}$  to GND)

20V

2) Internally Limited

Junction Temperature ( $T_J$ )

150°C

(Note 2)

Storage Temperature Range

-85°C to 150°C

Temperature

TO-220 (T) Package

SOT-223 (IMP) Package

ESD Tolerance (Note 3)

260°C, 10 sec

280°C, 4 sec please

2000V Distributors

**Operating Ratings** (Note 1)

Power Dissipation (Note

Input Voltage ( $V_{IN}$  to GND)

15V

Junction Temperature Range ( $T_J$ )(Note 2)

LM1117

0°C to 125°C

Lead

LM1117I

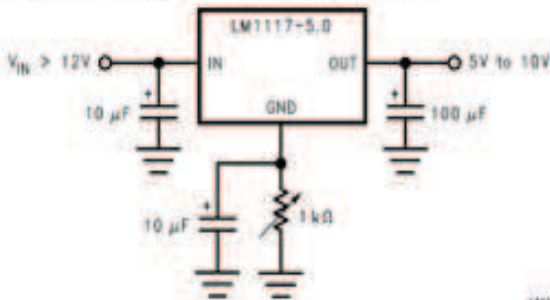
-40°C to 125°C

**LM1117 Electrical Characteristics**

Typicals and limits appearing in normal type apply for  $T_J = 25^\circ\text{C}$ . Limits appearing in **Boldface** type apply over the entire junction

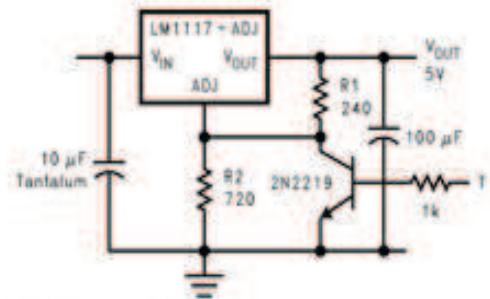
Symbol	Parameter	Conditions	Min (Note 5)	Typ (Note 4)	Max (Note 5)	Units		
$V_{REF}$	Reference Voltage	LM1117-ADJ $I_{OUT} = 10\text{mA}$ , $V_{IN}-V_{OUT} = 2\text{V}$ , $T_J = 25^\circ\text{C}$ $10\text{mA} \leq I_{OUT} \leq 800\text{mA}$ , $1.4\text{V} \leq V_{IN}-$	1.238 1.225	1.250 1.250	1.262 1.270	V $\nabla$		
$V_{OUT}$	Output Voltage	LM1117-1.8 $I_{OUT} = 10\text{mA}$ , $V_{IN} = 3.8\text{V}$ , $T_J =$ $25^\circ\text{C}$ $0 \leq I_{OUT} \leq 800\text{mA}$ , $3.2\text{V} \leq V_{IN} \leq 10\text{V}$	1.782 1.746	1.800 1.800	1.818 1.854	V $\nabla$		
		LM1117-2.5 $I_{OUT} = 10\text{mA}$ , $V_{IN} = 4.5\text{V}$ , $T_J =$ $25^\circ\text{C}$ $0 \leq I_{OUT} \leq 800\text{mA}$ , $3.9\text{V} \leq V_{IN} \leq 10\text{V}$	2.475 2.450	2.500 2.500	2.525 2.550	V $\nabla$		
		LM1117-2.85 $I_{OUT} = 10\text{mA}$ , $V_{IN} = 4.85\text{V}$ , $T_J =$ $25^\circ\text{C}$ $0 \leq I_{OUT} \leq 800\text{mA}$ , $4.25\text{V} \leq V_{IN} \leq$ $10\text{V}$	2.820 2.790 2.790	2.850 2.850 2.850	2.880 2.910 2.910	V $\nabla$ $\nabla$		
		LM1117-3.3 $I_{OUT} = 10\text{mA}$ , $V_{IN} = 5\text{V}$ , $T_J = 25^\circ\text{C}$ $0 \leq I_{OUT} \leq 800\text{mA}$ , $4.75\text{V} \leq V_{IN} \leq$	3.267 3.235	3.300 3.300	3.333 3.365	V $\nabla$		
		LM1117-5.0 $I_{OUT} = 10\text{mA}$ , $V_{IN} = 7\text{V}$ , $T_J = 25^\circ\text{C}$ $0 \leq I_{OUT} \leq 800\text{mA}$ , $6.5\text{V} \leq V_{IN} \leq 12\text{V}$	4.950 4.900	5.000 5.000	5.050 5.100	V $\nabla$		
		$\Delta V_{OUT}$	Line Regulation (Note 6)	LM1117-ADJ $I_{OUT} = 10\text{mA}$ , $1.5\text{V} \leq V_{IN}-V_{OUT}$		0.035	0.2	%
			LM1117-1.8 $I_{OUT} = 0\text{mA}$ , $3.2\text{V} \leq V_{IN} \leq 10\text{V}$		1	6	mV	
	LM1117-2.5 $I_{OUT} = 0\text{mA}$ , $3.9\text{V} \leq V_{IN} \leq 10\text{V}$		1	6	mV			
	LM1117-2.85 $I_{OUT} = 0\text{mA}$ , $4.25\text{V} \leq V_{IN} \leq 10\text{V}$		1	6	mV			
	LM1117-3.3 $I_{OUT} = 0\text{mA}$ , $4.75\text{V} \leq V_{IN} \leq 15\text{V}$		1	6	mV			
	LM1117-5.0 $I_{OUT} = 0\text{mA}$ , $6.5\text{V} \leq V_{IN} \leq 15\text{V}$		1	10	mV			

## Typical Application Circuits



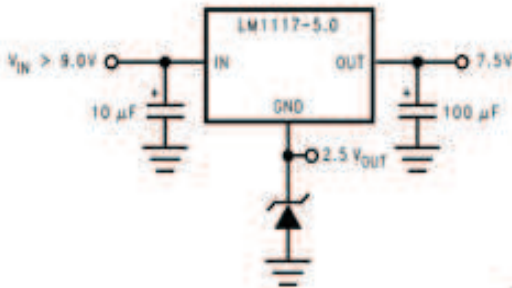
Adjusting Output of Fixed Regulators

10091900



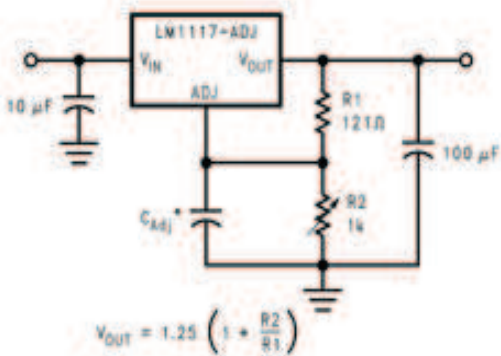
5V Logic Regulator with Electronic Shutdown\*

10091907



Regulator with Reference

10091901



\*  $C_{Adj}$  is optional, however it will improve ripple rejection.

1.25V to 10V Adjustable Regulator with Improved Ripple Rejection

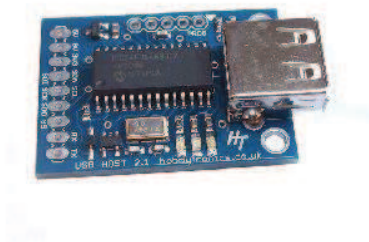
10091902

## HOJA DE DATOS MODULO USB-FLASH

### USB Board - Flash Drive Software

This software download is for our USB Board. It allows you to connect a USB Flash Memory Stick to the

Board and read/write files via a serial TTL interface. Ideal for data logging applications.



#### Features

Supports both the FAT16 and FAT32 file systems so can access large capacity memory sticks. Does not support Long File Names

The flash drive can be accessed via a PC Terminal Program (serial TTL connection required) or directly from a

Microcontroller using standard TTL Serial at the following baud rates

2400  
4800  
9600  
14400  
19200  
38400  
57600  
115200

#### The following Commands are available.

All commands should be terminated with a carriage return character (ascii 13 or 0x0D)

COMMAND	
<b>WRITE &lt;file&gt;</b>	Create [file] from input. Type input (or send from microcontroller). Terminate data entry with Control-Z
<b>APPEND &lt;file&gt;</b>	Append input to end of existing file. Type input (or send from microcontroller). Terminate data entry with Control-Z
<b>TYPE &lt;file&gt;</b>	Print contents of file, current dir only
<b>READ &lt;file&gt; [LINENO]</b>	Print a single line of file, current dir only. Useful for reading a configuration file.

<b>COPY &lt;file1&gt; &lt;file2&gt;</b>	Copy [file1] to [file2]
<b>DEL &lt;file&gt;</b>	Delete file, current dir only
<b>REN &lt;file1&gt; &lt;file2&gt;</b>	Rename [file1] to [file2]
<b>SIZE &lt;file&gt; [BYTE LINE]</b>	Return size of [file] in bytes or number of lines. The number of lines is useful when reading a configuration file.
<b>DIR &lt;name&gt;</b>	Change directory
<b>CD &lt;name&gt;</b>	
<b>MD &lt;name&gt;</b>	
<b>RD &lt;name&gt;</b>	
<b>DATE [yyyy-mm-</b>	Display or set the
<b>TIME [hh:mm:ss]</b>	Display or set the
<b>BAUD &lt;value&gt;</b>	Set Serial Port
<b>HELP or ?</b>	Display help
<b>WHO</b>	Display the VID

The following screen capture is the output from the HELP (?) command in terminal mode (see below)

```
Flash Drive Explorer v1.00

WRITE <file>           - create [file] from input
APPEND <file>          - append input to file
TYPE <file>            - print contents of file, current dir only
READ <file> [LINENO]   - print single line of file, current dir only
COPY <file1> <file2>   - copy [file1] to [file2]
DEL <file>             - delete file, current dir only
REN <file1> <file2>    - rename [file1] to [file2]
SIZE <file> [BYTE|LINE] - Return size of [file] in bytes or num lines

DIR <name>             - display directory
CD <name>              - change directory
MD <name>              - make directory
RD <name>              - remove directory

DATE [yyyy-mm-dd]     - display or set the date
TIME [hh:mm:ss]       - display or set the time (24 hr format)
BAUD <value>          - Set Serial Port Baud Rate
                       [2400|4800|9600|14400|19200|38400|57600|115200]

HELP or ?             - display help
WHO                   - display the VID and PID of the attached device
>
```

### Connections required for Flash Drive

- 5V power in
- 0V
- TX out
- RX in
- SS pin goes high when Flash drive inserted, low when removed

## HOJA DE DATOS MUDOLO XBEE SERIE 2

# XBee®/XBee-PRO® ZB RF Modules

---

### ZigBee RF Modules by Digi

**International** Models: XBEE2,  
XBEEPRO2, PRO S2B Hardware:  
S2 and S2B



Firmware Versions:

- 20xx - Coordinator - AT/Transparent Operation
- 21xx - Coordinator - API Operation
- 22xx - Router - AT/Transparent Operation
- 23xx - Router - API Operation
- 28xx - End Device - AT/Transparent Operation
- 29xx - End Device - API Operation



Digi International Inc.  
11001 Bren Road East  
Minnetonka, MN 55343  
877 912-3444 or 952 912-3444 <http://www.digi.com>



## Key Features

---

### High Performance, Low Cost

---

#### **XBee**

- Indoor/Urban: up to 133' (40 m)
- Outdoor line-of-sight: up to 400' (120 m)
- Transmit Power: 2 mW (3 dBm)
- Receiver Sensitivity: -96 dBm

#### **XBee-PRO (S2)**

- Indoor/Urban: up to 300' (90 m), 200' (60 m) for International variant
- Outdoor line-of-sight: up to 2 miles (3200 m), 5000' (1500 m) for International variant
- Transmit Power: 50mW (17dBm), 10mW (10dBm) for International variant
- Receiver Sensitivity: -102 dBm

#### **XBee-PRO (S2B)**

- Indoor/Urban: up to 300' (90 m), 200' (60 m) for International variant
- Outdoor line-of-sight: up to 2 miles (3200 m), 5000' (1500 m) for International variant
- Transmit Power: 63mW (18dBm), 10mW (10dBm) for International variant
- Receiver Sensitivity: -102 dBm

### Advanced Networking & Security

---

Retries and Acknowledgements  
DSSS (Direct Sequence Spread Spectrum)  
Each direct sequence channel has over

### Low Power

---

#### **XBee**

- TX Peak Current: 40 mA (@3.3 V)
- RX Current: 40 mA (@3.3 V)
- Power-down Current: < 1  $\mu$ A

#### **XBee-PRO (S2)**

- TX Peak Current: 295mA (170mA for international variant)
- RX Current: 45 mA (@3.3 V)
- Power-down Current: 3.5  $\mu$ A typical @ 25 degrees C

#### **XBee-PRO (S2B)**

- TX Peak Current: 205mA (117mA for international variant)
- RX Current: 47 mA (@3.3 V)
- Power-down Current: 3.5  $\mu$ A typical @ 25 degrees C

### Easy-to-Use

---

No configuration necessary for out-of box RF communications

AT and API Command Modes for configuring module parameters  
Small form factor  
Extensive command set  
Free X-CTU Software

## Worldwide Acceptance

---

FCC Approval (USA) Refer to Appendix A for FCC Requirements. Systems that contain XBee®/ XBee-PRO® ZB RF Modules inherit Digi Certifications.



ISM (Industrial, Scientific & Medical) 2.4 GHz frequency band

Manufactured under ISO 9001:2000 registered standards

XBee®/XBee-PRO® ZB RF Modules are optimized for use in US, Canada, Europe, Australia, and Japan (contact Digi for complete list of agency approvals).



## Specifications

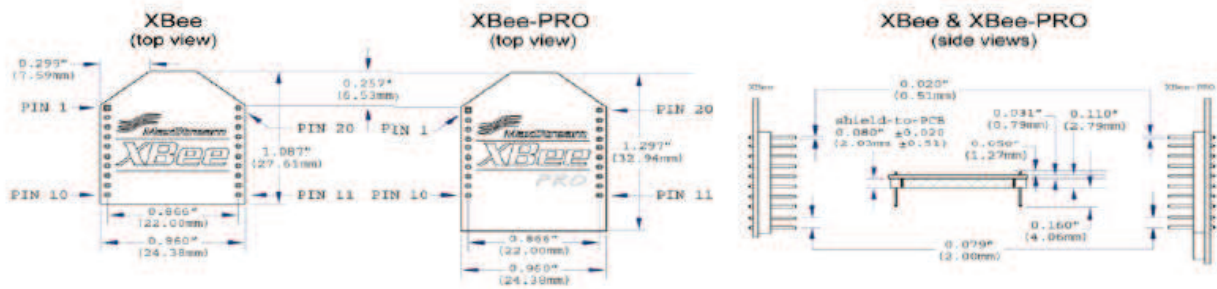
Specifications of the XBee®/XBee-PRO® ZB RF Module

Specification	XBee	XBee-PRO (S2)
Performance		
Indoor/Urban Range	up to 133 ft. (40 m)	Up to 300 ft. (90 m), up to 200 ft (60 m) international variant
Outdoor RF line-of-sight Range	up to 400 ft. (120 m)	Up to 2 miles (3200 m), up to 5000 ft (1500 m) international variant
Transmit Power Output	2mW (+3dBm), boost mode enabled 1.25mW (+1dBm), boost mode disabled	50mW (+17 dBm) 10mW (+10 dBm) for International variant
RF Data Rate	250,000 bps	250,000 bps
Data Throughput	up to 35000 bps (see chapter 4)	up to 35000 bps (see chapter 4)
Serial Interface Data Rate (software selectable)	1200 bps - 1 Mbps (non-standard baud rates also supported)	1200 bps - 1 Mbps (non-standard baud rates also supported)
Receiver Sensitivity	-96 dBm, boost mode enabled -95 dBm, boost mode disabled	-102 dBm
Power Requirements		
Supply Voltage	2.1 - 3.6 V	3.0 - 3.4 V
Operating Current (Transmit, max output power)	40mA (@ 3.3 V, boost mode enabled) 35mA (@ 3.3 V, boost mode disabled)	295mA (@3.3 V) 170mA (@3.3 V) international variant
Operating Current (Receive)	40mA (@ 3.3 V, boost mode enabled) 38mA (@ 3.3 V, boost mode disabled)	45 mA (@3.3 V)
Idle Current (Receiver off)	15mA	15mA
Power-down Current	< 1 uA @ 25°C	3.5 µA typical @ 25°C
General		
Operating Frequency		
Band	ISM 2.4 GHz	ISM 2.4 GHz
Dimensions	0.960" x 1.087" (2.438cm x 2.761cm)	0.960 x 1.297 (2.438cm x 3.294cm)
Operating Temperature	-40 to 85° C (industrial)	-40 to 85° C (industrial)
Antenna Options	Integrated Whip, Chip, RPSMA, or U.FL Connector	Integrated Whip, Chip, RPSMA, or U.FL Connector
Networking & Security		
Supported Network	Point-to-point, Point-to-multipoint, Peer-to-peer, and Mesh	Point-to-point, Point-to-multipoint, Peer-to-peer, and Mesh
Number of Channels	16 Direct Sequence Channels	14 Direct Sequence Channels
Channels	11 to 26	11 to 24
Addressing Options	PAN ID and Addresses, Cluster IDs and Endpoints (optional)	PAN ID and Addresses, Cluster IDs and Endpoints (optional)

Agency Approvals		
United States (FCC Part	FCC ID: OUIR-XBEE2	FCC ID: MCO-XBEEPRO2
Industry Canada (IC)	IC: 4214A-XBEE2	IC: 1846A-XBEEPRO2
Europe (CE)	ETSI	ETSI (International variant)

## Mechanical Drawings

Mechanical drawings of the XBee®/XBee-PRO® ZB RF Modules (antenna options not shown)



## Pin Signals

### Pin Assignments for the XBee-PRO Modules

(Low-asserted signals are distinguished with a horizontal line above signal name.)

Pin #	Name	Direction	Default State	Description
1	VCC	-	-	Power supply
2	DOUT	Output	Output	UART Data Out
3	DIN / <b>CONFIG</b>	Input	Input	UART Data In
4	DIO12	Both	Disabled	Digital I/O 12
5	<b>RESET</b>	Both	Open-Collector with pull-up	Module Reset (reset pulse must be at least 200 ns)
6	RSSI PWM / DIO10	Both	Output	RX Signal Strength Indicator / Digital IO
7	DIO11	Both	Input	Digital I/O 11
8	[reserved]	-	Disabled	Do not connect
9	DTR / SLEEP_RQ / DIO8	Both	Input	Pin Sleep Control Line or Digital IO 8
10	GND	-	-	Ground
11	DIO4	Both	Disabled	Digital I/O 4
12	<b>CTS</b> / DIO7	Both	Output	Clear-to-Send Flow Control or Digital I/O 7. CTS, if enabled, is an output.
13	<b>ON / SLEEP</b>	Output	Output	Module Status Indicator or Digital I/O 9
14	VREF	Input	-	Not used for EM250. Used for programmable secondary processor. For compatibility with other XBEE modules, we recommend connecting this pin voltage reference if Analog sampling is desired. Otherwise, connect to GND.
15	Associate / DIO5	Both	Output	Associated Indicator, Digital I/O 5
16	<b>RTS</b> / DIO6	Both	Input	Request-to-Send Flow Control, Digital I/O 6. RTS, if enabled, is an input.
17	AD3 / DIO3	Both	Disabled	Analog Input 3 or Digital I/O 3
18	AD2 / DIO2	Both	Disabled	Analog Input 2 or Digital I/O 2
19	AD1 / DIO1	Both	Disabled	Analog Input 1 or Digital I/O 1
20	AD0 / DIO0 / Commissioning Button	Both	Disabled	Analog Input 0, Digital IO 0, or Commissioning

- Signal Direction is specified with respect to the module
- See Design Notes section below for details on pin connections.

## EM250 Pin Mappings

---

The following table shows how the EM250 pins are used on the XBee.

EM250 Pin Number	XBee Pin Number	Other Usage
19 (GPIO 11)	16*	
20 (GPIO 12)	12*	
21 (GPIO 0)	15	
22 (GPIO 1)		<b>XBee</b> Tied to ground (module identification) <b>XBee-PRO (S2)</b> Low-asserting shutdown line for output power compensation circuitry. <b>XBee-PRO (S2B)</b> Used to communicate with Temp Sensor and control Shutdown for low power mode.
24 (GPIO 2)		<b>XBee</b> Not connected. Configured as output low. <b>XBee-PRO (S2)</b> Powers the output power compensation circuitry. <b>XBee-PRO (S2B)</b> Used to communicate with Temp Sensor and control Shutdown for low power mode.
25 (GPIO 3)	13	
26 (GPIO 4 / ADC 0)	20	Connected to pin 9 on 2x5 SIF header.
27 (GPIO 5 / ADC 1)	19	Connected to pin 10 on 2x5 SIF header.
29 (GPIO 6 / ADC 2)	18	
30 (GPIO 7 / ADC 3)	17	
31 (GPIO 8)	4	
32 (GPIO 9)	2*	
33 (GPIO 10)	3*	
34 (SIF_CLK)		Connected to pin 6 on 2x5 SIF header.
35 (SIF_MISO)		Connected to pin 2 on 2x5 SIF header.
36 (SIF_MOSI)		Connected to pin 4 on 2x5 SIF header.
37 (SIF_LOAD)		Connected to pin 7 on 2x5 SIF header.
40 (GPIO 16)	7	
41 (GPIO 15)	6	
42 (GPIO 14)	9	
43 (GPIO 13)	11	