



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA ELECTRÓNICA EN TELECOMUNICACIONES Y  
REDES

“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL DE ALARMAS  
COMUNITARIAS PARA EL BARRIO LA JOYA (CIUDAD RIOBAMBA)”

TESIS DE GRADO PREVIA OBTENCIÓN DEL TÍTULO DE

INGENIERO EN ELECTRÓNICA Y COMPUTACIÓN

PRESENTADO POR:

SANTOS DANILO LABRE GUANIN

RIOBAMBA – ECUADOR

2010

Agradezco a mi familia por estar en cada momento con su incentivo incondicional, a la vez un gran agradecimiento a mi tutor de tesis el Ing. Paul Romero por su apoyo para la realización de este trabajo, agradezco al Ing. Pablo Guevara por el empuje moral y palabras de apoyo, y a todos mis grandes compañeros que me ayudaron de una u otra manera para la finalización de esta tesis.

*Autor.*

Dedico este esfuerzo a mi padre Santos Wilfrido Labre y Ana María Guanín por ser la fortaleza y pilares de todo lo que soy y seré, palabras de apoyo, la persistencia y no rendirse ante todo obstáculo ayudaron a la culminación de este gran paso en mi vida, a mi hermano Giovanni, hermana Graciela, mi gran amiga Gladys y amigo Ricardo que siempre estuvieron junto a mí. En especial dedico a mis padres por su gran esfuerzo que han hecho para que alcance a culminar mi carrera, su paciencia y apoyo en cada paso que di.

NOMBRE	FIRMA	FECHA
Ing. MS.c. Iván Menes C.  DECANO DE LA FACULTAD DE  INFORMÁTICA Y ELECTRÓNICA	.....	.....
Ing. José Guerra  DIRECTOR DE LA ESCUELA  DE INGENIERÍA ELECTRÓNICA	.....	.....
Ing. Paúl Romero  DIRECTOR DE TESIS	.....	.....
Ing. Pablo Guevara  MIEMBRO DEL TRIBUNAL	.....	.....
Tlgo. Carlos Rodríguez  DIRECTOR DPTO.  DOCUMENTACION	.....	.....
NOTA DE LA TESIS	.....	

“Yo, **SANTOS DANILO LABRE GUANIN**, soy el responsable de las ideas, doctrinas y resultados expuestos en esta Tesis de Grado, y el patrimonio intelectual de la misma pertenecen a la Escuela Superior Politécnica de Chimborazo”

---

Santos Danilo Labre Guanín

# ÍNDICE GENERAL

## CAPÍTULO I: GENERALIDADES

1.1. ANTECEDENTES.....	20
1.2. JUSTIFICACIÓN .....	22
1.3. OBJETIVOS.....	23
1.3.1. Objetivo General .....	23
1.3.2. Objetivos Específicos .....	23

## CAPÍTULO II: FUNDAMENTO TEÓRICO

2.1. Línea Telefónica Convencional.....	24
2.1.1. Definición de señales Telefónica.....	25
2.1.1.1. Señal de invitación a marcar(Tono).....	25
2.1.1.2. Señal de ocupado .....	27
2.1.1.3. Señal de espera a respuesta .....	27
2.1.2. Desarrollo normal de una comunicación.....	28
2.1.3. Tonos DTMF (Dual Tone Multiplexed Frequency–frecuencia de los tonos multiplexados-).....	29
2.2. Microcontroladores .....	30
2.2.1 El Microcontrolador PIC18F4550 .....	32
2.2.1.1 Características Principales .....	32
2.2.1.2 Interrupción.....	36

2.2.1.3	Temporizador.....	37
2.2.1.4	Convertidor analógico/digital.....	38
2.2.1.5	Comunicación serie I <sup>2</sup> C con 18f4550.....	42
2.2.1.6	Protocolo del bus.....	45
2.2.1.7	Comunicación serie RS-232 con 18f4550.....	46
2.2.2	El Microcontrolador PIC16F628A.....	47
2.2.2.1	Características principales.....	48
2.2.2.2	Pines de I/O (Entrada/Salida).....	49
2.2.3	El Microcontrolador PIC16F876A.....	51
2.2.3.1	Organización de memoria.....	54
2.2.3.2	Organización de memoria de programa.....	54
2.2.3.3	Fichero de Registro de propósito general.....	56
2.2.3.4	Lectura de la memoria EEPROM de datos.....	56
2.2.3.5	Escritura de la memoria EEPROM de datos.....	56
2.2.3.6	USART en modo Asíncrono.....	56
2.2.3.7	Transmisión Asíncrono USART.....	57
2.2.3.8	Receptor Asíncrono USART.....	59
2.3.	Decodificador de doble tono multifrecuencia (DTMF MT8870).....	61
2.4.	Programas informáticos AUXILIARES.....	63
2.4.1.	Microcode Studio.....	63

2.4.2. Compilador PICBasic Pro.....	64
2.4.3. Compilador CCS C.....	64
2.4.3.1. Estructura de un programa .....	65
2.4.4. PICKit2.....	66
2.4.4.1. Conexion del puerto USB.....	66
2.4.4.2. LEDs de estado.....	66
2.4.4.3. Conector de programación.....	67
2.4.4.4. Aplicación de programación.....	67
2.4.4.5. Barra de Menú.....	68
2.4.5. PROTEUS 7.6 .....	69
2.4.5.1. Entorno Gráfico ISIS.....	70
2.4.5.2. Entorno Gráfico ARES.....	74

### **CAPÍTULO III: DISEÑO DEL SISTEMA**

3.1. REQUERIMIENTOS DEL SISTEMA.....	82
3.2. TRANSMISION.....	85
3.2.1. Etapa de alimentacion.....	85
3.2.2. Acoplamiento de señal.....	87
3.2.3. Amplificador de Audio.....	88
3.2.4. Detector de Línea Atendida.....	89
3.2.5. Etapa de control central.....	92



3.2.5.1 Programación de la Etapa de control central.....	93
3.3. RECEPTOR.....	95
3.3.1. Etapa de Alimentación.....	95
3.3.2. Etapa de Detección de Timbre.....	97
3.3.3. Etapa de identificación de Tonos.....	98
3.3.4. Etapa de Recepción de Datos.....	100
3.3.4.1 Programación de la Etapa de Recepción de Datos.....	101
3.3.5. Etapa de Almacenamiento de Datos.....	102
3.3.6. Etapa de Visualización de Información.....	105
3.3.7. Etapa de Ingreso de Información.....	106
3.3.7.1 Programación del proceso de Ingreso de Nuevo Usuario.....	106
3.3.7.2 Programación de Modificación de un Usuario.....	107
3.3.7.3 Programación de Borrado de un Usuario.....	108
3.3.7.4 Programación del Listado de Usuario.....	109
3.3.7.5 Programación del Listado de Alarmas Activadas.....	110
3.3.8. Etapa de Control Centralizado.....	112
3.3.9. Etapa de Control de Actuadores.....	113
3.3.9.1 Programación de la Etapa de Control de Actuadores .....	113
3.3.10. Etapa de Panel de Visualización.....	114

3.3.10.1 Programación del Panel de Visualización.....	116
---	-----

## **CAPÍTULO IV: IMPLEMENTACIÓN Y RESULTADOS EXPERIMENTALES**

4.1. IMPLEMENTACIÓN.....	117
4.2. TRANSMISIÓN.....	118
4.2.1. PCB del Circuito de Conectores de Alimentación.....	118
4.2.2. PCB del Circuito de Fuente de Alimentación.....	119
4.2.3. PCB del Circuito de Generación de Tonos.....	120
4.3. RECEPTOR.....	121
4.3.1. PCB del Circuito de Conectores de Alimentación.....	121
4.3.2. PCB del Circuito de Fuente de Alimentación.....	122
4.3.3. PCB del Circuito Detector de Timbre.....	123
4.3.4. PCB del Circuito de control Central de Procesamiento.....	124
4.3.5. PCB del Circuito del Panel de Visualización.....	125
4.4. RESULTADOS EXPERIMENTALES.....	127

CONCLUSIONES.

RESUMEN.

SUMMARY.

ANEXOS.

BIBLIOGRAFÍA.

## ÍNDICE DE FIGURAS

<b>Figura II.1.</b> Señal de tono telefónica.....	26
<b>Figura II.2.</b> Representación de señales de pulso.....	28
<b>Figura II.3.</b> Diagrama de pines PIC18F4550.....	34
<b>Figura II.4.</b> Diagrama de bloques PIC18F4550.....	35
<b>Figura II.5.</b> Diagrama de bloques para Interrupciones .....	37
<b>Figura II.6.</b> Diagrama de bloques Timer0 de 8 bits .....	37
<b>Figura II.7.</b> Diagrama de bloques Timer1 de 16 bits .....	38
<b>Figura II.8.</b> Diagrama de bloques A/D .....	39
<b>Figura II.9.</b> Modelo de entrada analógica.....	40
<b>Figura II.10.</b> Esquema circuital de conexión.....	43
<b>Figura II.11.</b> Conexión básica full dúplex entre PIC y PC .....	47
<b>Figura II.12.</b> Distribución de pines del PIC16F628A.....	49
<b>Figura II.13.</b> Distribución de pines del PIC16F876A.....	53
<b>Figura II.14.</b> Diagrama interno de Bloques del PIC16F876A.....	54
<b>Figura II.15.</b> Estructura de Memoria de Programa del PIC16F876A.....	55
<b>Figura II.16.</b> Diagrama de bloques de comunicación Asíncrona del PIC16F876A.....	57
<b>Figura II.17.</b> Diagrama de bloques de comunicación Asíncrona del PIC16F876A.....	59
<b>Figura II.18.</b> Diagrama eléctrico del DTMF MT8870 en modo de entrada única.....	62
<b>Figura II.19.</b> Pantalla Principal de Microcode.....	64
<b>Figura II.20.</b> Estructura de un programa modelo.....	65
<b>Figura II.21.</b> Vista Frontal de la Unidad PICKit2 .....	66
<b>Figura II.22.</b> Esquema de conexión del PICKit2.....	67
<b>Figura II.23.</b> Interface de la aplicación PICKit2.....	68
<b>Figura II.24.</b> Entorno de Trabajo Proteus.....	69

<b>Figura II.25.</b> Entorno de Trabajo del programa ISIS.....	70
<b>Figura II.26.</b> Submenús de trabajo del botón derecho del ratón.....	71
<b>Figura II.27.</b> a) Modo Componentes; b) Boton “pick”.....	71
<b>Figura II.28.</b> Ventana para la edición de componentes.....	72
<b>Figura II.29.</b> Los componentes añadidos.....	72
<b>Figura II.30.</b> Selección y Orientación del Componente.....	73
<b>Figura II.31.</b> Barra de rotación y reflexión.....	73
<b>Figura II.32.</b> Inicio de Cable.....	74
<b>Figura II.33.</b> Circuito “a mano”.....	74
<b>Figura II.34.</b> Unión eléctrica entre cables.....	74
<b>Figura II.35.</b> Modo de unión.....	74
<b>Figura II.36.</b> Entorno de ARES.....	75
<b>Figura II.37.</b> Herramientas de ARES.....	75
<b>Figura II.38.</b> Cuadro de Posicionamiento Manual.....	77
<b>Figura II.39.</b> Menú VER de ARES.....	78
<b>Figura II.40.</b> Área de visualización de coordenadas.....	78
<b>Figura II.41.</b> Vista de diseño en Ares.....	79
<b>Figura II.42.</b> Características de las Pistas.....	79
<b>Figura II.43.</b> Capa en cual diseñar.....	79
<b>Figura II.44.</b> Pistas ya terminadas.....	80
<b>Figura II.45.</b> Grosor de Pista.....	80
<b>Figura II.46.</b> Ventana de Impresión.....	81
<b>Figura III.47.</b> Diagrama de Bloques de Transmisor-Hardware.....	84
<b>Figura III.48.</b> Diagrama de Bloques de Receptor-Hardware.....	84
<b>Figura III.49.</b> Diagrama de Bloques de Visualización.....	85
<b>Figura III.50.</b> Diagrama de circuito alimentación-transmisor.....	86

<b>Figura III.51.</b> Diagrama de circuito de acoplamiento de señal.....	88
<b>Figura III.52.</b> Circuito interno del C.I. LM386 .....	89
<b>Figura III.53.</b> Circuito interno y modo de configuración del C.I. LM567 .....	90
<b>Figura III.54.</b> Esquema del Circuito Detector de Línea Atendida.....	91
<b>Figura III.55.</b> Diagrama circuital de la etapa de control central.....	92
<b>Figura III.56.</b> Diagrama de flujo del Transmisor de Alerta.....	94
<b>Figura III.57.</b> Diagrama circuital completo del sistema transmisor SAC .....	95
<b>Figura III.58.</b> Diagrama de circuito alimentación-receptor .....	96
<b>Figura III.59.</b> Diagrama circuital del detector de timbre.....	97
<b>Figura III.60.</b> Diagrama de circuito identificación de tonos. ....	99
<b>Figura III.61.</b> Gráfica de conexión para Recepción de datos.....	101
<b>Figura III.62.</b> Diagrama de Flujo para la identificación de tonos .....	102
<b>Figura III.63.</b> Diagrama de Estructura de Datos.....	102
<b>Figura III.64.</b> Diagrama de Grabación de Datos .....	103
<b>Figura III.65.</b> Diagrama de trama de Eliminación.....	103
<b>Figura III.66.</b> Diagrama de Descripción de Actualización .....	104
<b>Figura III.67.</b> Diagrama de Tramas para Visualización.....	104
<b>Figura III.68.</b> Gráfica de la Memoria EEPROM 24LC512.....	105
<b>Figura III.69.</b> Gráfica del LCD 16x4 segmentos .....	105
<b>Figura III.70.</b> Diagrama Procesos para Ingreso de Nuevo Usuario .....	107
<b>Figura III.71.</b> Diagrama Procesos para Ingreso de Nuevo Usuario.....	108
<b>Figura III.72.</b> Diagrama Procesos para Ingreso de Nuevo Usuario.....	109
<b>Figura III.73.</b> Diagrama de Procesos para Visualización de Usuarios.....	110
<b>Figura III.74.</b> Diagrama de Flujo para Visualización de Alarmas Activadas.....	111
<b>Figura III.75.</b> Gráfica de conexión para el Teclado 4x4.....	111
<b>Figura III.76.</b> Gráfica de conexión del microcontrolador 18F4550.....	112

<b>Figura III.77.</b> Gráfica del diagrama de conexión del microcontrolador 16F628A .....	113
<b>Figura III.78.</b> Diagrama de Flujo del funcionamiento de la etapa del Actuador.....	114
<b>Figura III.79.</b> Gráfica del diagrama de conexión del Panel de Visualización .....	116
<b>Figura III.80.</b> Gráfica del diagrama de Bloques del Panel de Visualización.....	116
<b>Figura IV.81.</b> Gráfica en 3D y pista del circuito de conectores de alimentación.. .....	118
<b>Figura IV.82.</b> Resultado de diseño de la plaqueta de conectores de alimentación.....	118
<b>Figura IV.83.</b> Gráfica en 3D y pista del circuito de Fuente de Alimentación .....	119
<b>Figura IV.84.</b> Gráfica resultante del circuito de Fuente de Alimentación .....	119
<b>Figura IV.85.</b> Gráfica en 3D y pista del circuito de Generación de Tonos.....	120
<b>Figura IV.86.</b> Gráfica resultante del circuito de Generación de Tonos .....	121
<b>Figura IV.87.</b> Gráfica en 3D y pista del circuito de conectores de Fuente de Alimentación... ..	121
<b>Figura IV.88.</b> Resultado del circuito de conectores de Fuente de Alimentación.....	122
<b>Figura IV.89.</b> Gráfica en 3D y pista del circuito de Fuente de Alimentación .....	122
<b>Figura IV.90.</b> Gráfica del circuito terminado de la Fuente de Alimentación.....	123
<b>Figura IV.91.</b> Gráfica en 3D y pista del Circuito Detector de Timbre .....	124
<b>Figura IV.92.</b> Gráfica en 3D y pista del Circuito Detector de Timbre .....	124
<b>Figura IV.93.</b> Gráfica en 3D y pista del Circuito de Control Central de Procesamiento.....	125
<b>Figura IV.94.</b> Diseño terminado de la plaqueta de Control Central d Procesamiento .....	125
<b>Figura IV.95.</b> Gráfica en 3D y pista del Circuito del Panel de Visualización .....	126
<b>Figura IV.96.</b> Diseño terminado del Panel de Visualización.....	126

## ÍNDICE DE TABLAS

<b>Tabla II.I.</b> Tabla de combinación de frecuencias para los correspondientes números .....	29
<b>Tabla II. II.</b> Especificación de las terminales más importantes del circuito MT8870 .....	62
<b>Tabla III.III.</b> Tabla de salidas binarias de MT8870 .....	63
<b>Tabla IV.IV.</b> Tabla de Estadística de Pruebas .....	128

## ÍNDICE DE ANEXOS

- Anexo A.** Manual de usuario
- Anexo B.** Descripción de cada pin del PIC16F876A
- Anexo C.** Datasheet del Microcontrolador PIC16F876A
- Anexo D.** Datasheet del Microcontrolador PIC16F628A
- Anexo E.** Datasheet del Microcontrolador PIC18F4550
- Anexo F.** Datasheet del CM8870
- Anexo G.** Datasheet del LM567
- Anexo H.** Datasheet del 4N25
- Anexo I.** Esquema del Sistema Completo
- Anexo J.** Código Fuente



## ÍNDICE DE ABREVIATURAS

<b>CNT:</b>	Consejo Nacional de Telecomunicaciones.
<b>UPC:</b>	Unidad de Policía Comunitaria
<b>DTMF:</b>	Dual Tone Multiplexed Frequency
<b>AD:</b>	Analógico a Digital
<b>DA:</b>	Digital a Analógico.
<b>RISC:</b>	Reduced Instruction set computer
<b>I<sup>2</sup>C:</b>	Interconexión de Circuitos Integrados
<b>EUSART:</b>	Canal de comunicación serial
<b>LCD:</b>	Display de Cristal Líquido
<b>USB:</b>	Bus Serial Universal
<b>CAD:</b>	Convertidor Analógico a Digital
<b>I/O:</b>	Entrada/Salida.
<b>PIN:</b>	Personal Identification Number
<b>PTC:</b>	Coficiente de Temperatura Negativo
<b>PCB:</b>	Print Circuit Board
<b>RAM:</b>	Memoria de acceso aleatorio.

<b>RISC:</b>	Computadores de Juego de Instrucciones Reducido.
<b>SDA:</b>	Sistema de datos
<b>SCL:</b>	Sistema de reloj
<b>GND:</b>	Masa
<b>FET:</b>	Transistor de efecto de campo
<b>USART:</b>	Universal Synchronous Asynchronous Receiver Transmitter
<b>ACK:</b>	Pulso de confirmación
<b>R/W:</b>	Lectura/Escritura
<b>LSB:</b>	Bit menos significativo
<b>DTE:</b>	Equipo terminal de datos
<b>DCE:</b>	Equipo de comunicación de datos
<b>TTL:</b>	Lógica transistor a transistor
<b>TXD:</b>	Transmisor de datos
<b>RXD:</b>	Receptor de datos
<b>MCU:</b>	Microcontrolador Microchip
<b>CCS:</b>	Custom Computer Services Incorporated.
<b>VSM:</b>	Virtual System Modelling.

## INTRODUCCIÓN

El desarrollo de las ciudades especialmente en las ciudades que son cabeceras provinciales en el Ecuador se ha realizado en un ritmo acelerado en la cual una consecuencia de esto es el crecimiento de la inseguridad en los diferentes sectores especialmente en las zonas céntricas.

Para contrarrestar este fenómeno el Gobierno ha aumentado el personal policial en estos sectores dando en ocasiones positivos resultados aunque en otros lugares a fallado por falta de colaboración de los moradores, ésta es una solución que por parte de la autoridades gubernamentales puede proporcionar a la ciudadanía, cabe destacar que cualquier sistema implementado para la solución de este problema funciona siempre y cuando haya una organización y coordinación de las partes involucradas.

El sistemas que se implemente ya sea organizacional como tecnológico que tenga como finalidad un apoyo para la seguridad ciudadana se debe establecer en lugares de mayor vulnerabilidad para tener un mejor control del sector, la coordinación de los diferentes entes que conforman la seguridad ciudadana deben tener una organización y coordinación previa.

El sistema que se implante es un sistema de alarmas en la cual realiza una llamada de alerta hacia una central que se encuentra localizado en un centro de ayuda como puede ser la policía, se basa en la llamada por medio de una línea telefónica y ubicación del lugar desde donde se realizó la llamada de emergencia.

# **CAPÍTULO I**

## **GENERALIDADES**

### **1.1. ANTECEDENTES**

El Barrio la Joya es uno de los más populares y tradicionales de la ciudad de Riobamba. Aunque no existe una demarcación precisa, los actuales directivos encabezados por el Dr. Geovanny Lema, Presidente, establecen que se extiende desde las calles 5 de junio hasta la Morona (incluye 6 calles horizontales) y desde la Olmedo hasta la Colombia (4 calles verticales).

Aún es un sector céntrico de la ciudad de Riobamba, situación que le ha afectado en el crecimiento del comercio y consiguientemente de la delincuencia lo que ha generado graves problemas de seguridad, desde hace aproximadamente 10 años. Existen algunos

moradores que inclusive expresan preocupadamente que se estaría convirtiendo en una “zona roja” debido al crecimiento de cantinas clandestinas, sitios de expendio de drogas, con lo cual la presencia de delincuentes prácticamente es cotidiana.

Los robos a domicilios, asaltos a plena luz del día, riñas callejeras, tránsito de ebrios e inclusive asesinatos, se han vuelto comunes. En una demarcación tan pequeña como la mencionada (aproximadamente 24 calles) se han producido dos asesinatos desde el año pasado.

Frente a este problema, algunos moradores de la parte baja del barrio, se han organizado e instalado una alarma rudimentaria. Sin embargo, la gran mayoría de los moradores están exentos de este sistema que de alguna manera contribuye a enfrentar a la delincuencia.

De otra parte, desde hace aproximadamente 3 años se ha instalado en el sector una unidad de la policía comunitaria integrada por 3 policías que se turnan en el servicio, siendo su aporte insuficiente ante el crecimiento delincencial, pues “no hay vecino que no haya sufrido un robo” o asalto, según manifiestan sus moradores.

Para enfrentar colectivamente la inseguridad que se vive en dicho sector, algunos moradores; por su propia iniciativa y en coordinación con la directiva del barrio, se están organizando decididamente, siendo así uno de los pioneros en impulsar desde la sociedad la lucha contra la delincuencia y la inseguridad. Hasta el momento ya han realizado dos reuniones masivas con las autoridades locales, incluidas la policía comunitaria, el Intendente de Policía e inclusive el Gobernador de la provincia. Al momento se prepara un Plan de Seguridad Ciudadana que incluye la participación de los

moradores en tareas de vigilancia mientras la Policía Comunitaria prepara un Plan de Capacitación para los moradores. En este contexto, los mismos moradores han solicitado la participación de la Escuela Superior Politécnica de Chimborazo por intermedio de la Escuela de Ingeniería Electrónica para el diseño e instalación de una Alarma Comunitaria, que será un complemento de las actividades que se realizaran en la zona.

El desarrollo del área técnica como es la Alarma Comunitaria pretende aportar al alcance del objetivo central del Sector, que es el de brindar mayor seguridad a los moradores. Cabe decir que este es un plan piloto en el tema de la vinculación de estudiantes con proyectos sociales y consiguientemente de instituciones prestigiosas como la ESPOCH para contribuir a la solución de problemas desde una perspectiva comunitaria.

## **1.2 JUSTIFICACION**

Hoy en día existen un sin número de sistemas utilizados con la finalidad de brindar seguridad a sectores barriales siendo estas de gran demanda, las cuales constan con sirenas, sistemas de monitoreo y un completo cuerpo de seguridad públicos o privados, donde los costos varían dependiendo de la complejidad y calidad de servicio que brindan, el sistema que se propone trata de aportar y ser un complemento más en la seguridad de barrios y sectores donde tienen un alto grado delincencial.

El sistema está diseñado para el aviso de una alerta en un sector específico hacia una unidad policial más cercana la cual se visualiza la dirección precisa en donde ocurre el evento.

### **1.3 OBJETIVOS**

#### **1.3.1 Objetivo General**

Diseñar e implementar un sistema de control de alarmas comunitarias para el barrio la Joya de la ciudad de Riobamba.

#### **1.3.2 Objetivos Específicos**

Estudiar y seleccionar los dispositivos que se van a utilizar para el desarrollo de la alarmar siendo estas su central y estaciones de aviso.

Analizar la forma de transmisión de señales mediante líneas y sus centrales telefónicas.

Diseñar e implementar un medio de visualización en la central localizada en la UPC (Unidad de Policía Comunitaria) para la localización rápida del lugar en donde está la estación que pide auxilio.

Programar un micro controlador para el envío de tonos por una línea telefónica.

Diseñar e implementar los circuitos electrónicos para el sistema de alarmas comunitarias.

## **CAPÍTULO II**

### **FUNDAMENTO TEÓRICO**

#### **2.1 LINEA TELEFONICA CONVENCIONAL**

La línea de teléfono presenta señales de audio que permiten identificar el proceso del estado de una comunicación telefónica. Estas señales pueden ser escuchadas a través del teléfono por el abonado.

Internamente las compañías de teléfono utilizan señales digitales para el direccionamiento y tratamiento de las llamadas telefónicas, pero esta información es solamente entre centrales telefónicas y no pueden ser vistas por el usuario normal.



Antiguamente las señales eran analógicas y algunas de ellas podían ser vistas a través del cambio de tensión en la línea, pero este sistema ya no es usado.

Las señales distintivas que se pueden escuchar con solo descolgar el teléfono, permiten al usuario determinar el estado de una llamada telefónica. Tales señales poseen un patrón común, su frecuencia. La cual es de aprox. 430 Hz, pero varían en su cadencia. También tienen un orden preestablecido, presentándose solo si determinadas condiciones ocurren y en un lapso de tiempo característico. Luego expiran y dejan de estar presentes. Las frecuencias, cadencias y duraciones de las señales varían de acuerdo a cada país.

### **2.1.1 Definiciones de señales Telefónicas**

Las empresas de teléfono informan el estado de una llamada a través de señales de audio, algunas de ellas las listamos ahora. Para más información remitirse a las normas telefónicas de su empresa, por lo general son muy similares, solo que varían sus cadencias, es decir el tiempo en que está presente el tono de 440 HZ.

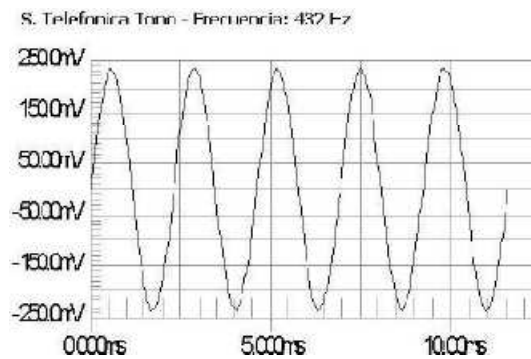
El las definiciones los valores que utilice son aproximaciones, debido que las frecuencias y duraciones de expiración de señales, varían de central a central telefónica y día a día. Según las mediciones efectuadas en un periodo de 6 meses, la frecuencia varía aproximadamente en un 7 %, del valor fijado como norma.

#### **2.1.1.1 Señal de invitación a marcar (Tono)**

Frecuencia aprox. 430 Hz +/- 20Hz

Emisión: Continua

**Duración total: 18 seg. +/- 2 seg.**



**Figura II.1.** Señal de tono telefónica

En la Figura II.1 vemos la señal de tono del teléfono, como si la estuviéramos midiendo con un osciloscopio. Observar la amplitud.

Cabe destacar que esta señal dura 18 +/- 2 seg. si no se efectúa ningún llamado (es decir no se digita ningún número), luego viene una señal de ocupado y posteriormente una señal de congestión (que combina varias frecuencias, a esta señal le llamaremos también señal de timeout), pasada estas señales el teléfono se queda mudo.

Cuando se digita un número en el teclado del teléfono (o discamos) la señal de tono desaparece, llevándose al teléfono a una “pausa interdigital” que no dura más de 20 segundos, es decir luego de marcar un número, este se debe completar con los números siguientes de lo contrario aparece la señal de ocupado, luego la de congestión y el teléfono se queda mudo.

Conviene experimentar con el teléfono midiendo los tiempos de cada señal y cuáles son las señales que le preceden.

### **2.1.1.2 Señal de ocupado**

Frecuencia aprox. 430 Hz +/- 20Hz

Emisión: 410 ms

Silencio: 270 ms

Duración total: 20 seg.

Esta señal está presente en varias ocasiones del llamado telefónico, pero en especial cuando el teléfono al que llamamos da ocupado.

### **2.1.1.3 Señal de espera a respuesta**

Frecuencia aprox. 430 Hz +/- 20Hz

Emisión: 1,1 seg.

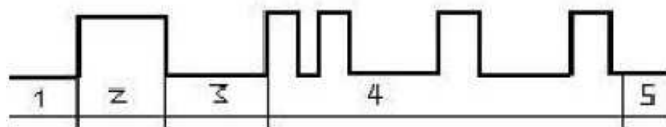
Silencio: 4 seg.

Duración total: 2 minutos

Esta señal aparece cuando al teléfono que llamamos está disponible, pero el usuario no ha contestado. Existen otras señales de información, pero que no son utilizadas por el circuito para su funcionamiento, incluso varían en frecuencia, llegando a tener más de una frecuencia a la vez (señales compuestas).

### 2.1.2 Desarrollo normal de una comunicación

Una vez identificadas las señales es posible, transformarlas en una secuencia de bits, que estarán presentes solo cuando aparece la frecuencia de 430 Hz.



**Figura II.2.** Representación de señales de pulso.

En la figura II.2. se muestra los pulsos o “UNO” lógico en donde representa la presencia de la señal telefónica de aproximadamente 430 Hz presente al descolgar el teléfono y durante otras etapas de la comunicación, como la señal de “espera a respuesta” o la señal de “ocupado”, pero con distintas cadencias. En esta secuencia de bits se muestra el proceso desde que el usuario levanta el teléfono hasta que el destinatario de la llamada contesta el teléfono.

DETALLAMIENTO:

- 1) Teléfono permanece colgado.
- 2) Tono de “invitación a marcar”.
- 3) Marcado del número de destino.
- 4) Señal de respuesta: 3 Opciones
  - a) Ocupado
  - b) Operadora

c) Espera a respuesta (Como se muestra en la Figura II.2.)

5) Time out o respuesta por parte del abonado (Como se muestra en la Figura II.2)

Luego de analizar esta secuencia de bits, notamos que el patrón de bits (que representan la presencia de la señal de 430 Hz) están presentes en la mayoría de las comunicaciones telefónicas, ya sea que de ocupado o espera de respuesta. Excepto cuando atiende la operadora o la persona que llamamos.

Por lo tanto es necesario realizar un circuito que detecte o “escuche” la presencia de estos bits en el paso numero 4) del proceso de llamado telefónico. Cuando los bits dejen de estar presentes, es porque la persona a la que llamamos atendió u ocurrió un time out.

### 2.1.3. Tonos DTMF (*Dual Tone Multiplexed Frequency* –frecuencia de dos tonos multiplexados–)

Utiliza un método bastante peculiar. Para que la central telefónica pueda detectar cada cifra marcada, lo que se envía es una combinación de dos señales senoidales de frecuencias muy determinadas, de entre siete frecuencias puras distintas. Cada cifra (y otros signos especiales) corresponde a una combinación de dos de esas frecuencias, según la Tabla II.1.

	1209Hz	1336Hz	1477Hz
697Hz	1	2	3
770Hz	4	5	6
852Hz	7	8	9
941Hz	*	0	#

**Tabla II.I** Tabla de combinación de frecuencias para los correspondientes números

Al combinar dos de estas señales senoidales obtenemos otra señal más compleja que es la que viaja por la línea telefónica. Luego, la central telefónica es capaz de volver a separar ambas señales y determinar su frecuencia, averiguando cual fue la cifra marcada.

## **2.2 MICROCONTROLADORES**

Hasta antes de la aparición de los microprocesadores (1971), la mayor parte de las aplicaciones digitales en electrónica se basaban en la llamada lógica cableada, es decir, si existía un problema este era analizado y se sintetizaba una función en base a la lógica de boole que era la solución al problema planteado.

Con la aparición de los microprocesadores, se varió el esquema de diseño de tal forma que un problema era descompuesto en una serie de tareas más simples, el microprocesador ejecutaba una serie de pasos o instrucciones para llevar a efecto cada una de las tareas, en ocasiones no era necesario volver a armar un circuito para solucionar otro problema sino que se cambiaba las instrucciones (programa) para obtener otra aplicación.

Desde luego el microprocesador es como el cerebro que ejecuta operaciones de índole aritméticas y lógicas por tanto no manejaba líneas externas (periféricos) más aún tampoco tenía un lugar donde almacenar el programa y los datos que necesitaba el programa para encontrar la respuesta al problema. El microprocesador buscaba una instrucción y la ejecutaba; al conjunto de circuitos (hardware) que daban el soporte necesario al microprocesador se le llamo sistema mínimo.

Con el pasar de los años el sistema mínimo se convirtió en un estándar, por otro lado la escala de integración mejoro y posibilito (1976) sintetizar en un solo chip un sistema mínimo, al cual se le llamo SISTEMA A que no era otra cosa que el primer microcontrolador.

En consecuencia definimos así a un microcontrolador; como un procesador con su sistema mínimo en un chip (incluye memoria para programa y datos, periféricos de entrada / salida, conversores de AD y DA, módulos especializados en la transmisión y recepción de datos). Desde luego que hay diferencias sustanciales como la arquitectura cerrada de un microcontrolador, en cambio en un microprocesador es abierta, podemos sumar nuevos dispositivos en hardware en función ha las necesidades que la aplicación demande.

Otra diferencia entre los microcontroladores y los microprocesadores es que los primeros cuentan con un set de instrucciones reducido en cambio la mayoría de los microprocesadores tienen mayor cantidad de instrucciones. Por otro lado la mayoría de los microcontroladores posee una arquitectura Harvard frente a una arquitectura Von Neuman de los microprocesadores.

Los microcontroladores se especializan en aplicaciones industriales para resolver problemas planteados específicos por ejemplo: los encontramos en los teclados o mouse de las computadoras, son el cerebro de electrodomésticos, también los encontramos en la industria automotriz, en el procesamiento de imagen y video.

Cabe señalar que los el aumento progresivo de la escala de integración y las técnicas de fabricación hacen que cada vez aparezcan microcontroladores mas poderosos y rápidos.

### 2.2.1 EL MICROCONTROLADOR PIC18F4550

El PIC 18F4550, pertenece a los microcontroladores PIC18 de gama alta. Posee una arquitectura RISC (reduced instruction set computer) de 16 bits longitud de instrucciones y 8 bits de datos.

Partiendo sobre todo que es un micro ampliamente utilizado como un microcontrolador “estándar” debido a sus innumerables características y potencia tiene incluido una memoria Flash USB y control de flujo de datos. Soporta USB low speed (1.5Mb/s) y full speed (12Mb/s) y USB V2.0, el cual es un atractivo complemento el poder incorporar por sí mismo una interfaz USB.

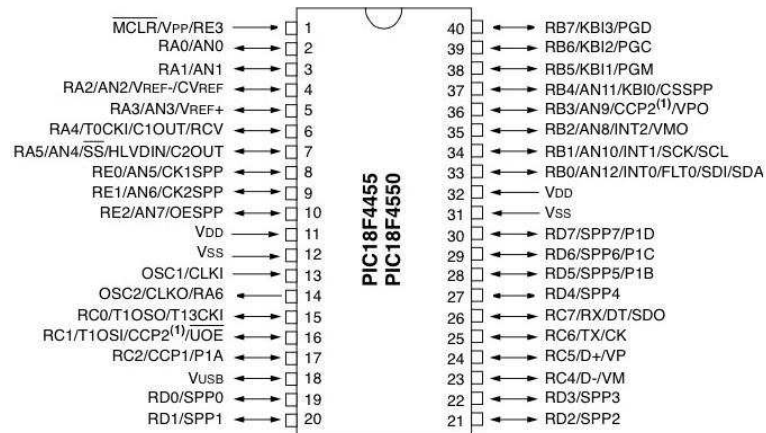
#### 2.2.1.1 Características Principales

Memoria Flash:	32Kbytes
Máximo número de instrucciones simples:	16384
Memoria SRAM:	2048 bytes
Memoria EEPROM:	256 bytes
Entradas / Salidas:	35
Número de entradas A/D:	13



Número de CCP:	1
Número de ECCP:	1
Soporta SPP:	Si
Soporta SPI:	Si
Soporta master I2C:	Si
Número de EAUSART:	1
Número de comparadores:	2
Número de temporizadores de 8 bits:	1
Número de temporizadores de 16 bits:	3
Universal Serial Bus (USB) module:	Si

Entre sus principales características destaca sobre todo su gran diversidad, con lo que se puede implementar muchas aplicaciones con tan sólo tener una buena distribución de sus entradas y salidas como se puede ver en la Figura II.3.



**Figura II.3.** Diagrama de pines PIC18F4550

La distribución de los registros del PIC18F4550 se puede visualizar en la Figura II.4. así como la longitud de cada registro y su correspondiente enlace de cada registro.

El bus de cada registro de los pines de entrada y salida están conectadas al bus de datos y así distribuido a los diferentes registros.

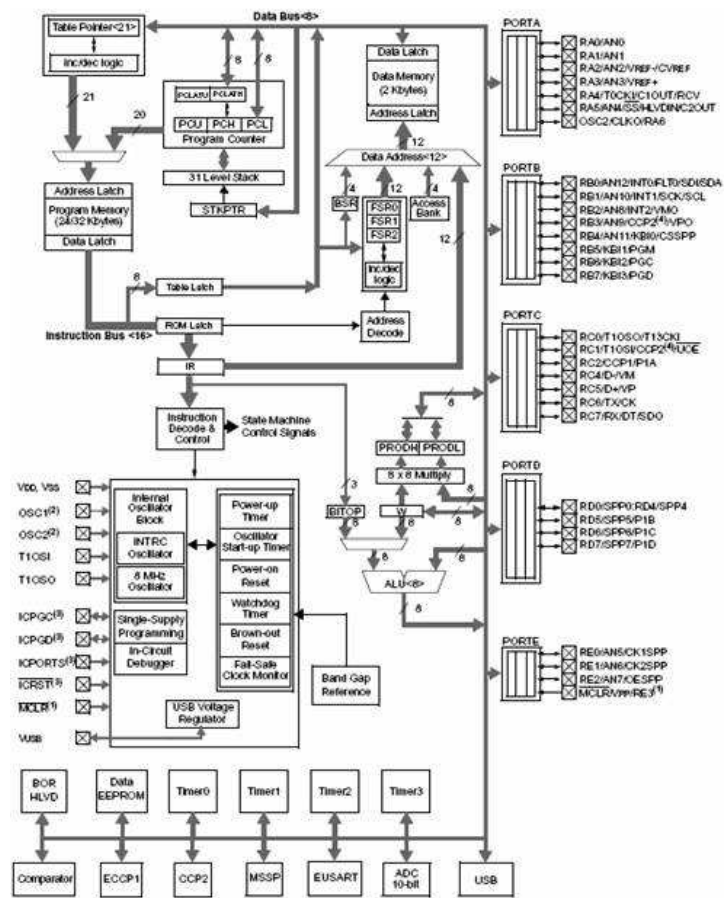


Figura II.4. Diagrama de bloques PIC18F4550

Microchip distribuye de forma general dos tipos de micros dependiendo del voltaje de alimentación:

Clase F: Voltaje típico (4.2 V a 5.5V)

Clase LF: Bajo voltaje (2.0 V a 5.5V)

Estos son exactamente iguales sólo que los micros de clase LF puedes ser usados con la nueva alimentación de 3.3V que actualmente y poco a poco se está imponiendo a los típicos 5V.

### **2.2.1.2 Interrupción**

Las interrupciones son tareas programas que el micro realiza cuando el flanco de interrupción se activa, con lo que el micro deja el programa principal y accede a una parte reservada de la memoria que se llama rutina de interrupción, donde una vez acabada la rutina de interrupción, se baja el flanco de interrupción que lo ha provocado y el micro continua el programa principal donde lo había dejado antes de ir a la rutina de interrupción. Una manera fácil de entenderlo se muestra un diagrama de bloques en la Figura II.5.

Las interrupciones en el micro pueden darse de varios tipos:

- Interrupciones externas.
- Interrupciones por desbordamiento del contador.
- Interrupciones de EUSART.
- Interrupciones USB.
- Interrupciones del CAD.
- Interrupciones por periféricos externos.

El micro puede tener varias interrupciones programadas a la vez, pero hay que tener en cuenta que una vez entra en una rutina de interrupción, el micro no puede acceder a otra interrupción hasta que la rutina de interrupción que se está ejecutando finalice. En el caso de que saltasen a la vez 2 o más interrupciones, el micro accedería aleatoriamente a una de ellas, es por ello que suele darse prioridad a las interrupciones si tenemos alguna rutina de interrupción más importantes que otras.

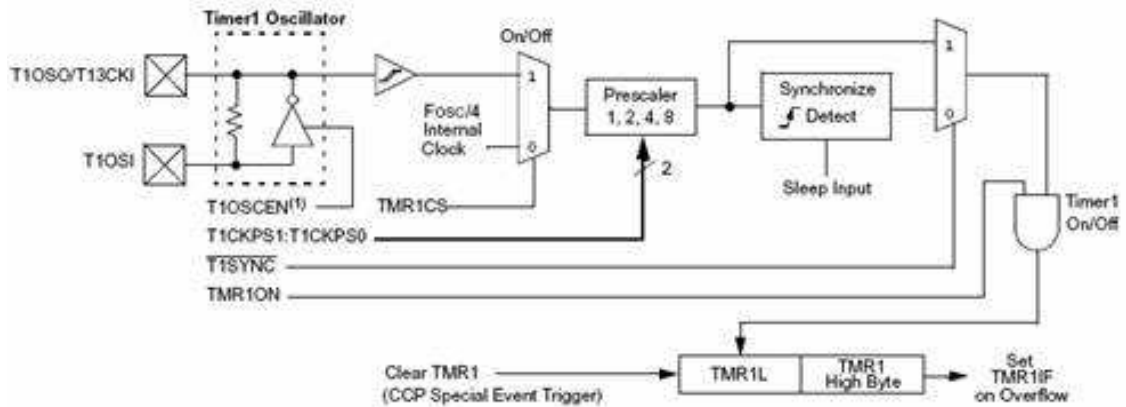


Figura II.5. Diagrama de bloques para Interrupciones

### 2.2.1.3 Temporizador

Los temporizadores son contadores que al activarlos empiezan una cuenta y cuando esta cuenta se acaba se activa el flanco de interrupción por el temporizador, entrando el micro en la rutina de interrupción del temporizador, esto se puede ver en la Figura II.6.

El PIC18F4550 tiene 4 temporizadores, de los cuales 1 de ellos es de 8 bits y el resto de una precisión de 16 bits.

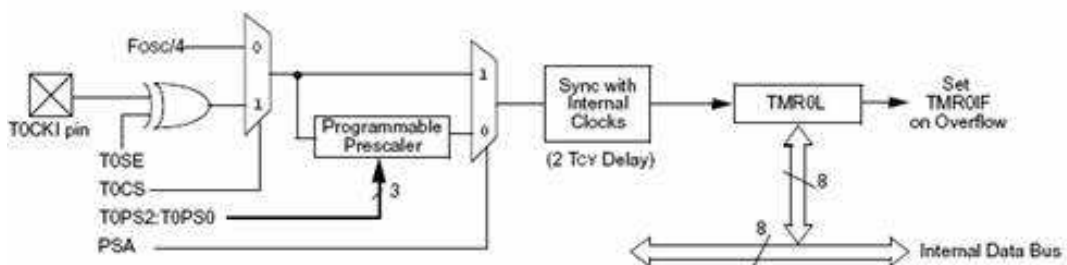
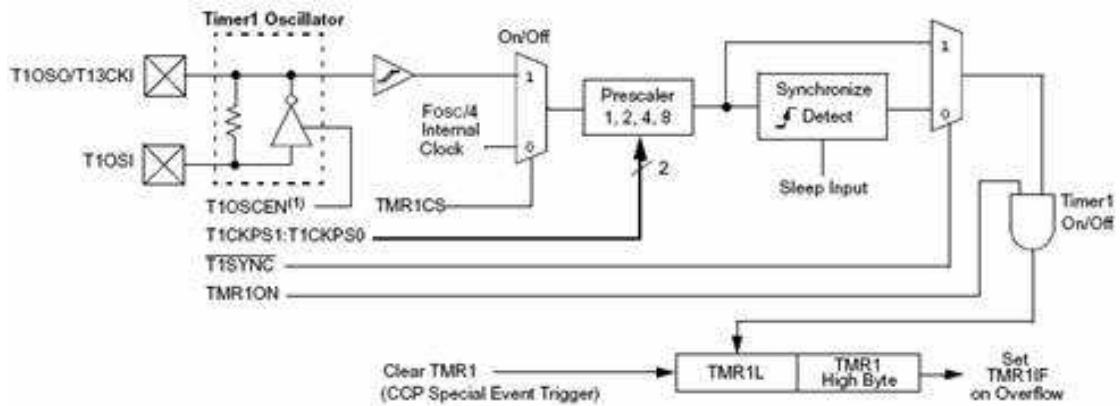


Figura II.6. Diagrama de bloques Timer0 de 8 bits

Existe la posibilidad de activar un preescaler en los temporizadores de forma que se pueda “alargar” la duración del temporizador, dependiendo del temporizador puede ser de 2,4,8 e incluso 16.



**Figura II.7.** Diagrama de bloques Timer1 de 16 bits

Resolución de los temporizadores:

Timer0 -> Temporizador configurable de 8 ó 16 bits.

Timer1 -> Temporizador de 16 bits.

Timer2 -> Temporizador de 8 bits.

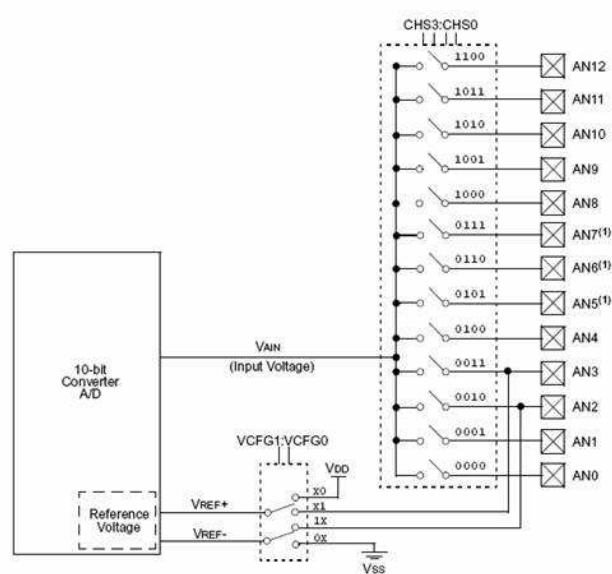
Timer3 -> Temporizador de 16 bits.

Esto se puede ver en la Figura II.7.

#### **2.2.1.4 Convertidor analógico/digital**

Microchip PIC18F4550 contiene 13 convertidores analógicos digitales como se ve en la Figura II.8, los cuales pueden ser seleccionados en modos de resolución de 8 ó 10 bits,

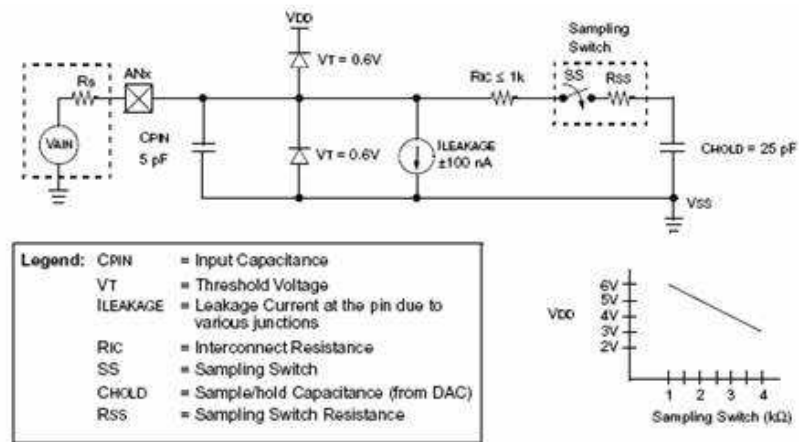
para ello antes habrá que configurar las entradas en modo CAD, pues ya que estas están por defecto como I/O.



**Figura II.8.** Diagrama de bloques A/D

Otra de sus características es que tiene dos patillas de referencia donde podemos dar la tensión de referencia para todas o algunas de las entradas del CAD.

Otra posibilidad es configurar el tiempo de adquisición de datos, pues tiene un registro habilitado especialmente para ello ya que en algunas ocasiones hay que esperar al interruptor de muestro se cierre y que el condensador (Chold) se descargue para poder hacer otra adquisición.



**Figura II.9.** Modelo de entrada analógica

Se recomienda que la máxima resistencia de entrada ( $R_s$ ) sea de 2.5K, pues sino la conversión no sería del todo fiable, por lo que habría que hacer una adaptación de impedancias entre las partes, un modelo de este circuito lo podemos ver en la Figura II.9.

El microcontrolador PIC18F4550 dispone de las siguientes memorias:

**Memoria de programa:** memoria flash interna de 32.768 bytes. Almacena instrucciones y constantes/datos, puede ser escrita/leída mediante un programador externo o durante la ejecución programa mediante unos punteros.

**Memoria RAM de datos:** Memoria SRAM interna de 2048 bytes en la que están incluidos los registros de función especial. Almacena datos de forma temporal durante la ejecución del programa. Puede ser escrita/leída en tiempo de ejecución mediante diversas instrucciones.



**Memoria EEPROM de datos:** Memoria no volátil de 256 bytes. Almacena datos que se deben conservar aun en ausencia de tensión de alimentación. Puede ser escrita/leída en tiempo de ejecución a través de registros.

**Pila:** bloque de 31 palabras de 21 bits. Almacena la dirección de la instrucción que debe ser ejecutada después de una interrupción o subrutina.

**Memoria de configuración:** memoria en la que se incluyen los bits de configuración (12 bytes de memoria flash) y los registros de identificación (2 bytes de memoria de solo lectura). Se trata de un bloque de memoria situado a partir de la posición 30000H de memoria de programa (más allá de la zona de memoria de programa de usuario). En esta memoria de configuración se incluyen:

**Bits de configuración:** contenidos en 12 bytes de memoria flash permiten la configuración de algunas opciones del microcontrolador como:

- Opciones del oscilador.
- Opciones de reset.
- Opciones del watchdog.
- Opciones de la circuitería de depuración y programación.
- Opciones de protección contra escritura de memoria de programa y memoria EEPROM de datos.

Estos bits se configuran generalmente durante la programación del microcontrolador, aunque también pueden ser leídos y modificados durante la ejecución del programa.

**Registros de identificación:** se trata de dos registros situados en las direcciones

3FFFFEH y 3FFFFFH que contienen información del modelo y revisión del dispositivo.

Son registros de solo lectura y no pueden ser modificados por el usuario.

### **2.2.1.5 Comunicación serie I<sup>2</sup>C con 18f4550**

Diseñado por Philips, este sistema de intercambio de información a través de tan solo dos cables permite a circuitos integrados y módulos OEM interactuar entre sí a velocidades relativamente lentas. Emplea comunicación serie, utilizando un conductor para manejar el timing (pulsos de reloj) y otro para intercambiar datos.

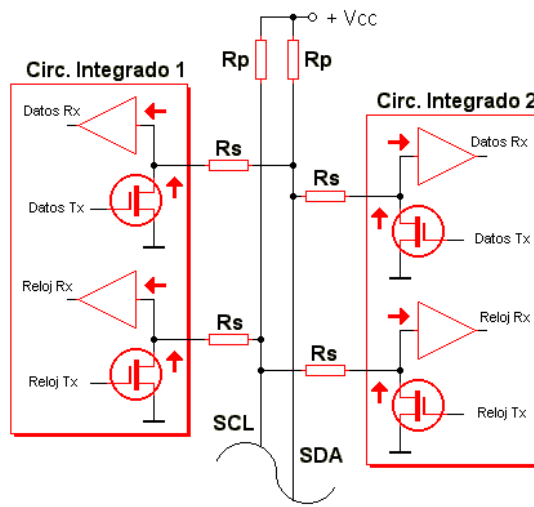
Este bus se basa en tres señales:

**SDA (System Data)** por la cual viajan los datos entre los dispositivos.

**SCL (System Clock)** por la cual transitan los pulsos de reloj que sincronizan el sistema.

**GND (Masa)** Interconectada entre todos los dispositivos "enganchados" al bus.

Las líneas SDA y SCL son del tipo drenador abierto, similares a las de colector abierto pero asociadas a un transistor de efecto de campo (ó FET). Se deben poner en estado alto (conectar a la alimentación por medio de resistores Pull-Up) para construir una estructura de bus tal que se permita conectar en paralelo múltiples entradas y salidas, la muestra de todo lo hablado tenemos en la Figura II.10.



**Figura II.10.** Esquema circuital de conexión

En el diagrama de la Figura II.10 se observa la configuración eléctrica básica del bus. Las dos líneas de comunicación disponen de niveles lógicos altos cuando están inactivas. Inicialmente el número de dispositivos que se puede conectar al bus es ilimitado, pero obsérvese que las líneas tienen una especificación máxima de 400pF en lo que respecta a capacidad de carga. La máxima velocidad de transmisión de datos que se puede obtener es de aproximadamente 100Kbits por segundo.

Las definiciones o términos utilizados en relación con las funciones del bus I<sup>2</sup>C son las siguientes:

**Maestro (Master):** Dispositivo que determina la temporización y la dirección del tráfico de datos en el bus. Es el único que aplica los pulsos de reloj en la línea SCL.

Cuando se conectan varios dispositivos maestros a un mismo bus la configuración obtenida se denomina "multi-maestro".

**Esclavo (Slave):** Cualquier dispositivo conectado al bus incapaz de generar pulsos de reloj. Reciben señales de comando y de reloj proveniente del dispositivo maestro.

**Bus Desocupado (Bus Free):** Estado en el cual ambas líneas (SDA y SCL) están inactivas, presentando un estado lógico alto. Únicamente en este momento es cuando un dispositivo maestro puede comenzar a hacer uso del bus.

**Comienzo (Start):** Sucede cuando un dispositivo maestro hace ocupación del bus, generando esta condición. La línea de datos (SDA) toma un estado bajo mientras que la línea de reloj (SCL) permanece alta.

**Parada (Stop):** Un dispositivo maestro puede generar esta condición dejando libre el bus. La línea de datos toma un estado lógico alto mientras que la de reloj permanece también en ese estado.

**Dato Válido (Valid Data):** Sucede cuando un dato presente en la línea SDA es estable mientras la línea SCL está a nivel lógico alto.

**Formato de Datos (Data Format):** La transmisión de datos a través de este bus consta de 8 bits de datos (ó 1 byte). A cada byte le sigue un noveno pulso de reloj durante el cual el dispositivo receptor del byte debe generar un pulso de reconocimiento, conocido como ACK (del inglés Acknowledge). Esto se logra situando la línea de datos a un nivel lógico bajo mientras transcurre el noveno pulso de reloj.

**Dirección (Address):** Cada dispositivo diseñado para funcionar en este bus dispone de su propia y única dirección de acceso, que viene pre-establecida por el fabricante. Hay dispositivos que permiten establecer externamente parte de la dirección de acceso. Esto permite que una serie del mismo tipo de dispositivos se puedan conectar en un mismo bus sin problemas de identificación. La dirección 00 es la denominada "de acceso general", por la cual responden todos los dispositivos conectados al bus.

**Lectura/Escritura (Bit R/W):** Cada dispositivo dispone de una dirección de 7 bits. El octavo bit (el menos significativo ó LSB) enviado durante la operación de direccionamiento corresponde al bit que indica el tipo de operación a realizar. Si este bit es alto el dispositivo maestro lee información proveniente de un dispositivo esclavo. En cambio, si este bit fuese bajo el dispositivo maestro escribe información en un dispositivo esclavo.

#### **2.2.1.6 Protocolo del Bus**

Como es lógico, para iniciar una comunicación entre dispositivos conectados al bus I<sup>2</sup>C se debe respetar un protocolo. Tan pronto como el bus esté libre, un dispositivo maestro puede ocuparlo generando una condición de inicio. El primer byte transmitido después de la condición de inicio contiene los siete bits que componen la dirección del dispositivo de destino seleccionado y un octavo bit correspondiente a la operación deseada (lectura o escritura). Si el dispositivo cuya dirección se apuntó en los siete bits está presente en el bus éste responde enviando el pulso de reconocimiento ó ACK. Seguidamente puede comenzar el intercambio de información entre los dispositivos.

Cuando la señal R/W está previamente a nivel lógico bajo, el dispositivo maestro envía datos al dispositivo esclavo hasta que deja de recibir los pulsos de reconocimiento, o hasta que se hayan transmitido todos los datos.

En el caso contrario, es decir cuando la señal R/W estaba a nivel lógico alto, el dispositivo maestro genera pulsos de reloj durante los cuales el dispositivo esclavo puede enviar datos. Luego de cada byte recibido el dispositivo maestro (que en este momento está recibiendo datos) genera un pulso de reconocimiento.

El dispositivo maestro puede dejar libre el bus generando una condición de parada (Stop). Si se desea seguir transmitiendo, el dispositivo maestro puede generar otra condición de inicio el lugar de una condición de parada. Esta nueva condición de inicio se denomina "inicio repetitivo" y se puede emplear para direccionar un dispositivo esclavo diferente ó para alterar el estado del bit de lectura/escritura (R/W).

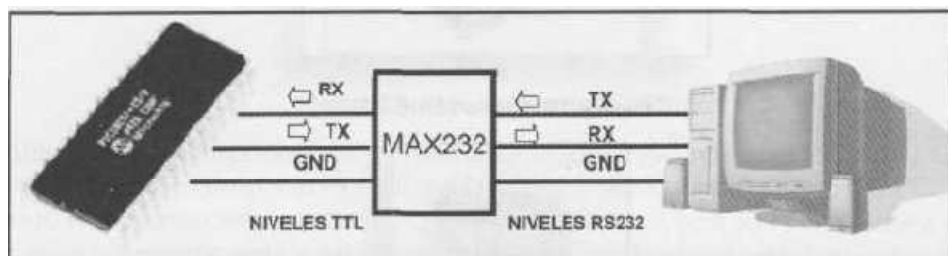
#### **2.2.1.7 Comunicación serie RS-232 con 18f4550**

La norma RS232 es la más habitual en la comunicación serie. Básicamente comunica un equipo de datos (DTE o Data Terminal Equipment) y el equipo de comunicación de datos (DCE o Data Communications Equipment).

Las características eléctricas de la señal en esta norma establecen que la longitud máxima entre el DTE y el DCE no debe ser superior a los 15 metros y la velocidad máxima de transmisión es de 20.000 bps. Los niveles lógicos no son compatibles TTL, deben situarse dentro de los siguientes rangos: 1 lógico entre -3V y -15V y 0 lógico

entre +3V y +15V. Se utilizan conectores de 25 patillas (DB 25) o de 9 patillas (DB 9) siendo asignado el conector macho al DTE y el conector hembra al DCE.

Para la comunicación full dúplex desde el USART del PIC, se debe conectar un mínimo número de señales, TXD y RXD así como la masa (GND). Los PIC utilizan señal TTL en el módulo USART por lo que se debe utilizar conversores de nivel a RS232, como el MAX 232, esto se ve en la Figura II.11.



**Figura II.11.** Conexión básica full dúplex entre PIC y PC

### 2.2.2 EL MICROCONTROLADOR PIC16F628A

El PIC 16F628 incorpora tres características importantes que son:

- Procesador tipo RISC (Procesador con un Conjunto Reducido de Instrucciones)
- Procesador segmentado.
- Arquitectura HARVARD

Con estos recursos el PIC es capaz de ejecutar instrucciones solamente en un ciclo de instrucción. Con la estructura segmentada se pueden realizar simultáneamente las dos

fases en que se descompone cada instrucción, ejecución de la instrucción y búsqueda de la siguiente.

La separación de los dos tipos de memoria son los pilares de la arquitectura Harvard, esto permite acceder en forma simultánea e independiente a la memoria de datos y a la de instrucciones. El tener memorias separadas permite que cada una tenga el ancho y tamaño más adecuado. Así en el PIC 16F628 el ancho de los datos es de un byte, mientras que la de las instrucciones es de 14 bits.

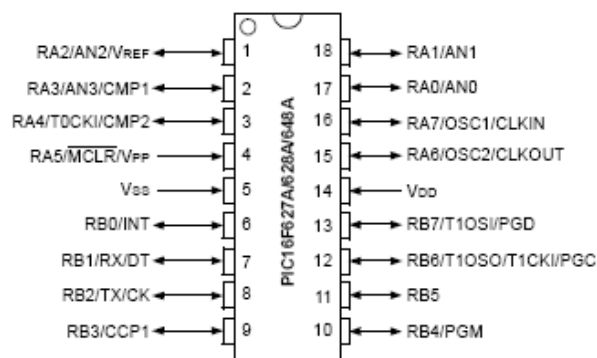
#### **2.2.2.1 Características principales**

- Conjunto reducido de instrucciones (RISC). Solamente 35 instrucciones que aprender a utilizar.
- Oscilador interno de 4MHz.
- Las instrucciones se ejecutan en un sólo ciclo de máquina excepto los saltos (*goto* y *call*), que requieren 2 ciclos. Aquí hay que especificar que un ciclo de máquina se lleva 4 ciclos de reloj, si se utiliza el reloj interno de 4MHz, los ciclos de máquina se realizarán con una frecuencia de 1MHz, es decir que cada instrucción se ejecutará en 1µs (microsegundo).
- Opera con una frecuencia de reloj de hasta 20 MHz (ciclo de máquina de 200 ns).
- Memoria de programa: 2048 locaciones de 14 bits.
- Memoria de datos: Memoria RAM de 224 bytes (8 bits por registro).
- Memoria EEPROM: 128 bytes (8 bits por registro).
- Stack de 8 niveles.



- 16 Terminales de I/O que soportan corrientes de hasta 25 mA.
- 3 Temporizadores.
- Módulos de comunicación serie, comparadores, PWM.
- Otra característica de los PICs es el manejo de los bancos de registros. En línea general, los registros se clasifican como de uso general (GPR) y de uso específico o de funciones especiales (SFR).
- Los registros de uso general pueden ser usados directamente por el usuario, sin existir restricciones. Pueden servir para almacenar resultados que se reciben desde el registro W (acumulador), datos que provienen de las puertas de entradas, etc.
- Los registros de uso específicos no pueden ser usados directamente por el usuario. Estos registros controlan prácticamente todo el funcionamiento del microcontrolador, pues toda la configuración necesaria para funcionamiento del microcontrolador es hecho a través de algún tipo de SFR.

### 2.2.2.2 Pines de I/O (Entrada/Salida)



**Figura II.12.** Distribución de pines del PIC16F628A

Todos los pines de la Figura II.12. se define posteriormente para tener una idea más global del funcionamiento y capacidades de este microcontrolador.

*PORTA: RA0-RA7:*

- Los pines RA0-RA4 y RA6-RA7 son bidireccionales y manejan señales TTL.
- El pin RA5 es una entrada Schmitt Trigger que sirve también para entrar en el modo de programación cuando se aplica una tensión igual a  $V_{pp}$  (13,4V mínimo).
- El terminal RA4 puede configurarse como reloj de entrada para el contador TMR0
- Los pines RA0-RA3 sirven de entrada para el comparador analógico.

*PORTB: RB0-RB7:*

- Los pines RB0-RB7 son bidireccionales y manejan señales TTL.
- Por software se pueden activar las resistencias de pull-up internas, que evitan el uso de resistencias externas en caso de que los terminales se utilicen como entrada (permite, en algunos casos, reducir el número de componentes externos).
- El pin RB0 se puede utilizar como entrada de pulsos para provocar una interrupción externa.

- Los pines RB4-RB7 están diseñados para detectar una interrupción por cambio de estado. Esta interrupción puede utilizarse para controlar un teclado matricial, por poner un ejemplo.

Otros pines:

- VDD: Pin de alimentación positiva. De 2 a 5,5 Vcc.
- VSS: Pin de alimentación negativa. Se conecta a tierra o a 0 Vcc.
- MCLR: Master Clear (Reset). Si el nivel lógico de este terminal es bajo (0 Vcc), el microcontrolador permanece inactivo. Este Reset se controla mediante la palabra de configuración del PIC.
- OSC1/CLKIN: Entrada de oscilador externo.
- OSC2/CLKOUT: Salida del oscilador. El PIC 16F628 dependiendo de cómo se configure puede proporcionar una salida de reloj por medio de este pin.

### **2.2.3. EL MICROCONTROLADOR PIC16F876A**

Este microcontrolador tiene como características principales:

- CPU tipo RISC de altas prestaciones.
- Repertorio de 35 instrucciones de una palabra.
- Todas las instrucciones son de un único ciclo, excepto las de salto, que llevan dos.
- Velocidad de trabajo de 20 Mhz, con un ciclo de instrucción de 200 ns.
- Memoria de programa tipo flash de 8 Kpalabras.
- Memoria de datos de 368 bytes.
- Memoria EEPROM de datos de 256 bytes.

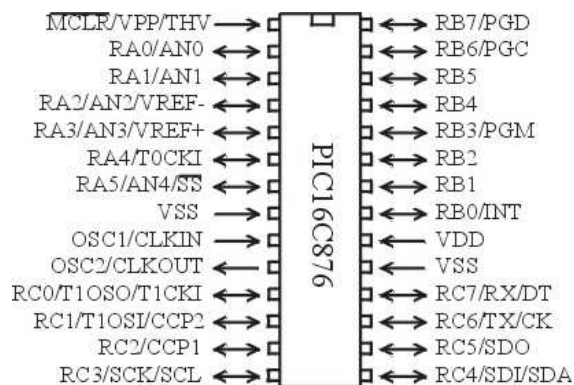
- Patillaje compatible con PIC16C73B/74B/76/77.
- Hasta 14 fuentes de interrupción.
- Pila por hardware de 8 niveles.
- Modos de direccionamiento directo, indirecto y relativo.
- Reset de conexión (POR).
- Temporización de conexión y temporización de inicio de oscilación.
- Circuito supervisor (watchdog).
- Código de protección programable.
- Tecnología de alta velocidad y bajo consumo en la memoria CMOS EEPROM/FLASH.
- Programación in-situ vía serie con dos patillas.
- Posibilidad de programación in-situ, vía serie, mediante tensión simple de 5 voltios.
- Acceso para lectura o escritura a la memoria de programa.
- Gran margen de alimentación entre 2 y 5,5 voltios.
- Corriente de salida de 25 mA.
- Bajo consumo:
  - Menor de 2 mA a 5V y 4 Mhz.
  - 20  $\mu$ A a 3V y 32 Khz.
  - Menor de 1  $\mu$ A en reposo.

Prestaciones de periféricos que posee este PIC son:

- Timer0: Temporizador-Contador de 8 bits, con Predivisor también de 8 bits.
- Timer1: Temporizador-Contador de 16 bits con Predivisor, que puede trabajar con reloj externo en el modo reposo (sleep).

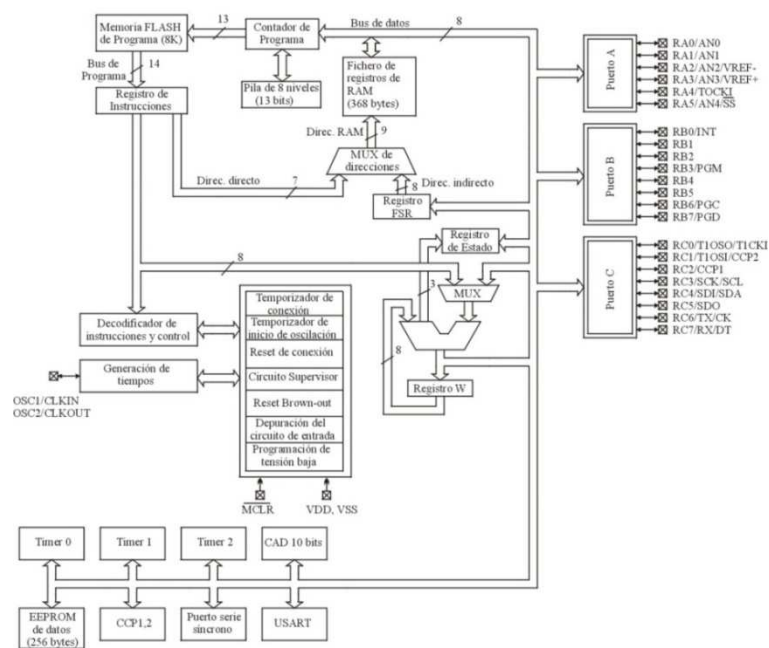
- Timer2: Temporizador-Contador de 8 bits con registro de período de la misma longitud, con Predivisor y Postdivisor.
- Dos módulos de Captura y Comparación y uno PWM (modulación por ancho de impulso:
  - La captura es de 16 bits, con resolución máxima de 12,5 ns.
  - La comparación es de 16 bits, con resolución máxima de 200 ns.
  - El bloque PWM tiene una resolución máxima de 10 bits.
- Convertidor multicanal analógico digital de 10 bits.
- Puerto serie síncrono (SSP), con modo Maestro (SPI) e I<sup>2</sup>C (maestro/servidor).
- Transmisor Receptor Universal Síncrono Asíncrono (USART/SCI) con detección de 9 bits y de dirección.
- Circuito de detección de BROWN-OUT (bajada de tensión).

Este microcontrolador consta de 28 pines tal y como se muestra en la Figura II.13.



**Figura II.13.** Distribución de pines del PIC16F876A

En lo que se refiere al diagrama interno de este microcontrolador se presenta a continuación en la Figura II.14. en la cual se muestra la forma y distribución de los bloques de datos, sus registros internos con las que cuenta, longitud de bits con el que cuenta cada trama de datos en los diversos bloques, la conexión y vinculación interna de los registros con los pines de entrada, salida, de control.



**Figura II.14.** Diagrama interno de Bloques del PIC16F876A

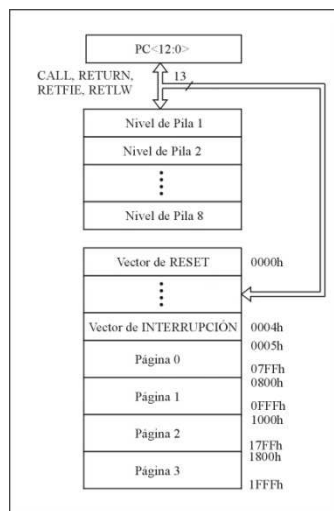
### 2.2.3.1. Organización de la memoria

Existen tres bloques de memoria: la Memoria de Programa, la Memoria de Datos y la EEPROM de datos. Las dos primeras poseen buses separados, lo que permite el acceso simultáneo.

### 2.2.3.2. Organización de la memoria de programa

El PIC16F876 tiene un Contador de Programa (PC) de 13 bits, capaz de direccionar una memoria de 8K palabras, siendo cada palabra de una longitud de 14 bits.

El vector de Reset se encuentra en la dirección 0000h, queriendo esto decir que tras un reset al dispositivo el PC se cargará con esa cantidad. El vector de la interrupción se encuentra situado en la dirección 0004h. En la Figura II.15. se representa esquemáticamente el mapa de memoria del chip.



**Figura II.15.** Estructura de Memoria de Programa del PIC16F876A

La Memoria de Datos se fracciona en cuatro bancos: banco 0, banco 1, banco 2 y banco 3.

Cada banco puede seleccionarse con los bits RP0 y RP1 del Registro de Funciones Especiales (SFR).

Cada banco tiene una extensión de 128 bytes (7Fh). Las posiciones más bajas de cada banco se reservan para el registro SFR. Tanto SFR como el GPR (Registro de Propósito General) se implementan como una RAM estática. Los registros que son muy utilizados dentro del SFR, se duplican en otros bancos para evitar los continuos cambios de éstos en los programas.

### **2.2.3.3. Fichero de Registro de propósito general**

Al fichero de registros de propósito general se puede acceder directamente o indirectamente a través del Fichero de Selección de Registros FSR.

### **2.2.3.4. Lectura de la memoria EEPROM de datos**

Para leer una posición de la memoria de datos, el usuario debe escribir en el registro EEADR la dirección deseada, borrar el bit de control EEPGD (EECON1<7>) y, entonces, activar el bit de control RD (EECON1<0>). El dato estará disponible, en el próximo ciclo de instrucción, en el registro EEDATA, por consiguiente podrá leerse en la próxima instrucción. EEDATA mantendrá este valor, hasta que se efectúe otra operación de lectura, o sea modificado expresamente por el programador.

### **2.2.3.5. Escritura de la memoria EEPROM de datos**

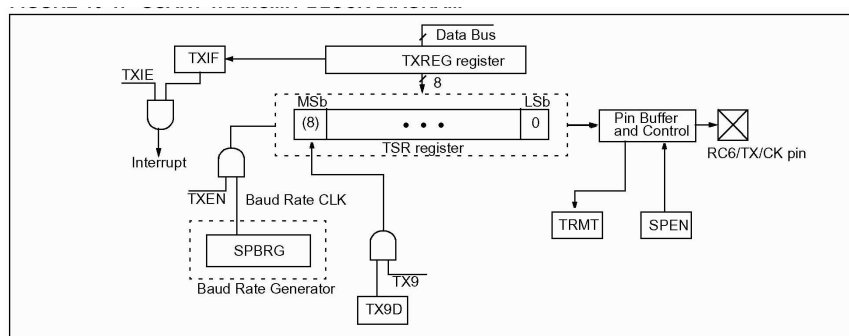
Para escribir en una posición de la memoria EEPROM de datos, primero hay que colocar la dirección en el registro EEADR, y después el dato a escribir, en el registro EEDATA.

### **2.2.3.6. USART en modo Asíncrono**

En este modo, la USART emplea el formato estándar “no retorno a cero” (NRZ) (un bit de start, ocho o nueve bits de datos y un bit de stop). El formato más común de datos es el de 8 bits, como se puede ver en la Figura II.16. Se utilizará un contador especializado, llamado generador de baudios, para obtener las frecuencias del oscilador. La USART transmite y recibe, en primer lugar, el bit menos significativo (LSb). Los bloques transmisor y receptor de la USART son funcionalmente independientes, pero usan el mismo formato de datos y la misma velocidad de transferencia. El generador de baudios produce una señal de reloj cada x16 o x64 de la velocidad de desplazamiento de bit, dependiendo del bit BRGH (TXSTA<2>). La paridad no



está soportada por el hardware, pero puede implementarse por software (y almacenarse como el noveno bit del dato). El modo Asíncrono se para durante el estado SLEEP, y se selecciona borrando el bit llamado SYNC (TXSTA<4>).



**Figura II.16.** Diagrama de bloques de comunicación Asíncrona del PIC16F876A

El módulo Asíncrono USART consta de los siguientes elementos importantes:

- Generador de velocidad de baudios (generador de baudios o BRG).
- Circuito de captura.
- Transmisor asíncrono.
- Receptor asíncrono.

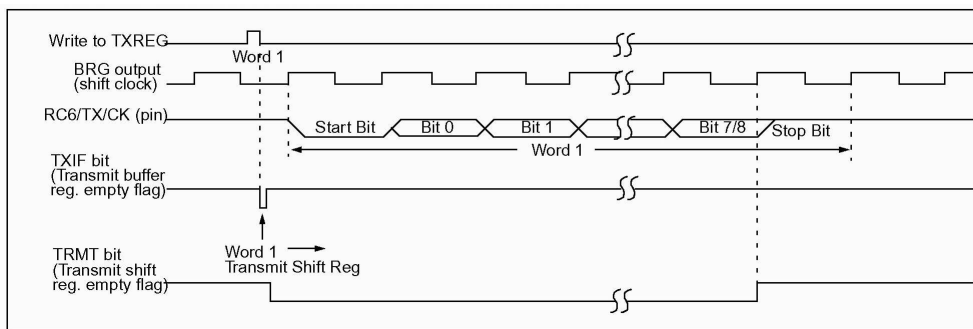
### 2.2.3.7. Transmisor Asíncrono USART

El corazón del transmisor es el registro de desplazamiento de transmisión serie (TSR). El registro de desplazamiento obtiene su dato del buffer de transmisión de lectura-escritura, TXREG. El registro TXREG no se carga hasta que el bit de STOP, de la carga previa, ha sido transmitido. Tan pronto como el bit de STOP se transmite, el TSR se carga con el nuevo dato proveniente del registro TXREG (si está disponible). Enseguida que el registro TXREG transmite su contenido al registro TSR (sucede en un ciclo  $T_{CY}$ ), el registro TXREG queda vacío, y el bit indicador TXIF (PIR1<4>) se activa. Esta interrupción puede habilitarse o

inhibirse activando o borrando, respectivamente, el bit de habilitación TXIE (PIE1<4>). El bit indicador se activará, independientemente del estado del bit de habilitación TXIE, y no podrá borrarse por software. Solamente se borrará cuando un nuevo dato se cargue en el registro TXREG. Mientras el bit indicador TXIF señala el estado del registro TXREG, otro bit, TRMT (TXSTA<1>), muestra el estado del registro TSR. El bit de estado TRMT es de lectura, y se activa cuando el registro TSR está vacío. Ninguna interrupción está unida a este bit, de manera que el usuario debe consultar este bit para determinar si el registro TSR está vacío, todo esto se puede ver en la Figura II.17.

La transmisión se habilita activando el bit de habilitación TXEN (TXSTA<5>). La transmisión actual no ocurrirá, hasta que el registro TXREG sea cargado con el dato, y el generador de baudios (BRG) haya producido el reloj de desplazamiento. La transmisión puede detenerse, primero cargando el registro TXREG, y activando después el bit de habilitación TXEN. Normalmente, cuando la transmisión se ha iniciado primero, el registro TSR está vacío. En este punto, transferir al registro TXREG dará como resultado una transferencia inmediata de aquél a TSR, quedando un registro TXREG vacío. Por esto, es posible una transferencia en oposición (back-to-back). El borrado de TXEN durante una transmisión provocará el abandono de ésta y la inicialización del transmisor. Como resultado, la patilla RC6/TX/CK quedará en alta impedancia.

Para seleccionar una transmisión a 9 bits, el bit TX9 (TXSTA<6>) deberá activarse, y el noveno bit deberá escribirse en TX9D (TXSTA<0>). El noveno bit se escribirá antes de cargar los 8 bits de datos en el registro TXREG. Esto es necesario, porque al escribir un dato en el registro TXREG puede tener lugar una transferencia inmediata del dato al registro TSR (si TSR está vacío). En este caso, se habrá cargado un dato incorrecto de nueve bits en el registro TSR.



**Figura II.17.** Diagrama de Pulsos de comunicación Asíncrona del PIC16F876A

Los pasos a seguir para activar una transmisión Asíncrona, son:

1. Inicializar el registro SPBRG para la adecuada cadencia de bits. Si se desea un valor alto, activar el bit BRGH.
2. Habilitar el puerto serie asíncrono borrando el bit SYNC y activando el bit SPEN.
3. Si se desea interrupción, activar el bit de habilitación TXIE.
4. Si se desea una transmisión en 9 bits, activar el bit TX9.
5. Habilitar la transmisión activando el bit TXEN, que también activará el bit TXIF.
6. Si se selecciona transmisión a 9 bits, el noveno bit se cargará en TX9D.

Cargar el dato en el registro TXREG (se inicia la transmisión).

### 2.2.3.8. Receptor Asíncrono USART

El dato se recibe en la patilla RC7/RX/DT y se conduce hasta el bloque Data Recovery (Recuperación de dato). Este bloque es un circuito desplazador de alta velocidad que opera con una cadencia x16, considerando que el desplazador del receptor serie principal opera a  $F_{osc}$ .

Al seleccionar el modo asíncrono, la recepción se habilita activando el bit CREN (RCSTA<4>).

El corazón del receptor es el registro de desplazamiento serie de recepción (RSR). Después de la captura del bit de STOP, el dato recibido en el registro RSR se transfiere al registro RCREG (si está vacío). Si la transferencia se completa, se activa el bit indicador RCIF (PIR1<5>). La interrupción actual puede habilitarse-inhibirse, activando-borrando el bit de habilitación RCIE (PIE1<5>). El bit RCIF es de lectura, y se borrará por hardware. Esto se producirá cuando el registro RCREG se ha leído y queda vacío. El RCREG es un registro de doble buffer (una FIFO de dos posiciones). Por esto es posible, simultáneamente, cargar RCREG desde la CPU, transferir a RCREG desde el RSR, y comenzar a desplazar en RSR desde la patilla RX. Al detectar el bit de STOP que entra por la patilla RX, si el registro RCREG está todavía lleno, se activará el bit de error por desbordamiento OERR (RCSTA<1>). En este caso, la palabra existente en RSR se perderá. El registro RCREG puede leerse dos veces seguidas para recuperar los dos bytes de la FIFO. El bit OERR será borrado por software, borrando CREN. Si el bit OERR está en 1, se inhibe la transferencia desde el registro RSR hasta el registro RCREG, por lo que es vital borrar el bit OERR, si estuviese activado. El error de concordancia FERR (RCSTA<2>) se activa si un bit de STOP se lee en nivel bajo. El bit FERR y el noveno bit recibido se memorizan en el mismo canal que el dato recibido. Leyendo RCREG se cargarán los bits RX9D y FERR con los nuevos valores. Por consiguiente, es necesario que el usuario lea el registro RCSTA antes de leer el registro RCREG, para no perder los antiguos valores de FERR y RX9D.

Para programar Recepción Asíncrona, hay que seguir los pasos siguientes:

1. Inicializar el registro SPBRG con la apropiada cadencia. Si se desea alta velocidad, poner a 1 el bit BRGH.
2. Habilitar el puerto serie asíncrono, borrando el bit SYNC y activando el bit SPEN.
3. Si se desea interrupción, activar el bit RCIE.
4. Si se desea recepción de 9 bits, activar RX9.

5. Habilitar la recepción, activando el bit CREN.
6. El indicador RCIF se activará cuando se complete la recepción. Esto generará una interrupción, si está activado el bit RCIE.
7. Leer el registro RCSTA para obtener el valor del noveno bit (si está habilitado el modo) y determinar si ha existido error durante la recepción.
8. Leer los 8 bits del dato recibido, leyendo el registro RCREG.

Si ha ocurrido algún error, cancelar éste borrando el bit de habilitación CREN.

### **2.3. Decodificador de doble tono multifrecuencial (DTMF MT8870)**

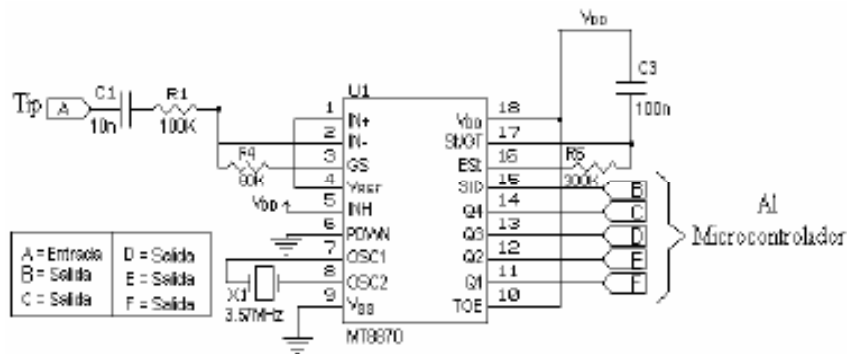
Para este bloque se utilizó el circuito integrado MT8870, el cual es un decodificador de doble tono multifrecuencial receptor; las funciones principales que realiza son filtrar e identificar los tonos de entrada [13]. La etapa de filtrado emplea filtros selectivos para separar el conjunto de frecuencias altas y el conjunto de frecuencias bajas. Además, ésta etapa incluye un rechazo a las señales cuyas frecuencias se encuentren entre 350 y 440 Hz, evitando así la posible decodificación de tonos no deseados. De manera similar, verifica el tiempo de duración de la señal (esto es conocido como la *condición de reconocimiento* de un caracter) y también verifica el tiempo mínimo de pausa entre un caracter con otro. En la etapa de decodificación, emplea técnicas de conteo digital para detectar y decodificar los 16 pares de dobles tonos y proporcionar como resultado en su salida un grupo de datos de cuatro bits. Finalmente, cuenta con una terminal de salida que puede ser utilizada por el sistema de control para detectar la presencia de un nuevo dato, es decir, de una terminal de interrupción para el microcontrolador.

Las terminales más importantes de este circuito se muestran en la Tabla II.2.

Terminales	Entrada / salida	Función
OSC1, OSC2	Entrada	Oscilador de 3.57 MHz
Vdd y Vss	Entrada	Alimentación eléctrica de 5 V
TOE	Entrada	Control del tercer estado (activo en 0 V)
Q1-Q4	Salida	Bus de datos de 4 bits
StD	Salida	Indicador de fin de de codificación de un doble tono (activo en 5 V)

**Tabla II.II.** Especificación de las terminales más importantes del circuito MT8870

Por otro lado, en la Figura II.18 se muestra el diagrama eléctrico del MT8870 en modo de entrada única; los componentes externos que se utilizan son los que recomienda el fabricante.



**Figura II.18.** Diagrama eléctrico del DTMF MT8870 en modo de entrada única

En la Tabla II.3 podemos visualizar la combinación binaria de salida que obtenemos dependiendo de la frecuencia que tenemos de entrada.

F <sub>LOW</sub>	F <sub>HIGH</sub>	KEY	TOW	Q <sub>4</sub>	Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>
697	1209	1	H	0	0	0	1
697	1336	2	H	0	0	1	0
697	1477	3	H	0	0	1	1
770	1209	4	H	0	1	0	0
770	1336	5	H	0	1	0	1
770	1477	6	H	0	1	1	0
852	1209	7	H	0	1	1	1
852	1336	8	H	1	0	0	0
852	1477	9	H	1	0	0	1
941	1209	0	H	1	0	1	0
941	1336	.	H	1	0	1	1
941	1477	#	H	1	1	0	0
697	1633	A	H	1	1	0	1
770	1633	B	H	1	1	1	0
852	1633	C	H	1	1	1	1
941	1633	D	H	0	0	0	0
-	-	ANY	L	Z	Z	Z	Z
L = logic Low, H = Logic High, Z = High Impedance							

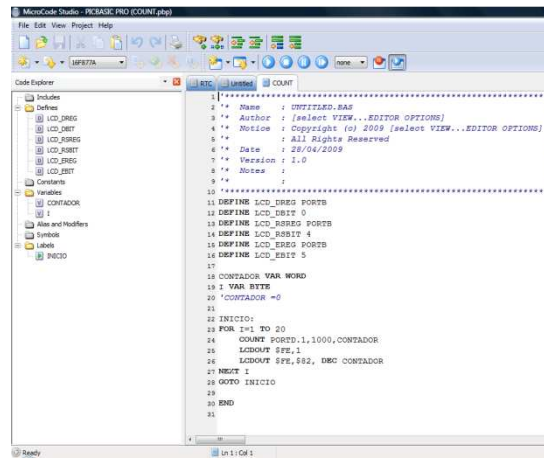
**Tabla II.III.** Tabla de salidas binarias de MT8870

## 2.4. Programas informáticos AUXILIARES

Se han utilizado algunos lenguajes de programación y la combinación de estos han ayudado al desarrollo de la tesis en cuestión, cabe decir que la utilización de cada una de estas dependió del hardware con el cual se va a trabajar.

### 2.4.1 Microcode Studio

Microcode es el software editor en programación en alto nivel como lo es el bloc de notas de Windows, pero con la diferencia que está hecho exclusivamente para facilitar la programación de los microcontroladores PIC, en la Figura II.19 podemos observar la pantalla principal del Microcode Studio, en la parte izquierda podemos encontrar las diferentes partes de nuestro proyecto también podemos elegir el tipo de microcontrolador a utilizar, si el programa está bien escrito y sin fallas compilara y mostrara en la parte inferior izquierda el numero de palabras que está utilizando el PIC, y enseguida se creara tres archivos: .mac, .asm y el .hex, este último es el más importante para el PIC y es el que se cargara en el.



**Figura II.19.** Pantalla Principal de Microcode

### 2.4.2. Compilador PICBasic Pro

Este es el compilador que nos facilita la programación de los PIC, ya que Microcode es un editor, el compilador es el encargado de generar el archivo hexadecimal .HEX, necesario para poder grabar en el microcontrolador PIC.

### 2.4.3. Compilador CCS C

El Compilador C de CCS ha sido desarrollado específicamente para PIC MCU, obteniendo la máxima optimización del compilador con estos dispositivos. Dispone de una amplia librería de funciones predefinidas, comandos de preprocesado y ejemplos. Además, suministra los controladores (drivers) para diversos dispositivos como LCD, convertidores AD, relojes de tiempo real, EEPROM serie, etc. Las características generales de este compilador y más información adicional se pueden encontrar en la dirección <http://www.ccsinfo.com>.

Un compilador convierte el lenguaje de alto nivel a instrucciones en código máquina; un cross-compiler es un compilador que funciona en un procesador (normalmente en un



PC) diferente a procesador objeto. El compilador CCS C es un cross-compiler. Los programas son editados y compilados a instrucciones máquina en el entorno de trabajo del PC, el código máquina puede ser cargado del PC al sistema PIC mediante el ICD2 (o mediante cualquier programador) y puede ser depurado (puntos de ruptura, paso a paso, etc.) desde el entorno de trabajo del PC.

El CCS C es C estándar y, además de las directivas estándar (*#include*, etc.), suministra unas directivas específicas para PIC (*#include*, etc.); además incluye funciones específicas (*bit\_set()*, etc.). Se suministra con un editor que permita controlar la sintaxis del programa.

### 2.4.3.1. Estructura de un programa

Para escribir un programa en C con el CCS C se deben tener en cuenta una serie de elementos básicos de su estructura (Figura II.20).

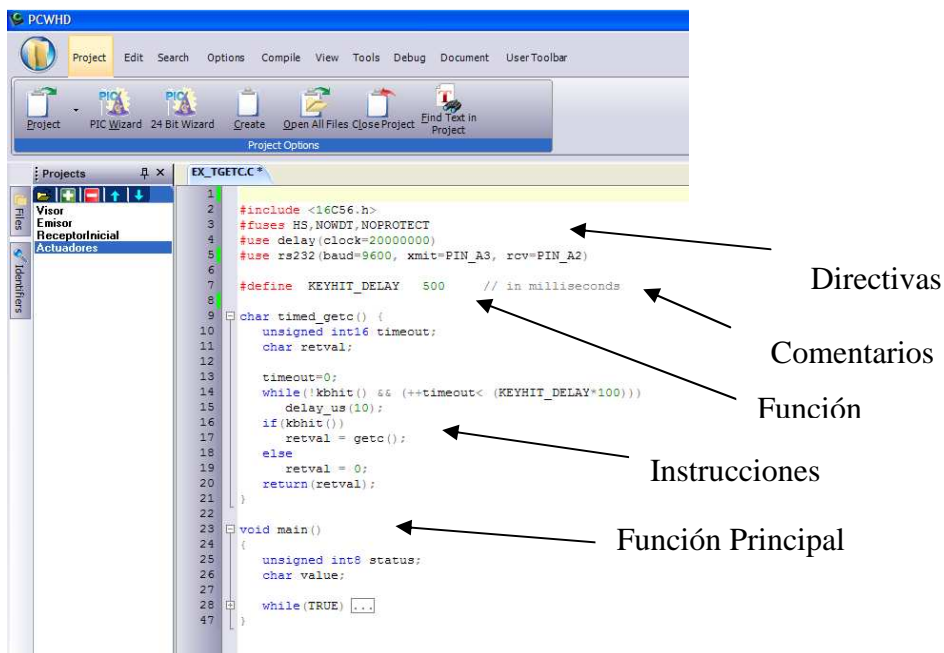


Figura II.20. Estructura de un programa modelo

#### 2.4.4. PICKit 2

El programador PicKit2, es una herramienta de programación para desarrollo de bajo costo. Es capaz de programar la mayoría de los microcontroladores y memorias seriales EEPROM de Microchip. Para los dispositivos específicos vea el léeme, incluido en el CD ROM PicKit2 Starter kit.

La unidad del PicKit2 se muestra en la Figura II.21:



**Figura II.21.** Vista Frontal de la Unidad PICKit2

##### 2.4.4.1. Conexión del puerto USB:

El puerto de conexión USB, es un conector del tipo mini-B. Conecta el PicKit2 a la PC usando el cable suministrado.

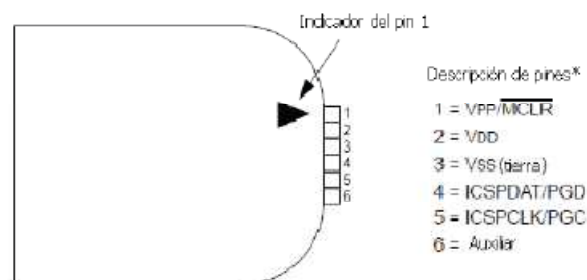
##### 2.4.4.2. LEDs de estado:

Los leds de estado indican el estado del PicKit2.

- Power (Verde) - La alimentación es aplicada al PicKit2 vía USB.
- Target (Amarillo) - El PicKit2 está alimentando un dispositivo.
- Busy (Rojo) - El PicKit2 está ocupado con una función en progreso, tal como la programación.

### 2.4.4.3. Conector de programación:

El conector de programación es de 6 pines con un espacio entre pines de .1" o .25mm que se conecta al dispositivo a programar. Ver las especificaciones de los pines de salida en la Figura II.22.



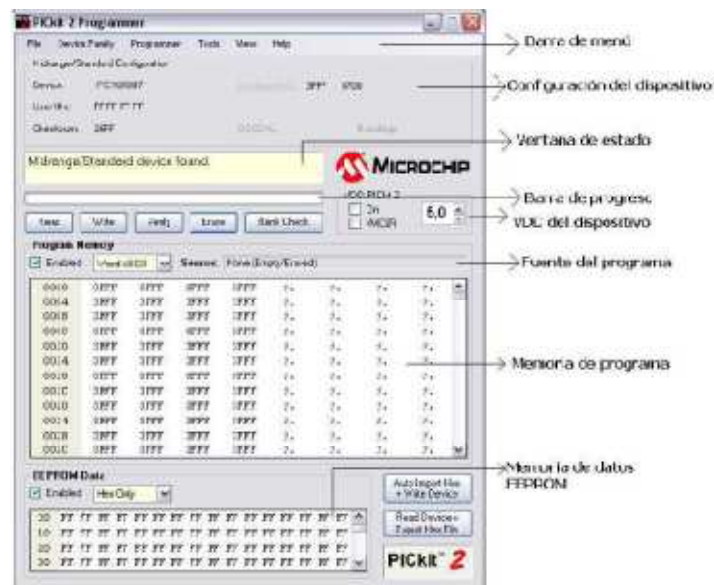
**Figura II.22.** Esquema de conexión del PICKit2

Las funciones de los pines de programación son diferentes para programar EEPROM y HCS. Ver el archivo léeme en (*Help/readme*) incluido en el software de programación del PicKit2 para estos pines.

### 2.4.4.4. Aplicación de programación:

La aplicación de programación te permite programar todos los dispositivos soportados en el archivo léeme del PicKit2. La interface de programación aparece como se muestra en la Figura II.23. Los controles han sido listados en las siguientes secciones.

Para más información sobre como instalar y usar la aplicación de programación del PicKit2 vea el capítulo: PicKit2 Comenzando a programar.



**Figura II.23.** Interface de la aplicación PICKit2

#### 2.4.4.5. Barra de menú:

En la barra de menú seleccionamos varias de las funciones de la aplicación de programación del PicKit2. Un sumario de esas funciones es:

##### *Archivo (File)*

##### Importar HEX – (Import HEX):

Importa un archivo HEX para programar. El formato de archivo HEX, INHX32 está soportado.

##### Exportar HEX – (Export HEX):

Exporta un archivo HEX leído desde el dispositivo. El archivo HEX es creado en el formato INHX32.

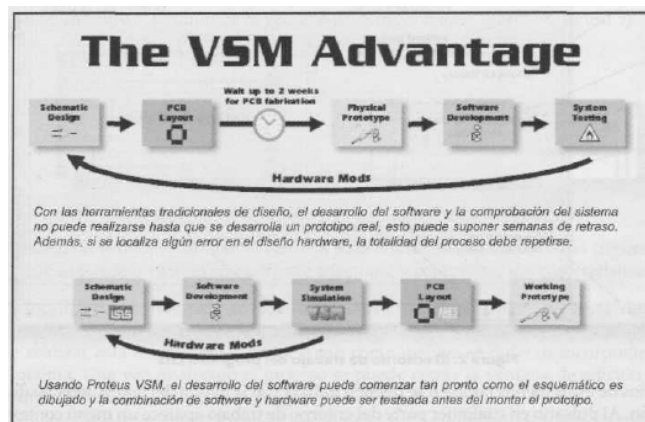
Historial de archivo – (File History):

Los últimos 4 archivos HEX abiertos son mostrados con la dirección de donde se encuentran. Estos archivos recién abiertos son seleccionados para la rápida importación. Note que el historial de archivo se va a iniciar en blanco en una nueva instalación hasta que un archivo HEX sea importado.

**Exit:** Sale del programa.

### 2.4.5. PROTEUS 7.6

El entorno de diseño electrónico PROTEUS VSM de LABCENTER ELECTRONICS ofrece la posibilidad de simular código microcontrolador de alto y bajo nivel y, simultáneamente, con la simulación en modo mixto de SPICE. Esto permite el diseño tanto a nivel hardware como software y realizar la simulación en un mismo y único entorno. Para ello, se suministran tres potentes subentornos como son el ISIS para el diseño gráfico, VSM (Virtual System Modelling) para la simulación y el ARES para el diseño de placas, ver Figura II.24.



**Figura II.24.** Entorno de Trabajo Proteus.

### 2.4.5.1. Entorno Gráfico ISIS

ISIS es un potente programa de diseño electrónico que permite realizar esquemas que pueden ser simulados en el entorno VSM o pasados a un circuito impreso ya en el entorno ARES.

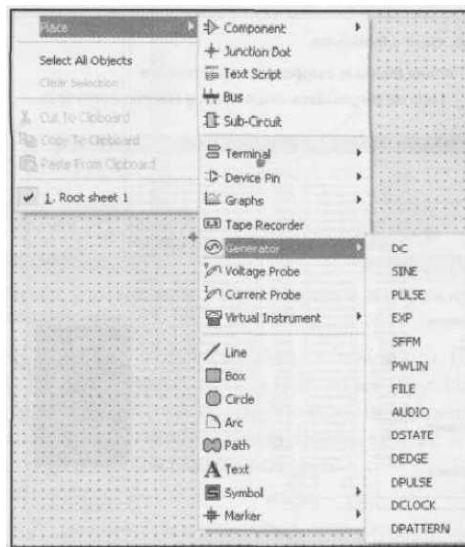
Posee una buena colección de librerías de modelos tanto para dibujar, simular o para las placas. Además, permite la creación de nuevos componentes, su modelización para su simulación e, incluso, la posibilidad de solicitar al fabricante que cree un nuevo modelo.

Sin entrar profundamente en cómo utilizar dicho programa, a continuación se explican las bases para dibujar cualquier circuito electrónico. El programa ISIS posee un entorno de trabajo como se puede ver en la Figura II.25. formado por distintas barras de herramientas y la ventana de trabajo.



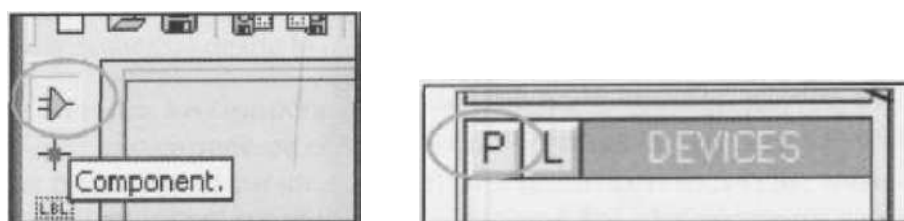
**Figura II.25.** Entorno de Trabajo del programa ISIS

Varios de estos menús también se pueden utilizar con la ayuda del botón derecho del ratón. Al pulsarlo en cualquier parte del entorno de trabajo aparece un menú contextual donde se pueden ir obteniendo los distintos submenús de trabajo como se puede ver en la Figura II.26.



**Figura II.26.** Submenús de trabajo del botón derecho del ratón.

Para dibujar, lo primero es colocar los distintos componentes en la hoja de trabajo. Para ello, se selecciona el modo componentes como de muestra en la Figura II.27.a y, acto seguido, realizar una pulsación sobre el botón P de la ventana de componentes y librerías en la cual se muestra en la Figura II.27.b.



**Figura II.27.** a) Modo Componentes; b) Botón “pick”

Tras activar el botón P se abre la ventana para la edición de componentes (Figura II.28.)

donde se puede buscar el componente adecuado y comprobar sus características.

Al localizar el componente adecuado se realiza una doble pulsación en él, de tal forma

que aparezca en la ventana de componentes y librerías (Figura II.29). Se puede realizar

esta acción tantas veces como componentes se quieran incorporar al esquema. Una vez

finalizado el proceso se puede cerrar la ventana de edición de componentes.

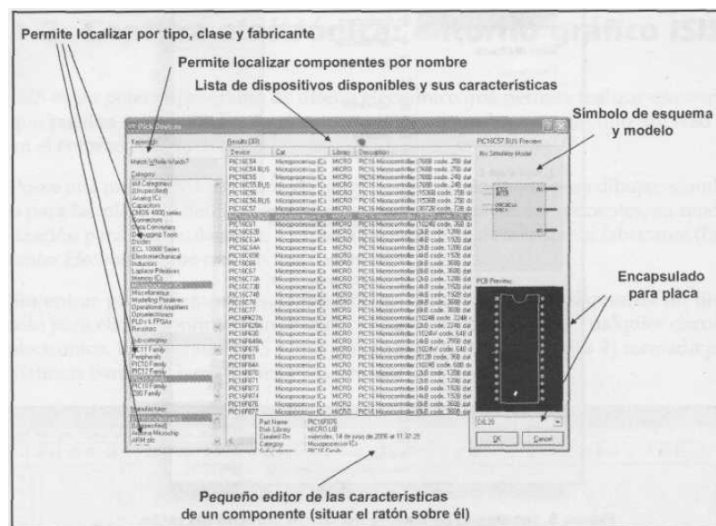


Figura II.28. Ventana para la edición de componentes.

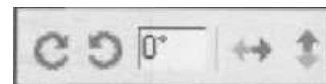


Figura II.29. Los componentes añadidos



Para situar un componente en el esquema tan sólo debemos seleccionarlo de la lista. Al hacerlo se puede comprobar su orientación (tal como se representará en el esquema) en la ventana de edición como se muestra en la Figura II.30. Si deseamos modificar la rotación o la reflexión del componente podemos acceder a ello a través de la barra de herramientas correspondiente (Figura II.31).

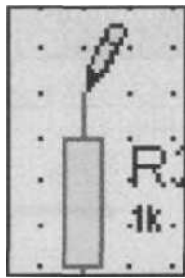
Haciendo de esta forma, “todos” los componentes de la lista tendrán la misma orientación (si se desea orientar un único componente debemos hacerlo una vez ya situado en el esquema).



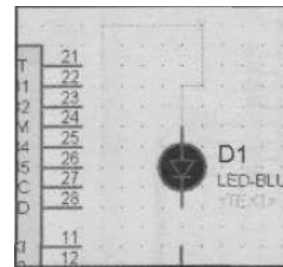
**Figura II.30.** Selección y Orientación del componente

**Figura II.31.** Barra de rotación y reflexión

Para unir los componentes con cables hay que situarse en los extremos de los terminales, el cursor se convierte en un lápiz verde como se ve en la Figura II.32. Ahora se pueden ejecutar dos acciones o ir marcando el camino hasta el destino con distintas pulsaciones como se ve en la Figura II.33 o realizar, directamente, una pulsación en el destino y dejar que ISIS realice el camino. Para ello, debe estar activada la herramienta **TOOLS**→**Wire Autorouter**.

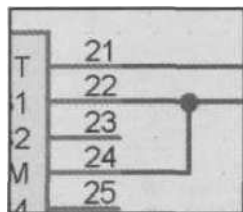


**Figura II.32.** Inicio de Cable.



**Figura II.33.** Circuito “a mano”

Las uniones entre cables se pueden realizar de forma automática. Para ello, mientras se traza un camino debemos realizar una pulsación sobre el cable objeto de la unión eléctrica como se ve en la Figura II.34. También se pueden realizar de forma manual mediante el modo de unión (Figura II.35); en este modo ten sólo hay que ir haciendo pulsaciones sobre los puntos donde deseamos realizar la unión.



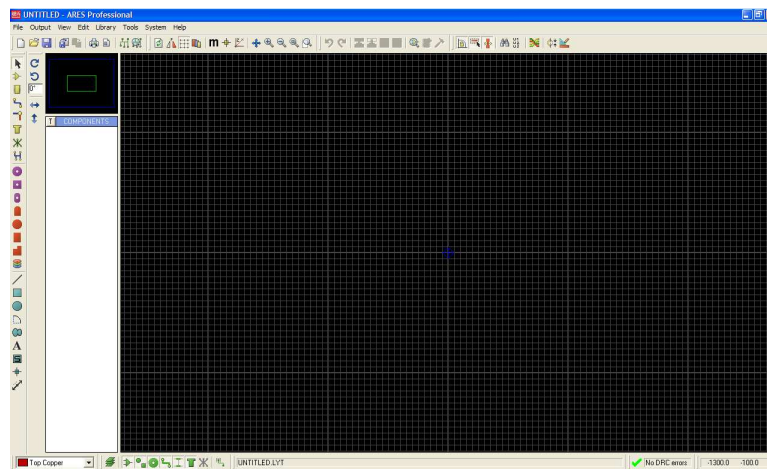
**Figura II.34.** Unión eléctrica entre cables



**Figura II.35.** Modo de unión

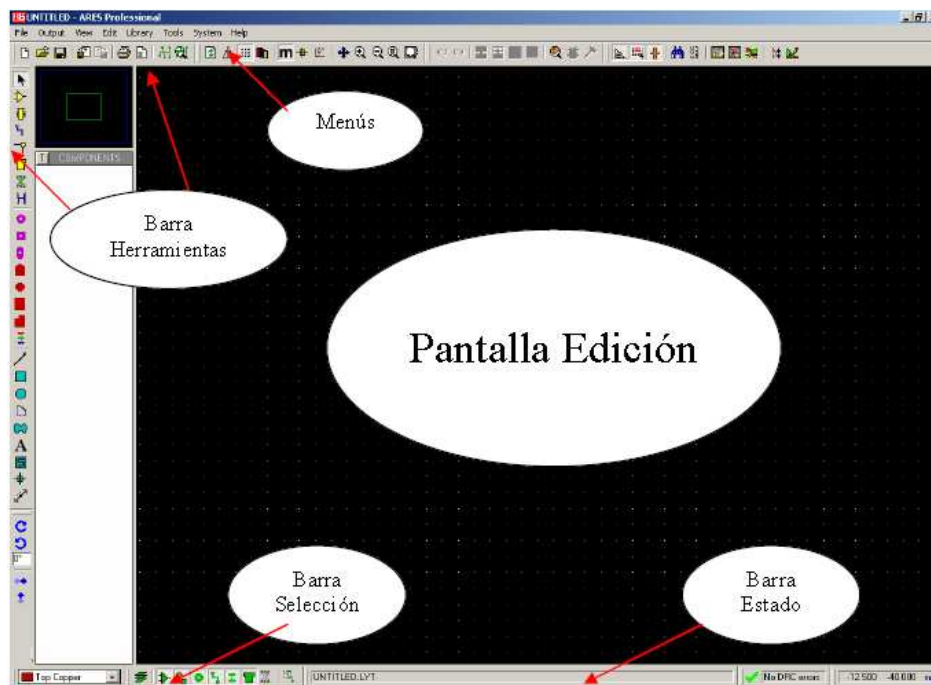
#### 2.4.5.2. Entorno Gráfico ARES

Esta herramienta es muy indispensable si desea fabricar un circuito impreso, para ello una vez que se ha diseñado un circuito en el Entorno ISIS se presiona el icono ARES, se presentará una pantalla similar a la Figura II.36.



**Figura II.36.** Entorno de ARES

El entorno de trabajo de ARES se muestra en la Figura II.37. en la cual se muestra los componentes y herramientas con que cuenta el software.



**Figura II.37.** Herramientas de ARES

*Menús:* Encontramos los menús para guardar el diseño, crear los ficheros GERBER, distintas opciones de visualización, rutado, opciones, etc.

*Barra Herramientas:* Selección de las herramientas para la creación de las pistas, comprobación del netlist, etc.


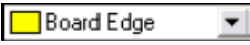
*Barra Selección:* nos permitirá seleccionar la capa de trabajo, dentro de la pantalla de Edición, también podemos seleccionar los elementos que deseemos, etc.

*Barra de Estado:* Muestra la posición en la que se encuentra el cursor, esta se puede dar en Th(mils) o milímetros (mm). La relación que hay entre estas medidas es la siguiente: 40mils=1mm, de forma que con esta relación ya se puede dar una idea de las medidas.

*Pantalla de Edición:* Aquí se realiza el diseño del circuito la diagramado en ISIS.

Dentro de la barra de herramientas, se dispone de la herramienta “*Package Mode*”, que nos permitirá escoger el encapsulado que deseemos añadir.

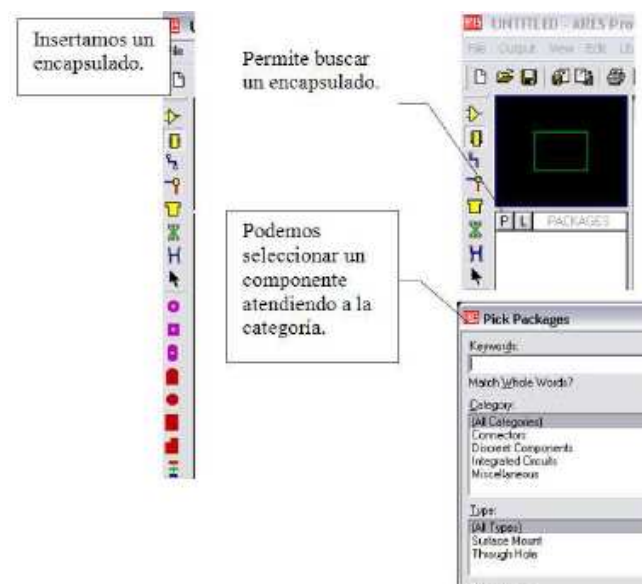
### Creación de la placa PCB.

En primer lugar se crea la delimitación de la placa PCB, se debe escoger la herramienta “2D Graphics Box”  , y seleccionar la capa , con esto ya se puede realizar el borde de la PCB.

### Posicionamiento de los Componentes.

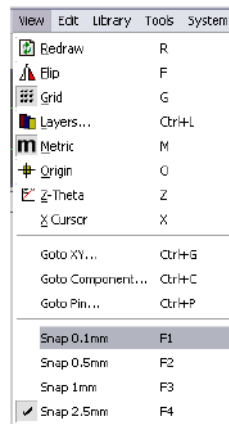
Partiendo del hecho de que todos los componentes que se tiene en la placa disponen de encapsulado posteriormente se pasa a realizar el posicionamiento de los mismos, para ello existen 2 opciones:

- Posicionamiento Automático.- Realizado por ARES, para lo cual bastará con seleccionar dentro del menú Tools la herramienta “Auto Placer”. Para poder utilizar esta herramienta, es necesario haber definido previamente los límites de nuestra placa.
- Posicionamiento Manual: Mediante los botones presentados en la siguiente Figura II.38.



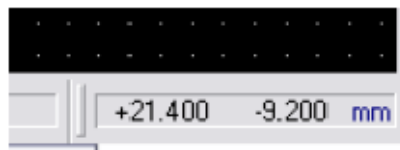
**Figura II.38.** Cuadro de Posicionamiento Manual

Para mejorar la posición de los componentes así como el rutado de las pistas se recomienda usar un grip (distancia entre puntos) adecuado, para ello se debe dirigir al menú VER como se puede observar en la Figura II.39.



**Figura II.39.** Menú VER de ARES

Se puede ver la distancia entre componentes así como la medida de las pistas etc, mediante las coordenadas que aparecen en la parte inferior derecha de la pantalla, como se muestra en la siguiente Figura II.40.

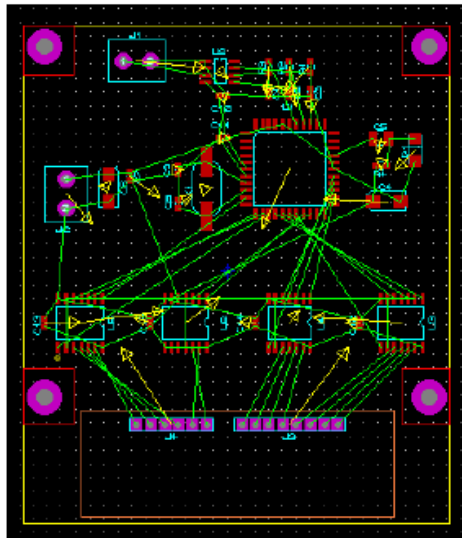


**Figura II.40.** Área de visualización de coordenadas

Por defecto estas medidas aparecerán en milis (Th), si queremos pasarlas a mm bastará con seleccionarlo en la barra de herramientas.

### Rutado de las pistas.

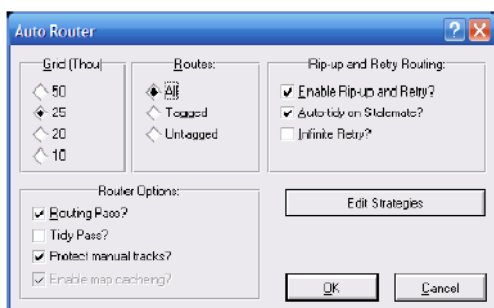
Con los componentes ya dispuestos en su posición, pasamos a realizar el rutado de las pistas. Se debe tener en cuenta que las líneas “verdes” son las líneas correspondientes a las uniones (netlist) entre los componentes (ver Figura II.41.), es decir las líneas que indican qué está conectado con qué.



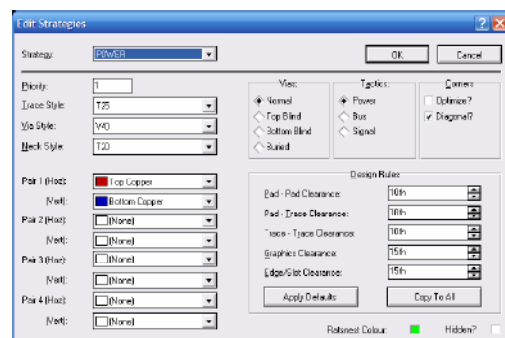
**Figura II.41.** Vista de diseño en Ares

Para realizar el rutado de las pistas también se dispone de dos opciones:

- Rutado Automático: Mediante la herramienta “*Auto Router*” del menú Tool. Al pulsarla se nos mostrará una pantalla en la que indicaremos el grip, las pistas a rutar... etc. Lo más destacable de esta ventana será el botón de *Edit Strategies* que nos permitirá indicar las ventanas de la Figura II.42 y la Figura II.43.



**Figura II.42.** Características de las Pistas



**Figura II.43.** Capa en cual diseñar

La estrategia de rutado a seguir, el estilo de la traza, vías....etc.


Podemos elegir si deseamos realizar el rutado a una sola capa o si por el contrario vamos a utilizar más de una, para ello debemos realizar lo siguiente:

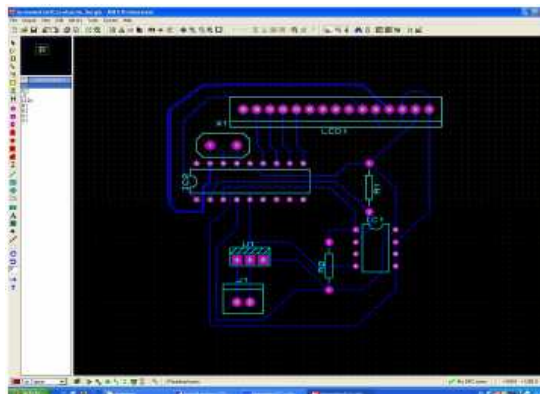
Si vamos a trabajar con una sola capa, dentro del primer PAR de capas (Horz-Vert) seleccionamos como horizontal la BOT Copper y en Vertical seleccionaremos NONE. Si deseamos 2 o más capas en vez de seleccionar NONE pre seguiríamos con la capa que nos interesase.

También se debe configurar las opciones de REGLAS DE DISEÑO, para lo cual debemos tener en cuenta que la relación entre los milis y los mm es la siguiente: 40milis=1mm.

Una vez realizado esto si aceptamos pasará a realizar el rutado automáticamente.

- Rutado Manual: Para ello solo tendremos que pinchar sobre el icono “Track mode” de

la barra de herramientas  y elegimos el grosor de la pista con el cual trabajaremos como se ve en la Figura II.45 y el diseño ya terminado puede verse como la Figura II.44.



**Figura II.44.** Pistas ya terminadas

CE	TRACES
	DEFAULT
	DR.DCC
	FANOUT
	REJE?
	T8
	T10
	T12
	T16
	T20
	<b>T25</b>
	T30
	T40
	T50
	T60
	T70
	T80
	T90
	T100
	T200
	T300
	T400
	T500

**Figura II.45.** Grosor de Pista

Una vez realizado todos los ajustes puede imprimirlo, para ello abra la pestaña *Output* y de un clic en *Print..* como se ve en la Figura II.46



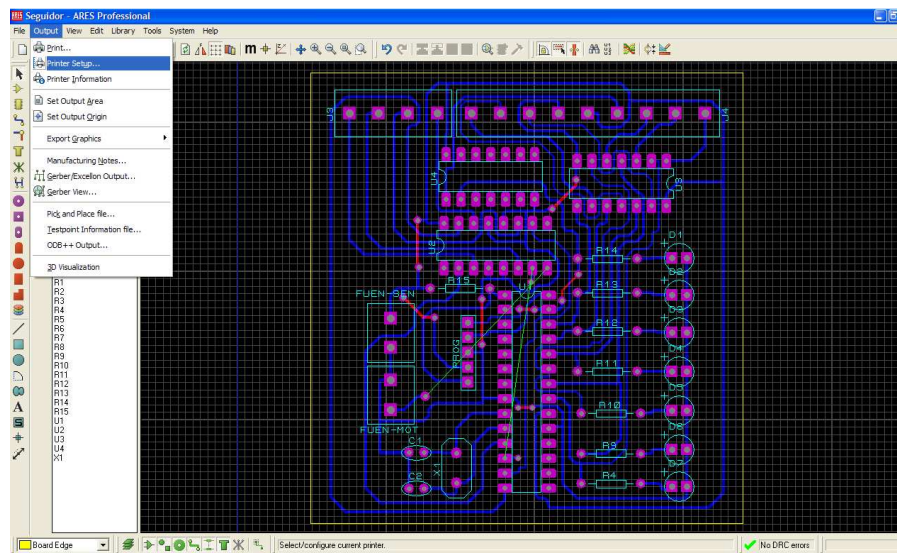


Figura II.46. Ventana de Impresión.

## **CAPÍTULO III**

### **DISEÑO DEL SISTEMA**

#### **3.1 REQUERIMIENTOS DEL SISTEMA**

En este capítulo se presenta el diseño de hardware como software para la implementación del Sistema de Control de Alarmas Comunitarias o también llamada SAC.

Este sistema está desarrollado como una ayuda para el aviso de alertas posibles o existentes en el barrio, esta realiza una llamada telefónica hacia la Unidad de Policía Comunitaria, el dispositivo transmisor que se encuentra en la vivienda al ser accionado

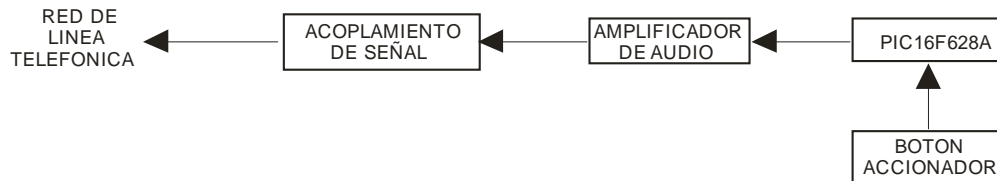
realiza una llamada telefónica enviando tonos únicos para la identificación de la vivienda que acciono la llamada de auxilio, en la UPC se encuentra una central que consta de una pequeña base de datos en la cual está almacenado los nombres, direcciones y teléfonos de los usuarios que tengan los dispositivos de transmisión.

En la central consta de un diseño el cual es muy versátil para su instalación el cual basta con un adaptador 12Volt. de corriente continua, un conector para la línea telefónica convencional y otro conector adicional para la conexión del panel de visualización que es un modulo complementario, también está diseñado para que se pueda realizar todas las acciones posibles para una correcta gestión de usuarios como es la de ingreso, eliminación, actualización y visualización, cabe destacar que también consta de una base de datos en la cual se almacena todas las alarmas que se han accionado con sus respectiva información.

Al igual que a la central, el transmisor se alimenta con un voltaje de 12 volt. Y una conexión hacia la línea telefónica convencional.

El diseño del sistema esta implementado en función del siguiente diagrama de bloques de la Figura III. 47. en la cual se muestra un diagrama de bloques de la etapa de transmisión.

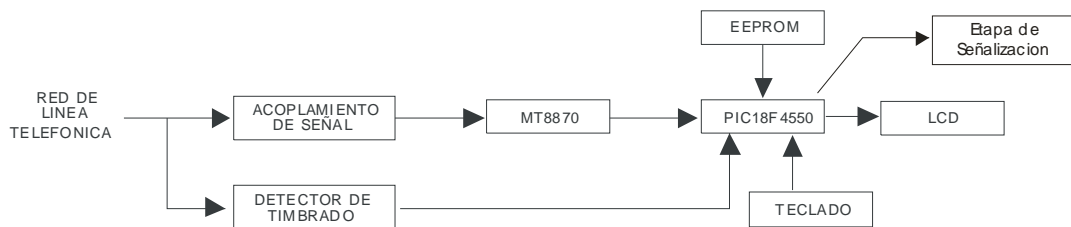
### TRANSMISOR



**Figura III.47.** Diagrama de Bloques de Transmisor-Hardware

### RECEPTOR

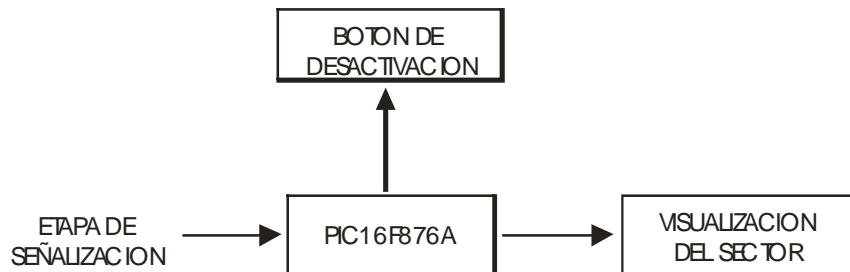
En la Figura III.48. se muestra el diagrama de bloques del equipo de recepción en la cual se describe d una manera más global en modo de funcionamiento y etapas de procesamiento de la información y señales.



**Figura III.48.** Diagrama de Bloques de Receptor-Hardware

Otra etapa adicional en el receptor es la visualización de los sectores en un mapa visual en la cual basta con la conexión directa con el receptor principal, cabe decir que esta etapa no es indispensable su conexión para el funcionamiento del sistema.

El esquema funcional de esta última etapa se muestra en la Figura III.49.



**Figura III.49.** Diagrama de Bloques de Visualización

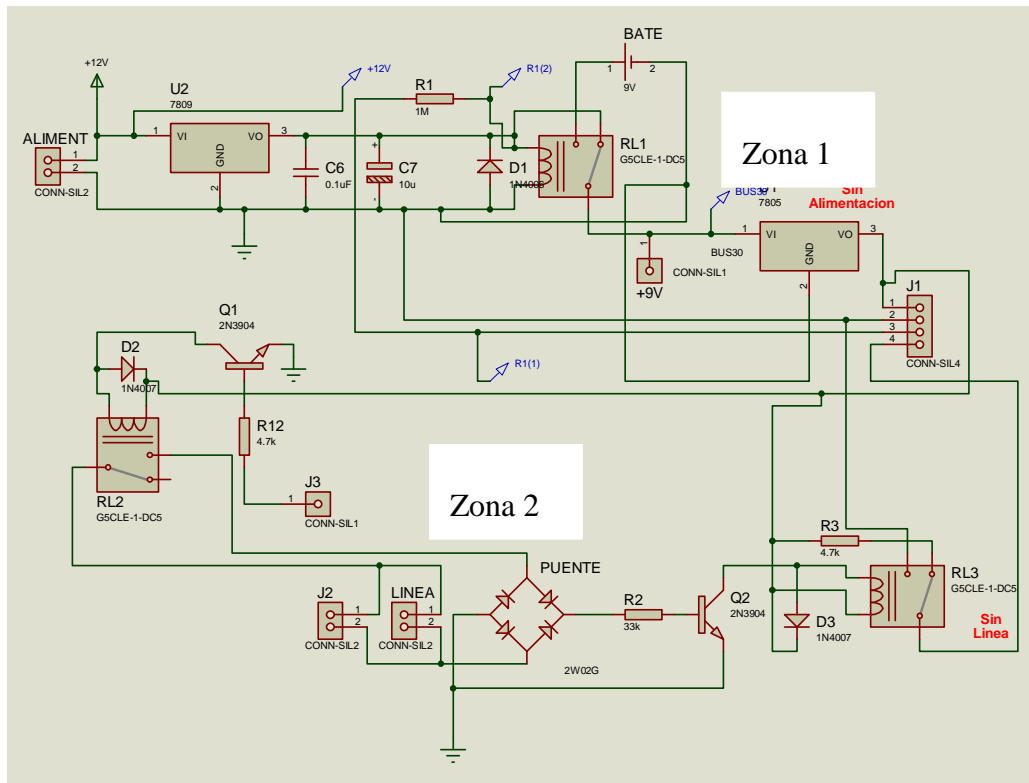
## 3.2 TRANSMISOR

### 3.2.1 Etapa de Alimentación

Para iniciar la descripción del funcionamiento del sistema empezaremos a describir el método de alimentación, control de respaldo de energía y alertar que se le han puesto al sistema.

La alimentación del sistema es de 12 V. con un consumo de corriente de 0.5 amperios, la cual se puede obtener de un adaptador de corriente continua de 12V. a 500mA

En una forma general se puede decir que para su accionamiento se debe pulsar el botón de encendido por 3 segundos, el motivo de esto es debido a que hay momentos que por accidente se pulsará este botón y se accionará la alarma, para evitar esos casos se ha determinado que el pulso por 3 segundos realice el proceso de envío de alerta para su posterior acción de auxilio.



**Figura III.50.** Diagrama de circuito alimentación-transmisor

En la Zona 1 como podemos ver en la Figura III.50. que tenemos la entrada de alimentación que consta de un nivelador de voltaje de 9 volt. positivos con sus respectivos capacitores de filtrado de voltaje, seguidamente se encuentra un relé de 9 volt. con la finalidad de que se accione o conmute a otra fuente de alimentación como es este caso a una batería de 9 volt. cuando haya ausencia de alimentación eléctrica convencional.

En la Zona 2 se diseñó un diagrama simple con la finalidad de obtener una señal lógica que nos represente el estado de la línea telefónica (con línea/sin línea), la línea telefónica está conectada a un puente de diodos para transformar la señal senusoidal que se encuentra en la red telefónica en una señal de corriente continua, antes del puente rectificador se le ha conectado a las líneas de conmutación de un relé de 5Volt., el

propósito es que cuando la línea no esté en uso el circuito este escuchando permanentemente el estado de la línea telefónica y en el momento que se accione la alarma esta deje de escuchar y deje la línea libre para que el circuito genere los tonos respectivos de llamada sin que haya conflictos con la etapa de aviso de alerta de línea.

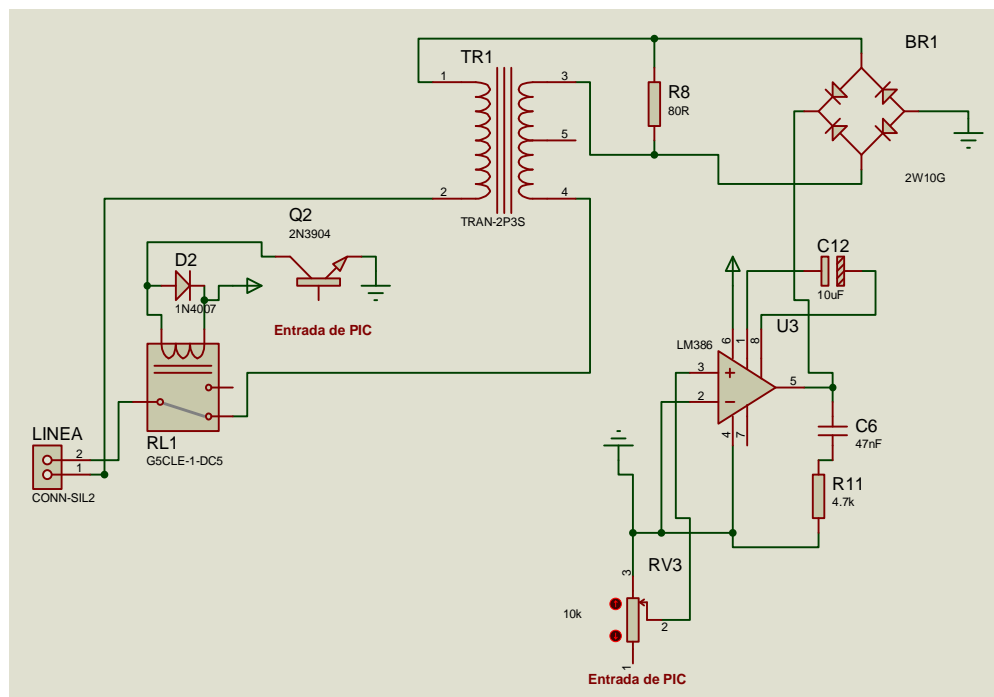
El relé RL3 tiene la finalidad de que cuando haya servicio telefónico conmute 5 volt. a su línea de salida y esta sea dirigida hacia el microcontrolador, pero cuando no haya servicio esta conmutara 0 volt. re-dirigiéndolo al microcontrolador y así dando la potestad al microcontrolador de avisar al usuario el estado de la línea.

En la zona entre el relé de 9 volt. y los capacitores de filtrado se toma una línea de indicación de estado el cual es dirigido hacia el microcontrolador para que este último haga la acción correspondiente de aviso de falta de energía.

### **3.2.2 Acoplamiento de señal**

Para el diseño del sistema como se ve la Figura III.51 se ha basado en diagramas esquemáticos usados en diversos circuitos de acoplamiento, para empezar se ha usado un relé de dos posiciones el cual determinará el estado de la línea, ésta es controlada por medio del microprocesador PIC16F628A, el cual será accionado en base a un pulsador, una vez accionado la alarma el relé es conmutado y la red telefónica es conectado al resto del circuito, sin antes pasar por un transformador de acoplamiento el cual tiene una impedancia de 600 Ohms a 400 Hz y una relación 1:1, que es similar al trafo usado por los módems, seguidamente se encuentra una resistencia en paralelo de 680 Ohms con una potencia de disipación de 2 vatios, con la finalidad de que cuando se quiera realizar la llamada la resistencia se conecte en paralelo con la línea telefónica y realice el

descolgado de la línea telefónica , posteriormente se le ha colocado un rectificador de voltaje, como es un puente de diodos, el cual nos ayuda a tener una sola polarización de la línea telefónica sin tener que preocuparnos por el sentido de los cables de conexión de la línea telefónica.



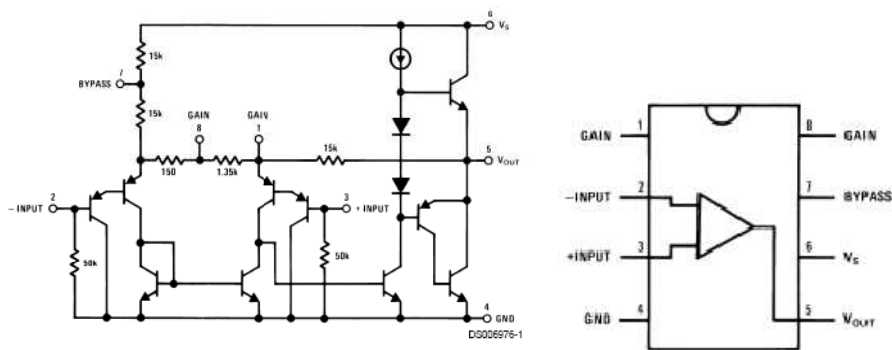
**Figura III.51.** Diagrama de circuito de acoplamiento de señal

### 3.2.3 Amplificador de Audio

Posteriormente a la etapa acoplamiento de la señal se colocó un amplificador de audio para obtener una ganancia de 200dB por cada tono los tonos emitidos por el microcontrolador esta ganancia se obtiene mediante el CI. LM386 el cual en su interior consta de transistores en cascada como se puede ver en la Figura III.52, estos tonos son usados para que la central telefónica sepa que número telefónico le está llamando, y para que posterior a la llamada y atención de la misma se proceda al envío de los tonos



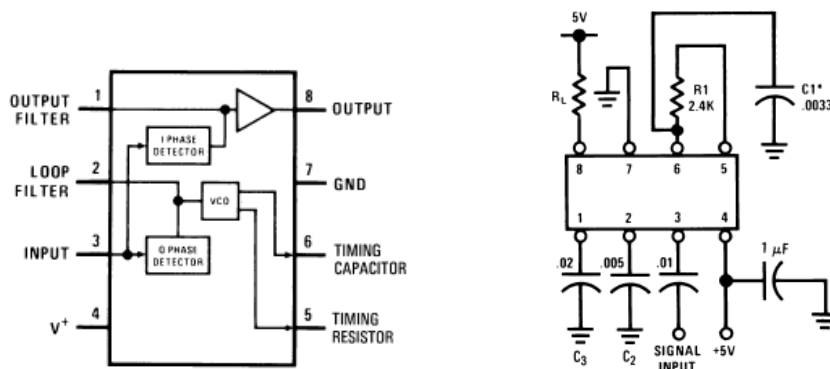
identificadores del dispositivo emisor, cabe decir que también se colocó otro amplificador en estado escucha en el mismo transmisor con la finalidad de obtener las frecuencias de timbrado que se realizan al otro lado de la línea y saber si fue contestado o no.



**Figura III.52.** Circuito interno del C.I. LM386

### 3.2.4 Detector de Línea Atendida

Esta etapa consta de un amplificador de audio el cual escucha los tonos que la central telefónica emite al número que se está llamando, seguidamente se ha colocado un CI. LM567 el cual es un detector de frecuencias (Figura III.53), esta última es con la finalidad de detectar el tono de timbrado del lado receptor, siendo esta una frecuencia de 430 Hz +/- 20Hz, con una frecuencia de 1,1 segundos, esta frecuencia es calibrada según los cálculos que se han realizado según la casa fabricante de este CI.



**Figura III.53.** Circuito interno y modo de configuración del C.I. LM567

La frecuencia central del tono a detectar está dada por la siguiente ecuación:

$$f_0 \cong \frac{1}{1.1 R_1 C_1} \quad Ec.1$$

- La frecuencia está dada en Hz
- La capacitancia esta en Faradios
- Y la resistencia en ohmios

Reemplazando valores se obtiene:

Si la capacitancia es de 0.1uF

$$R_1 = \frac{1}{1.1 * 430 * 0.0000001} = 21141.649 \text{ ohm.}$$

Para la afinación de la frecuencia se ha colocado un potenciómetro de 50K.

El ancho de banda de filtrado está dada por:

$$BW = 1070 \sqrt{\frac{V_f}{f_0 C_2}} \text{ in \% of } f_0 \quad Ec.2$$

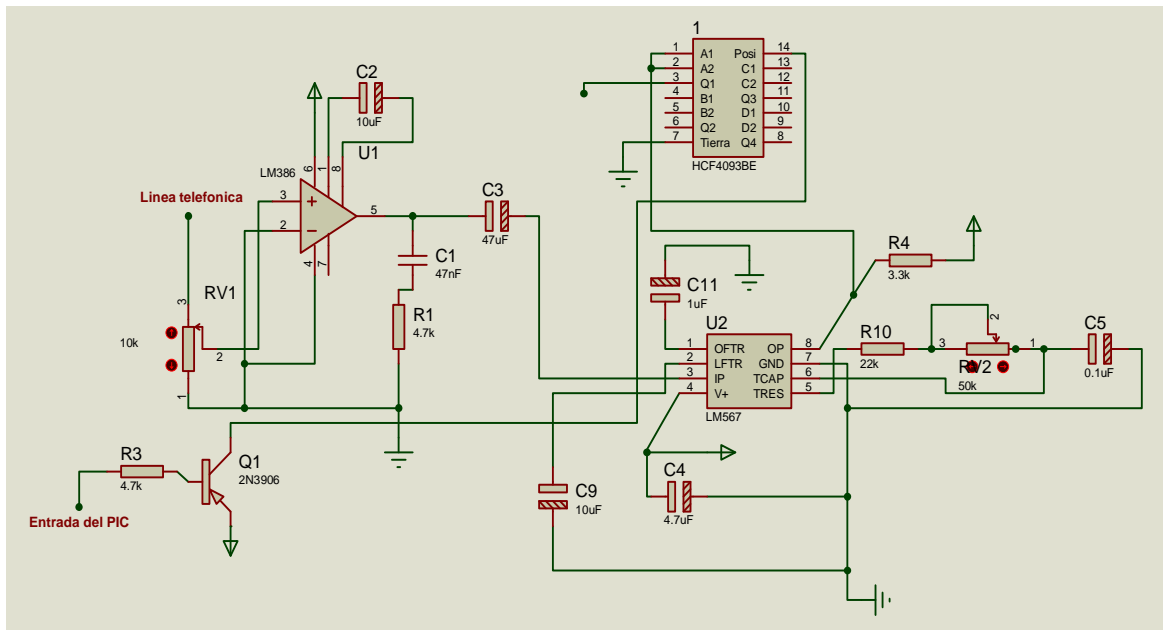
Reemplazando valores se obtiene:

El voltaje rms de entrada es de 200mV y el Capacitor 2 es de 10uF.

$$1070 \sqrt{\frac{0.2}{430 * 0.00001}} = 32,7 \%$$

El ancho de banda es el  $\pm 32,7 \%$  de la frecuencia central.

El detector brinda una salida de voltaje mayor a 3 voltio cuando se ha detectado la frecuencia deseada, la salida de este detector de tonos está conectado un CI HCF4093BE con la finalidad de convertirla la señal a TTL para el ingreso al PIC, el diseño de este circuito lo podemos ver en la Figura III.54



**Figura III.54.** Esquema del Circuito Detector de Línea Atendida

El CI. HCF4093BE es habilitado posteriormente haberse realizado la llamada telefónica debido a que el sistema detectaría los tonos que el mismo PIC emite para la realización de la llamada.

### 3.2.5 Etapa de control central

Todas las etapas anteriores son centralizadas hacia un PIC16F628A el cual está programado para la emisión de Tonos y recepción de señales que indique el estado de la línea y activación de la misma, este circuito también está encargado de la presentación del estado de todo el proceso mediante la ayuda de LED's, estos últimos nos mostrara si el mensaje de alarma se ha enviado, si la línea telefónica a sido deshabilitada, si la luz eléctrica de alimentación se ha suspendido, siendo en este último caso, el dispositivo cuenta con una batería de alimentación como respaldo, este puede alimentar al circuito por 5 horas aproximadamente.

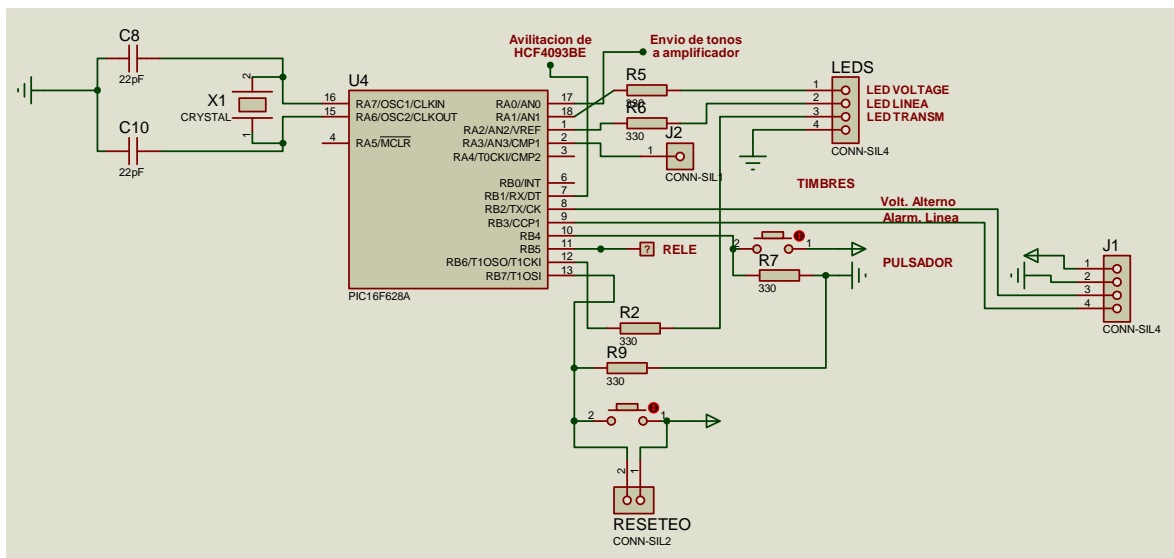


Figura III.55. Diagrama circuital de la etapa de control central

El microcontrolador como podemos ver en la Figura II.55 está utilizando un oscilador externo de cristal a 20MHz, la razón es que la generación de los tonos es de mayor precisión y oscilación a esta frecuencia. La programación de este microcontrolador se le realizó con ayuda del software Microcode, el cual trabaja con un compilador Basic. El motivo de la utilización de este programa fue la facilidad para la generación de tonos DTMF.

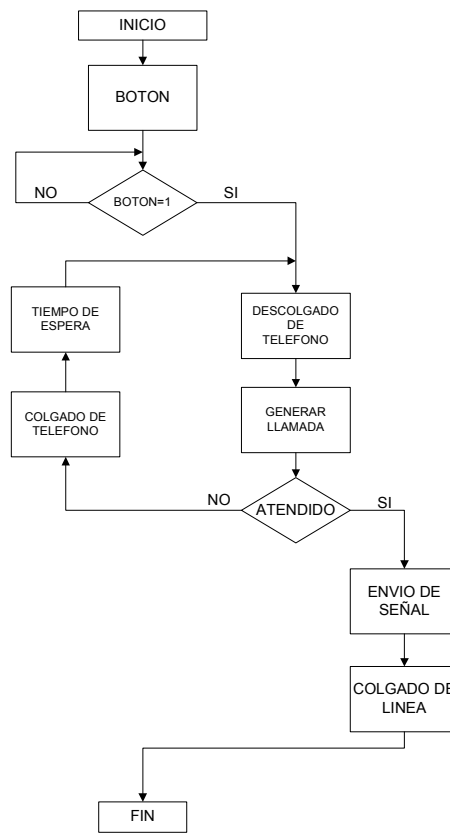
Cabe resaltar que los niveles de voltaje con las que se está trabajando son todos digitales, salvo la señal de tono que emite el microcontrolador, por ende la configuración del PIC es que todas las entradas sean digitales (CMCON=0).

El rango de voltaje que emite el microcontrolador con los tonos es bajo por ello es la razón que se ha utilizado un amplificador de audio.

#### 4.2.5.1 Programación de la Etapa de control central

El programa de control que utiliza todo este sistema es muy largo por ello es que se muestra a continuación un diagrama de bloques que es la manera más simple y efectiva de representar un sistema de programación completa.

En la Figura III.56. se muestra el diagrama de bloques que describe el funcionamiento del sistema de transmisión:



**Figura III.56.** Diagrama de flujo del Transmisor de Alerta

Una vez especificado las etapas del sistema, en la Figura III.57 se muestra el diagrama del circuito completo, cabe decir que para la conexión del switch de activación se le ha colocado una resistencia de 220 Ohmios para bajar la corriente de consumo que soporta el microcontrolador.

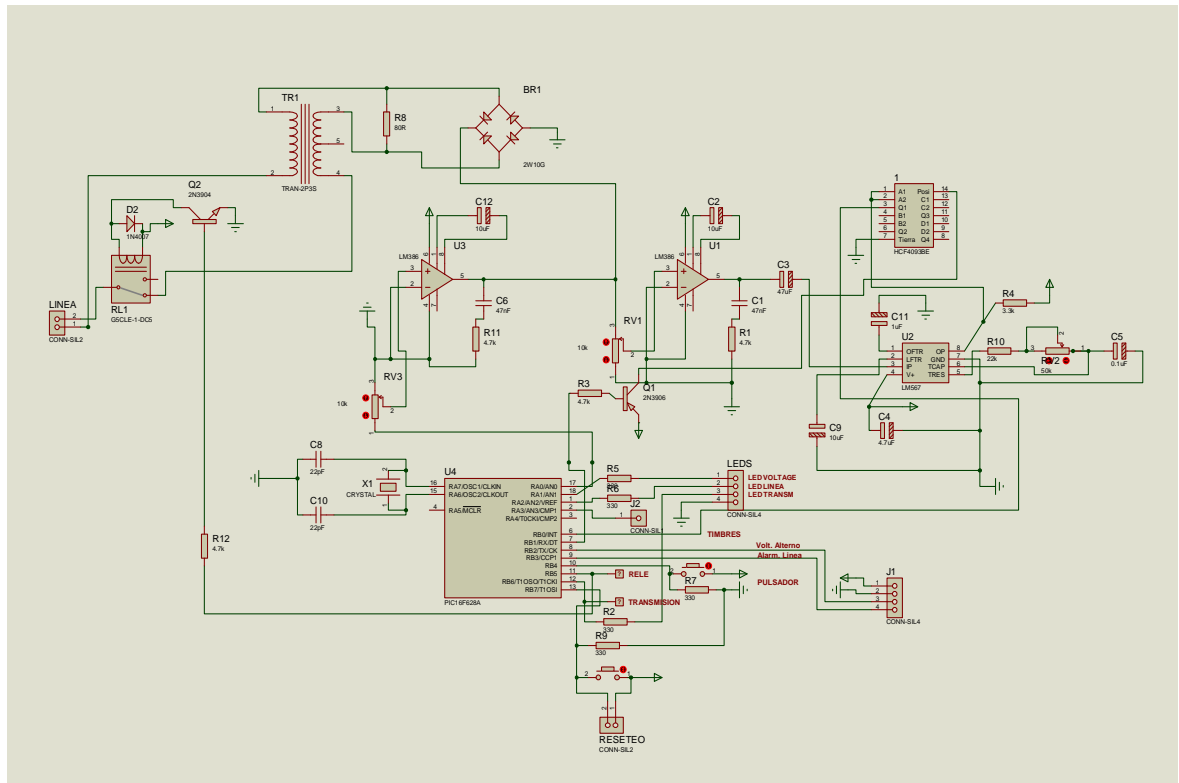


Figura III.57. Diagrama circuital completo del sistema transmisor SAC

### 3.3 RECEPTOR

#### 3.3.1 Etapa de Alimentación

En esta etapa a diferencia de la Etapa de Alimentación explicada en el Transmisor se tiene unas variantes importantes pero con la misma lógica que la anterior, lo idéntico del circuito anterior es la Zona 1, que tiene el mismo esquema que el transmisor, cabe decir que la alimentación del transmisor es de un alto consumo de corriente debido a los componentes electrónicos que posee, la fuente de alimentación debe superar el 1 Amperio para tener un correcto funcionamiento del circuito, el esquema de este circuito lo podemos ver en la Figura III.58.

En la Zona 2 la diferencia de la Zona 2 del transmisor es que no se puso el relé entre la línea telefónica y el puente rectificador, después del puente rectificador se coloca una resistencia de 33K ohmios y un transistor para el accionamiento del relé para cuando no exista la alimentación de la línea telefónica.

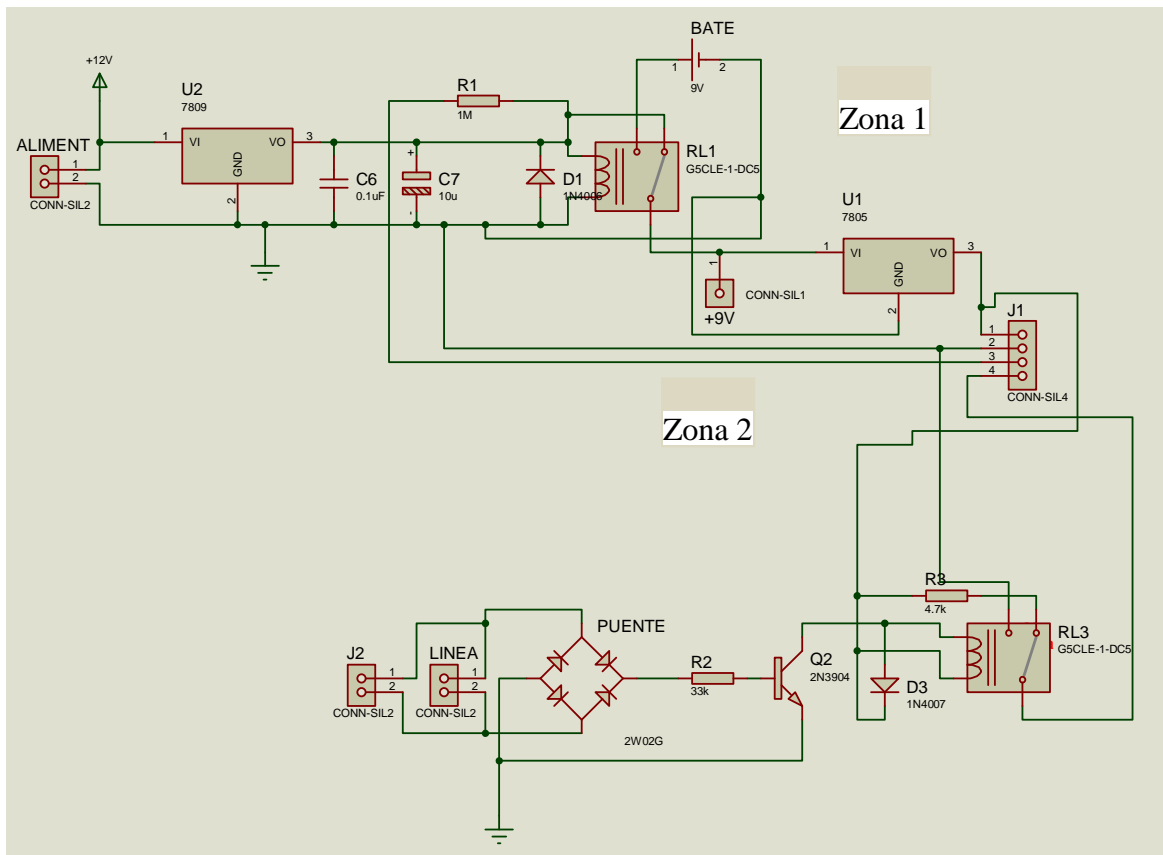
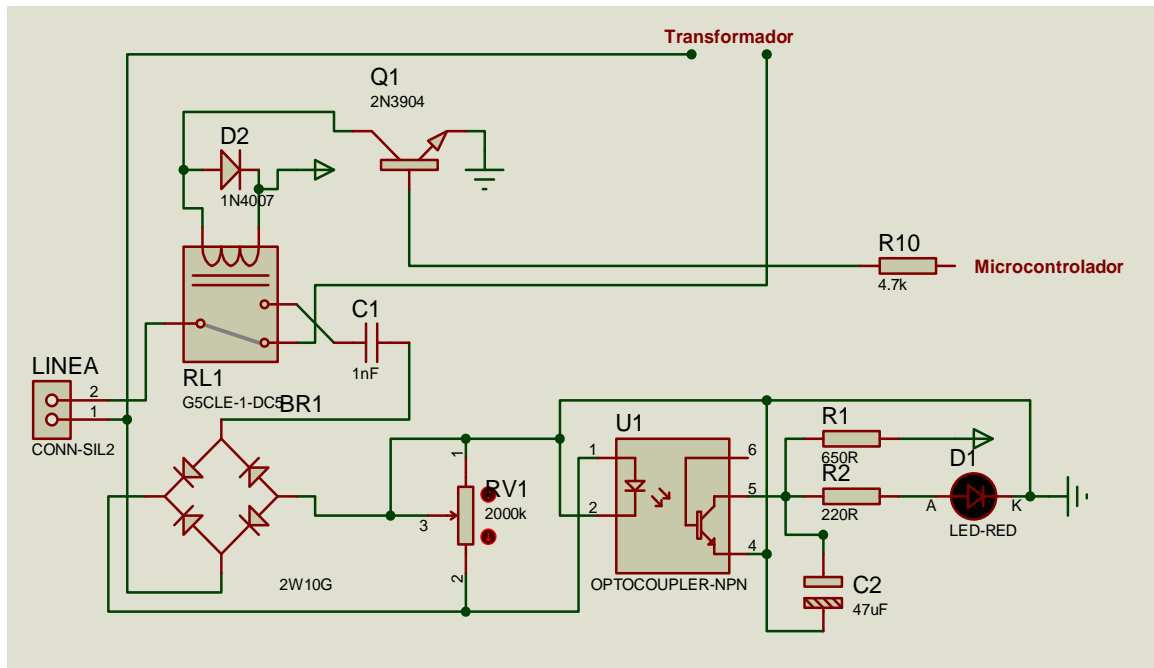


Figura III.58. Diagrama de circuito alimentación-receptor



### 3.3.2 Etapa de Detección de Timbre



**Figura III.59.** Diagrama circuital del detector de timbre.

En esta etapa como podemos ver en la Figura II.59 la línea telefónica está conectada a un relé el cual cuando no está activado esta conmuta a un puente rectificador con la finalidad de tener una sola polarización como resultado, posteriormente se ha conectado a un potenciómetro de 2 Mega Ohmios, el capacitor C1 ayuda a filtrar las frecuencias y a aislar las frecuencias de audio con la alimentación de la línea, esta frecuencia de audio es dirigida a un optoacoplador 4N25 en la cual consta de un diodo emisor y un foto transistor el cual ayuda a aislar y transformar la onda de frecuencia a una señal digital, el funcionamiento de este CI. es la de detectar una frecuencia de timbre, cuando esta se ha detectado la salida del CI. que es su pin 5 envía un voltaje menor a 1 voltio y cuando no sea a detectado ninguna frecuencia de audio su salida es de 5 voltios siendo esta señal la enviada al microcontrolador.

El capacitor C2 se lo ha colocado con la finalidad de eliminar el parpadeo que genera la señal sinusoidal recibida por el CI.

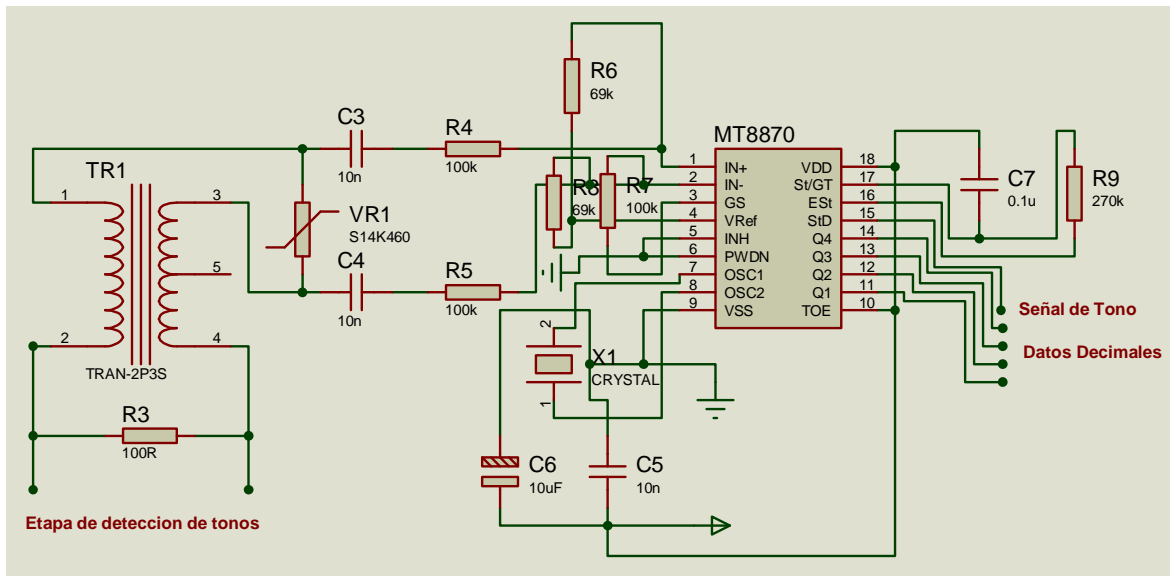
El relé es activado solo cuando se ha detectado el timbre de llamada, esta es redirigido al transformador de acoplamiento y a la resistencia en paralelo para que sea atendida la llamada.

### **3.3.3 Etapa de identificación de Tonos.**

La etapa siguiente se dedica a la recolección y reconocimiento de tonos enviados por los emisores, estos tonos son transformados a una numeración decimal para poder ser comparados con una base de datos que se ha creado en el sistema y proceder a las siguientes etapas.

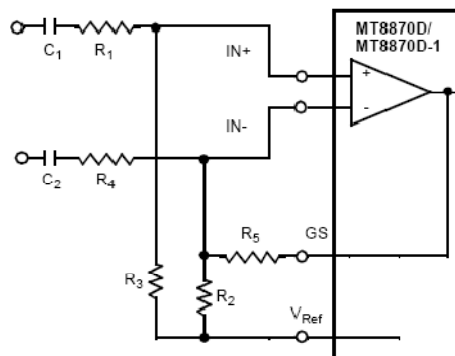
Tenemos una salida adicional que nos brinda el CI. MT8870 con la finalidad de saber cuándo se ha detectado un tono que identifique al transmisor, esta es la salida StD que es el pin 15.

Si en esta salida se ha detectado un uno lógico significa que existe y tono de entrada y si es un cero lógico es lo contrario, el esquema completo de esta etapa se muestra en la Figura III.60.



**Figura III.60.** Diagrama de circuito identificación de tonos.

Este circuito integrado necesita un oscilador de cristal de 3.579545 Mz. Y tiene una configuración de de filtrado de entrada según las ecuaciones brindadas por la casa fabricante como es:



$$C_1=C_2=10 \text{ nF}$$

$$R_1=R_4=R_5=100 \text{ k}\Omega$$

$$R_2=60\text{k}\Omega, R_3=37.5 \text{ k}\Omega$$

$$R_3 = \frac{R_2 R_5}{R_2 + R_5}$$

$$\text{VOLTAGE GAIN } (A_v \text{ diff}) = \frac{R_5}{R_1}$$

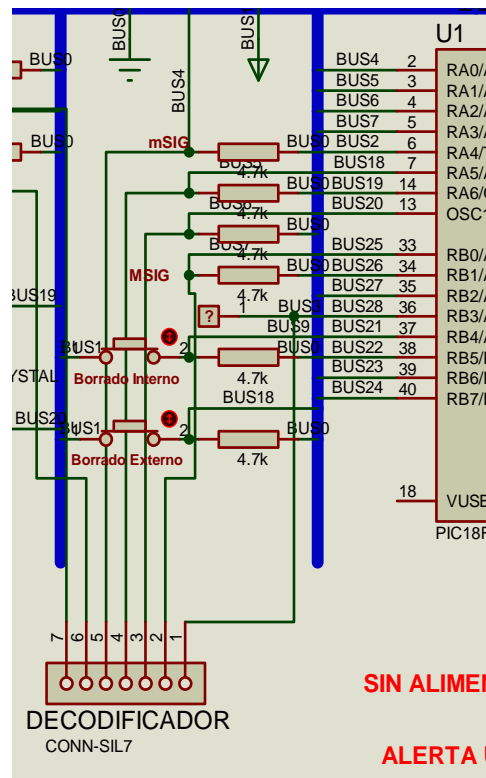
Con los cálculos realizados los resultados son:

Voltaje de Ganancia=1

La polarización de la línea telefónica para la conexión a este C.I. no importa ya que está diseñado para la conexión directa.

#### **3.3.4** Etapa de Recepción de Datos

Una vez determinado el número decimal enviado en forma de tonos, el MT8870 envía mediante un bus de 4 bits en paralelo hacia un microcontrolador 18F4550, los datos que llegan son en forma binaria, uno de los trabajos del microcontrolador es la de convertir el valor binario a un valor decimal. Los valores que llegan son valores de un solo dígito, en el caso que el valor enviado por el transmisor sea de 2 dígitos el microcontrolador realiza el debido proceso para la formación del dato decimal, para reconocer la llegada de los dígitos el microcontrolador se basa en el bit de aviso StD que proviene del C.I. MT8870, una vez que se han recibido los dos dígitos el microcontrolador realiza la acción de liberado de la línea, los pines a los que son conectados los 4 bits de datos son del RA0 al RA3 siendo el menos significativo el RA0, el pin al que se encuentra conectado la señal StD es el RA4, estos datos se pueden visualizar en la Figura III.61.

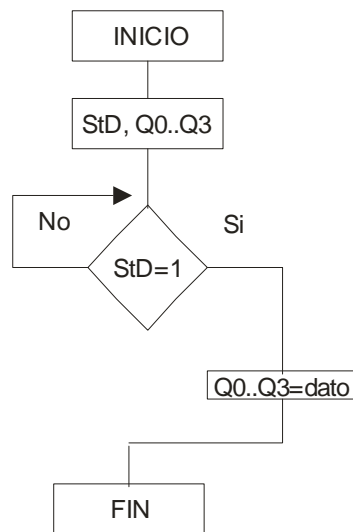


**Figura III.61.** Gráfica de conexión para Recepción de datos

### 3.3.4.1 Programación de la Etapa de Recepción de Datos

La programación de esta etapa se muestra en el siguiente diagrama de bloques en donde todos los procesos principales que se realizan en el momento en que se detecta un tono entrante.

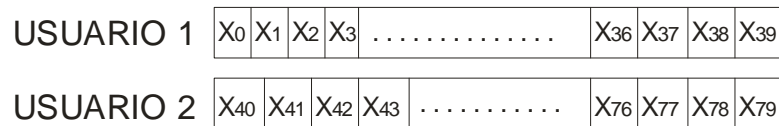
La comunicación con el microcontrolador se basa en el diagrama de flujo (Figura III.62.):



**Figura III.62.** Diagrama de Flujo para la identificación de tonos

### 3.3.5 Etapa de Almacenamiento de Datos

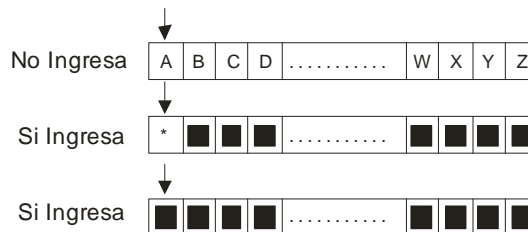
El SAC necesita un sistema de almacenamiento de datos en el cual consta de los nombres de usuarios, direcciones, numero de teléfonos y el código de casa para la identificación, el método que la información está almacenado es el que se muestra en la Figura III.63, en cual el espacio de memoria que esta ocupa es el más reducido posible, el espacio de memoria ocupada por cada usuario es de 40 Bytes.



**Figura III.63.** Diagrama de Estructura de Datos.

Para mejorar la gestión de memoria y aprovechamiento de la misma se ha ideado un método de marcado para la información que ha sido borrada o modificada, la descripción de cada paso se realiza de la siguiente manera:

*Ingreso de Datos:* Para el ingreso de datos el sistema verifica el espacio de memoria cada 40 Bytes, el dato es grabado si en ese espacio de memoria se encuentra un valor decimal de “255” o en forma de carácter “■”, ó el símbolo “\*” (Ver Figura III.64).



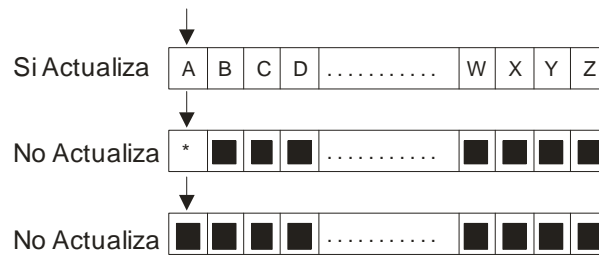
**Figura III.64.** Diagrama de Grabación de Datos

*Eliminación de Datos:* Para realizar la eliminación de datos se procedió primero a la visualización de los datos en el lector LCD, una vez visualizado el usuario que se desea eliminar se esperará la confirmación de la eliminación de la misma, una vez realizada la confirmación, el primer byte del bloque de datos se le sobre-escribe con el dato string "\*" y los 39 bytes siguientes se los reemplaza por el carácter "■" (Ver Figura III.65).



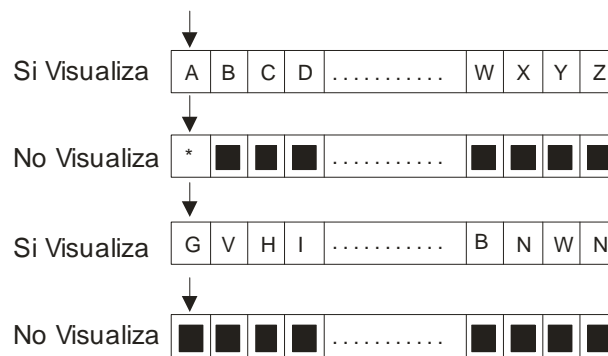
**Figura III.65.** Diagrama de trama de Eliminación

*Actualización de Datos:* Para la actualización de datos se procedió al igual que la eliminación en la visualización y la posterior confirmación, se pedirá el ingreso de la nueva información, esta última se sobre-escribirá a la información que anteriormente se encontraba en los bytes correspondientes (Ver Figura III.66).



**Figura III.66.** Diagrama de Descripción de Actualización

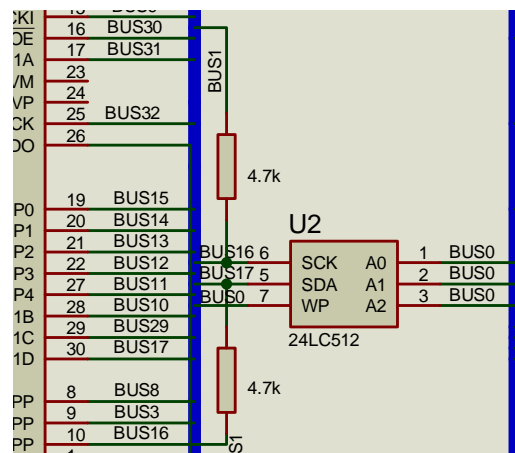
*Visualización de Datos:* En la visualización de datos el microcontrolador busca el primer byte cada 40 bytes, en el caso que el primer byte sea diferente de “■” ó “\*” se visualizará desde ese byte hasta los siguientes 40 (Ver Figura III.67).



**Figura III.67.** Diagrama de Tramas para Visualización

El medio de almacenamiento que se utilizo es una memoria eeprom 24LC512 como se puede ver en la Figura III.68, se escogió este tipo de memoria debido a su facilidad de configuración y capacidad de memoria.



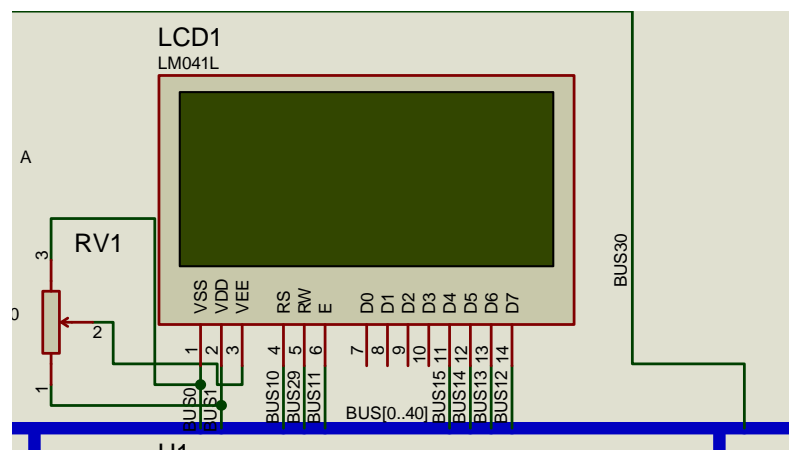


**Figura III.68.** Gráfica de la Memoria EEPROM 24LC512

### 3.3.6 Etapa de Visualización de Información

La manera de visualización el cual se está empleando es mediante un LDC de 4x16 segmentos (Ver Figura III.69).

La información que se muestra es todo acerca de los usuarios, nombre, dirección, número telefónico, también se muestra información como son las alertas de alimentación y cortes de línea telefónica.



**Figura III.69.** Gráfica del LCD 16x4 segmentos

### 3.3.7 Etapa de Ingreso de Información

En esta etapa se usa un teclado matricial 4x4 usado para el ingreso, actualización y visualización de la información de cada usuario y alarmas activadas.

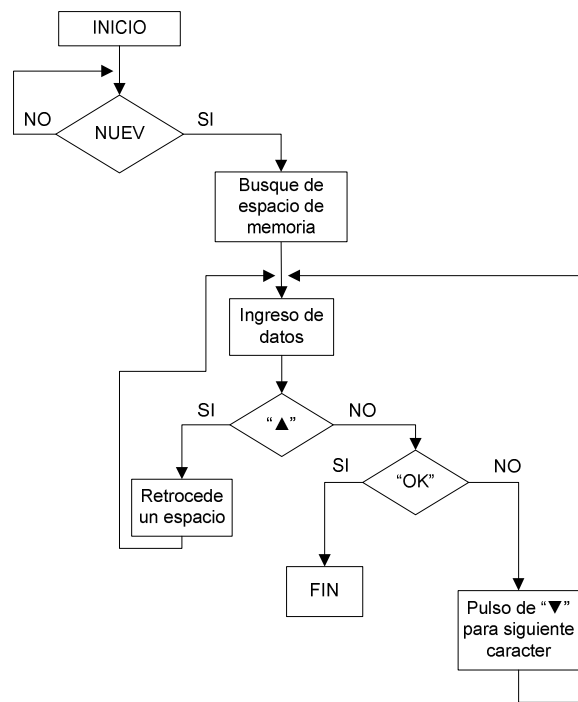
Para confirmar cualquier ingreso, actualización o eliminación siempre se necesitara una confirmación, la tecla de confirmación es el OK, así mismo para la desactivación de las alarmas.

Los botones con los caracteres "^" y "V" son usados dependiendo de en qué procesos nos encontremos.

*Proceso de Ingreso de Nuevo Usuario:* Cuando se elija esta opción automáticamente en el visualizador se verá los datos que hay que ingresarlo, los datos se ingresaran caracter a caracter en forma de escritura de un celular, para la escritura del siguiente caracter se pulsará la tecla "V" y en el caso que se quiera corregir una letra mal escrita se pulsara la tecla "^" tantas veces hasta llegar a la posición del caracter a corregir, cuando se acabe de llenar la información de una línea se presionara la tecla "OK" para la confirmación.

#### 3.3.7.1 Programación del proceso de Ingreso de Nuevo Usuario

En la Figura III.70. siguiente se muestra el proceso total de la etapa de Ingreso de información de una manera muy clara y precisa usando un diagrama de flujo, este diagrama muestra todo el proceso de de busque en la memoria espacios vacíos, y esperando a que se teclee los caracteres correspondientes con su respectiva confirmación de la información.



**Figura III.70.** Diagrama Procesos para Ingreso de Nuevo Usuario

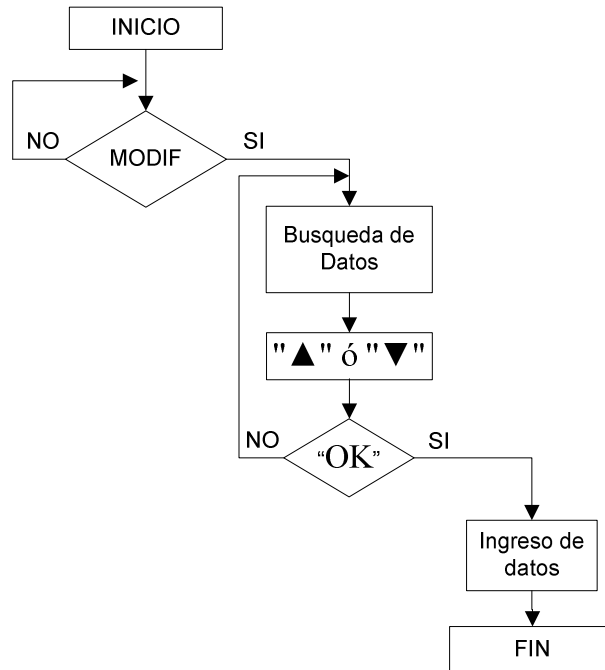
*Proceso de Modificación de un Usuario:* En esta opción nos ayudamos del proceso de listado de usuarios ya que mediante los cursores hacia arriba y hacia abajo se selecciona el usuario que se desea modificar.

Al hallar el usuario a modificar se presiona el botón de confirmación que es el "OK".

Posteriormente se pedirá el ingreso de los nuevos datos del usuario, el cual se ingresan los datos caracter a caracter tal y como se procedió al ingreso de un nuevo usuario. Al terminar el ingreso de los datos se visualizará un aviso de "Guardando datos".

### 3.3.7.2 Programación de Modificación de un Usuario

En el siguiente diagrama de bloques se especifica el proceso de ingreso de usuarios y la confirmación de la información ingresada, se puede visualizar el diagrama de bloques de este proceso en la Figura III.71.

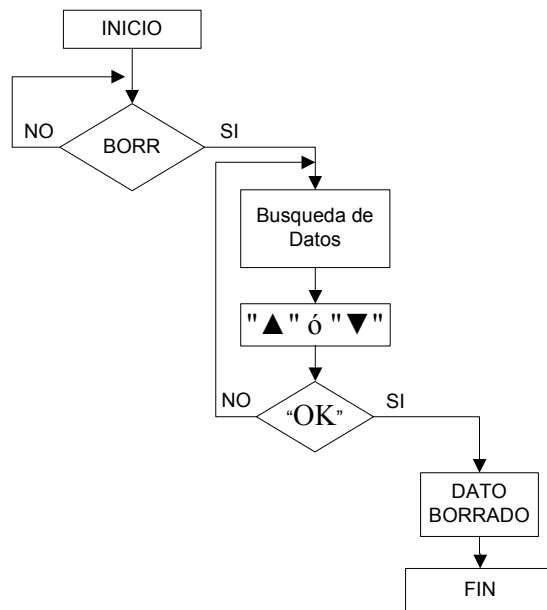


**Figura III.71.** Diagrama Procesos para Ingreso de Nuevo Usuario

*Proceso de Borrado de un Usuario:* Al igual que el proceso de actualización de datos este proceso se inicia empleando el proceso de listado de Usuarios, una vez hallado al usuario de presionará la tecla de confirmación y aparecerá un aviso de "Usuario Borrado" y por último se presionará nuevamente el "OK".

### 3.3.7.3 Programación de Borrado de un Usuario

La programación de la etapa de Borrado de un Usuario se le representa en el diagrama de bloques de la Figura III.72. en la cual se muestra de una manera bien clara todo el proceso de borrado y la estructura del programa.



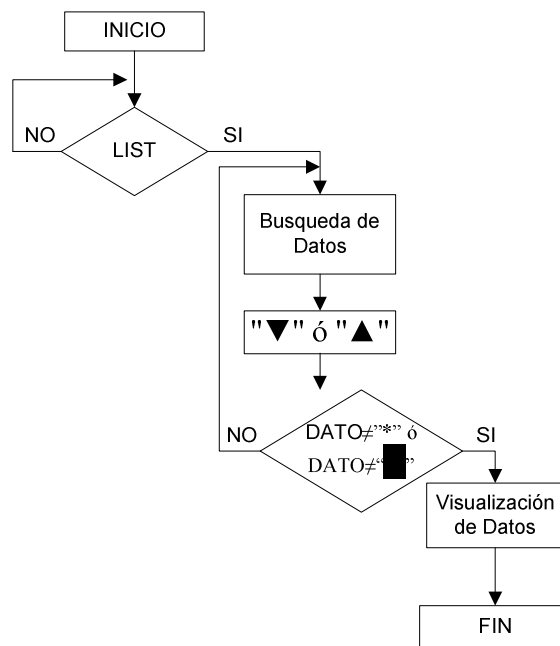
**Figura III.72.** Diagrama Procesos para Borrado de Usuario

*Proceso de Listado de Usuario:* En este proceso se listara todos los usuarios que se encuentran almacenados con sus respectivos números telefónicos. Para este proceso se necesita estar fuera de cualquier proceso, para el listado se debe presionar la tecla "0" o marcado con la palabra "LIST".

Para salir del listado de usuarios basta con presionar el botón "OK".

#### 3.3.7.4 Programación del Listado de Usuario

El esquema del programa realizado para el listado de usuarios se presenta en la Figura III.73 para una mejor muy aclara mediante el uso de un diagrama de flujo, el programa se puede visualizar en los anexos adjuntos al documento.

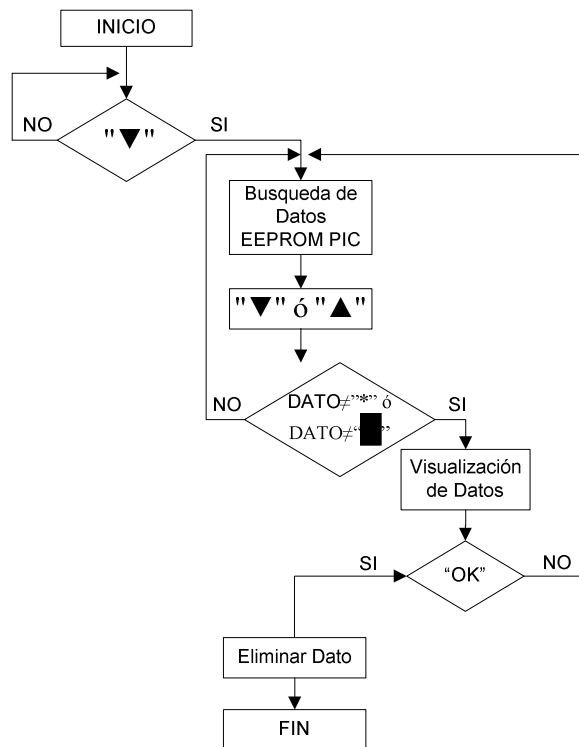


**Figura III.73.** Diagrama de Procesos para Visualización de Usuarios

*Proceso de Listado de Alarmas Activadas:* El sistema posee de una pequeña base de datos creada en el mismo eeprom interna del microcontrolador PIC18F4550 con la finalidad de almacenar en esta el listado de usuarios que han accionado la alarma, cabe decir que esta lista es posible borrarla, para la visualización de esta lista basta con que se esté fuera de cualquier proceso y presionar el cursor que marca para abajo ("V") y se visualizará la información que se desea, para la eliminación de un usuario de esta lista basta que en el momento que este mostrando un usuario de esta lista se presione el botón "OK" y automáticamente se eliminara el usuario de esta lista.

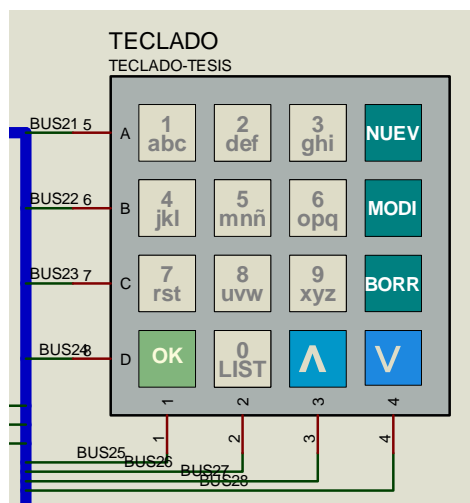
### 3.3.7.5 Programación del Listado de Alarmas Activadas.

El programa que describe esta etapa se presenta en la Figura III.74. en una forma de diagramas de flujo con la finalidad de tener una idea más global de toda la programación que se desarrollo para obtener esta finalidad.



**Figura III.74.** Diagrama de Flujo para Visualización de Alarmas Activadas

La configuración del teclado se muestra en la Figura III.75 con sus respectivos conectores con el microcontrolador (RB0...RB7)



**Figura III.75.** Gráfica de conexión para el Teclado 4x4

### 3.3.8 Etapa de Control Centralizado

Todos los procesos sea de gestión de información como el envío de señales de control para la desactivación y activación de los diferentes dispositivo, en envío de señales de control son controlados por un microcontrolador de gama alta, la razón por la que se escogió este microcontrolador fue por su capacidad de memoria de programa que es de un alto nivel, sus características de comunicación serial, su eficiencia y capacidad de puertos, se puede ver el esquema de esta etapa en la Figura III.76.

El código realizado en este microcontrolador es sumamente extenso por lo que solo se mostrara un diagrama de flujo de los procesos que realiza.

La programación se realizó en lenguaje C debido a que la misma aprovecha en gran parte el espacio de memoria de programa.

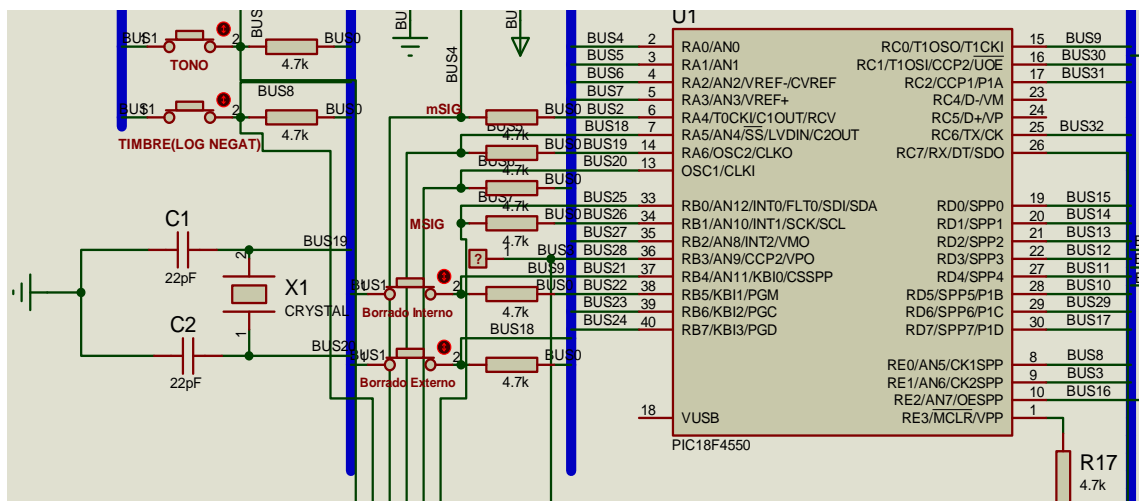
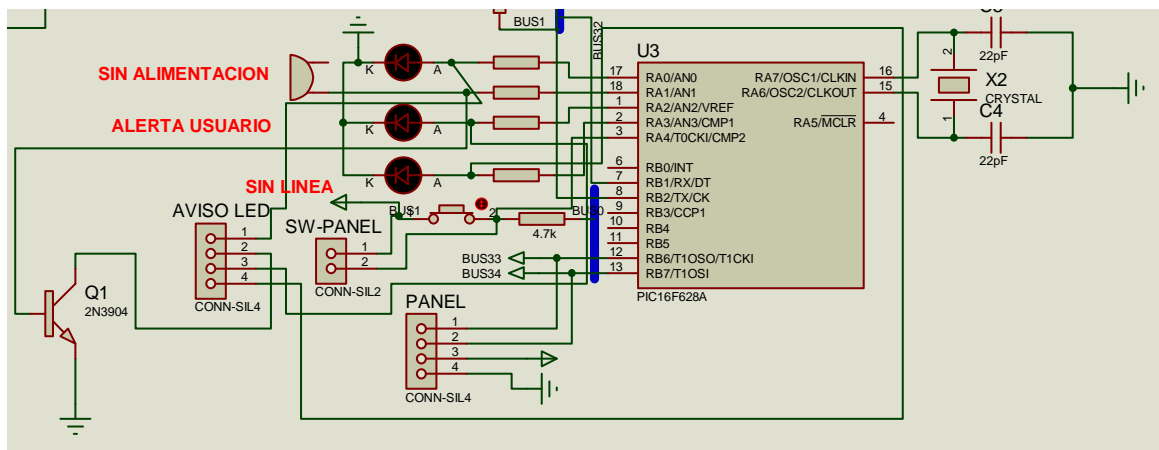


Figura III.76. Gráfica de conexión del microcontrolador 18F4550



### 3.3.9 Etapa de Control de Actuadores

Esta siguiente etapa es encargada de la señalización y avisos de las distintas alarmas como es la de la Activación de una Alarma por un Usuario, Alarma de Falta de Alimentación, Alarma por corte de Línea Telefónica, y el encendido de un Buzzer ya que aparte de existir un aviso visual existe un aviso sonoro, el diagrama circuital se puede ver en la Figura III.77.

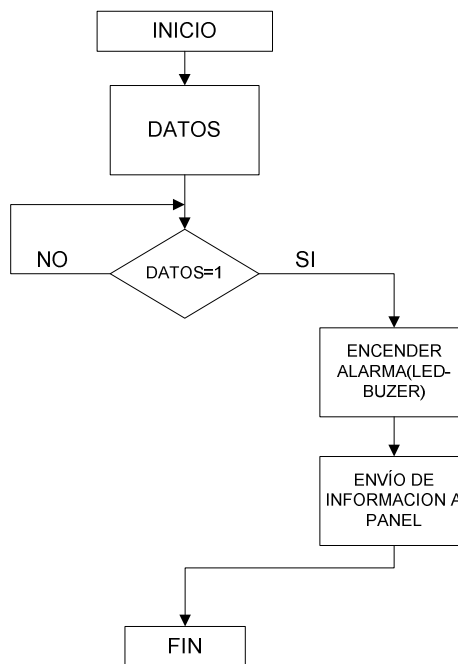


**Figura III.77.** Gráfica del diagrama de conexión del microcontrolador 16F628A

El envío de activación de alarmas desde el microcontrolador central al microcontrolador de actuadores se realiza mediante una comunicación serial usando los puertos que poseen los mismos para la tecnología RS232.

#### 3.3.9.1 Programación de la Etapa de Control de Actuadores

Para un mayor entendimiento del proceso de la que se encarga esta etapa se puede ver el diagrama de bloques que se muestra en la Figura III.78.



**Figura III.78.** Diagrama de Flujo del funcionamiento de la etapa del Actuador

En esta etapa también se encarga del envío serial de la información acerca de las cuadras en donde está ubicado la casa que activó la alarma. Estos datos son enviados vía serialmente a la etapa de panel de visualización.

### **3.3.10** Etapa de Panel de Visualización

La base de esta etapa es un microcontrolador PIC16F876A en la cual se recibe datos serialmente en la que consta información de las cuadras en la que está localizada la alerta, el diagrama circuital se puede ver en la Figura III.79.

La información que le llega a esta etapa es guardada en su memoria EEPROM interna con la finalidad de almacenar las alarmas activadas y desde aquí activar las cuadras que emiten la alerta. Cabe decir que la localización de cada casa que cuenta con el sistema de alerta al momento de su registro se debe registrar de un orden de numeración de calles

ya establecido para una mejor facilidad de procesamiento de datos, cada calle con la que cuenta el barrio es numerada.

Para el ingreso de la información de un nuevo usuario en el momento en que se pida la información de la CALLE1 de deberá ingresar el número correspondiente de la calle principal en la que se encuentra la casa, cuando se pida la CALLE2 y CALLE3 se ingresara en número correspondiente de las calles laterales en las que se localiza la casa a registrar.

Una vez que la alarma de una casa es activada se visualizara las cuadras en donde está localizada la casa en cuestión mediante el encendido de unos Led's en las respectivas cuadras, estas permanecerán encendidas hasta que se presione un botón que se encuentra en el lado izquierdo del panel de visualización.

Se debe mencionar que una vez que se ha visualizado el sector de alerta en el caso que llegue otra nueva alarma en otro sector, ésta última se visualizará junto con la anterior y así sucesivamente hasta que se presione el botón de reseteo.

El motivo por la que se emplea este microcontrolador es que consta de 28 pines, ya que son utilizadas para el encendido de 36 indicadores LD's, se es posible ya que se usa la teoría de multiplexación de salidas, así se ahorran salidas y se optimiza espacio de cableado dentro del panel.

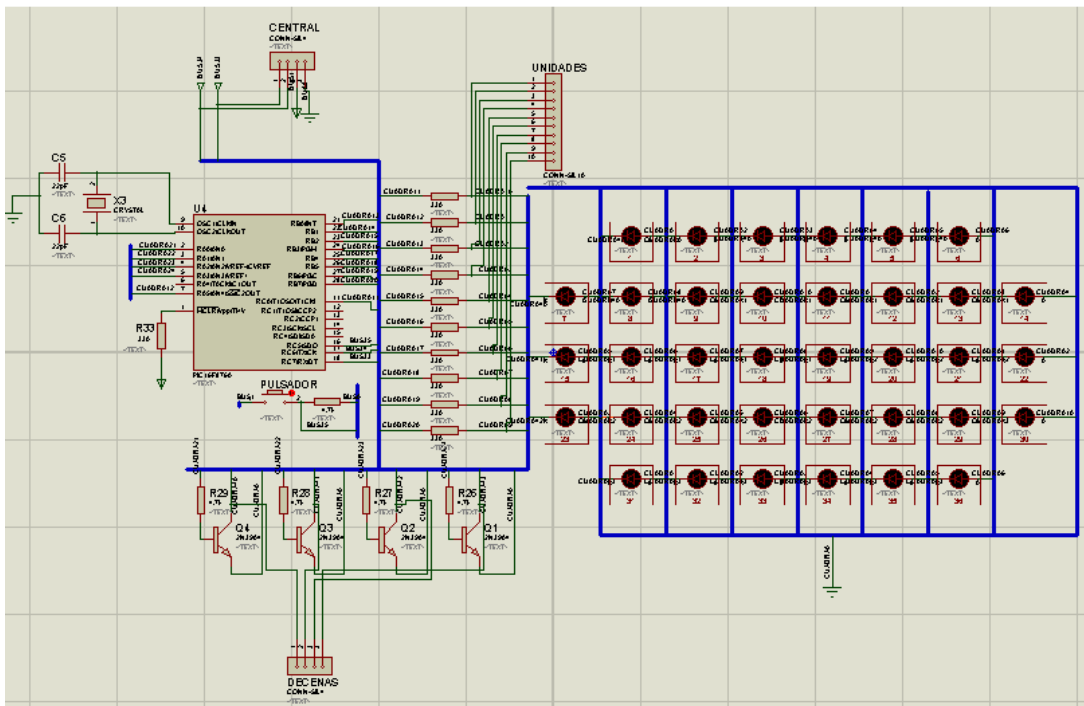


Figura III.79. Gráfica del diagrama de conexión del Panel de Visualización

### 3.3.10.1 Programación del Panel de Visualización.

Posteriormente en la Figura III.80 se muestra un diagrama de bloques del funcionamiento de esta etapa el cual describe toda la programación que se ha realizado.

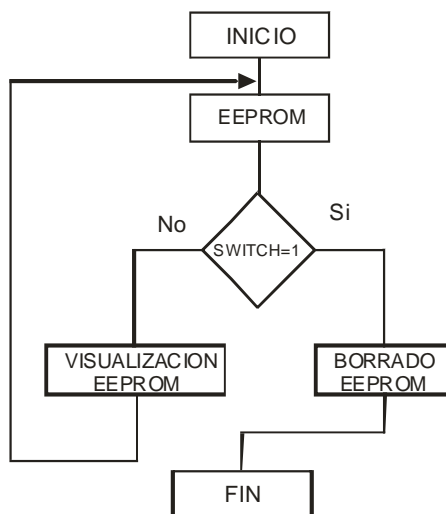


Figura III.80. Gráfica del diagrama de Bloques del Panel de Visualización

## **CAPÍTULO IV**

### **IMPLEMENTACIÓN Y RESULTADOS EXPERIMENTALES**

#### **4.1 IMPLEMENTACIÓN**

El sistema fue diseñado con todos los parámetros que se puede requerir para un funcionamiento optimo y una presentación excelente, aquí se indica los circuitos pasados en una placa de cobre, para implementar dichos circuitos se utilizo el programa Ares de Proteus 7.6, software muy potente y fácil de manejarlo.

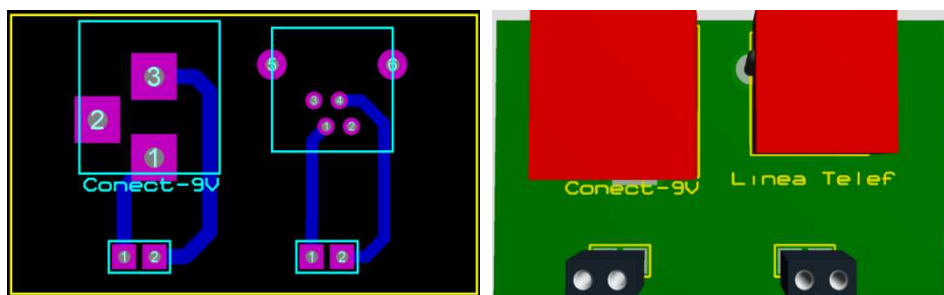
Las gráficas siguientes son las plaquetas diseñadas con el software antes mencionado, junto a estos se encuentran las graficas en 3D de las mismas.

Cabe decir que algunos de los componentes fueron creadas por el usuario ya que el software no tubo incluido en sus librerías como son algunos de los conectores de alimentación y conectores de transformadores.

## 4.2 TRANSMISION.-

### 4.2.1 PCB del Circuito de Conectores de Alimentación

Esta placa consta de conectores para la alimentación de 12 voltios y un conector para la línea telefónica RJ11. En la Figura IV.81 se puede ver el diseño de la misma junto con la grafica en 3D de la plaqueta.



**Figura IV.81.** Gráfica en 3D y pista del circuito de conectores de alimentación-  
Transmisor

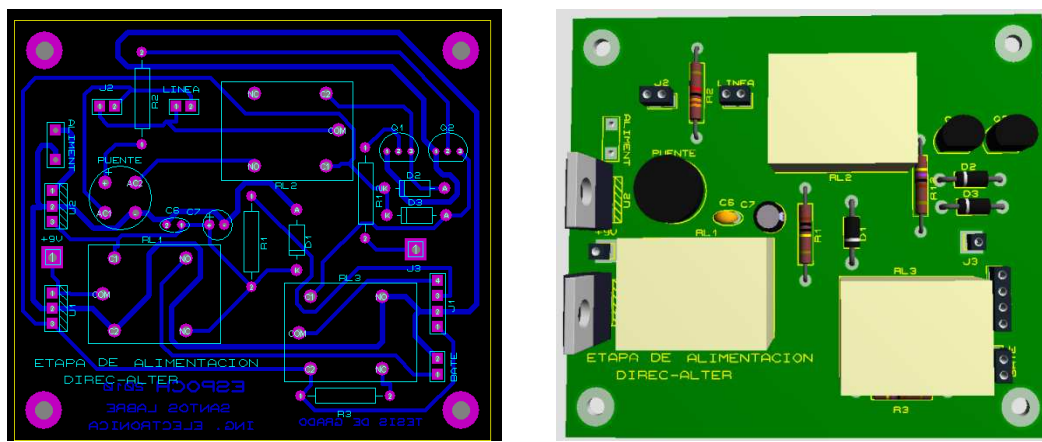
El la grafica IV.82 se puede visualizar la plaqueta resultante con sus respectivos componentes.



**Figura IV.82.** Resultado de diseño de la plaqueta de conectores de alimentación-  
Transmisor

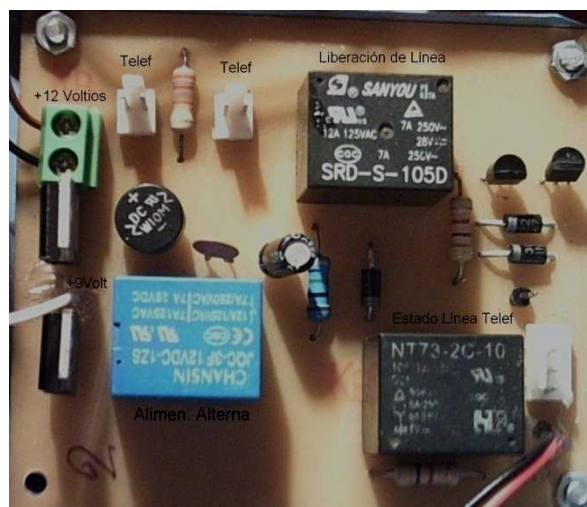
#### 4.2.2 PCB del Circuito de Fuente de Alimentación

En la Figura IV.83 se muestra la gráfica del diseño de la plaqueta del circuito de Fuente de Alimentación y su visualización en 3D del mismo, en la cual se observan 3 relés como componentes principales para la captación de los diferentes estados de señales, sea de alimentación, estado de línea, y liberado de línea telefónica.



**Figura IV.83.** Gráfica en 3D y pista del circuito de Fuente de Alimentación-Transmisor

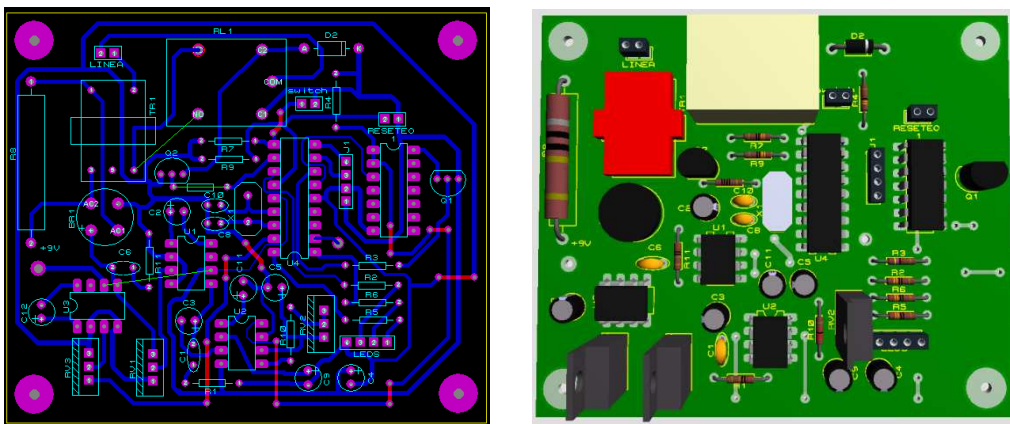
En la figura IV.84 Podemos mostrar el resultado del diseño de la plaqueta mencionada con los resultados deseados.



**Figura IV.84.** Gráfica resultante del circuito de Fuente de Alimentación-Transmisor

#### 4.2.3 PBC del Circuito de Generación de Tonos

En la Figura IV.85. podemos ver la gráfica en 3D y la pista del circuito de Generación de Tonos en la cual se encarga de la generación de tonos y captación de señales de timbrado, junto con la etapa de visualización de alertas de alimentación, cortes de línea y transmisión de datos.

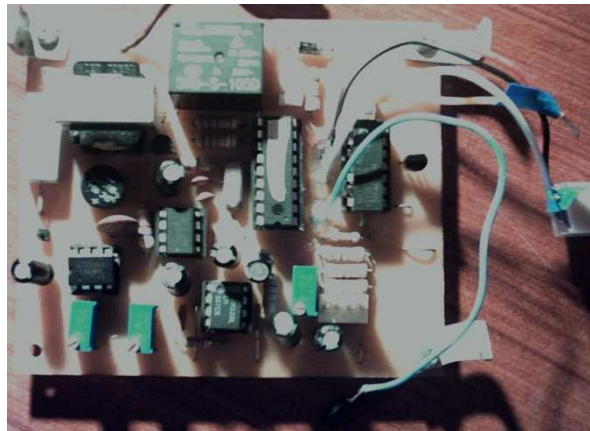


**Figura IV.85.** Gráfica en 3D y pista del circuito de Generación de Tonos-Transmisor

En la Figura IV.86. se puede ver el resultado de la plaqueta anteriormente mencionada, en la grafica se puede ver los conectores ya implementado y cables de comunicación para las diferentes etapas.

En la cual podemos ver claramente los componentes importantes y vitales del transmisor como es el microcontrolador 16f628a, amplificadores LM567 y un relé que realiza la etapa de descolgado de línea.



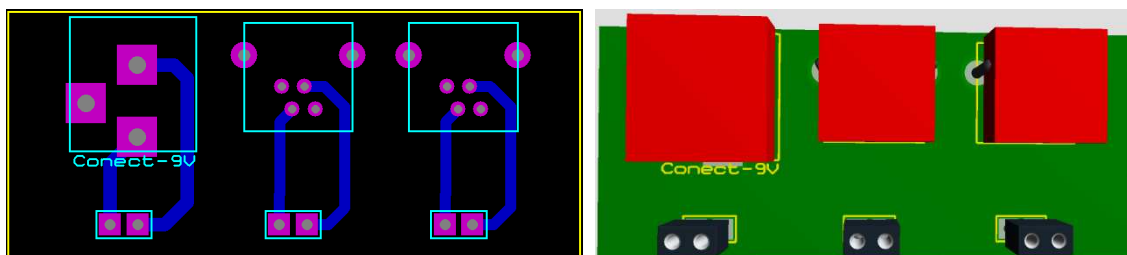


**Figura IV.86.** Gráfica resultante del circuito de Generación de Tonos-Transmisor

### 4.3 RECEPTOR.-

#### 4.3.1 PCB del Circuito de Conectores de Alimentación

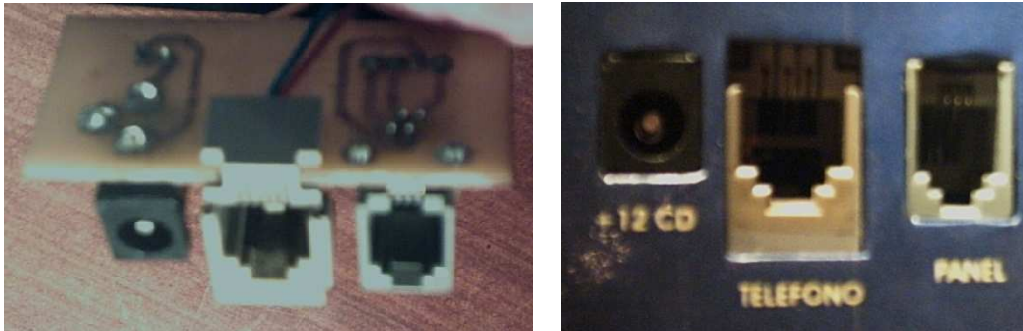
En la Figura IV.87 se puede ver la gráfica en 3D y la pista diseñada para la plaqueta de los conectores de alimentación y línea telefónica para la Central de Recepción.



**Figura IV.87.** Gráfica en 3D y pista del circuito de conectores de Fuente de Alimentación-Receptor

El resulta del diseño de la plaqueta de conectores antes mencionado se muestra en la Figura IV.88, en la cual se muestra el diseño en la plaqueta y colocada en la central receptora lista para su conexión, el conector que se ha etiquetado como “PANEL” se lo

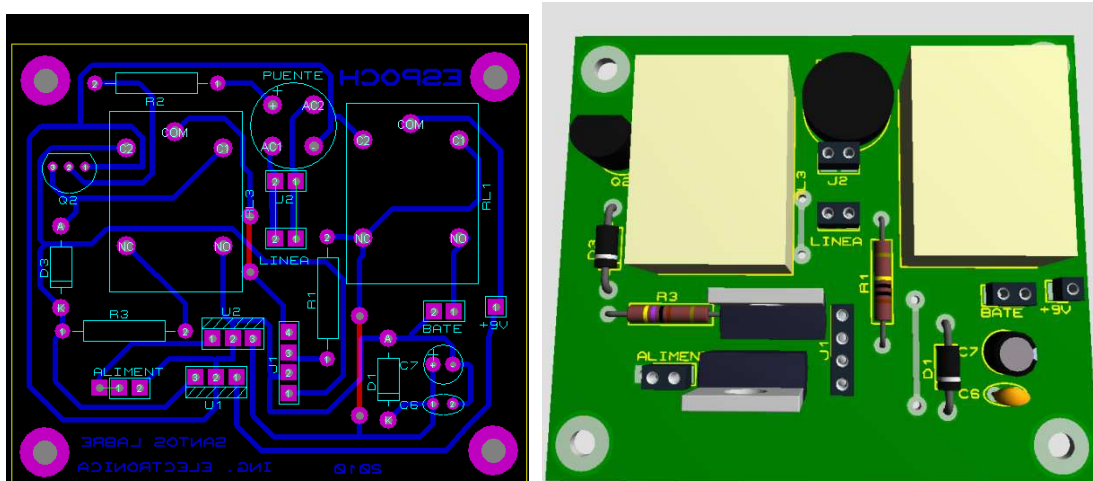
utilizara para la conexión con un panel externo con la finalidad de mejorar la ubicación de la cara que se encuentra en alerta.



**Figura IV.88.** Resultado del circuito de conectores de Fuente de Alimentación-Receptor

#### 4.3.2 PCB del Circuito de Fuente de Alimentación

Tal como se muestra en la Figura IV.89 la parte fundamental de esta plaqueta son los dos Relés en la cual son empleados para la detección de alarmas que son el corte de la línea telefónica y el corte de energía eléctrica.



**Figura IV.89.** Gráfica en 3D y pista del circuito de Fuente de Alimentación-Receptor

En la Figura IV.90 se muestra el resultado de la plaqueta que se ha diseñado con las especificaciones anteriormente mencionadas.

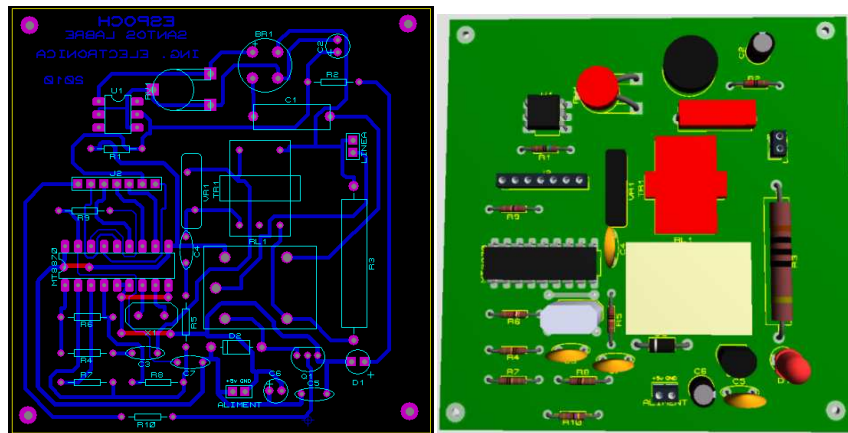


**Figura IV.90.** Gráfica del circuito terminado de la Fuente de Alimentación-Receptor

#### **4.3.3** *PBC del Circuito Detector de Timbre*

Como se muestra en la Figura IV.91 la parte importante es el CI MT8880 que es el que codifica las señales de tonos con la que se determina en forma decimal el número de casa que esta activando la alarma y un Relé que conmuta la identificación de estos dígitos.

En estas graficas se muestra el diagrama de las pistas impresas y el resultado en 3D de este circuito.



**Figura IV.91.** Gráfica en 3D y pista del Circuito Detector de Timbre

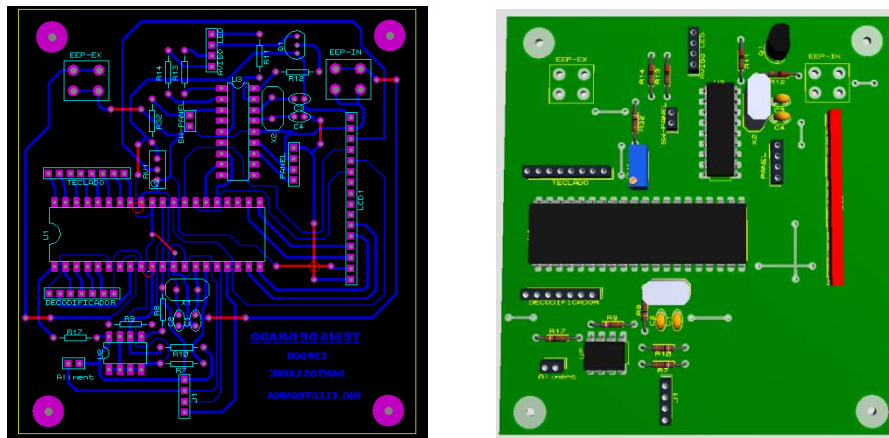
En la Figura IV.92 se muestra el circuito resultante e implementado en la plaqueta, esta es colocada en la caja de la Central Receptora, cabe decir que para la conexión y comunicación con los demás módulos se realizan por medio de conectores Molex para tener una mejor sujeción y evitar desconexiones accidentales.



**Figura IV.92.** Gráfica en 3D y pista del Circuito Detector de Timbre

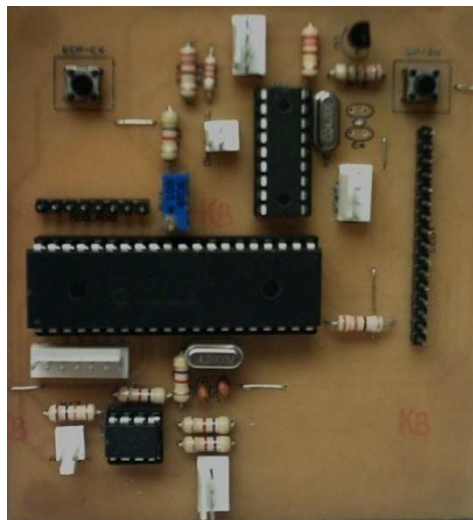
#### **4.3.4** PBC del Circuito de Control Central de Procesamiento

El cerebro principal de esta etapa o en sí de todo el sistema es el PIC18F4550 que consta de 40 pines, como se ve en la Figura IV.93. en la que se ve el diseño de la pista y su resultante en 3D del diseño.



**Figura IV.93.** Gráfica en 3D y pista del Circuito de Control Central de Procesamiento.

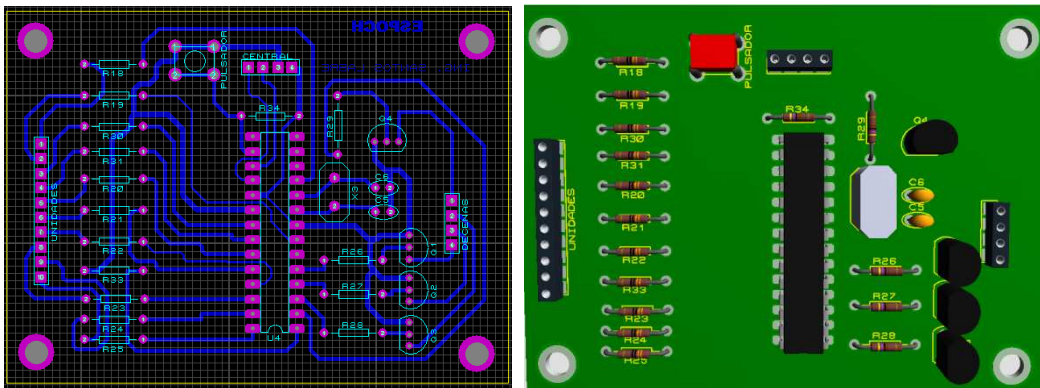
El circuito ya implementado se muestra en la Figura IV.94, en la que se puede visualizar todos los componentes, entre los más importantes es el PIC18F4550, el PIC16F628A y los correspondientes conectores para el teclado matricial y el visualizador LCD.



**Figura IV.94.** Diseño terminado de la plaqueta de Control Central d Procesamiento.

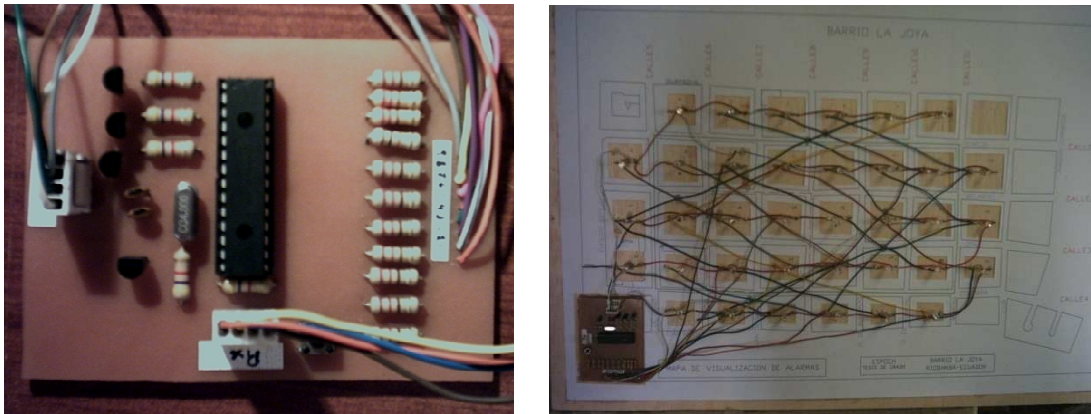
#### 4.3.5 PBC del Circuito del Panel de Visualización

En la Figura IV.95. se muestra el diseño de la plaqueta del panel de Visualización, puede verse el diseño de la pista y una gráfica en 3D, como parte muy importante de esta etapa se puede ver el microcontrolador PIC16F876A.



**Figura IV.95.** Gráfica en 3D y pista del Circuito del Panel de Visualización.

El caro resultado de este diseño lo podemos ver en la Figura IV.96. en la cual se muestra todos los componentes antes mencionados, junto con la conexión de los indicadores y croquis del sector a cubrir.



**Figura IV.96.** Diseño terminado del Panel de Visualización.

Para pasar el circuito a la placa se utilizo papel fotográfico o también llamado papel glosi siendo este una excelente alternativa para imprimir las pistas conductoras en una placa de cobre, obteniendo los resultados óptimos de todos los circuitos antes mencionados.

#### **4.4 RESULTADOS EXPERIMENTALES**

El consumo de corriente y tiempos de retardo en el accionamiento de la alarma se especifica a continuación:

##### *Sistema Transmisor*

###### **En Reposo:**

Corriente: 0,056 Amperios

Voltaje de consumo: 12 Voltios

###### **Alarma encendida:**

Corriente: 0,188 Amperios

Voltaje de consumo: 12 Voltios

###### **Tiempo de retardo de activación:**

Tiempo de activación: 2 segundos

Tiempo de espera antes de envío de señal: 16 segundos

Tiempo de transmisión de datos: 2 segundos

Duración de llamada completa desde que se presiona el botón: 20 segundos

##### *Corriente de consumo del Transmisor*

###### **En Reposo:**

Corriente: 0,082 Amperios

Voltaje de consumo: 12 Voltios

###### **Alarma encendida de falta de alimentación o línea:**

Corriente: 0,122 Amperios

Voltaje de consumo: 12 Voltios

###### **Tiempo de retardo de activación de alarma:**

Tiempo para realizar el descolgado: 2 segundos

Tiempo de espera tiempo de espera para visualización de panel: 4 segundos

Tiempo de duración de llamada: 8 segundos

Total de recepción de alarma 11 segundos

El sistema SAC tarda alrededor 20 segundos en alertar u sector en peligro, desde que se presiona la alerta hasta que se visualiza el sector en alerta en el panel de visualización, esto puede aumentar dependiendo si la línea telefónica del receptor esté o no ocupada.

A continuación en la Tabla IV.IV. se presenta datos estadísticos de eficiencia del sistema tomadas por medio de un determinado número de pruebas realizadas en distintas circunstancias.

Número de Pruebas realizadas	Número de aciertos efectivos
10	9

**Tabla IV.IV.** Tabla de Estadística de Pruebas



Cabe decir que el porcentaje fallido fue por motivo que la línea con la que se trabajo en la central no fue una línea dedicada y en esos momentos se receiptó una llamada convencional.



## **CONCLUSIONES**

Los dispositivos que se han estudiado y seleccionado para el desarrollo de la alarma han sido el uso de microcontroladores como dispositivos pasivos.

La señal que se ha transmitido por la línea telefónica es en base en tonos telefónicos en la cual tenemos resultados muy favorables.

Se ha comprobado que la utilización de microcontroladores tiene una amplia gama de utilidades como es la del envío de señales de tono, de esta manera da un hincapié al desarrollo de nuevos sistemas más avanzados con la misma técnica.

El sistema esta constando de dos medios de visualización de datos y localización de alertas, siendo estas por medio de un LCD al igual que por medio de luces indicadoras que ayudan a su fácil comprensión y un panel de visualización esquemático implementado en un panel de visualización en la que consta de el diagrama completo de todo el barrio que va abarcar, el acoplamiento del LCD al microcontrolador no tuvo grandes problemas y su correspondiente manejo en el envío y captura de datos del mismo.

Se ha diseñado e implementado el circuito electrónico para el sistema de alarmas comunitarias en base a programas de simulación y programación de microcontroladores como es el lenguaje de programación C en la cual tiene grandes ventajas debido a su

aprovechamiento de memoria de programa y su mejor manipulación de puertos, cabe decir que para su mejor aprovechamiento de las características de un microcontrolador de gama Alta es más favorable que la programación sea en C.

## **RECOMENDACIONES**

La manipulación de equipo de recepción se debe regir en base al manual proporcionado por el diseñador para obtener el mejor rendimiento del equipo y para la aclaración de incógnitas que pueden aparecer en el transcurso de su uso.

En la central de recepción no consta de switch de encendido por motivos de seguridad, este dispositivo es para uso de las 24 horas al día, por ello en el caso que se necesite realizar un apagado de emergencia se recomienda la desconexión de su batería interna y de la desconexión del adaptador externo.

En el caso que se realice un ingreso erróneo de la información de un usuario se recomienda realizar una posterior modificación del mismo o una eliminación del mismo, ya que esta información servirá para la localización del sector de emergencia en el panel de visualización.

Para que sea enviado una alerta de alarma la línea del transmisor no debe estar en uso ya que si lo está no llegará al destino de la central las señales deseadas, de esta misma manera la central no debe estar en uso por el mismo motivo que el transmisor.

En el caso que el número de usuarios rebase la capacidad de la memoria EEPROM esta puede ser cambiada por otra de mayor capacidad sin problemas teniendo en cuenta que se debe realizar un nuevo ingreso de usuarios en el otro dispositivo.

El transmisor sirve solo para enviar señales de alerta dirigidos solo a la central de auxilio pero esta señal no puede ser retransmitidos para los demás usuarios ya que los transmisores que se encuentran distribuidos en el barrio no cuentan con un sistema de recepción.

En el caso de fallo del sistema se recomienda realizar la desconexión y conexión del sistema así para que el sistema borre todo proceso con mal funcionamiento.

Si se necesita realizar un borrado rápido por emergencia de la lista de usuarios o como de alarmas efectuadas se puede proceder a presionar los botones correspondientes internos que se encuentran en la parte interna del receptor.

Para el funcionamiento de la alarma no es indispensable que el panel de visualización esté conectado, ya que en el visor de la central mostrará el nombre del usuario que activó la llamada de emergencia y el número correspondiente de este usuario.

En el caso que el dispositivo transmisor de la alarma sea cambiada de casa, los datos de esta deben ser actualizados en la central junto con el nombre del nuevo usuario y principalmente la dirección de ubicación ya que ya que si no se lo hace en el momento en que se realice la activación de la alarma esta avisara con la dirección y nombre del usuario anterior.

Se recomienda tener una línea dedicada para la central receptora ya que al momento en que se reciba una llamada de que no sea de alerta, esta última será recibida por la central esperando las señales de ubicación de alarma, en la cual no llegara por ningún motivo y

eso hará que la central proceda a realizar un colgado de la línea, y no será atendida por el usuario con quien se quiere hablar.

## **RESÚMEN**

Se diseñó e implementó un Sistema de Alarma Comunitaria (SAC) para el barrio la Joya de la Ciudad de Riobamba Provincia de Chimborazo, con la finalidad de optimizar y mejorar las llamadas de alerta posibles desde el barrio hacia la policía.

Para el desarrollo del SAC se emplearon software de programación CCS PCWH, Microcode Basic y Proteus 7.6, computador personal, dispositivos electrónicos. La estructura de la Central está construida de placas de fibra de vidrio junto con sus respectivas estaciones de alerta, las mismas que generan la señal de auxilio tan solo presionando un botón, que genera una llamada telefónica hacia el receptor que visualiza la información de la estación que genero la alerta, todo el sistema está basado en microcontroladores y su medio de comunicación son líneas telefónicas.

El alcance del sistema está diseñado para dar cobertura a 4 cuadras del barrio con posibilidad de ampliación.

Este sistema es posible ser usado en cualquier lugar de la ciudad mientras se disponga de una línea telefónica para la transmisión y otra para la recepción.

## **SUMMARY**

If was design and implemented a Community Alarm System (SAC) for the La Joya neighbourhood, city of Riobamba, Chimborazo Province, in order to optimize and improve the possible distress calls from the neighbourhood to the police.

For the development of the SAC were used PCWH CCS software programming, and Proteus, Microcode Basic 7.6, computer, electronic devices. The central structure is fibreglass panel, along with their warning station which generates the same distress signal just by a button that generates a call to the receiver which displays the station information that generated the alert, the whole system is based on microcontrollers and their media are telephone lines. The scope of the system is designed to cover four blocks from the neighbourhood with possible extension.

This system can be used anywhere in the city when there is no phone line for transmission and another for the reception.

## **ANEXOS**



## **Anexo A.** Manual de usuario

### TRANSMISOR

El equipo transmisor del sistema SAC se encuentra en una estructura portable en la que consta de visores de alarmas (LED's) con fácil interpretación, un botón de encendido de alarma, un switch de encendido del equipo y conectores de alimentación como de línea telefónica.

*Vista frontal.*

Se observa claramente los LED's indicadores y el botón de encendido de alerta, en la cual con su respectiva señalización de cada una de las mismas.



Para la activación de la alarma se procede a presionar el botón de color rojo por 3 segundos en la cual realiza la llamada de alarma hacia la central.



Una vez presionado esta realiza la llamada y avisa a la central que hay una alerta, cuando el transmisor envía la llamada de alerta se enciende el indicador de “TRANSMITIENDO” hasta que finalice la transmisión, para posteriormente liberar la línea telefónica, en el caso que no se realice la transmisión de la alerta ya sea por motivo s de línea ocupada la misma espera un tiempo para realizar remarcado automático y así hasta que se realice la llamada con éxito de la alarma.

Si existe un corte de Luz este dispositivo tiene internamente una batería de respaldo en la cual es usada en ese momento y se encenderá una luz indicadora de “ALIMENTACION ALTERNA”, pero aun así el dispositivo funcionará con normalidad ya que la alimentación eléctrica es distinta a la alimentación telefónica.



En el caso que se haya producido un corte de la línea telefónica se encenderá el indicador de “ALERTA DE LINEA”, si es este caso no se podrá realizar ninguna llamada de alerta hasta que se tenga el servicio telefónico completamente.



*Vista Lateral.*

Aquí se puede visualizar los conectores de alimentación y de la línea telefónica, también el switch de encendido del dispositivo., cabe decir que en esta parte también se encuentra el número del dispositivo en la cual se debe basar para su registro en el sistema.



La conexión de este dispositivo es de la alimentación como de la línea telefónica.

La alimentación es de 12 voltios de corriente continua con un amperaje de 500mA. de la cual se lo puede obtener de un adaptador de corriente con los parámetros antes mencionados.

La línea telefónica es del tipo convencional brindada por la CNT (Consejo Nacional de Telecomunicaciones).

## CENTRAL RECEPTORA

*Vista Frontal.*

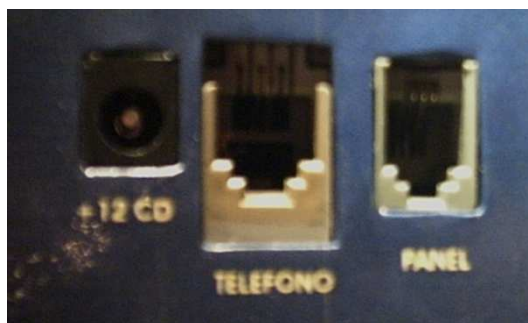
En la siguiente grafica se muestra el dibujo de la central receptora en la que se puede visualizar.



En el momento que el sistema está en reposo presenta un mensaje en pantalla “SISTEMA COMUNITARIO”, el sistema posee al igual que un LCD 16x4 luces indicadoras en las que nos presenta en estado de la línea como el de la alimentación eléctrica al igual que los equipos transmisores. Más adelante se definirá el manejo e interpretación del sistema.

### *Vista lateral*

En la parte lateral se tiene 3 conectores (Alimentación, Línea Telefónica y Panel de Visualización) esto se puede ver en la siguiente figura.



En el conector de alimentación se puede suministrar un voltaje de 9 a 12 voltios de corriente continua de la que se puede obtener de un adaptador que proporcione ese voltaje a 1 Amperio.

La conexión de la línea telefónica se lo hace en el conector que está claramente marcada con la palabra “TELEFONO”, esta es conectada a una línea convencional telefónica proporcionada por la CNT.

El siguiente conector es usado para la comunicación con el panel de visualización, cabe decir que este panel no es esencial para el funcionamiento del sistema.

### *Registro de un nuevo usuario*

Para el registro de un nuevo usuario de ante mano se debe tener el nombre del Usuario, número de teléfono, nombres de calles en la que se ubica la casa en donde se va a colocar el transmisor de alarmas, estas direcciones son; calle principal en la que se ubica, y las dos calles transversales.



Una vez que se tiene los datos antes mencionados se debe presionar el botón color blanco que se encuentra en la parte superior derecha con la palabra “NUEV”. Una vez presionada la aparecerá los campos que tiene que llenar, cabe decir que los datos a ingresar se lo hace por medio del teclado que se encuentra a lado derecho del equipo, las teclas están claramente marcadas con las letras correspondientes, el funcionamiento del teclado es muy parecido a la de un celular, es decir que cada botón tiene tres caracteres posibles para ser utilizados, junto con un valor numérico usado para el ingreso del número telefónico y dirección de calles que posteriormente hablaremos de ello.



Para cambiar la letra que se desea ingresar se presiona el botón que contiene la letra, en el caso que la letra no sea la que se encuentra en la primera posición del botón se presionara las veces correspondientes el mismo botón hasta que aparezca en el visor LCD la letra deseada.

Para escribir la siguiente letra se pulsara la tecla “▼” que se encuentra a lado inferior derecho, y así se colocará el cursor a la siguiente posición para proceder a escribir la letra siguiente.



En el caso que se quiera corregir una letra mal escrita en la posición anterior se presiona la tecla ” ▲” y después ingresar la letra deseada, una vez corregido para volver a la posición en blanco deseada se presionara dos veces la tecla “▼” para seguir con el ingreso de datos.

Cuando se ha finalizado de ingresar un dato completo, para su confirmación se deberá presionar la tecla “OK” que se encuentra a lado inferior izquierdo.



Cuando nos pida el ingreso de la dirección ésta debe ser ingresado en forma de numero, la numeración se basa en la siguiente tabla.

CALLE	NUMERO
OLMEDO	01
VILLAROEL	02
CHILE	03
COLOMBIA	04
5 DE JUNIO	05
TARQUI	06
J. D VELAZCO	07
S. DE BENALCAZAR	08
P. DE ALVARADO	09
D. DE ALMAGRO	10
MORONA	11

Cuando se pida en ingreso de la “CALLE1” esta debe ser la calle principal en donde está situada la casa que cuenta con el dispositivo transmisor, en el caso que se pida la “CALLE2” y “CALLE3” estas son las calles laterales en la que se encuentra la casa. Cabe decir que en el caso que corresponda una calle que este dentro del rango del 1 a 9 se deberá ingresar el dicho número precedido por el “0”.





Una vez ya ingresado los datos aparecerá un mensaje “USUARIO GRABADO” el cual confirmará que los datos han sido guardados con éxito. Para salir de este mensaje se debe presionar el botón “OK” y así regresar al estado de reposo del sistema.



#### *Modificación de un Usuario*

En esta opción se puede modificar la información de un usuario ya registrado. Para eso se debe presionar el botón “MOD” que se encuentra en la parte de abajo del botón de ingreso de nuevo usuario.



Una vez elegido la opción de modificación se visualizará los datos del primer usuario, para la elección del usuario a modificar de deberá presionar en botón “▼” para visualizar el siguiente usuario, este botón se presionará hasta que se llegue a la información deseada para la modificación, después se debe presionar el botón “OK”

para proceder a la modificación. Los datos se deberán ingresar de la misma manera como el ingreso de un nuevo usuario.

En el caso que se haya pasado el usuario accidentalmente se puede retroceder la visualización presionando el botón "▲".

### *Eliminación de un Usuario*

Esta opción tiene la finalidad de eliminar un usuario registrado, basta con presionar el botón "BORR" que se encuentra en la parte de abajo del botón de modificación.



Una vez elegido la opción aparecerá la información del primer usuario en la cual se deberá elegir al usuario a eliminar de la misma manera que se lo hizo para realizar la modificación de un usuario.

En el caso que se haya pasado el usuario accidentalmente se puede retroceder la visualización presionando el botón "▲".

### *Visualización del Listado de Usuarios*

Para visualizar los usuarios registrados se deberá estar en el estado de reposo de la central. Una vez en este estado se debe presionar el botón marcado con el número “0” ó “LIST” que se encuentra en la parte inferior del teclado.



Al momento de presionar se visualizará la información del primer usuario registrado, para visualizar la información del siguiente usuario se debe presionar el botón “▼”, en el caso que se quiera visualizar la información del usuario anterior se debe presionar el botón “▲”.

#### *Visualización del Historial de Alarmas Activadas*

Para la visualización de las alarmas activadas se debe estar en el estado de reposo y posteriormente presionar el botón “▼” en la cual se visualizará la información de la última alarma activada.



El orden de visualización va desde la última alarma activada hasta la primera, cabe decir que este listado puede ser borrado en el momento en que se presione el botón “OK”, en la cual se borrara del historial los datos del usuario que se encuentra visible en el LCD.

En el caso que no se quiera borrar el historial se debera llegar al final de la lista y presionar el botón “OK” para regresar al estado de reposo del sistema.

#### *Encendido de Alarma.*

Cuando se reciva una señal de alerta la central activara una señal visual, como es de la visualizacion de la información de la persona que activó la alarma junto con una alerta sonora y un encendido luminoso de un LED.



Tambien se podrá visualizar el sector que está en alerta en el Panel de Visualización.



Para el apagado de la alerta se debe presionar el botón “OK” el cual apagara la alerta sonora, el LED de alerta y la informacion del usuario que activo la alarma.

### *Panel de Visualización*

En el panel se encuentra la grafica del barrio a cubrir en la cual se encuentran en cada cuadra unas luces indicadoras, las que se encenderán al momento de recibir una señal de alerta indicandonos el sector presiso en donde esta la casa que activo la alarma.



En el momento que se active la alarma a pesar de que se presione el botón para el apagado de las señales de alerta el panel va a seguir mostrando en sector preciso de alerta hasta que se presione el botón Rojo que se encuentra a lado derecho del panel.



El panel está diseñado para mostrar las diferentes alarmas que se puedan activar, los LED indicadores indican los sectores en alerta en el cual en el caso que no se presione el botón de borrado de alertas que se encuentra en el panel el panel puede recibir nuevas señales de alerta y mostrarlos sin que se pierda los anteriores.

**Anexo B.** Descripción de cada pin del PIC16F876A

PIN	DESCRIPCIÓN
OSC1/CLKIN(9)	Entrada de cristal oscilador/entrada reloj externo.
OSC2/CLKOUT(10)	Salida del oscilador. Esta patilla saca $\frac{1}{4}$ de la frecuencia de reloj de la patilla OSC1, e indica la duración del ciclo de instrucción.
MCLR/VPP/THV(1)	Entrada de reset por nivel bajo, entrada de tensión de programación o alta tensión de prueba de control.
RA0/AN0(2)	RA0 puede ser, también, la entrada analógica 0.
RA1/AN1(3)	RA1 puede ser, también, la entrada analógica 1.
RA2/AN2/VREF-(4)	RA2 puede ser la entrada analógica 2 o la tensión negativa de referencia analógica.
RA3/AN3/VREF+(5)	RA3 puede ser la entrada analógica 3 o la tensión positiva de referencia analógica.
RA4/T0CKI (6)	RA4 puede ser, también, la entrada de reloj del Temporizador 0. La salida es en drenador abierto.

RA5/SS#/AN4(7)	RA5 puede ser, también, la entrada analógica 4 o la selección de servidor para el puerto serie síncrono.
RB0/INT(21)	RB0 puede ser, también, la interrupción externa.
RB1(22)	RB1 puede ser, también, la interrupción externa.
RB2(23)	RB2 puede ser, también, la interrupción externa.
RB3/PGM(24)	RB3 puede ser, también, la entrada de programación.
RB4(25)	Patilla de cambio de interrupción activa.
RB5(26)	Patilla de cambio de interrupción activa.
RB6/PGC(27)	Cambio de interrupción activa o reloj para programación en serie.
RB7/PGD(28)	Cambio de interrupción activa o reloj para programación en serie.
RC0/T1OSO/T1CKI(11)	RC0 puede ser, también, la salida de oscilación del Timer1 o la entrada de reloj del Timer1.
RC1/T1OSI/CCP2(12)	RC1 puede ser, también, la entrada de oscilación del Timer1, la entrada de Captura2, salida de Comparación2 o salida de PWM2.

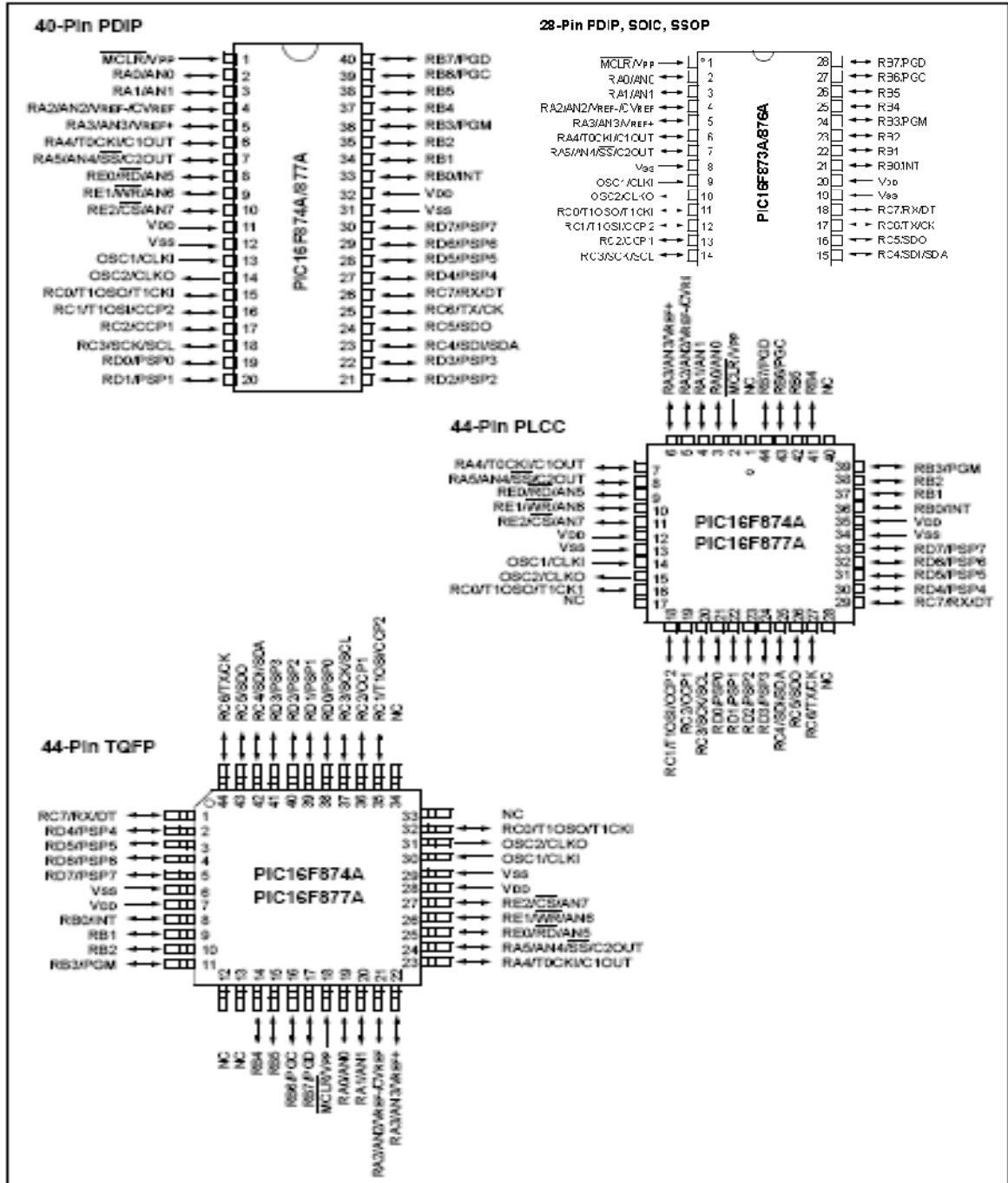


RC2/CCP1(13)	RC2 puede ser, también, la entrada de Captura1, salida de Comparación1 o salida de PWM1.
RC3/SCK/SCL(14)	RC3 puede ser, también, entrada de reloj síncrono serie o salida del bloque SPI en modo I <sup>2</sup> C.SPI e I2C.
RC4/SDI/SDA(15)	RC4 puede ser, también, el dato de entrada en modo SPI o el dato de entrada salida en modo I <sup>2</sup> C.
RC5/SDO(16)	RC5 puede ser, también, el dato de salida en modo SPI.
RC6/TX/CK(17)	RC6 puede ser también, el reloj de la USART.
RC7/RX/DT(18)	RC7 puede ser, también, el dato de la USART.
Vss(8,19)	Conexión de 0V para circuitos lógicos y entradas/salidas.
VDD(20)	Alimentación positiva para lógica y entradas/salidas.

Anexo C. Datasheet del Microcontrolador PIC16F876A

# PIC16F87XA

## Pin Diagrams (Continued)



# PIC16F87XA

FIGURE 2-3: PIC16F876A/877A REGISTER FILE MAP

File Address	File Address	File Address	File Address
Indirect addr. <sup>(1)</sup> 00h	Indirect addr. <sup>(1)</sup> 80h	Indirect addr. <sup>(1)</sup> 100h	Indirect addr. <sup>(1)</sup> 180h
TMR0 01h	OPTION_REG 81h	TMR0 101h	OPTION_REG 181h
PCL 02h	PCL 82h	PCL 102h	PCL 182h
STATUS 03h	STATUS 83h	STATUS 103h	STATUS 183h
FSR 04h	FSR 84h	FSR 104h	FSR 184h
PORTA 05h	TRISA 85h	105h	185h
PORTB 06h	TRISB 86h	PORTB 106h	TRISB 186h
PORTC 07h	TRISC 87h	107h	187h
PORTD <sup>(1)</sup> 08h	TRISD <sup>(1)</sup> 88h	108h	188h
PORTE <sup>(1)</sup> 09h	TRISE <sup>(1)</sup> 89h	109h	189h
PCLATH 0Ah	PCLATH 8Ah	PCLATH 10Ah	PCLATH 18Ah
INTCON 0Bh	INTCON 8Bh	INTCON 10Bh	INTCON 18Bh
PIR1 0Ch	PIE1 8Ch	EEDATA 10Ch	EECON1 18Ch
PIR2 0Dh	PIE2 8Dh	EEADR 10Dh	EECON2 18Dh
TMR1L 0Eh	PCON 8Eh	EEDATH 10Eh	Reserved <sup>(2)</sup> 18Eh
TMR1H 0Fh	8Fh	EEADRH 10Fh	Reserved <sup>(2)</sup> 18Fh
T1CON 10h	90h	110h	190h
TMR2 11h	SSPCON2 91h	111h	191h
T2CON 12h	FR2 92h	112h	192h
SSPBUF 13h	SSPADD 93h	113h	193h
SSPCON 14h	SSPSTAT 94h	114h	194h
CCPR1L 15h	95h	115h	195h
CCPR1H 16h	96h	116h	196h
CCP1CON 17h	97h	General Purpose Register 16 Bytes 117h	General Purpose Register 16 Bytes 197h
RCSTA 18h	TXSTA 98h	118h	198h
TXREG 19h	SPBRG 99h	119h	199h
RCREG 1Ah	9Ah	11Ah	19Ah
CCPR2L 1Bh	9Bh	11Bh	19Bh
CCPR2H 1Ch	CMCON 9Ch	11Ch	19Ch
CCP2CON 1Dh	CVRCON 9Dh	11Dh	19Dh
ADRESH 1Eh	ADRESL 9Eh	11Eh	19Eh
ADCON0 1Fh	ADCON1 9Fh	11Fh	19Fh
20h	A0h	120h	1A0h
General Purpose Register 96 Bytes	General Purpose Register 80 Bytes	General Purpose Register 80 Bytes	General Purpose Register 80 Bytes
7Fh	EFh	16Fh	1EFh
Bank 0	accesses 70h-7Fh	accesses 70h-7Fh	accesses 70h-7Fh
	7Fh	17Fh	1FFh
	Bank 1	Bank 2	Bank 3

Unimplemented data memory locations, read as '0'.  
<sup>\*</sup> Not a physical register.

**Note 1:** These registers are not implemented on the PIC16F876A.  
**Note 2:** These registers are reserved; maintain these registers clear.

# PIC16F87XA

## 2.2.2 SPECIAL FUNCTION REGISTERS

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

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

TABLE 2-1: SPECIAL FUNCTION REGISTER SUMMARY

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Details on page:		
<b>Bank 0</b>													
00h <sup>(2)</sup>	INDF	Addressing this location uses contents of FSR to address data memory (not a physical register)								0000 0000	31, 150		
01h	TMR0	Timer0 Module Register								xxxx xxxx	55, 150		
02h <sup>(2)</sup>	PCL	Program Counter (PC) Least Significant Byte								0000 0000	30, 150		
03h <sup>(2)</sup>	STATUS	IRP	RP1	RP0	TO	PD	Z	DC	C	0001 1xxxx	22, 150		
04h <sup>(2)</sup>	FSR	Indirect Data Memory Address Pointer								xxxx xxxx	31, 150		
05h	PORTA	—	—	PORTA Data Latch when written; PORTA pins when read								-- 0x 0000	43, 150
06h	PORTB	PORTB Data Latch when written; PORTB pins when read								xxxx xxxx	45, 150		
07h	PORTC	PORTC Data Latch when written; PORTC pins when read								xxxx xxxx	47, 150		
08h <sup>(4)</sup>	PORTD	PORTD Data Latch when written; PORTD pins when read								xxxx xxxx	48, 150		
09h <sup>(4)</sup>	PORTE	—	—	—	—	—	RE2	RE1	RE0	--- -xxx	49, 150		
0Ah <sup>(1,3)</sup>	PCLATH	—	—	—	Write Buffer for the upper 5 bits of the Program Counter					-- 0 0000	30, 150		
0Bh <sup>(2)</sup>	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	24, 150		
0Ch	PIR1	PSPIF <sup>(2)</sup>	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	26, 150		
0Dh	PIR2	—	CMIF	—	EEIF	BCLIF	—	—	CCP2IF	-0-0 0-0	28, 150		
0Eh	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	60, 150		
0Fh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	60, 150		
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	-- 00 0000	57, 150		
11h	TMR2	Timer2 Module Register								0000 0000	62, 150		
12h	T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	61, 150		
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	79, 150		
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	82, 82, 150		
15h	CCPR1L	Capture/Compare/PWM Register 1 (LSB)								xxxx xxxx	63, 150		
16h	CCPR1H	Capture/Compare/PWM Register 1 (MSB)								xxxx xxxx	63, 150		
17h	CCP1CON	—	—	CCP1X	CCP1Y	CCP1M3	CCP1M2	CCP1M1	CCP1M0	-- 00 0000	64, 150		
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RXSD	0000 000x	112, 150		
19h	TXREG	USART Transmit Data Register								0000 0000	118, 150		
1Ah	RCREG	USART Receive Data Register								0000 0000	118, 150		
1Bh	CCPR2L	Capture/Compare/PWM Register 2 (LSB)								xxxx xxxx	63, 150		
1Ch	CCPR2H	Capture/Compare/PWM Register 2 (MSB)								xxxx xxxx	63, 150		
1Dh	CCP2CON	—	—	CCP2X	CCP2Y	CCP2M3	CCP2M2	CCP2M1	CCP2M0	-- 00 0000	64, 150		
1Eh	ADRESH	A/D Result Register High Byte								xxxx xxxx	133, 150		
1Fh	ADCON0	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON	0000 00-0	127, 150		

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

- Note 1: The upper byte of the program counter is not directly accessible. PCLATH is a holding register for the PC<12:8>, whose contents are transferred to the upper byte of the program counter.
- Note 2: Bits PSPIE and PSPIF are reserved on PIC16F873A/876A devices; always maintain these bits clear.
- Note 3: These registers can be addressed from any bank.
- Note 4: PORTD, PORTE, TRISD and TRISE are not implemented on PIC16F873A/876A devices, read as '0'.
- Note 5: Bit 4 of EEDR implemented only on the PIC16F876A/877A devices.

## 3.0 DATA EEPROM AND FLASH PROGRAM MEMORY

The data EEPROM and Flash program memory is readable and writable during normal operation (over the full V<sub>DD</sub> range). This memory is not directly mapped in the register file space. Instead, it is indirectly addressed through the Special Function Registers. There are six SFRs used to read and write this memory:

- EECON1
- EECON2
- EEDATA
- EEDATH
- EEADR
- EEADRH

When interacting to the data memory block, EEDATA holds the 8-bit data to read/write and EEADR holds the address of the EEPROM location being accessed. These devices have 128 or 256 bytes of data EEPROM (depending on the device), with an address range from 00h to FFh. On devices with 128 bytes, addresses from 80h to FFh are not implemented and will wrap around to the beginning of data EEPROM memory. When writing to unimplemented locations, the on-chip charge pump will be turned off.

When interacting to the program memory block, the EEDATA and EEDATH registers form a two-byte word that holds the 14-bit data to read/write and the EEADR and EEADRH registers form a two-byte word that holds the 13-bit address of the program memory location being accessed. These devices have 4 or 8K words of program Flash, with an address range from 0000h to 0FFFh for the PIC16F873A/871A and 0000h to 1FFFh for the PIC16F876A/877A. Addresses above the range of the respective device will wrap around to the beginning of program memory.

The EEPROM data memory allows single-byte read and write. The Flash program memory allows single-word read and four-word block writes. Program memory write operations automatically perform an erase-before-write on blocks of four words. A byte write to data EEPROM memory automatically erases the location and writes the new data (erase-before-write).

The write time is controlled by an on-chip timer. The write/erase voltages are generated by an on-chip charge pump, rated to operate over the voltage range of the device for byte or word operations.

When the device is code-protected, the CPU may continue to read and write the data EEPROM memory. Depending on the settings of the write-protect bits, the device may or may not be able to write certain blocks of the program memory; however, reads of the program memory are allowed. When code-protected, the device programmer can no longer access data or program memory; this does NOT inhibit internal reads or writes.

## 3.1 EEADRH and EEADRH

The EEADRH:EEADR register pair can address up to a maximum of 256 bytes of data EEPROM or up to a maximum of 8K words of program EEPROM. When selecting a data address value, only the LS byte of the address is written to the EEADR register. When selecting a program address value, the MS byte of the address is written to the EEADRH register and the LS byte is written to the EEADR register.

If the device contains less memory than the full address range of the address register pair, the Most Significant bits of the registers are not implemented. For example, if the device has 128 bytes of data EEPROM, the Most Significant bit of EEADR is not implemented and access to data EEPROM.

## 3.2 EECON1 and EECON2 Registers

EECON1 is the control register for memory accesses.

Control bit, EEPGD, determines if the access will be a program or data memory access. When clear, as it is when reset, any subsequent operations will operate on the data memory. When set, any subsequent operations will operate on the program memory.

Control bits, RD and WR, initiate read and write or erase, respectively. These bits cannot be cleared, only set, in software. They are cleared in hardware at completion of the read or write operation. The inability to clear the WR bit in software prevents the accidental, premature termination of a write operation.

The WREN bit, when set, will allow a write or erase operation. On power-up, the WREN bit is clear. The WRRERR bit is set when a write (or erase) operation is interrupted by a MCLR or a WDT Time-out Reset during normal operation. In these situations following Reset, the user can check the WRRERR bit and rewrite the location. The data and address will be unchanged in the EEDATA and EEADR registers.

Interrupt flag bit, EEIF, in the PIR2 register, is set when the write is complete. It must be cleared in software.

EECON2 is not a physical register. Reading EECON2 will read all '1's. The EECON2 register is used to clock the PIC16F87XA into the EEPROM write sequence.

**Note:** The write programming algorithm for Flash program memory has been changed. On previous PIC16F87X devices, Flash programming was done in single-word erase/write cycles. The newer PIC16F87XA devices use a four-word erase/write cycle. See Section 3.6 "Writing to Flash Program Memory" for more information.

# PIC16F87XA

## 9.4 I<sup>2</sup>C Mode

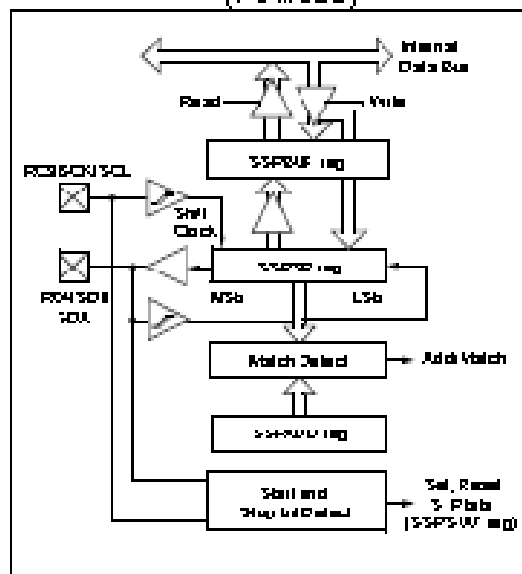
The MSSP module in I<sup>2</sup>C mode fully implements all master and slave functions (including general call support) and provides interrupts on Start and Stop bits in hardware to determine a free bus (in Master mode only). The MSSP module implements the standard mode specifications, as well as 7-bit and 10-bit addressing.

Two pins are used for data transfer:

- Serial clock (SCL) – RC3/SCK/SCL
- Serial data (SDA) – RC4/SDA/DA

The user must configure the se pins as inputs or outputs through the TRISC-4,3- bits.

FIGURE 9-7: MSSP BLOCK DIAGRAM (I<sup>2</sup>C MODE)



## 9.4.1 REGISTER SET

The MSSP module has six registers for I<sup>2</sup>C operation. These are:

- MSSP Control Register (SSPCON)
- MSSP Control Register2 (SSPCON2)
- MSSP Status Register (SSPSTAT)
- Serial Receive/Transmit Buffer Register (SSPBUF)
- MSSP Shift Register (SSPSR) – Not directly accessible
- MSSP Address Register (SSPAD0)

SSPCON, SSPCON2 and SSPSTAT are the control and status registers in I<sup>2</sup>C mode operation. The SSPCON and SSPCON2 registers are readable and writable. The lower six bits of the SSPSTAT are read-only. The upper two bits of the SSPSTAT are read/write.

SSPSR is the shift register used for shifting data in or out. SSPBUF is the buffer register to which data bytes are written to or read from.

SSPAD0 register holds the slave device address when the SSP is configured in I<sup>2</sup>C Slave mode. When the SSP is configured in Master mode, the lower seven bits of SSPADD act as the baud rate generator reload value.

In receive operations, SSPSR and SSPBUF together create a double-buffered receiver. When SSPSR receives a complete byte, it is transferred to SSPBUF and the SSPIF interrupt is set.

During transmission, the SSPBUF is not double-buffered. A write to SSPBUF will write to both SSPBUF and SSPSR.

# PIC16F87XA

## 9.4.2 OPERATION

The MSSP module functions are enabled by setting MSSP Enable bit, **SSPEN** (**SSPCON<5>**).

The **SSPCON** register allows control of the I<sup>2</sup>C operation. Four mode selection bits (**SSPCON<3:0>**) allow one of the following I<sup>2</sup>C modes to be selected:

- I<sup>2</sup>C Master mode, **cbck = OSC/4** (**SSPADD + 1**)
- I<sup>2</sup>C Slave mode (7-bit address)
- I<sup>2</sup>C Slave mode (10-bit address)
- I<sup>2</sup>C Slave mode (7-bit address) with Start and Stop bit interrupts enabled
- I<sup>2</sup>C Slave mode (10-bit address) with Start and Stop bit interrupts enabled
- I<sup>2</sup>C Firmware Controlled Master mode, **slave is idle**

Selection of any I<sup>2</sup>C mode, with the **SSPEN** bit set, forces the SCL and SDA pins to be open-drain, provided these pins are programmed to inputs by setting the appropriate TRISC bits. To insure proper operation of the module, pull-up resistors must be provided externally to the SCL and SDA pins.

## 9.4.3 SLAVE MODE

In Slave mode, the SCL and SDA pins must be configured as inputs (TRISC<4:3> set). The MSSP module will override the input state with the output data when required (slave-transmission).

The I<sup>2</sup>C slave mode hardware will always generate an interrupt on an address match. Through the mode select bits, the user can also choose to interrupt on Start and Stop bits.

When an address is matched, or the data transfer after an address match is received, the hardware automatically will generate the Acknowledge (ACK) pulse and load the **SSPBUF** register with the received value currently in the **SSPSR** register.

Any combination of the following conditions will cause the MSSP module not to give the ACK pulse.

- The bus error bit, **BF** (**SSPSTAT<3>**), was set before the transfer was received.
- The bus error bit, **SSPOV** (**SSPCON<6>**), was set before the transfer was received.

In this case, the **SSPSR** register value is not loaded into the **SSPBUF**, but bit **SSPIF** (**PIR1<3>**) is set. The **BF** bits cleared by reading the **SSPIF** register, while bit **SSPOV** is cleared through software.

The SCL clock input must have a minimum high and low for proper operation. The high and low times of the I<sup>2</sup>C specification, as well as the requirements of the MSSP module, are shown in timing parameter #100 and parameter #101.

## 9.4.3.1 Addressing

Once the MSSP module has been enabled, it waits for a Start condition to occur. Following the Start condition, the 8 bits are shifted into the **SSPSR** register. All incoming bits are sampled with the rising edge of the clock (SCL) line. The value of register (**SSPSR<7:0>**) is compared to the value of the **SSPADD** register. The address is compared on the falling edge of the eighth clock (SCL) pulse. If the addresses match, and the **BF** and **SSPOV** bits are clear, the following events occur:

1. The **SSPSR** register value is loaded into the **SSPBUF** register.
2. The Buffer Full bit, **BF**, is set.
3. An **ACK** pulse is generated.
4. MSSP interrupt flag bit, **SSPIF** (**PIR1<3>**), is set (interrupt is generated if enabled) on the falling edge of the ninth SCL pulse.

In 10-bit Address mode, two address bytes need to be received by the slave. The five Most Significant bits (MSBs) of the first address byte specify if this is a 10-bit address. Bit **R/A0** (**SSPSTAT<2>**) must specify a write so the slave device will receive the second address byte. For a 10-bit address, the first byte would equal '11110' as 'as', where 'as' and 'as' are the two MSBs of the address. The sequence of events for 10-bit address is as follows, with steps 7 through 9 for the slave-transmission.

1. Receive first (high) byte of address (bits **SSPIF**, **BF** and **D/LUA** (**SSPSTAT<1>**) are set).
2. Update the **SSPADD** register with second (low) byte of address (clears bit **UA** and releases the SCL line).
3. Read the **SSPBUF** register (clears bit **BF**) and clear flag bit **SSPIF**.
4. Receive second (low) byte of address (bits **SSPIF**, **BF** and **UA** are set).
5. Update the **SSPADD** register with the first (high) byte of address. If match releases SCL line, this will clear **D/LUA**.
6. Read the **SSPBUF** register (clears bit **BF**) and clear flag bit **SSPIF**.
7. Receive Repeated Start condition.
8. Receive first (high) byte of address (bits **SSPIF** and **BF** are set).
9. Read the **SSPBUF** register (clears bit **BF**) and clear flag bit **SSPIF**.

## 9.4.3.2 Reception

When the  $\overline{R/W}$  bit of the address byte is clear and an address match occurs, the  $\overline{R/W}$  bit of the SSPSTAT register is cleared. The received address is loaded into the SSPBUF register and the SDA line is held low ( $\overline{ACK}$ ).

When the address byte overflow condition exists, then the No Acknowledge ( $\overline{ACK}$ ) pulse is given. An overflow condition is defined as either bit BF (SSPSTAT-0) is set or bit SSPOV (SSPCON-6) is set.

An MSSP interrupt is generated for each data transfer byte. Flag bit SSPIF (PIR1-3) must be cleared in software. The SSPSTAT register is used to determine the status of the byte.

If SEN is enabled (SSPCON-0 = 1), RC3/SCK/SCL will be held low (clock stretch) following each data transfer. The clock must be released by setting bit CKP (SSPCON-4). See Section 9.4.4 "Clock Stretching" for more detail.

## 9.4.3.3 Transmission

When the  $\overline{R/W}$  bit of the incoming address byte is set and an address match occurs, the  $\overline{R/W}$  bit of the SSPSTAT register is set. The received address is loaded into the SSPBUF register. The  $\overline{ACK}$  pulse will be sent on the ninth bit and pin RC3/SCK/SCL is held low regardless of SEN (see Section 9.4.4 "Clock Stretching" for more detail). By stretching the clock, the master will be unable to assert another clock pulse until the slave is done preparing the transmit data. The transmit data must be loaded into the SSPBUF register, which also loads the SSPSR register. Then pin RC3/SCK/SCL should be enabled by setting bit CKP (SSPCON-4). The eight data bits are shifted out on the falling edge of the SCL input. This ensures that the SDA signal is valid during the SCL high time (Figure 9-9).

The  $\overline{ACK}$  pulse from the master receiver is latched on the rising edge of the ninth SCL input pulse. If the SDA line is high ( $\overline{ACK}$ ), then the data transfer is complete. In this case, when the  $\overline{ACK}$  is latched by the slave, the slave logic is reset (it sets SSPSTAT register) and the slave monitors for another occurrence of the Start bit. If the SDA line was low ( $\overline{ACK}$ ), the next transmit data must be loaded into the SSPBUF register. Again, pin RC3/SCK/SCL must be enabled by setting bit CKP.

An MSSP interrupt is generated for each data transfer byte. The SSPIF bit must be cleared in software and the SSPSTAT register is used to determine the status of the byte. The SSPIF bit is set on the falling edge of the ninth clock pulse.



# PIC16F87XA

FIGURE 9-8: I<sup>2</sup>C SLAVE MODE TIMING WITH SEN = 0 (RECEPTION, 7-BIT ADDRESS)

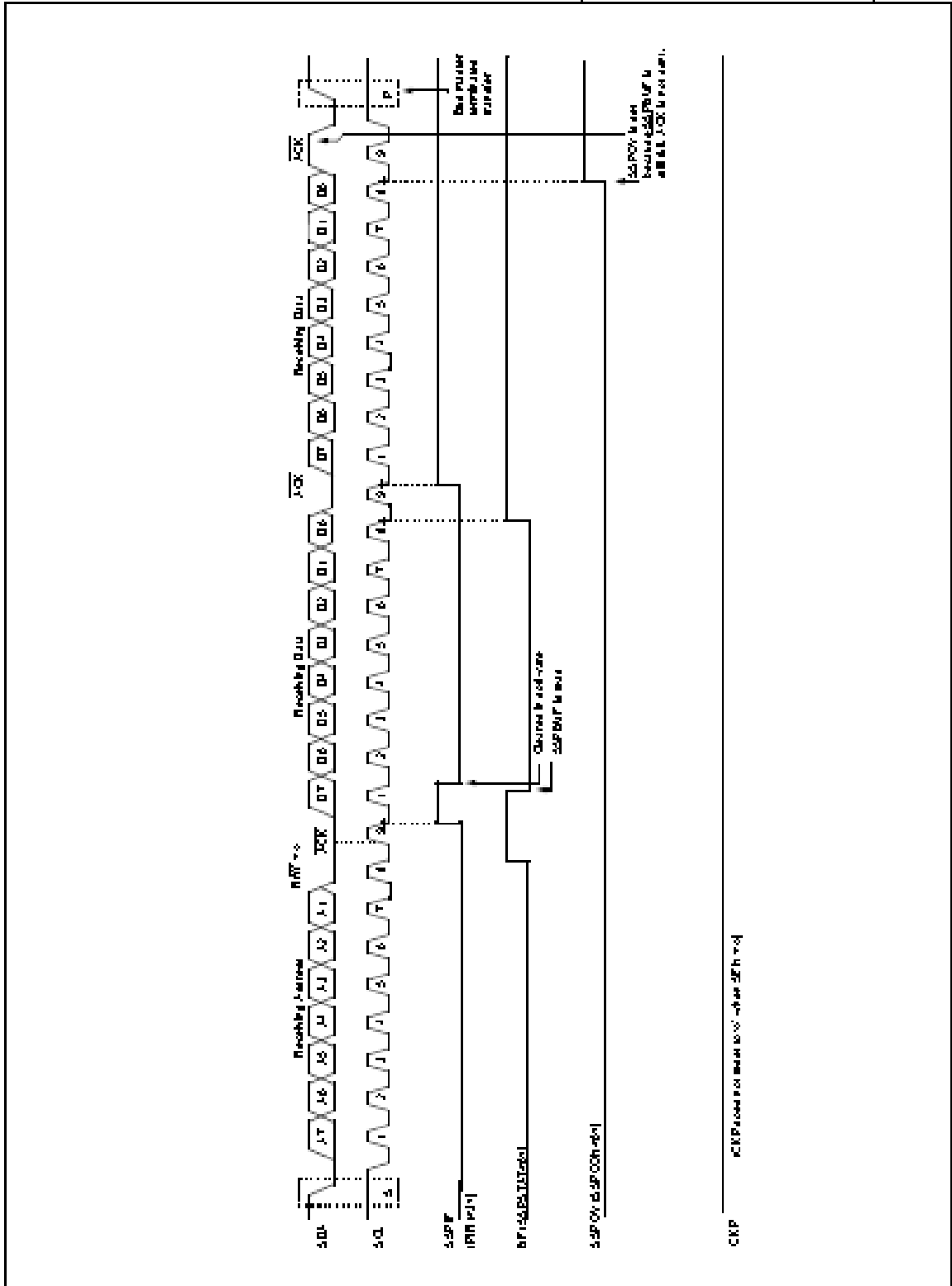
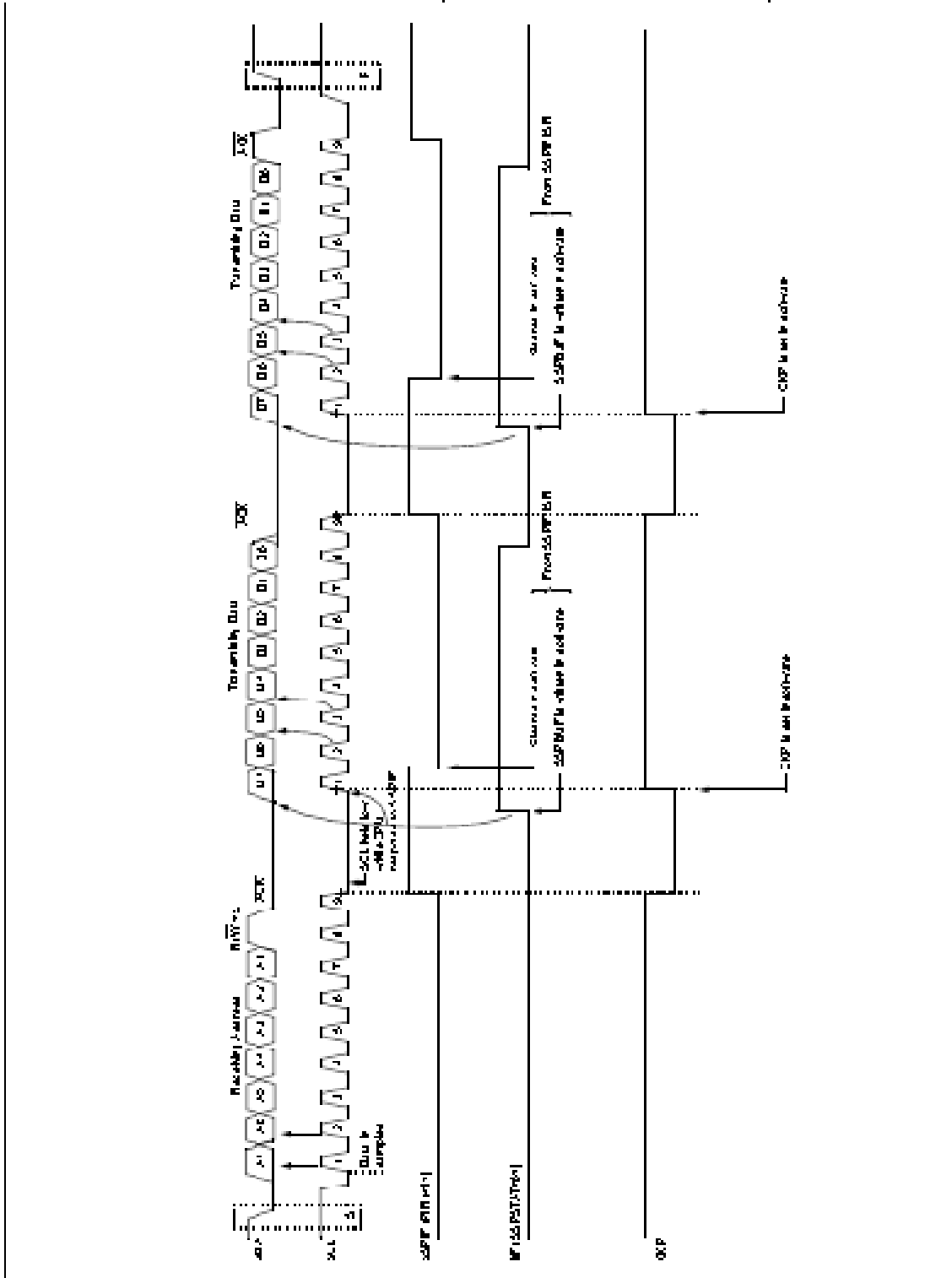


FIGURE 9-9: I<sup>2</sup>C SLAVE MODE TIMING (TRANSMISSION, 7-BIT ADDRESS)



# PIC16F87XA

## 17.0 ELECTRICAL CHARACTERISTICS

### Absolute Maximum Ratings †

Ambient temperature under bias.....	-55 to +125°C
Storage temperature.....	-65°C to +150°C
Voltage on any pin with respect to V <sub>SS</sub> (except V <sub>DD</sub> , $\overline{\text{MCLR}}$ , and RA4).....	-0.3V to (V <sub>DD</sub> + 0.3V)
Voltage on V <sub>DD</sub> with respect to V <sub>SS</sub> .....	-0.3 to +7.5V
Voltage on $\overline{\text{MCLR}}$ with respect to V <sub>SS</sub> (Note 2).....	0 to +14V
Voltage on RA4 with respect to V <sub>SS</sub> .....	0 to +8.5V
Total power dissipation (Note 1).....	1.0W
Maximum current out of V <sub>SS</sub> pin.....	300 mA
Maximum current into V <sub>DD</sub> pin.....	250 mA
Input clamp current, I <sub>IK</sub> (V <sub>I</sub> < 0 or V <sub>I</sub> > V <sub>DD</sub> ).....	± 20 mA
Output clamp current, I <sub>OK</sub> (V <sub>O</sub> < 0 or V <sub>O</sub> > V <sub>DD</sub> ).....	± 20 mA
Maximum output current sunk by any I/O pin.....	25 mA
Maximum output current sourced by any I/O pin.....	25 mA
Maximum current sunk by PORTA, PORTB and PORTE (combined) (Note 3).....	200 mA
Maximum current sourced by PORTA, PORTB and PORTE (combined) (Note 3).....	200 mA
Maximum current sunk by PORTC and PORTD (combined) (Note 3).....	200 mA
Maximum current sourced by PORTC and PORTD (combined) (Note 3).....	200 mA

Note 1: Power dissipation is calculated as follows:  $P_{dis} = V_{DD} \times (I_{DD} - \sum I_{OH}) + \sum \{(V_{DD} - V_{OH}) \times I_{OH}\} + \sum (V_{OL} \times I_{OL})$

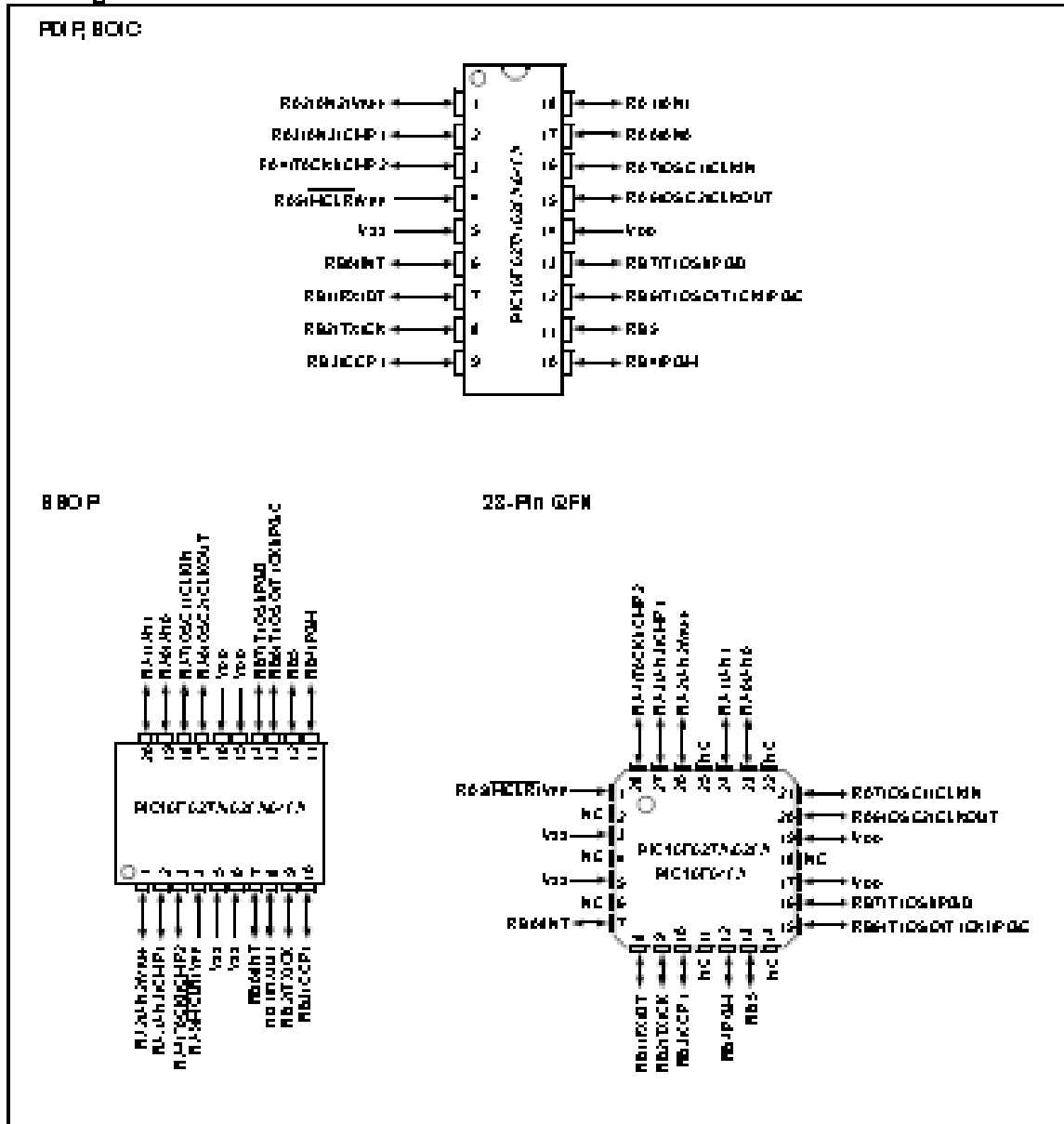
2: Voltage spikes below V<sub>SS</sub> at the  $\overline{\text{MCLR}}$  pin, inducing currents greater than 60 mA, may cause latch-up. Thus, a series resistor of 50-100Ω should be used when applying a "low" level to the  $\overline{\text{MCLR}}$  pin rather than pulling this pin directly to V<sub>SS</sub>.

3: PORTD and PORTE are not implemented on PIC16F873A/876A devices.

† NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

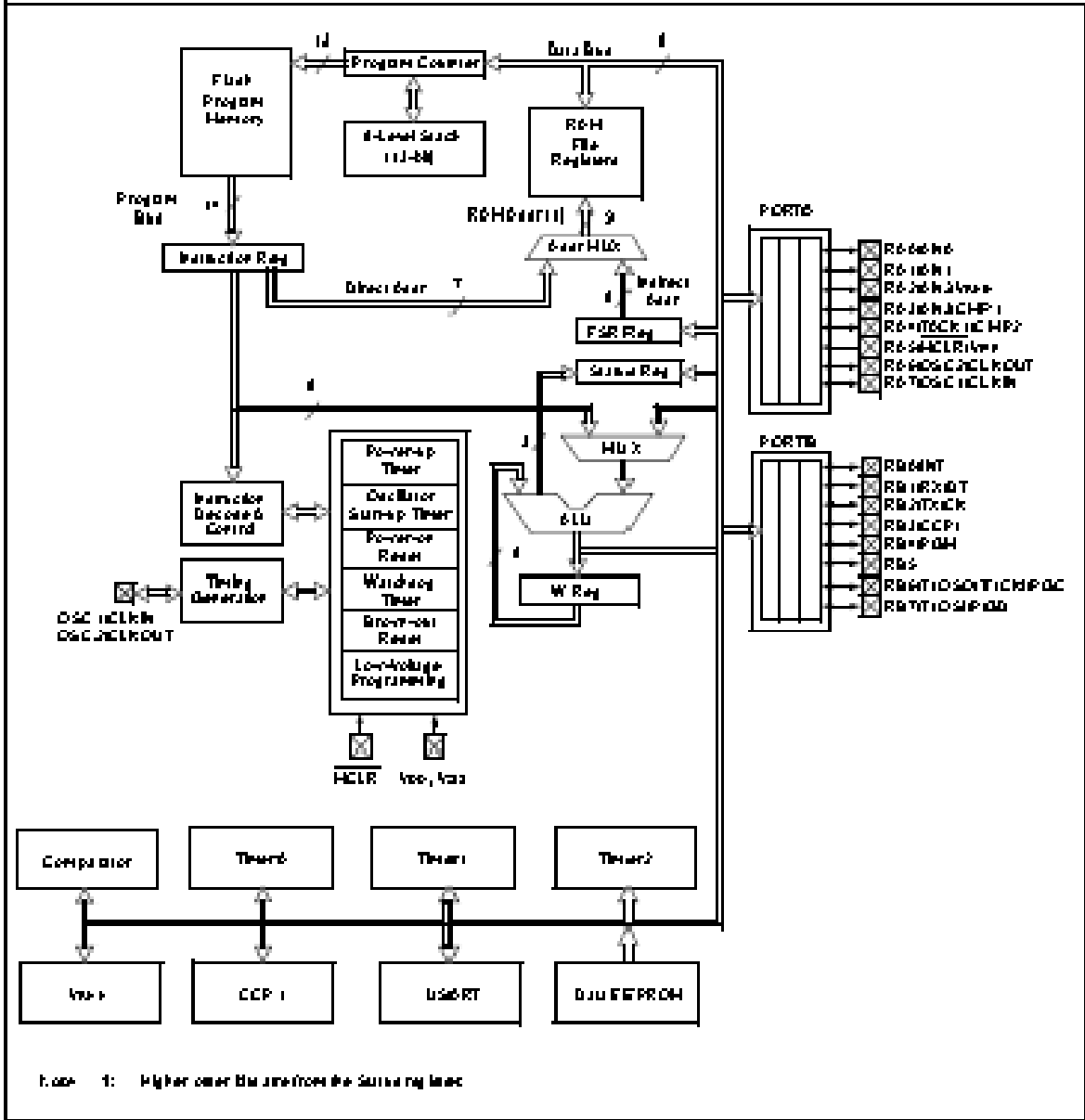
# PIC16F627A/628A/648A

Pin Diagram 1



# PIC16F627A/628A/648A

FIGURE 3-1: BLOCK DIAGRAM



# PIC16F627A/628A/648A

TABLE 3-2: PIC16F627A/628A/648A PINOUT DESCRIPTION

Name	Function	Input Type	Output Type	Description
RA0/AN0	RA0	ST	CMOS	Bidirectional I/O port I
	AN0	AN	—	Analog comparator input
RA1/AN1	RA1	ST	CMOS	Bidirectional I/O port I
	AN1	AN	—	Analog comparator input
RA2/AN2/VREF	RA2	ST	CMOS	Bidirectional I/O port I
	AN2	AN	—	Analog comparator input
	VREF	—	AN	VREF output
RA3/AN3/COMP1	RA3	ST	CMOS	Bidirectional I/O port I
	AN3	AN	—	Analog comparator input
	COMP1	—	CMOS	Comparator 1 output
RA4/TDCK/COMP2	RA4	ST	OD	Bidirectional I/O port I
	TDCK	ST	—	Timer/D clock input
	COMP2	—	OD	Comparator 2 output
RA5/MCLR/VPP	RA5	ST	—	Input port I
	MCLR	ST	—	Master clear. When configured as MCLR, this pin is an active low Reset to the device. Voltage on MCLR/VPP must not exceed VDD during normal device operation.
	VPP	—	—	Programming voltage input
RA6/OSC2/CLKOUT	RA6	ST	CMOS	Bidirectional I/O port I
	OSC2	—	XTAL	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode.
	CLKOUT	—	CMOS	In RC/INTOSC mode, OSC2 pin can output CLKOUT, which has 1/4 the frequency of OSC1.
RA7/OSC1/CLKIN	RA7	ST	CMOS	Bidirectional I/O port I
	OSC1	XTAL	—	Oscillator crystal input
	CLKIN	ST	—	External clock source input. RC biasing pin.
RB0/INT	RB0	TTL	CMOS	Bidirectional I/O port I. Can be software programmed for internal weak pullup.
	INT	ST	—	External interrupt I
RB1/RX/DT	RB1	TTL	CMOS	Bidirectional I/O port I. Can be software programmed for internal weak pullup.
	RX	ST	—	USART receive pin
	DT	ST	CMOS	Synchronous data I/O
RB2/TX/CK	RB2	TTL	CMOS	Bidirectional I/O port I. Can be software programmed for internal weak pullup.
	TX	—	CMOS	USART transmit pin
	CK	ST	CMOS	Synchronous clock I/O
RB3/CCP1	RB3	TTL	CMOS	Bidirectional I/O port I. Can be software programmed for internal weak pullup.
	CCP1	ST	CMOS	Capture/Compare/PWM I/O

Legend: O = Output  
 — = Not used  
 TTL = TTL Input

CMOS = CMOS Output  
 I = Input  
 OD = Open Drain Output

P = Power  
 ST = Schmitt Trigger Input  
 AN = Analog

# PIC16F627A/628A/648A

TABLE 3-2: PIC16F627A/628A/648A PINOUT DESCRIPTION (CONTINUED)

Name	Function	Input Type	Output Type	Description
RB4/PGM	RB4	TTL	CMOS	Bidirectional I/O port. Interrupt-on-pin change. Can be software programmed for internal weak pull-up.
	PGM	ST	—	Low-voltage programming input pin. When low-voltage programming is enabled, the interrupt-on-pin change and weak pull-up resistor are disabled.
RB5	RB5	TTL	CMOS	Bidirectional I/O port. Interrupt-on-pin change. Can be software programmed for internal weak pull-up.
RB6/TIO80/TICKI/PGC	RB6	TTL	CMOS	Bidirectional I/O port. Interrupt-on-pin change. Can be software programmed for internal weak pull-up.
	TIO80	—	XTAL	Timer1 oscillator output
	TICKI	ST	—	Timer1 clock input
	PGC	ST	—	CSP™ programming clock
RB7/TIO8I/PGD	RB7	TTL	CMOS	Bidirectional I/O port. Interrupt-on-pin change. Can be software programmed for internal weak pull-up.
	TIO8I	XTAL	—	Timer1 oscillator input
	PGD	ST	CMOS	CSP data I/O
VSS	VSS	Power	—	Ground reference for logic and I/O pins
VDD	VDD	Power	—	Positive supply for logic and I/O pins

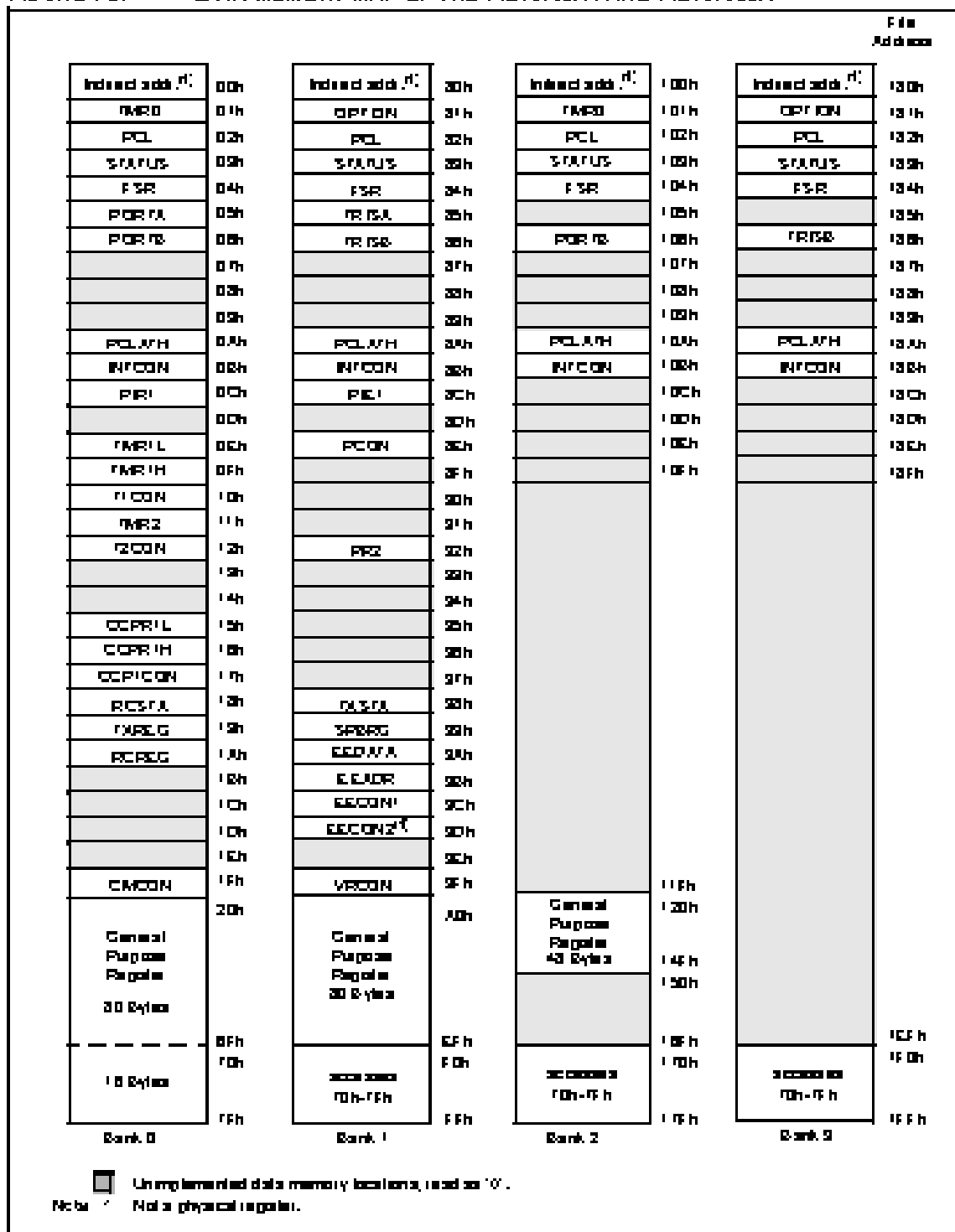
Legend: O = Output  
 — = Not used  
 TTL = TTL Input

CMOS = CMOS Output  
 I = Input  
 OD = Open Drain Output

P = Power  
 ST = Schmitt Trigger Input  
 AN = Analog

# PIC16F627A/628A/648A

FIGURE 4-2: DATA MEMORY MAP OF THE PIC16F627A AND PIC16F628A





# PIC16F627A/628A/648A

## 12.0 UNIVERSAL SYNCHRONOUS ASYNCHRONOUS RECEIVER TRANSMITTER (USART) MODULE

The Universal Synchronous Asynchronous Receiver Transmitter (USART) is also known as a Serial Communications Interface (SCI). The USART can be configured as a full-duplex asynchronous system that can communicate with peripheral devices such as CRT terminals and personal computers, or it can be configured as a half-duplex synchronous system that can communicate with peripheral devices such as A/D or D/A integrated circuits, Serial EEPROMs, etc.

The USART can be configured in the following modes:

- Asynchronous (full-duplex)
- Synchronous – Master (half-duplex)
- Synchronous – Slave (half-duplex)

Bit SPEN (RCSTR<7>) and bit TRISB<21> have to be set in order to configure pins RB2/TXCK and RB1/RXD/T as the Universal Synchronous Asynchronous Receiver Transmitter.

Register 12-1 shows the Transmitt Status and Control Register (TXSTA) and Register 12-2 shows the Receive Status and Control Register (RCSTA).

REGISTER 12-1: TXSTA – TRANSMIT STATUS AND CONTROL REGISTER (ADDRESS: 98h)

RAWD	RAWD	RAWD	RAWD	U-0	RAWD	R-1	RAWD	
CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TXSD	
bit 7					bit 0			

bit 7	CSRC: Clock Source Select bit <u>Asynchronous mode</u> Don't care <u>Synchronous mode</u> 1 = Master mode (Clock generated internally from BRG) 0 = Slave mode (Clock from external source)
bit 6	TX9: 9-bit Transmitt Enable bit 1 = Selects 9-bit transmission 0 = Selects 8-bit transmission
bit 5	TXEN: Transmitt Enable bit <sup>1</sup> 1 = Transmitt enabled 0 = Transmitt disabled
bit 4	SYNC: USART Mode Select bit 1 = Synchronous mode 0 = Asynchronous mode
bit 3	Unimplemented: Read as '0'
bit 2	BRGH: High Baud Rate Select bit <u>Asynchronous mode</u> 1 = High speed 0 = Low speed <u>Synchronous mode</u> Unused in this mode
bit 1	TRMT: Transmitt (Shift) Register Status bit 1 = TSR empty 0 = TSR full
bit 0	TXSD: 9th bit of transmitt data. Can be parity bit. Note 1: SPEN/REN overrides TXEN in SYNC mode.

### Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared
		x = Bit is unknown

# PIC16F627A/628A/648A

## 12.1 USART Baud Rate Generator (BRG)

The BRG supports both the Asynchronous and Synchronous modes of the USART. It is a dedicated 8-bit baud rate generator. The SPBRG register controls the period of a free-running 8-bit timer. In Asynchronous mode, bit BRGH (TXSTX<2>) also controls the baud rate. In Synchronous mode, bit BRGH is ignored. Table 12-1 shows the formula for computation of the baud rate for different USART modes, which only apply in Master mode (internal clock).

Given the desired baud rate and  $F_{osc}$ , the nearest integer value for the SPBRG register can be calculated using the formula in Table 12-1. From this, the error in baud rate can be determined.

Example 12-1 shows the calculation of the baud rate error for the following conditions:

- $F_{osc} = 16\text{ MHz}$
- Desired Baud Rate = 9600
- BRGH = 0
- SYNC = 0

EQUATION 12-1: CALCULATING BAUD RATE ERROR

$$\begin{aligned} \text{Desired Baud Rate} &= \frac{F_{osc}}{64(x+1)} \\ 9600 &= \frac{16000000}{64(x+1)} \\ x &= 23.071 \\ \text{Calculated Baud Rate} &= \frac{16000000}{64(23+1)} = 9615 \\ \text{Error} &= \frac{(\text{Calculated Baud Rate} - \text{Desired Baud Rate})}{\text{Desired Baud Rate}} \\ &= \frac{9615 - 9600}{9600} = 0.16\% \end{aligned}$$

It may be advantageous to use the high baud rate (BRGH = 1) even for slower baud clocks. This is because the  $F_{osc}/(16(x+1))$  equation can reduce the baud rate error in some cases.

Writing a new value to the SPBRG register causes the BRG timer to be reset (or cleared) and ensures the BRG does not wait for a timer overflow before outputting the new baud rate.

The data on the RB1R<0:7> pins is sampled three times by a majority detect circuit to determine if a high or a low level is present at the RX pin.

TABLE 12-1: BAUD RATE FORMULA

SYNC	BRGH = 0 (Low Speed)	BRGH = 1 (High Speed)
0	(Asynchronous) Baud Rate = $F_{osc}/(64(x+1))$	Baud Rate = $F_{osc}/(160(x+1))$
1	(Synchronous) Baud Rate = $F_{osc}/(40(x+1))$	N/A

Legend: x = value in SPBRG (0 to 255)

TABLE 12-2: REGISTERS ASSOCIATED WITH BAUD RATE GENERATOR

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on PIC16	Value on all other PIC16s
20h	TXSTX	TXSR	TXB	TXEN	SYNC	—	BRGH	TRMT	TXRD	0000 - 010	0000 - 010
18h	RCSX	SPEN	RXB	SPEN	CREN	ADEN	FERR	DEFF	RXSD	0000 000x	0000 000x
20h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented (used as '0'). Shaded cells are not used for the BRG.

# PIC16F627A/628A/648A

## 12.2.2 USART ASYNCHRONOUS RECEIVER

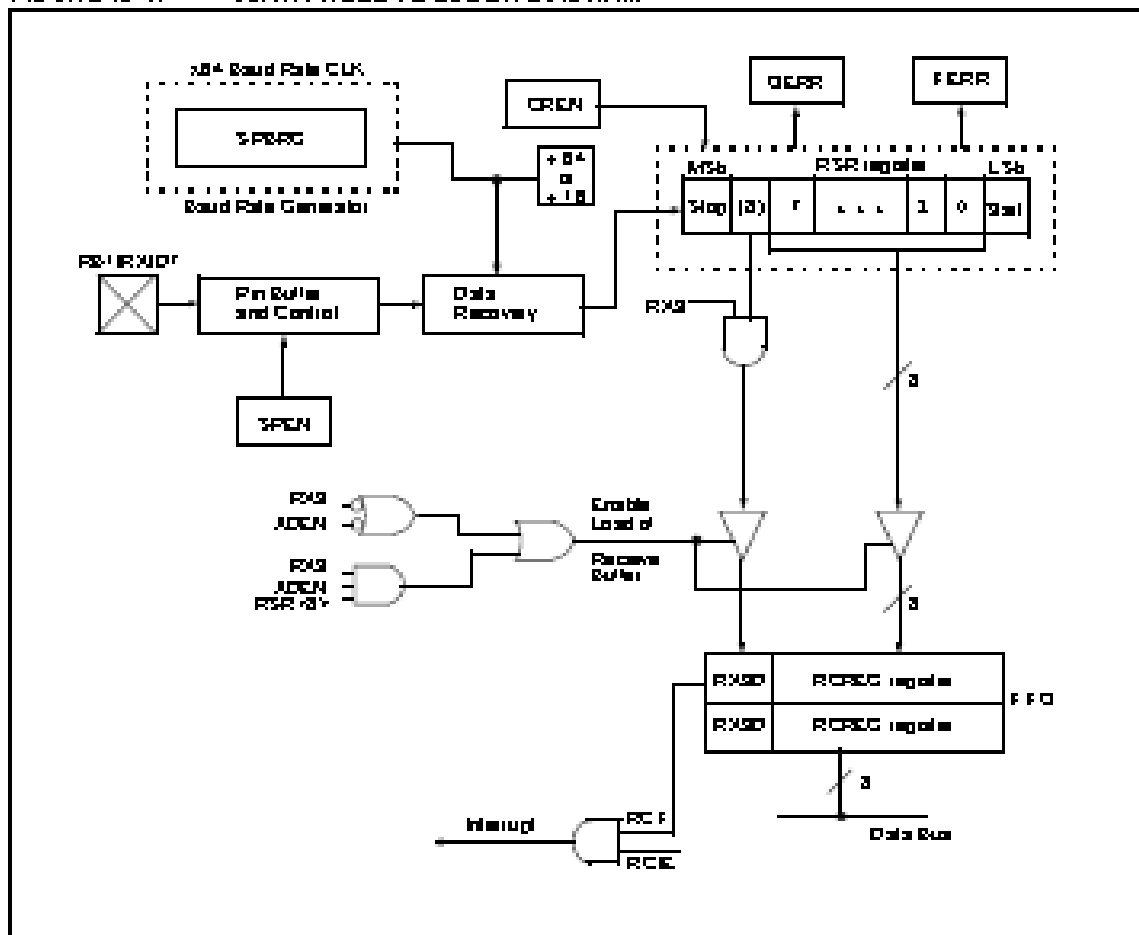
The receiver block diagram is shown in Figure 12-4. The data is received on the RB1/RX0DTPin and drives the data recovery block. The data recovery block is actually a high-speed shifter operating at  $\times 16$  times the baud rate, whereas the main receive serial shifter operates at the bit rate or at  $F_{osc}$ .

When Asynchronous mode is selected, reception is enabled by setting bit OREN (RCSTx:0).

The heart of the receiver is the Receive (Serial) Shift Register (RSR). After sampling the Stop bit, the received data in the RSR is transferred to the RCREG register (if it is empty). If the transfer is complete, flag bit RCIF (PIR1:5) is set. The actual interrupt can be enabled/disabled by setting/clearing enable bit RCIE (PIE1:5). Flag bit RCIF is a read-only bit, which is cleared by the hardware. It is cleared when the RCREG register has been read and is empty. The RCREG is a

double buffered register (i.e., it is a two-deep FIFO). It is possible for two bytes of data to be received and transferred to the RCREG FIFO and a third byte begin shifting to the RSR register. On the detection of the Stop bit of the third byte, if the RCREG register is still full, then overrun error bit OERR (RCSTx:1) will be set. The word in the RSR will be lost. The RCREG register can be read twice to retrieve the two bytes in the FIFO. Overrun bit OERR has to be cleared in software. This is done by resetting the receive logic (OREN is cleared and then set). If bit OERR is set, transfers from the RSR register to the RCREG register are inhibited, so it is essential to clear error bit OERR if it is set. Framing error bit FERR (RCSTx:2) is set if a Stop bit is detected as clear. Bit FERR and the 9th receive bit are buffered the same way as the receive data. Reading the RCREG will load bits RXSD and FERR with new values, therefore it is essential for the user to read the RCSTA register before reading RCREG register in order not to lose the old FERR and RXSD information.

FIGURE 12-4: USART RECEIVE BLOCK DIAGRAM



# PIC16F627A/628A/648A

## 13.0 DATA EEPROM MEMORY

The EEPROM data memory is readable and writable during normal operation (VDD VDD range). This memory is not directly mapped in the register file space. The data is indirectly addressed through the Special Function Registers (SFRs). There are four SFRs used to read and write this memory. These registers are:

- EECON1
- EECON2 (Not a physically implemented register)
- EEDATA
- EEADR

EEDATA holds the 8-bit data for read/write and EEADR holds the address of the EEPROM location being accessed. PIC16F627A/628A devices have 128 bytes of data EEPROM with an address range from 0h to 7Fh. The PIC16F648A device has 256 bytes of data EEPROM with an address range from 0h to FFh.

The EEPROM data memory allows byte read and write. A byte write automatically erases the location and writes the new data (erase before write). The EEPROM data memory is rated for high erase/write cycles. The write time is controlled by an on-chip timer. The write time will vary with voltage and temperature, as well as from chip-to-chip. Please refer to AC specifications for exact limits.

When the device is code-protected, the CPU can continue to read and write the data EEPROM memory. A device programmer can no longer access this memory.

Additional information on the data EEPROM is available in the PIC® *16C48 Range Reference Manual* (DS33023).

REGISTER 13-1: EEDATA – EEPROM DATA REGISTER (ADDRESS: 9A1)

R/W/x	R/W/x	R/W/x	R/W/x	R/W/x	R/W/x	R/W/x	R/W/x
EEDAT7	EEDAT6	EEDAT5	EEDAT4	EEDAT3	EEDAT2	EEDAT1	EEDAT0
b117							b110

b117-0 EEDATn: Byte used to Write to or Read from data EEPROM memory location.

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared    x = Bit is unknown

REGISTER 13-2: EEADR – EEPROM ADDRESS REGISTER (ADDRESS: 9B1)

R/W/x	R/W/x	R/W/x	R/W/x	R/W/x	R/W/x	R/W/x	R/W/x
EEADR7	EEADR6	EEADR5	EEADR4	EEADR3	EEADR2	EEADR1	EEADR0
b117							b110

b117 PIC16F627A/628A  
Unimplemented. Address: Must be set to '0'

PIC16F648A  
EEADR: Set to '1' specifies top 128 locations (128-255) of EEPROM Read/Write Operation  
b116-0 EEADR: Specifies one of 128 locations of EEPROM Read/Write Operation

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared    x = Bit is unknown

# PIC16F627A/628A/648A

## 13.3 Reading the EEPROM Data Memory

To read data from any location, the user must write the address to the EEAR register and then set control bit RD (EEDC041-04). The data is available, in the very next cycle, in the EEDATA register; therefore it can be read in the next instruction. EEDATA will hold this value until another read or until it is written to by the user (during a write operation).

### EXAMPLE 13-1: DATA EEPROM READ

RSP	STATUS, RFO	:Bank 1
MOVM	CON PIR2_P0DC	:
MOVW	EEDADR	:Address to read
RSP	EEDCON1, RD	:EE Read
MOVW	EEDATA, W	:W = EEDATA
RSP	STATUS, RFO	:Bank 0

## 13.4 Writing to the EEPROM Data Memory

To write an EEPROM data location, the user must first write the address to the EEAR register and the data to the EEDATA register. Then the user must follow a specific sequence to initiate the write (see Table 13-1).

### EXAMPLE 13-2: DATA EEPROM WRITE

RSP	STATUS, RFO	:Bank 1
RSP	EEDCON1, WREN	:Enable write
RSP	INTCON, GIE	:Disable INTs.
LFPSZ	INTCON, GIE	:See AN179
GOTO	5-2	
MOVW	55h	:
MOVW	EEDCON2	:Write 55h
MOVW	AAh	:
MOVW	EEDCON2	:Write AAh
RSP	EEDCON1, WR	:Set WR bit
		:begin write
RSP	INTCON, GIE	:Enable INTs.

The write will not initiate if the above sequence is not followed exactly (write 55h to EEDCON2, write AAh to EEDCON2, then set WR bit) for each byte. We strongly recommend that interrupts be disabled during this code segment. A cycle count is executed during the required sequence. Any number that is not equal to the required cycles to execute the required sequence will cause the data not to be written into the EEPROM.

Additionally, the WREN bit in EEDCON1 must be set to enable write. This mechanism prevents accidental writes to data EEPROM due to error (unexpected) code execution (i.e., lost programs). The user should keep the WREN bit clear at all times, except when updating EEPROM. The WREN bit is not cleared by hardware.

After a write sequence has been initiated, clearing the WREN bit will not affect this write cycle. The WR bit will be inhibited from being set unless the WREN bit is set.

At the completion of the write cycle, the WR bit is cleared in hardware and the EE Write Complete Interrupt Flag bit (EEIF) is set. The user can either enable this interrupt or poll this bit. The EEIF bit in the PIR1 registers must be cleared by software.

## 13.5 Write Verify

Depending on the application, good programming practice may dictate that the value written to the Data EEPROM should be verified (Example 13-3) to the desired value to be written. This should be used in applications where an EEPROM bit will be stressed near the specification limit.

### EXAMPLE 13-3: WRITE VERIFY

RSP	STATUS, RFO	:Bank 1
MOVW	EEDATA, W	
RSP	EEDCON1, RD	:Read the
		:value written
:		
:		:Is the value written (in W reg) and
:		:read (in EEDATA) the same?
:		
SWHWP	EEDATA, W	:Is difference of
SWPZC	STATUS, Z	:0? Not the answer
ORWF	WRTED_P0R	:YES, Good write
:		:Continue program

## 13.6 Protection Against Spurious Write

There are conditions when the device may not want to write to the data EEPROM memory. To protect against spurious EEPROM writes, various mechanisms have been built in. On power-up, WREN is cleared. Also when enabled, the Power-up Timer (22 ms duration) prevents EEPROM write.

The write inhibit sequence and the WREN bit together help prevent an accidental write during brown-out, power glitch or software malfunction.

## Anexo E. Datasheet del Microcontrolador PIC18F4550



# MICROCHIP PIC18F2455/2550/4455/4550

## 28/40/44-Pin, High-Performance, Enhanced Flash, USB Microcontrollers with nanoWatt Technology

### Universal Serial Bus Features:

- USB V2.0 Compliant
- Low Speed (1.5 Mb/s) and Full Speed (12 Mb/s)
- Supports Control, Interrupt, Isochronous and Bulk Transfers
- Supports up to 32 Endpoints (16 bidirectional)
- 1-Kbyte Dual Access RAM for USB
- On-Chip USB Transceiver with On-Chip Voltage Regulator
- Interface for Off-Chip USB Transceiver
- Streaming Parallel Port (SPP) for USB streaming transfers (40/44-pin devices only)

### Power-Managed Modes:

- Run: CPU on, peripherals on
- Idle: CPU off, peripherals on
- Sleep: CPU off, peripherals off
- Idle mode currents down to 5.8  $\mu\text{A}$  typical
- Sleep mode currents down to 0.1  $\mu\text{A}$  typical
- Timer1 Oscillator: 1.1  $\mu\text{A}$  typical, 32 kHz, 2V
- Watchdog Timer: 2.1  $\mu\text{A}$  typical
- Two-Speed Oscillator Start-up

### Flexible Oscillator Structure:

- Four Crystal modes, including High Precision PLL for USB
- Two External Clock modes, up to 48 MHz
- Internal Oscillator Block:
  - 8 user-selectable frequencies, from 31 kHz to 8 MHz
  - User-tunable to compensate for frequency drift
- Secondary Oscillator using Timer1 @ 32 kHz
- Dual Oscillator options allow microcontroller and USB module to run at different clock speeds
- Fail-Safe Clock Monitor:
  - Allows for safe shutdown if any clock stops

### Peripheral Highlights:

- High-Current Sink/Source: 25 mA/25 mA
- Three External Interrupts
- Four Timer modules (Timer0 to Timer3)
- Up to 2 Capture/Compare/PWM (CCP) modules:
  - Capture is: 16-bit, max. resolution 5.2 ns ( $T_{OV} \times 16$ )
  - Compare is: 16-bit, max. resolution 83.3 ns ( $T_{OV}$ )
  - PWM output: PWM resolution is 1 to 10-bit
- Enhanced Capture/Compare/PWM (ECCP) module:
  - Multiple output modes
  - Selectable polarity
  - Programmable dead time
  - Auto-shutdown and auto-restart
- Enhanced USART module:
  - LIN bus support
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I<sup>2</sup>C™ Master and Slave modes
- 10-bit, up to 13-channel Analog-to-Digital Converter module (A/D) with Programmable Acquisition Time
- Dual Analog Comparators with Input Multiplexing

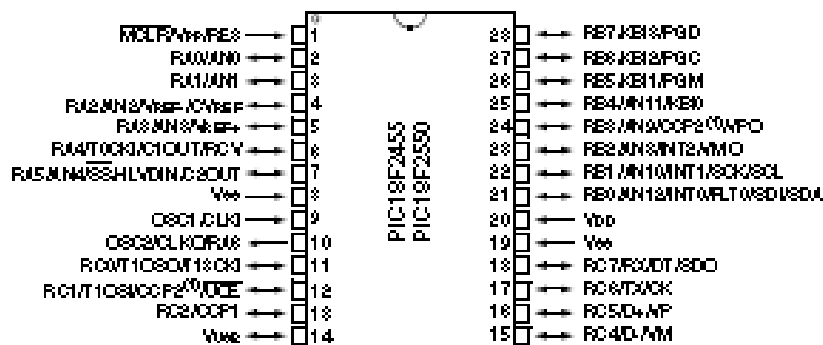
### Special Microcontroller Features:

- C Compiler Optimized Architecture with optional Extended Instruction Set
- 100,000 Erase/Write Cycle Enhanced Flash Program Memory typical
- 1,000,000 Erase/Write Cycle Data EEPROM Memory typical
- Flash/Data EEPROM Retention: > 40 years
- Self-Programmable under Software Control
- Priority Levels for Interrupts
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
  - Programmable period from 41 ms to 131 s
- Programmable Code Protection
- Single-Supply 5V In-Circuit Serial Programming™ (ICSP™) via two pins
- In-Circuit Debug (ICD) via two pins
- Optional dedicated ICSP port (44-pin devices only)
- Wide Operating Voltage Range (2.0V to 5.5V)

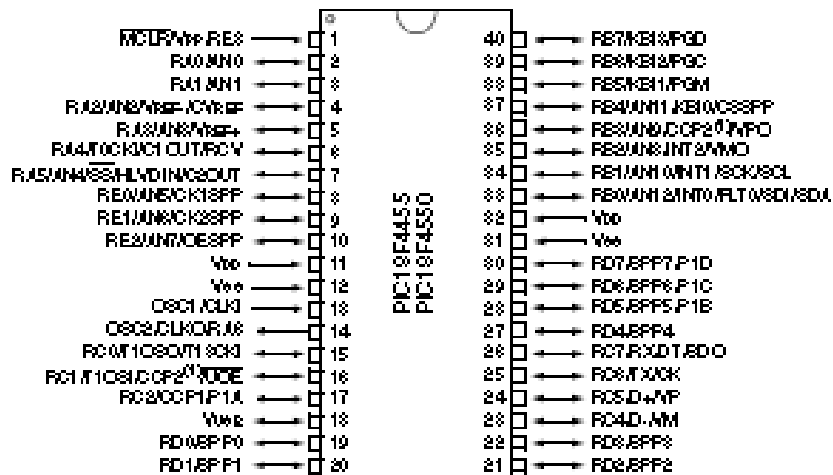
Device	Program Memory		Data Memory		ID	10-Bit A/D (ch)	CCP/ECCP (PWM)	SPP	MSSP		EUSART	Comparators	Timers & 16-Bit
	Flash (bytes)	Single-Word Instructions	SRAM (bytes)	EEPROM (bytes)					SP	Master I <sup>2</sup> C™			
PIC18F2455	24K	12288	2048	256	24	10	2/0	No	Y	Y	1	2	1/3
PIC18F2550	32K	16384	2048	256	24	10	2/0	No	Y	Y	1	2	1/3
PIC18F4455	24K	12288	2048	256	35	13	1/1	Yes	Y	Y	1	2	1/3
PIC18F4550	32K	16384	2048	256	35	13	1/1	Yes	Y	Y	1	2	1/3

## Pin Diagrams

### 28-Pin PDIP, SOIC



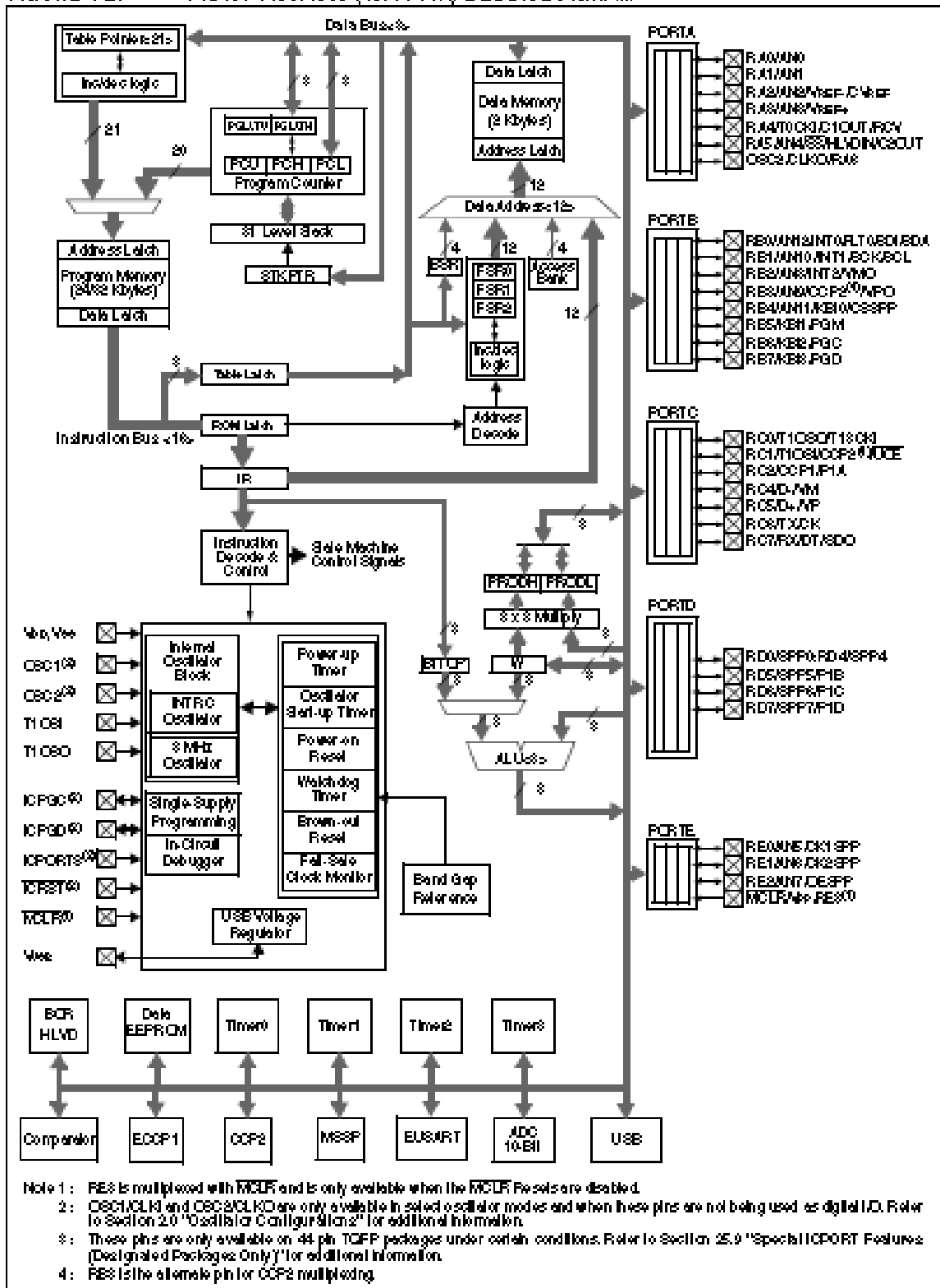
### 40-Pin PDIP



Note 1: RE3 is the alternate pin for CP2 multiplexing.

# PIC18F2455/2550/4455/4550

FIGURE 1-2: PIC18F4455/4550 (40/44-PIN) BLOCK DIAGRAM





# PIC18F2455/2550/4455/4550

## 9.0 INTERRUPTS

The PIC18F2455/2550/4455/4550 devices have multiple interrupt sources and an interrupt priority feature that allows each interrupt source to be assigned a high priority level or a low priority level. The high priority interrupt vector is at 000008h and the low priority interrupt vector is at 000018h. High priority interrupt events will interrupt any low priority interrupts that may be in progress.

There are ten registers which are used to control interrupt operation. These registers are:

- RCON
- INTCON
- INTCON2
- INTCON3
- PIR1, PIR2
- PIE1, PIE2
- IPR1, IPR2

It is recommended that the Microchip header files supplied with MPLAB® IDE be used for the symbolic bit names in these registers. This allows the assembler/compiler to automatically take care of the placement of these bits within the specified register.

Each interrupt source has three bits to control its operation. The functions of these bits are:

- Flag bit to indicate that an interrupt event occurred
- Enable bit that allows program execution to branch to the interrupt vector address when the flag bit is set
- Priority bit to select high priority or low priority

The interrupt priority feature is enabled by setting the IPEN bit (RCON<7>). When interrupt priority is enabled, there are two bits which enable interrupts globally. Setting the GIEH bit (INTCON<7>) enables all interrupts that have the priority bit set (high priority). Setting the GIEL bit (INTCON<6>) enables all interrupts that have the priority bit cleared (low priority). When the interrupt flag, enable bit and appropriate global interrupt enable bit are set, the interrupt will vector immediately to address 000008h or 000018h, depending on the priority bit setting. Individual interrupts can be disabled through their corresponding enable bits.

When the IPEN bit is cleared (default state), the interrupt priority feature is disabled and interrupts are compatible with PICmicro® mid-range devices. In Compatibility mode, the interrupt priority bits for each source have no effect. INTCON<6> is the PEIE bit which enables/disables all peripheral interrupt sources. INTCON<7> is the GIE bit which enables/disables all interrupt sources. All interrupts branch to address 000008h in Compatibility mode.

When an interrupt is responded to, the global interrupt enable bit is cleared to disable further interrupts. If the IPEN bit is cleared, this is the GIE bit. If interrupt priority levels are used, this will be either the GIEH or GIEL bit. High priority interrupt sources can interrupt a low priority interrupt. Low priority interrupts are not processed while high priority interrupts are in progress.

The return address is pushed onto the stack and the PC is loaded with the interrupt vector address (000008h or 000018h). Once in the Interrupt Service Routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bits must be cleared in software before re-enabling interrupts to avoid recursive interrupts.

The "return from interrupt" instruction, RETFIE, exits the interrupt routine and sets the GIE bit (GIEH or GIEL if priority levels are used) which re-enables interrupts.

For external interrupt events, such as the INT pins or the PORTB input change interrupt, the interrupt latency will be three to four instruction cycles. The exact latency is the same for one or two-cycle instructions. Individual interrupt flag bits are set regardless of the status of their corresponding enable bit or the GIE bit.

<b>Note:</b> Do not use the <code>MOVWF</code> instruction to modify any of the interrupt control registers while any interrupt is enabled. Doing so may cause erratic microcontroller behavior.
--

### 9.1 USB Interrupts

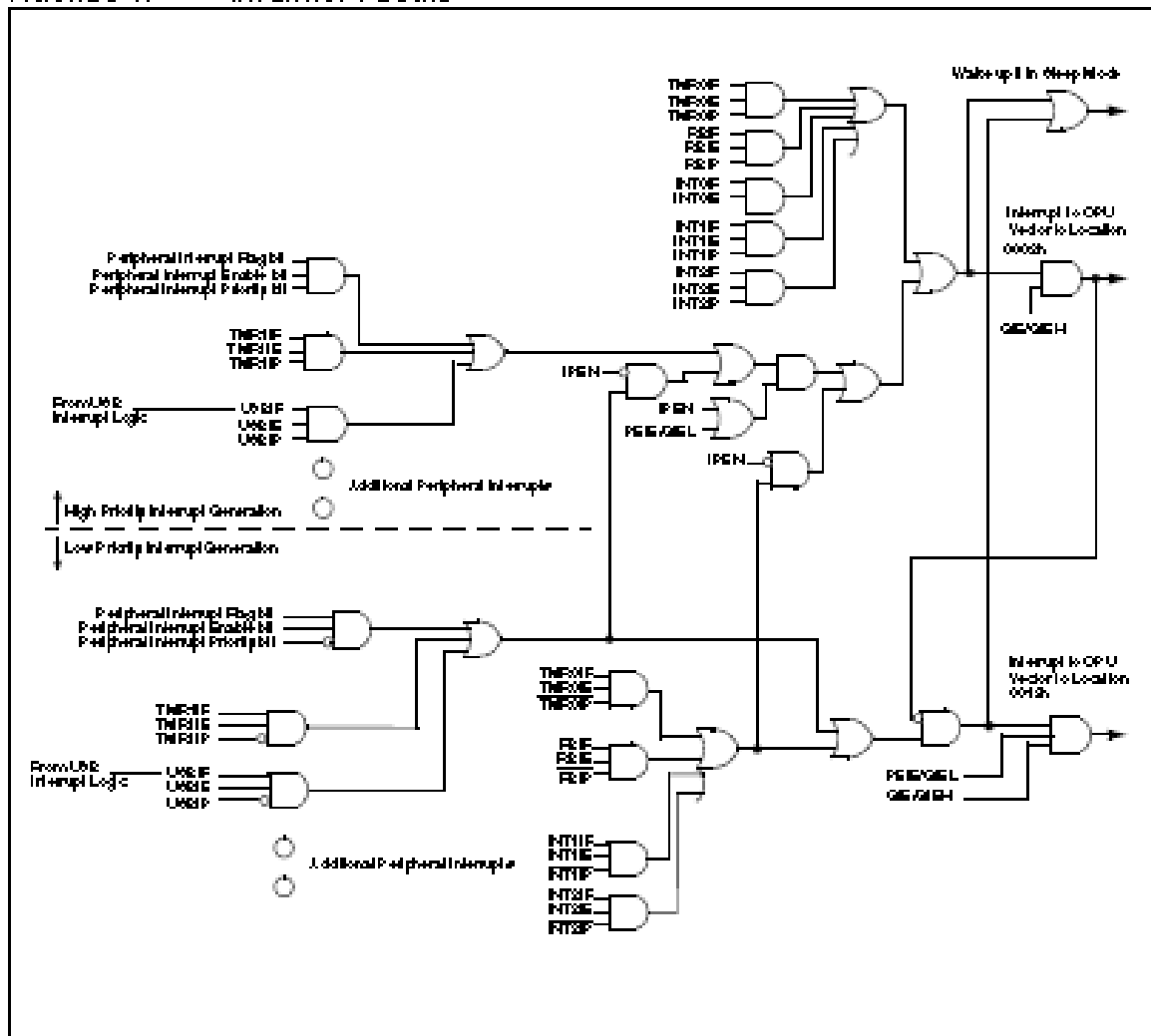
Unlike other peripherals, the USB module is capable of generating a wide range of interrupts for many types of events. These include several types of normal communication and status events and several module level error events.

To handle these events, the USB module is equipped with its own interrupt logic. The logic functions in a manner similar to the microcontroller level interrupt funnel, with each interrupt source having separate flag and enable bits. All events are funneled to a single device level interrupt, USBIF (PIR2<5>). Unlike the device level interrupt logic, the individual USB interrupt events cannot be individually assigned their own priority. This is determined at the device level interrupt funnel for all USB events by the USBIP bit.

For additional details on USB interrupt logic, refer to Section 17.5 "USB Interrupts".

# PIC18F2455/2550/4455/4550

FIGURE 9-1: INTERRUPT LOGIC



# PIC18F2455/2550/4455/4550

## 20.3 EUSART Synchronous Master Mode

The Synchronous Master mode is entered by setting the CSRC bit (TXSTA<7>). In this mode, the data is transmitted in a half-duplex manner (i.e., transmission and reception do not occur at the same time). When transmitting data, the reception is inhibited and vice versa. Synchronous mode is entered by setting bit SYNC (TXSTA<4>). In addition, enable bit, SPEN (RCSTA<7>), is set in order to configure the TX and RX pins to CK (clock) and DT (data) lines, respectively.

The Master mode indicates that the processor transmits the master clock on the CK line.

Clock polarity (CK) is selected with the TXCKP bit (BAUDCON<1>). Setting TXCKP sets the Idle state on CK as high, while clearing the bit sets the Idle state as low. Data polarity (DT) is selected with the RXDTP bit (BAUDCON<6>). Setting RXDTP sets the Idle state on DT as high, while clearing the bit sets the Idle state as low. DT is sampled when CK returns to its idle state. This option is provided to support Microwire devices with this module.

### 20.3.1 EUSART SYNCHRONOUS MASTER TRANSMISSION

The EUSART transmitter block diagram is shown in Figure 20-3. The heart of the transmitter is the Transmit Serial Shift Register (TSR). The Shift register obtains its data from the Read/Write Transmit Buffer register, TXREG. The TXREG register is loaded with data in software. The TSR register is not loaded until the last bit has been transmitted from the previous load. As soon as the last bit is transmitted, the TSR is loaded with new data from the TXREG (if available).

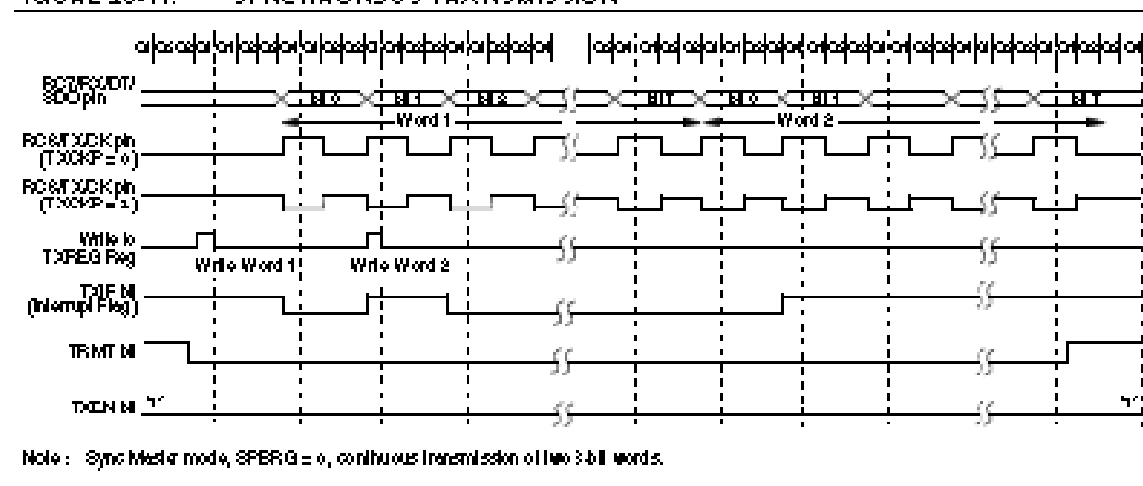
Once the TXREG register transfers the data to the TSR register (occurs in one Tcy), the TXREG is empty and the TXIF flag bit (PIR1<4>) is set. The interrupt can be enabled or disabled by setting or clearing the interrupt enable bit, TXIE (PIE1<4>). TXIF is set regardless of the state of enable bit TXIE; it cannot be cleared in software. It will reset only when new data is loaded into the TXREG register.

Write flag bit, TXIF, indicates the status of the TXREG register, another bit, TRMT (TXSTA<1>), shows the status of the TSR register. TRMT is a read-only bit which is set when the TSR is empty. No interrupt logic is tied to this bit so the user has to poll this bit in order to determine if the TSR register is empty. The TSR is not mapped in data memory so it is not available to the user.

To set up a Synchronous Master Transmission:

1. Initialize the SPBRGH/SPBRG registers for the appropriate baud rate. Set or clear the BRG16 bit, as required, to achieve the desired baud rate.
2. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
3. If the signal from the CK pin is to be inverted, set the TXCKP bit. If the signal from the DT pin is to be inverted, set the RXDTP bit.
4. If interrupts are desired, set enable bit TXIE.
5. If 9-bit transmission is desired, set bit TX9.
6. Enable the transmission by setting bit TXEN.
7. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
8. Start transmission by loading data to the TXREG register.
9. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7:6>) are set.

FIGURE 20-11: SYNCHRONOUS TRANSMISSION



# PIC18F2455/2550/4455/4550

## 20.3.2 EUSART SYNCHRONOUS MASTER RECEPTION

Once Synchronous mode is selected, reception is enabled by setting either the Single Receive Enable bit, SREN (RCSTA<5>), or the Continuous Receive Enable bit, CREN (RCSTA<4>). Data is sampled on the RX pin on the falling edge of the clock.

If enable bit SREN is set, only a single word is received. If enable bit CREN is set, the reception is continuous until CREN is cleared. If both bits are set, then CREN takes precedence.

To set up a Synchronous Master Reception:

1. Initialize the SPBRGH:SPBRG registers for the appropriate baud rate. Set or clear the BRG16 bit, as required, to achieve the desired baud rate.
2. Enable the synchronous master serial port by setting bits SYNC, SPEN and CSRC.
3. Ensure bits CREN and SREN are clear.
4. If the signal from the CK pin is to be inverted, set the TXCKP bit. If the signal from the DT pin is to be inverted, set the RXDTP bit.
5. If interrupts are desired, set enable bit RCIE.
6. If 9-bit reception is desired, set bit RX9.
7. If a single reception is required, set bit SREN. For continuous reception, set bit CREN.
8. Interrupt flag bit, RCIF, will be set when reception is complete and an interrupt will be generated if the enable bit, RCIE, was set.
9. Read the RCSTA register to get the 8th bit (if enabled) and determine if any error occurred during reception.
10. Read the 8-bit received data by reading the RCREG register.
11. If any error occurred, clear the error by clearing bit CREN.
12. If using interrupts, ensure that the GIE and PEIE bits in the INTCON register (INTCON<7,6>) are set.

FIGURE 20-13: SYNCHRONOUS RECEPTION (MASTER MODE, SREN)

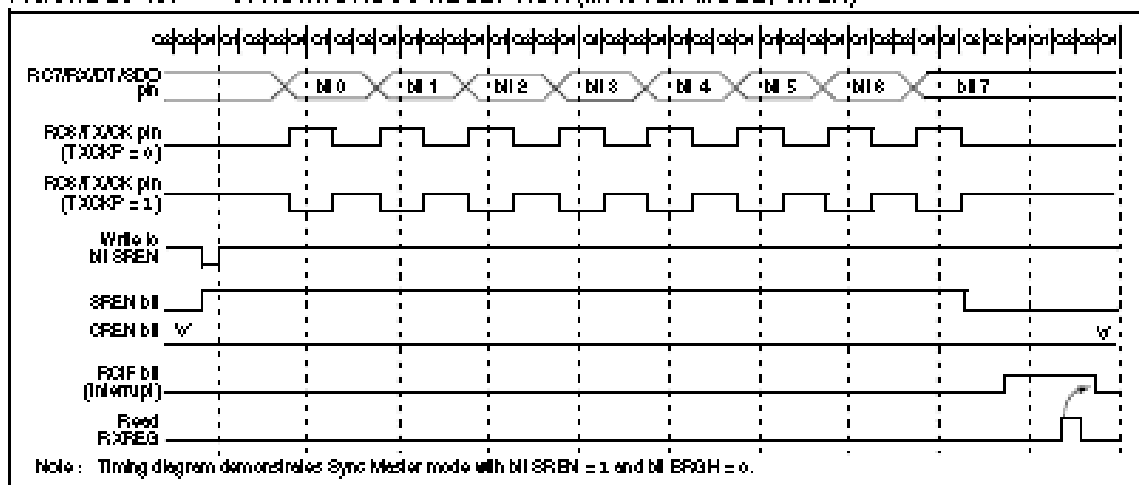


TABLE 20-8: REGISTERS ASSOCIATED WITH SYNCHRONOUS MASTER RECEPTION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	51
PIR1	SPPIE <sup>(1)</sup>	ADIF	RCIF	TXIF	SSPIE	CCP1IE	TMR2IE	TMR1IE	54
PIE1	SPPIE <sup>(1)</sup>	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	54
IPR1	SPPIR <sup>(1)</sup>	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	54
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OVER	RX9D	53
RCREG	EUSART Receive Register								53
TXSTA	CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D	53
BAUDCON	ABDQVF	RCIDL	RXDTP	TXCKP	BRG16	—	WUE	ABDEN	53
SPBRGH	EUSART Baud Rate Generator Register High Byte								53
SPBRG	EUSART Baud Rate Generator Register Low Byte								53

Legend: — = unimplemented, read as '0'. Shaded cells are not used for synchronous master reception.

Note 1: Reserved in 28-pin devices; always maintain these bits clear.

## Anexo F. Datasheet del CM8870

### CMOS Integrated DTMF Receiver

#### Features

- + Full DTMF receiver
- + Less than 35mW power consumption
- + Industrial temperature range
- + Uses quartz crystal or ceramic resonators
- + Adjustable acquisition and release times
- + 18-pin DIP, 18-pin DIP EIAJ, 18-pin SOIC, 20-pin PLCC
- + CM8870C
  - Power down mode
  - Inhibit mode
  - Buffered OSC3 output (PLCC package only)
- + CM8870C is fully compatible with CM8870 for 18-pin devices by grounding pins 5 and 6

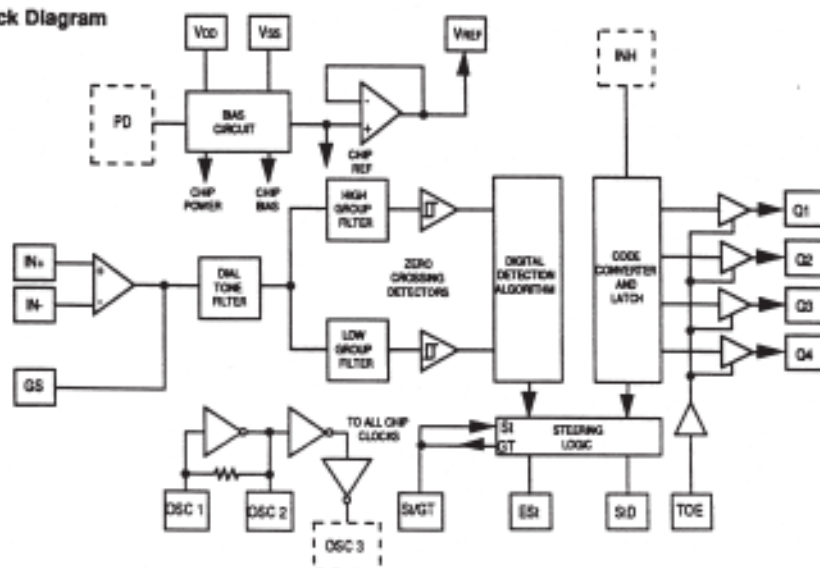
#### Applications

- + PABX
- + Central office
- + Mobile radio
- + Remote control
- + Remote data entry
- + Call limiting
- + Telephone answering systems
- + Paging systems

#### Product Description

The CAMD CM8870/70C provides full DTMF receiver capability by integrating both the bandsplit filter and digital decoder functions into a single 18-pin DIP, SOIC, or 20-pin PLCC package. The CM8870/70C is manufactured using state-of-the-art CMOS process technology for low power consumption (35mW, max.) and precise data handling. The filter section uses a switched capacitor technique for both high and low group filters and dial tone rejection. The CM8870/70C decoder uses digital counting techniques for the detection and decoding of all 16 DTMF tone pairs into a 4-bit code. This DTMF receiver minimizes external component count by providing an on-chip differential input amplifier, clock generator, and a latched three-state interface bus. The on-chip clock generator requires only a low cost TV crystal or ceramic resonator as an external component.

Block Diagram



Pin Function Table

PIN FUNCTION		
Name	Description	
IN+	Non-inverting Input	Connection to the front-end differential amplifier
IN-	Inverting Input	
GS	Gain Select	Gives access to output of front-end differential amplifier for connection of feedback resistor.
V <sub>REF</sub>	Reference voltage output (nominally V <sub>DD</sub> /2). May be used to bias the inputs at mid-rail.	
INH	Inhibits detection of tones represents keys A, B, C, and D	
OSC	Digital buffered oscillator output.	
PD	Power Down	Logic high powers down the device and inhibits the oscillator.
OSCI	Clock Input	3.579546 MHz crystal connected between these pins completes internal oscillator.
OSCZ	Clock Output	
V <sub>SS</sub>	Negative power supply (normally connected to GND).	
TCE	Three-state output enable (input). Logic high enables the outputs Q <sub>1</sub> -Q <sub>4</sub> . Internal pull-up.	
Q <sub>4</sub> Q <sub>3</sub> Q <sub>2</sub> Q <sub>1</sub>	Three-state outputs. When enabled by TCE, provides the code corresponding to the last valid tone pair received. (See Fig. 2).	
StD	Delayed steering output. Presents a logic high when a received tone pair has been registered and the output latch is updated. Returns to logic low when the voltage on St/GT falls below V <sub>TH</sub> .	
Est	Early steering output. Presents a logic high immediately when the digital algorithm detects a recognizable tone pair (signal condition). Any momentary loss of signal condition will cause Est to return to a logic low.	
St/Gt	Steering input/guard time output (bidirectional). A voltage greater than V <sub>TH</sub> detected at St causes the device to register the detected tone pair. The GT output acts to reset the external steering time constant, and its state is a function of Est and the voltage on St. (See Fig. 2)	
V <sub>DD</sub>	Positive power supply.	
IC	Internal Connection.	Must be tied to V <sub>SS</sub> (for SSPB configuration only)

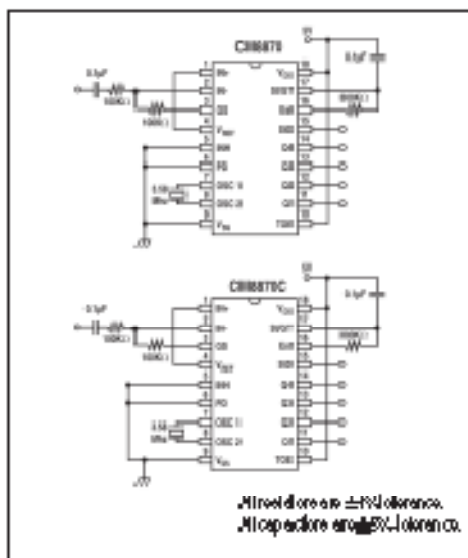


Figure 1.  
Single Ended Input Configuration

F <sub>LOW</sub>	F <sub>HIGH</sub>	KEY	TOW	Q <sub>4</sub>	Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>
89.7	120.9	1	H	0	0	0	1
89.7	133.0	2	H	0	0	1	0
89.7	147.7	3	H	0	0	1	1
77.0	120.9	4	H	0	1	0	0
77.0	133.0	5	H	0	1	0	1
77.0	147.7	6	H	0	1	1	0
85.2	120.9	7	H	0	1	1	1
85.2	133.0	8	H	1	0	0	0
85.2	147.7	9	H	1	0	0	1
94.8	120.9	0	H	1	0	1	0
94.8	133.0	.	H	1	0	1	1
94.8	147.7	A	H	1	1	0	0
89.7	103.3	A	H	1	1	0	1
77.0	103.3	B	H	1	1	1	0
85.2	103.3	C	H	1	1	1	1
94.8	103.3	D	H	0	0	0	0
.	.	ANY	L	Z	Z	Z	Z

L = Logic Low, H = Logic High, Z = High Impedance

Figure 2.  
Functional Diode Table

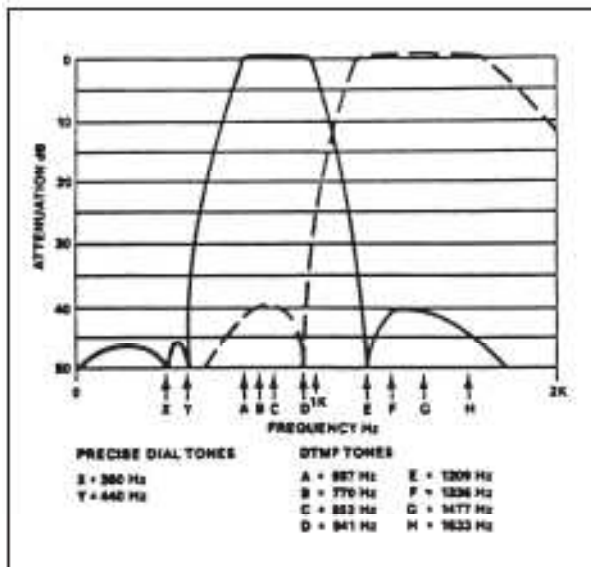


Figure 3. Typical Filter Characteristic

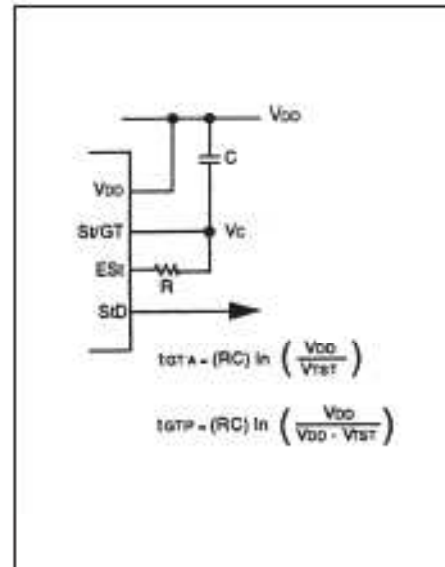


Figure 4. Basic Steering Circuit

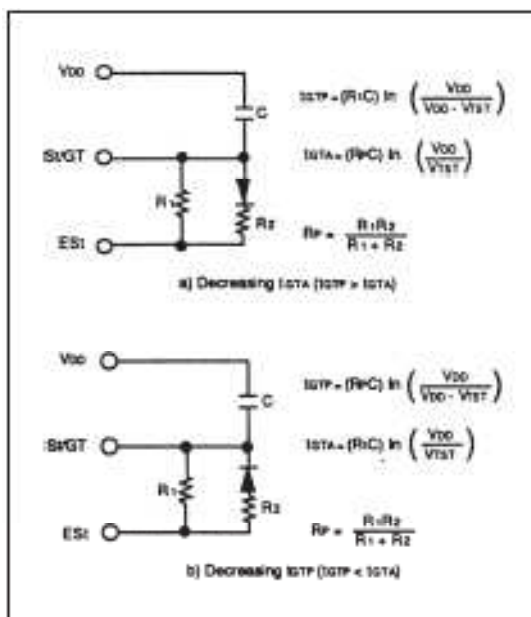


Figure 5. Guard Time Adjustment

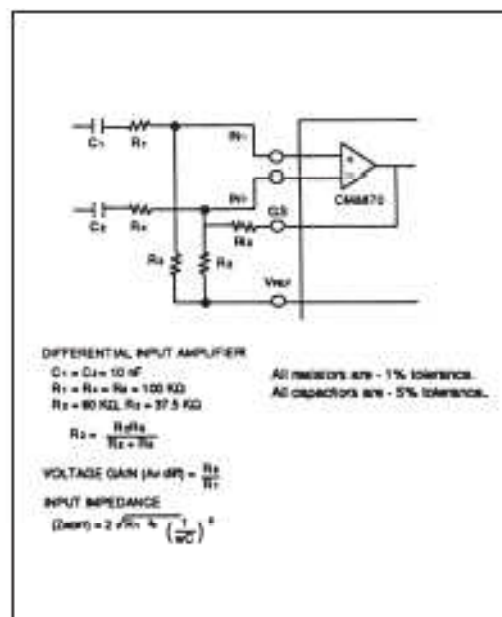


Figure 6. Differential Input Configuration



February 2003

## LM567/LM567C Tone Decoder

### General Description

The LM567 and LM567C are general purpose tone decoders designed to provide a saturated transistor switch to ground when an input signal is present within the passband. The circuit consists of an I and Q detector driven by a voltage controlled oscillator which determines the center frequency of the decoder. External components are used to independently set center frequency, bandwidth and output delay.

### Features

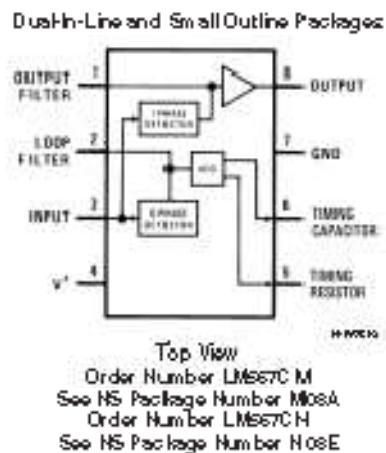
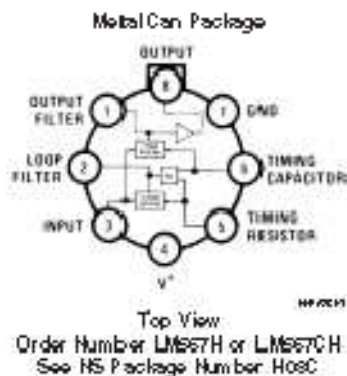
- 20 to 1 frequency range with an external resistor
- Logic compatible output with 100 mA current sinking capability
- Bandwidth adjustable from 0 to 14%

- High rejection of out of band signals and noise
- Immunity to false signals
- Highly stable center frequency
- Center frequency adjustable from 0.01 Hz to 500 kHz

### Applications

- Touch tone decoding
- Precision oscillator
- Frequency monitoring and control
- Wide band FSK demodulation
- Ultrasonic controls
- Carrier current remote controls
- Communications paging decoders

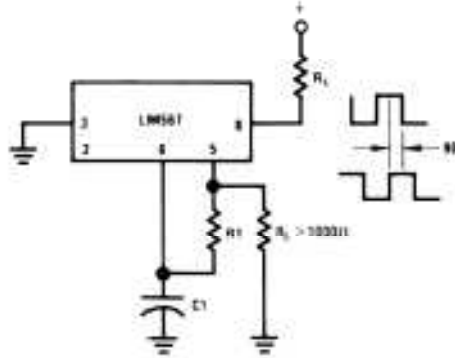
### Connection Diagrams





## Typical Applications (Continued)

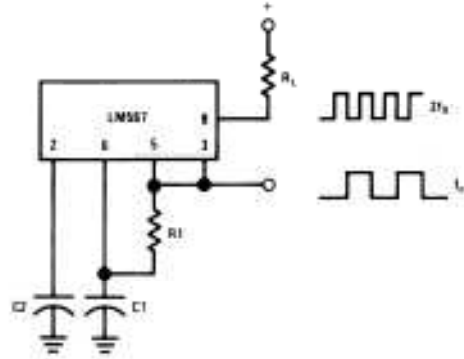
Oscillator with Quadrature Output



Connect Pin 3 to  $\pm 25V$  to Invert Output

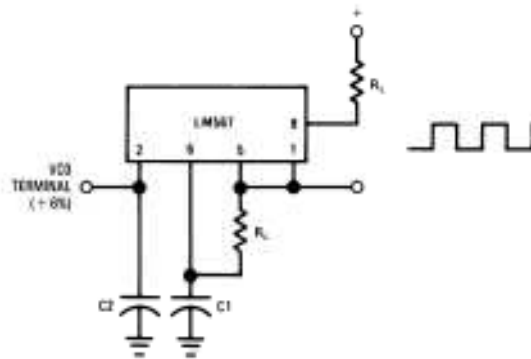
HW 70-1

Oscillator with Double Frequency Output



HW 70-2

Precision Oscillator Drive 100 mA Loads



HW 70-3

## Anexo H. Datasheet del 4N25

### MOTOROLA SEMICONDUCTOR TECHNICAL DATA

Order this document  
by 4N25/D



## 6-Pin DIP Optoisolators Transistor Output

The 4N25/A, 4N25, 4N27 and 4N28 devices consist of a gallium arsenide infrared emitting diode optically coupled to a monolithic silicon phototransistor detector.

- Most Economical Optoisolator Choice for Medium Speed, Switching Applications
- Meets or Exceeds ALL JEDEC Registered Specifications
- To order devices that are tested and marked per VDE 0884 requirements, the suffix "V" must be included at end of part number. VDE 0884 is a 3-bit option.

#### Applications

- General Purpose Switching Circuits
- Interfacing and coupling systems of different potentials and impedances
- I/O Interfacing
- Solid State Relays

#### MAXIMUM RATINGS (T<sub>A</sub> = 25°C unless otherwise noted)

Rating	Symbol	Value	Unit
--------	--------	-------	------

#### INPUT LED

Reverse Voltage	V <sub>R</sub>	3	Volts
Forward Current — Continuous	I <sub>F</sub>	60	mA
LED Power Dissipation @ T <sub>A</sub> = 25°C with Negligible Power in Output Detector Derate about 25°C	P <sub>D</sub>	120	mW
		1.41	mW/°C

#### OUTPUT TRANSISTOR

Collector-Emitter Voltage	V <sub>CE0</sub>	30	Volts
Emitter-Collector Voltage	V <sub>EC0</sub>	7	Volts
Collector-Base Voltage	V <sub>CB0</sub>	70	Volts
Collector Current — Continuous	I <sub>C</sub>	150	mA
Detector Power Dissipation @ T <sub>A</sub> = 25°C with Negligible Power in Input LED Derate about 25°C	P <sub>D</sub>	150	mW
		1.76	mW/°C

#### TOTAL DEVICE

Isolation Surge Voltage <sup>(1)</sup> (Peak ac Voltage, 60 Hz, 1 sec Duration)	V <sub>ISO</sub>	7500	V <sub>ac(pk)</sub>
Total Device Power Dissipation @ T <sub>A</sub> = 25°C Derate about 25°C	P <sub>D</sub>	250	mW
		2.94	mW/°C
Ambient Operating Temperature Range <sup>(2)</sup>	T <sub>A</sub>	-55 to +100	°C
Storage Temperature Range <sup>(2)</sup>	T <sub>stg</sub>	-55 to +150	°C
Soldering Temperature (10 sec, 1/16" from case)	T <sub>L</sub>	350	°C

1. Isolation surge voltage is an internal device dielectric breakdown rating. For its test, Pins 1 and 2 are common, and Pins 4, 5 and 6 are common.
2. Refer to Quality and Reliability Section in Data Book for information on test conditions.

<sup>(1)</sup> Turned devices are Motorola recommended choice for future use and best overall value. Global Optoisolator is a trademark of Motorola, Inc.

**4N25\***  
**4N25A\***  
**4N26\***  
(CIR = 20% Min)  
**4N27**  
**4N28**  
(CIR = 70% Min)

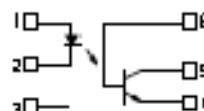
Motorola Turned Devices

STYLE 1 PLASTIC



STANDARD THRU HOLE  
CASE 730A-04

#### SCHEMATIC



PN 1. LED ANODE  
2. LED CATHODE  
3. NC.  
4. EMITTER  
5. COLLECTOR  
6. BASE

**4M25 4M26 4M27 4M28**

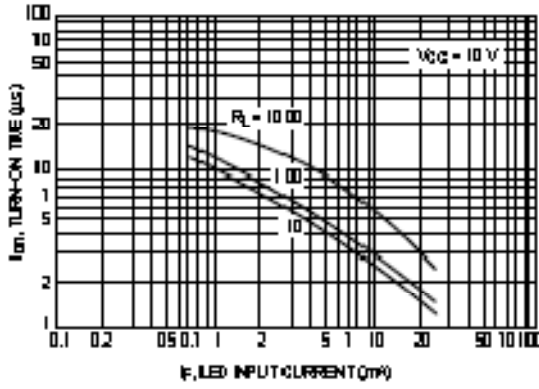


Figure 7. Turn-On Switching Times (Typical Values)

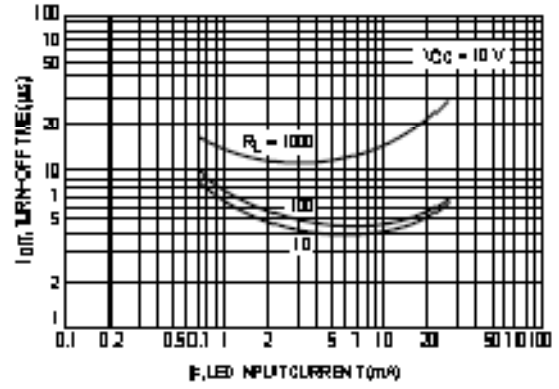


Figure 8. Turn-Off Switching Times (Typical Values)

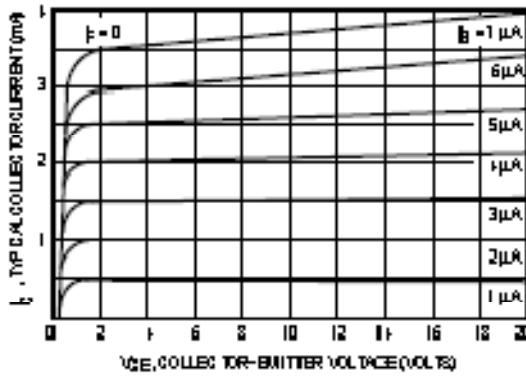


Figure 9. DC Current Gain (Detector Only)

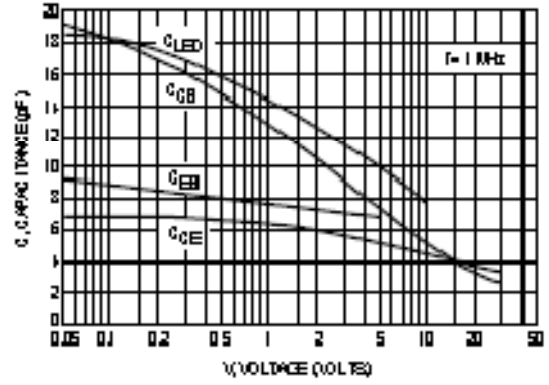


Figure 10. Capacitance versus Voltage

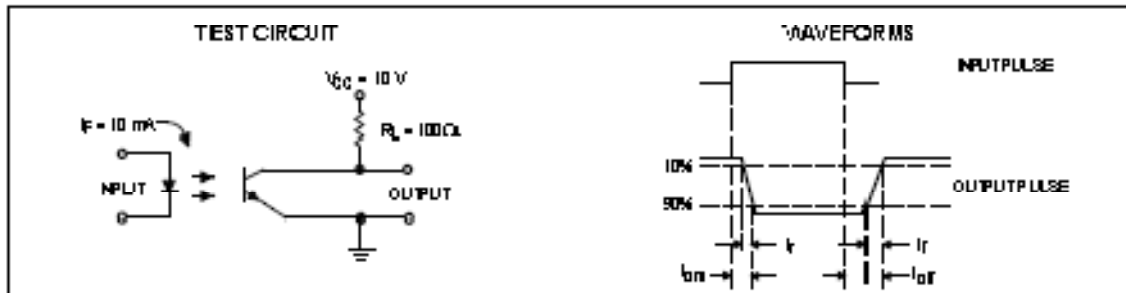
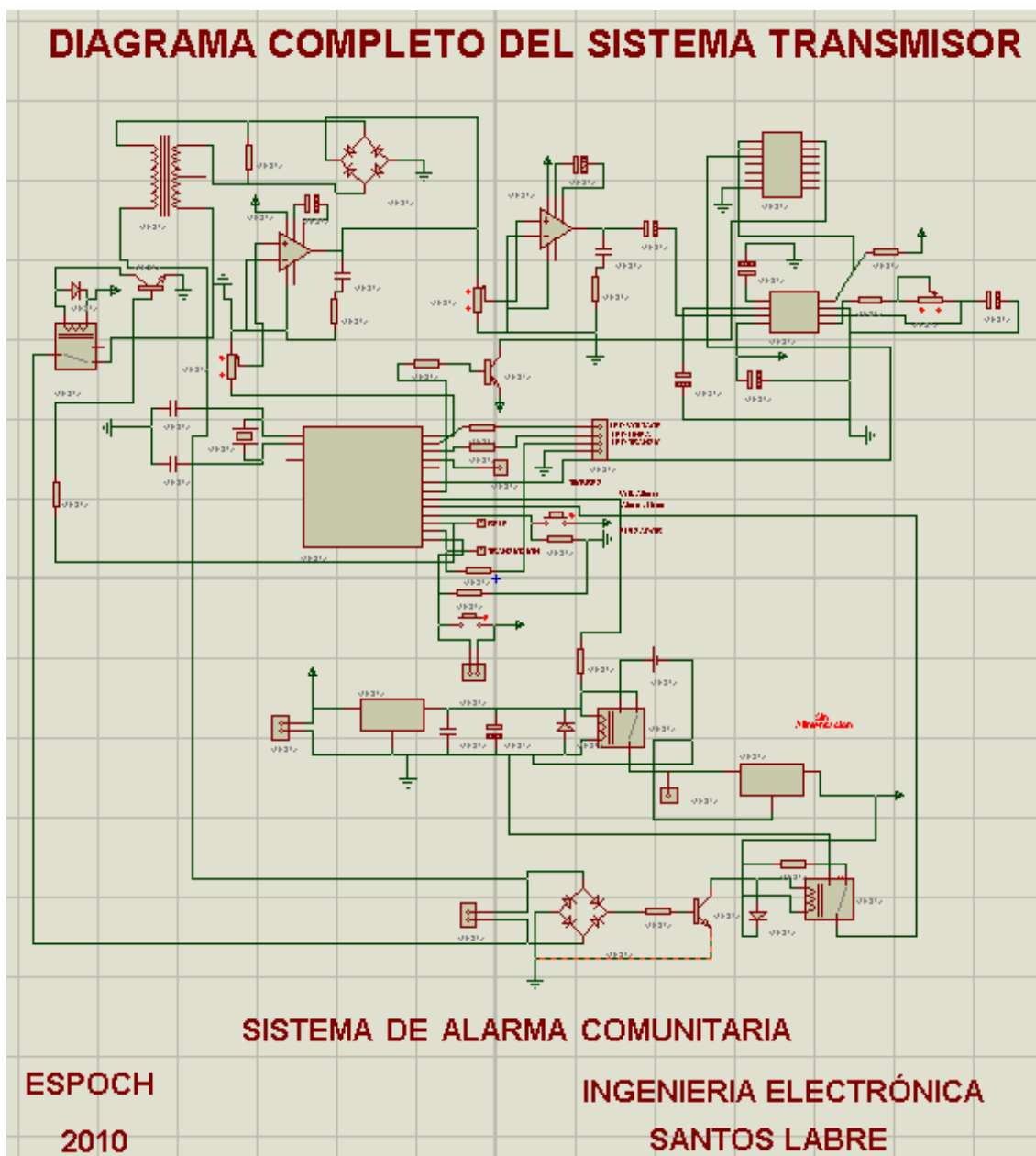


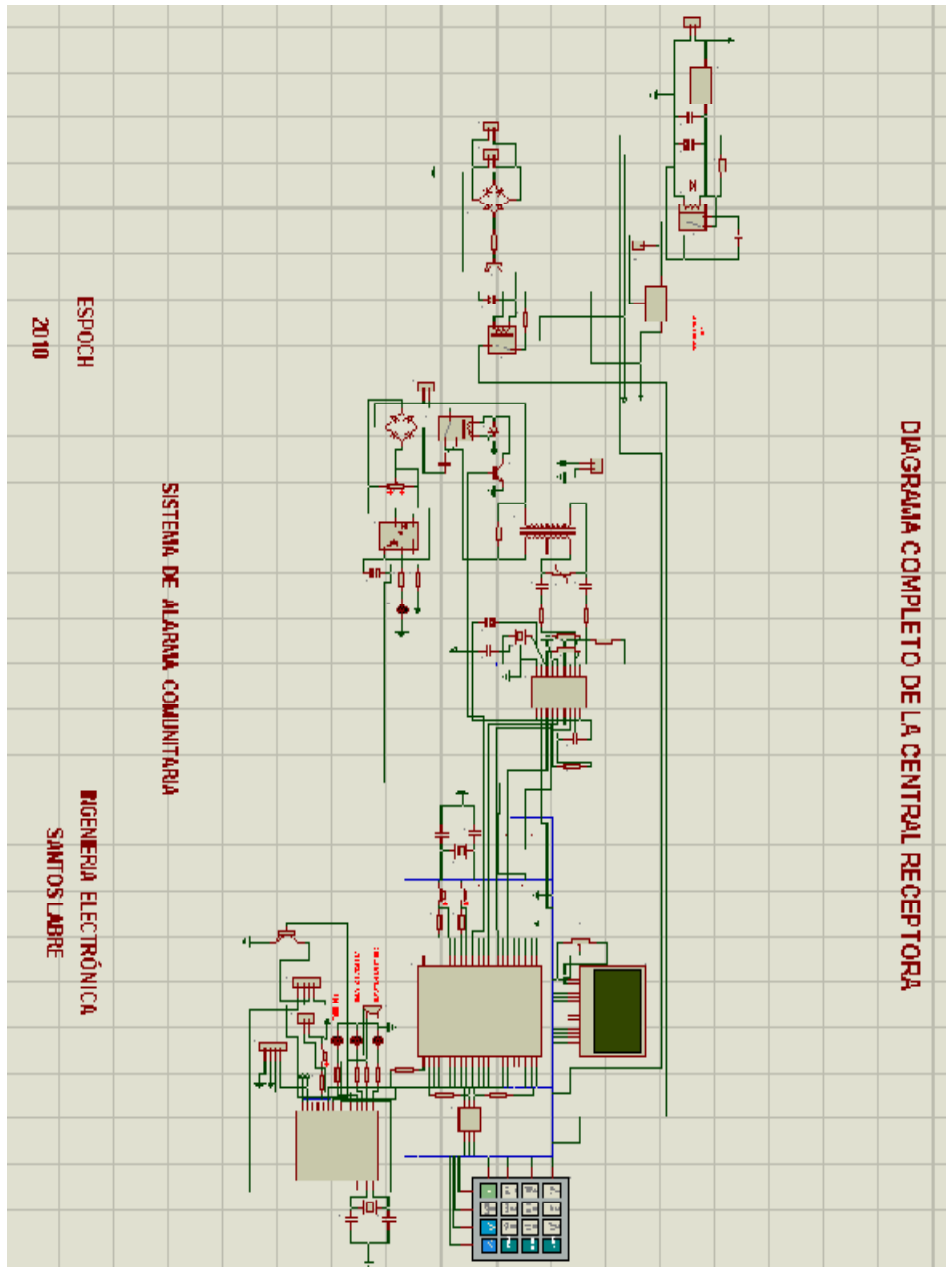
Figure 11. Switching Time Test Circuit and Waveforms

**Anexo I.** Esquema del Sistema Completo

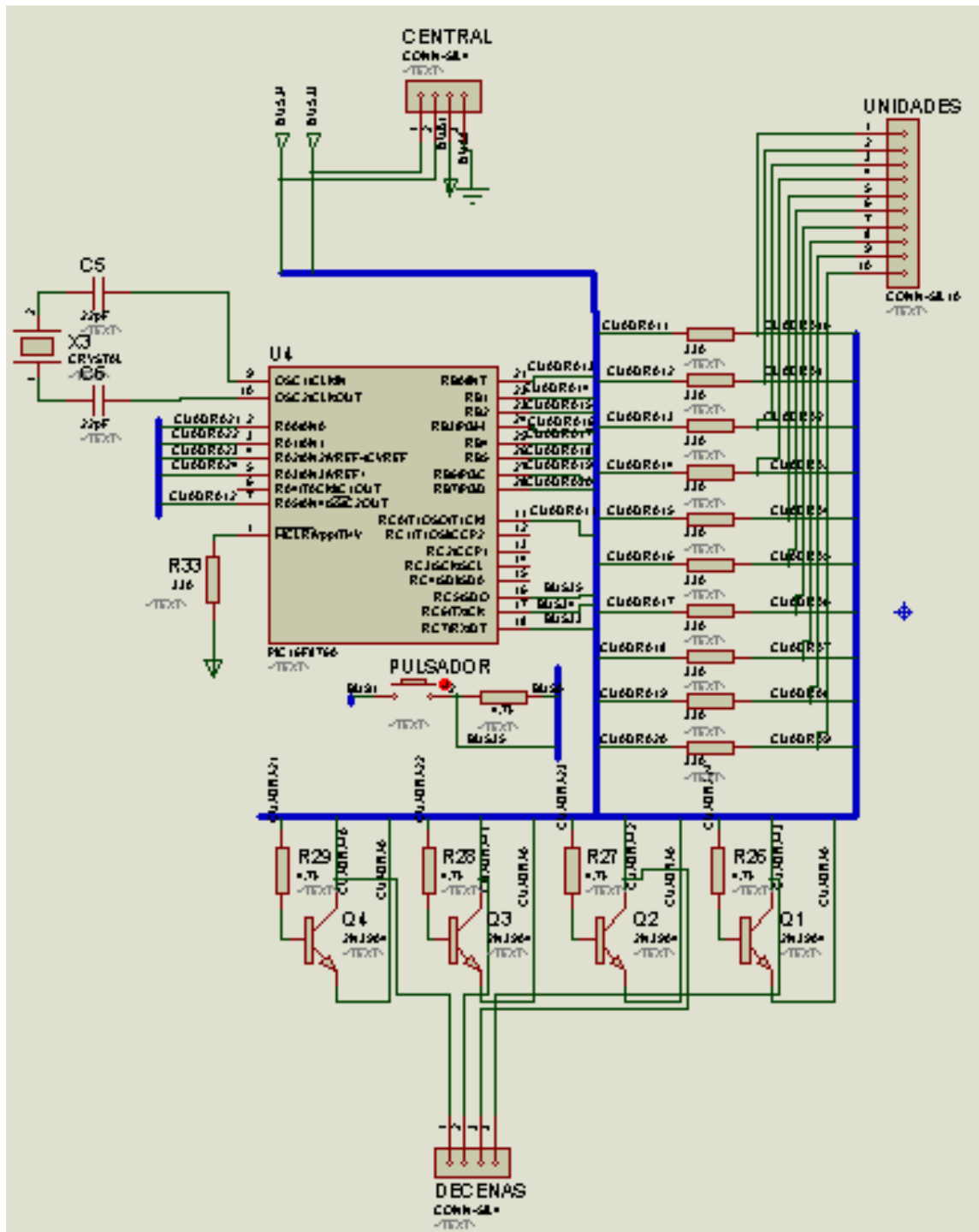
**TRANSMISOR**



RECEPTO



# PANEL DE VISUALIZACIÓN



## **Anexo J.** Código Fuente

### *TRANSMISOR*

```
@ device HS_OSC
@ device MCLR_OFF
@ device WDT_OFF
cmcon=7
define osc 20
CONTADOR VAR BYTE
alarma VAR BYTE
bandera var byte
pulsador var portb.4
bandera=0
portb.1=0
portb.2=0
portb.5=0
pulsador=0
contador=0
alarma=0
high portb.1
inicio:
if pulsador=1 then
  pause 2000
  if pulsador=1 then
    repetir:
      HIGH portb.5 ;RELE
      pause 1000
      dtmfout porta.0, [2,6,0,4,6,3,3]
      contador=0
      low portb.1
      for bandera=0 to 30
        IF PORTB.0=1 THEN
          CONTADOR=CONTADOR+1
          pause(500)
          repetir1:
            IF PORTB.0=1 THEN
              goto repetir1
            endif
          else
            pause 500
          ENDIF
        next
      high portb.1
```

```

    if (CONTADOR=0) then
        high portb.6
        dtmfout porta.0, [0,2]
        pause 1000
        low portb.6
        LOW portb.5
        CONTADOR=0
        goto salto
    else
        low portb.6
        LOW portb.5
        CONTADOR=0
        pause 3000
        goto repetir
    endif
endif
endif
salto:
    if portb.2=0 then ;led de voltage alterno
        if alarma=0 then
            high porta.1
        endif
    else
        low porta.1
        alarma =0
    endif
    if portb.3=0 then ;led de alarma de sin línea
        if alarma=0 then
            high porta.2
        endif
    else
        low porta.2
        alarma=0
    endif
    if portb.7=1 then ;led de alarma de sin línea
        LOW porta.1
        low porta.2
        low portb.5
        alarma=1
    endif
    goto inicio
end

```

*Receptor-Control central*



```

#include <18F4550.h>
#fuses XT, MCLR,NOWDT
#use delay(clock=4000000)
#use I2C(master,sda=PIN_D7,scl=PIN_E2)
#use rs232(baud=9600,bits=8, parity=N, xmit=PIN_C6,rcv=PIN_C7, long_data)
#include <LCD1.c>
#define use_portb_kbd TRUE
#include <LCD_NEW.c>
#byte porta=0x05
SHORT INT MENSAJE=1;
int grabacion, contador, accion, accion2, elemento=0,
digito_palabra=0,luz=0,LINEA=0;
long int lugar_eeprom,memoria=0,badera_salto;
byte casa() ....
void escritura_eeprom(long int address, BYTE data) .....
BYTE lectura_eeprom(long int address) .....
void borrado_eeprom() .....
void borrado_eeprom_pic() .....
long int posicion_grabar() .....
long int posicion_grabar_eeprom() .....
void informacion() .....
void proceso_modificar() ....
void proceso_eliminar() .....
void busqueda_casa(byte casa1,casa2) .....
void sensado_timbre() ....
void main(){
    int i,x,con,conta_tecla=0;
    char tecla,letra,tecla2,actuador;
    byte letra1,casa1,casa2;
    set_tris_a(0b111111);
    set_tris_e(0b101);
    OUTPUT_LOW(PIN_C2);
    lcd_init();
    kbd_init();
    port_b_pullups(TRUE);
    lcd_putc('\f');
    output_high(PIN_C1);
    While(true){
        if (input(pin_A5)==1){
            borrado_eeprom();
        }
        if (input(pin_C0)==1){
            borrado_eeprom_pic();
        }
        IF (MENSAJE==1) .....
//-----SENSADO DE TIMBRE-----

```

```

    sensado_timbre();
//-----SENSADO DE TECLADO-----
    tecla=kbd_getc();
    if(tecla!=0){
        switch(tecla){
            case 'A' :
                MENSAJE=0;
                accion=1;
                informacion();
                break;
            case 'B' :
                MENSAJE=0;
                accion=2;
                proceso_modificar();
                break;
            case 'C' :
                MENSAJE=0;
                accion=3;
                proceso_eliminar();
                break;
            case 'D' :{
                if ((accion==0)||((accion==5)) ....
                if ((accion==1)||((accion==2)&&(accion2==1))) .....
                if (((accion==2)&&(accion2==0))||((accion==4)||((accion==3)) ....
                finlista2:
                break;
                }
            case '0' :{ ..... }
            case '*' :{ ..... }
            case '#' :{ ..... }
            default :{ ..... }
        }
    }

```

### *Receptor-Actuadores*

```

#include <16F628a.h>
#fuses
XT,NOPUT,NOWDT,NOMCLR,NOBROWNOUT,NOLVP,NOPROTECT
#use delay(clock=4000000)
#use rs232(baud=9600,bits=8,parity=N ,rcv=PIN_B1, xmit=PIN_B2)
#use rs232(baud=9600,bits=8,parity=N , rcv=PIN_B7,
xmit=PIN_B6,stream=PC,FORCE_SW)
int const ruta0 [14]={ 7,1,8,2,9,3,10,4,11,5,12,6,13,14};
int const ruta1[16]={ 7,15,8,16,9,17,10,18,11,19,12,20,13,21,14,22};

```

```

int const ruta2[16]={ 15,23,16,24,17,25,18,26,19,27,20,28,21,29,22,30};
int const ruta3[14]={ 23,24,35,25,32,26,33,27,24,28,35,29,36,30};
int const ruta4[8]={ 1,7,8,15,16,23,24,31};
int const ruta5[10]={ 1,2,8,9,16,17,24,25,31,32};
int const ruta6[10]={ 2,3,9,10,17,18,25,36,32,33};
int const ruta7[10]={ 3,4,10,17,18,19,26,27,33,34};
int const ruta8[10]={ 4,5,11,12,19,20,27,28,34,35};
int const ruta9[10]={ 5,6,12,13,26,21,28,29,35,36};
int const ruta10[8]={ 6,13,14,21,22,29,30,36};
int direccion1[16];
int direccion2[16];
int direccion3[16];
char valor;
byte caracter,letra;
long int lugar;
int posicion,calle=0,calle1=0,calle2=0,calle3=0;
int calle1_largo=0,calle2_largo=0,calle3_largo=0;
void longitud(int calle1_a,calle2_a,calle3_a){ ..... }
void busqueda(){ ..... }
int proceso(int posicion,char letra){ ..... }
#int_RDA
void interrupcion_rx(){
    if (kbhit()){
        valor=getc();
        switch (valor){ ..... }
    }
}
void main(){
    int i;
    output_low(PIN_A0);
    output_low(PIN_A1);
    output_low(PIN_A2);
    output_low(PIN_A3);
    enable_interrupts(INT_RDA);
    enable_interrupts(GLOBAL);
    lugar=0;
    WHILE(TRUE){
        if (input(pin_A4)==1){
            fputc('C', PC);
            output_low(PIN_A0); //Desactivar alarma ENERGIA

```

```

output_low(PIN_A1); //Desactivar alarma BUZZER
output_low(PIN_A2); //Desactivar alarma USUARIO
output_low(PIN_A3); //Desactivar alarma SIN LINEA
valor=0;
posicion=0;
letra=0;
calle=0;
calle1=0;
calle2=0;
calle3=0;
caracter=0;
calle1_largo=0;
calle2_largo=0;
calle3_largo=0;
for (i=0;i<=15;i++){
    direccion1[i]=0;
}
for (i=0;i<=15;i++){
    direccion2[i]=0;
}
for (i=0;i<=15;i++){
    direccion3[i]=0;
}
repetir:
if (input(pin_A4)==1){
    goto repetir;
}
}
}
}

```

### *Panel de Visualización*

```

#include <16F876a.h>
#fuses XT,NOWDT,NOPUT,NOBROWNOUT,NOLVP,NOPROTECT
#use delay(clock=4000000)
#use rs232(baud=9600,bits=8,parity=N ,rcv=PIN_C7, xmit=PIN_C6)
int contador=0;
char valor;
char letra[2];

```

```

char letra3[2];
long int puntero;
byte letra2=0;

#int_RDA
void RDA_isr(){
long int posi=0;
letra2=0;
valor=getchar();
if (contador==0){
switch (valor){
case '0': .....
case '1': .....
case '2': .....
case '3': .....
case 'C': .....
valor=0;
posi=0;
repetir:
letra2 = read_eeprom(posi);
if (letra2!=255){.....}
posi=posi-1;
letra2 = read_eeprom(posi);
while (letra2!=255){.....}
letra3[0]=255;
letra3[1]=255;
puntero=0;
break;
}
}
else {
switch (valor){.....}
posi=puntero;
write_eeprom(posi, letra[0]);
write_eeprom(posi+1, letra[1]);
puntero=puntero+2;
letra[0]=0;
letra[1]=0;
}
}

```

```

void encendido(char letra3_1,letra3_2){
    long int i;
    switch (letra3_1){ .....}
    switch (letra3_2){ .....}
    for (i=0;i<500;i++){ .....}
    output_LOW(PIN_A0);
    output_LOW(PIN_A1);
    output_LOW(PIN_A2);
    output_LOW(PIN_A3);
    output_LOW(PIN_C0);
    output_LOW(PIN_A5);
    output_LOW(PIN_B0);
    output_LOW(PIN_B1);
    output_LOW(PIN_B2);
    output_LOW(PIN_B3);
    output_LOW(PIN_B4);
    output_LOW(PIN_B5);
    output_LOW(PIN_B6);
    output_LOW(PIN_B7);
}
void main(){
    long int posi=0;
    int i;
    letra[0]=0;
    letra[1]=0;
    letra3[0]=255;
    letra3[1]=255;
    set_tris_A(0x00);
    set_tris_B(0x00);
    enable_interrupts(INT_RDA);
    enable_interrupts(GLOBAL);
    output_LOW(PIN_A0);
    output_LOW(PIN_A1);
    output_LOW(PIN_A2);
    output_LOW(PIN_A3);
    output_LOW(PIN_C0);
    output_LOW(PIN_A5);
    output_LOW(PIN_B0);
    output_LOW(PIN_B1);
    output_LOW(PIN_B2);

```

```

output_LOW(PIN_B3);
output_LOW(PIN_B4);
output_LOW(PIN_B5);
output_LOW(PIN_B6);
output_LOW(PIN_B7);
puntero=0;
repetir3:
letra2 = read_eeprom(puntero);
if (letra2!=255){ .....}
WHILE(TRUE){
if (!kbhit()){ .....}
    if (input(pin_C5)==1){ .....}
        repetir2:
        output_LOW(PIN_A0);
        output_LOW(PIN_A1);
        output_LOW(PIN_A2);
        output_LOW(PIN_A3);
        output_LOW(PIN_A4);
        output_LOW(PIN_A5);
        output_LOW(PIN_B0);
        output_LOW(PIN_B1);
        output_LOW(PIN_B2);
        output_LOW(PIN_B3);
        output_LOW(PIN_B4);
        output_LOW(PIN_B5);
        output_LOW(PIN_B6);
        output_LOW(PIN_B7);
        valor=0;
        posi=0;
        letra[0]=0;
        letra[1]=0;
        puntero=0;
    }
}
}

```

## BIBLIOGRAFÍA

1. GARCIA, EDUARDO. Compilador C CCS y Simulador Proteus para Microcontroladores PIC, 2008. 83-220 p.

2. REYES, CARLOS A. Aprenda Rápidamente a Programar Microcontroladores.

Quito: Gráficas Ayerve, 2004. 193 p.

3. CONTROL A TRAVEZ DE LINEA TELEFONICA

[http://www.elotrolado.net/hilo\\_Control-a-traves-de-la-linea-telefonica\\_82165](http://www.elotrolado.net/hilo_Control-a-traves-de-la-linea-telefonica_82165)

Julio 2009.

4. DECODIFICADOR DTMF

<http://www.pablin.com.ar/electron/circuito/telefon/decdtmf/index.htm>

Julio 2009.

5. INDICADOR DE LINEA TELEFONICA

<http://www.pablin.com.ar/electron/circuito/telefon/indtele3/index.htm>

Julio 2009.

6. REPETIDOR LUMINOSO DE TIMBRE TELEFONICO

<http://www.pablin.com.ar/electron/circuito/telefon/timblamp/index.htm>

Agosto 2009.



## 7. REMOTE DTMF DECODER

[http://www.g8dhe.com/newlogic/dtmf\\_remote\\_decoder.htm](http://www.g8dhe.com/newlogic/dtmf_remote_decoder.htm)

Agosto 2009.

## 8. CALLER LINE IDENTIFICATION(CLI)

[http://www.picbasic.nl/frameload\\_uk.htm?http://www.picbasic.nl/numm\\_ermelder\\_uk.htm](http://www.picbasic.nl/frameload_uk.htm?http://www.picbasic.nl/numm_ermelder_uk.htm)

Agosto 2009.

## 9. TELEPHONE LINE AUDIO INTERFACE CIRCUITS

<http://www.epanorama.net/circuits/teleinterface.html>

Agosto 2009.

## 10. TELEPHONE CALL LOGGER

[http://www.ozitronics.com/docs/k164\\_sc.pdf](http://www.ozitronics.com/docs/k164_sc.pdf)

Septiembre 2009

## 11. AUDIO DTMF DECODER

<http://www.ozitronics.com/docs/k153.pdf>

Octubre 2009

## GLOSARIO

**Microcontrolador:** Un microcontrolador es un sistema cerrado que contiene un computador completo y de prestaciones limitadas que no se puede modificar, todas las partes están contenidas en su interior y solo salen al exterior las líneas a los periféricos.

**Microprocesador:** Un microprocesador es un sistema abierto con el que puede construirse un computador con las características que se desee, acoplándose los módulos necesarios.

**Oscilador electrónico:** Un oscilador es un circuito electrónico que produce una señal electrónica repetitiva, a menudo una onda senoidal o una onda cuadrada, que permite el funcionamiento del microcontrolador.

**RS-232:** En telecomunicaciones es un estándar para la conexión serial de señales de datos binarias entre dos dispositivos electrónicos, generalmente usado en puertos seriales de computadoras.

**Señal de Timeout:** Señal que se presenta en una línea telefónica cuando al abonado al que se lo llama al central telefónica avisa que esta la línea congestionada.