



UNIVERSIDAD NACIONAL AUTÓNOMA DE  
MÉXICO

---

**FACULTAD DE INGENIERIA**

**DISEÑO DE UN SISTEMA DE MONITOREO Y CONTROL  
PARA CASA HABITACIÓN BASADO EN DTMF Y WEB.**

**T E S I S**

QUE PARA OBTENER EL GRADO ACADÉMICO DE

**INGENIERO MECANICO ELECTRICISTA**

P R E S E N T A N

Gómez Figueroa Silvia Liliana  
Medina Jiménez Jorge  
Muñoz Sánchez Daniel Jonathan  
Núñez Villaluz Pablo  
Pérez González Jorge

DIRECTOR DE TESIS: M. EN I. JORGE VALERIANO ASSEM

MÉXICO, D. F.

2010

“El hombre no llega a ser hombre más que por la  
educación. No es más que lo que la educación  
hace de él.  
Inmanuel Kant

“Los hombres han nacido los unos para los otros;  
edúcales y padécelos”  
Marco Aurelio

## **Dedicatorias y Agradecimientos**

A dios por iluminarme el camino y estar siempre conmigo

A mis padres Blanca y Carlos por enseñarme el amor al estudio y apoyarme siempre en mis proyectos

A mis hijos Carlos Andrés y Victor Manuel por el tiempo que les robe y ser mi compañía siempre

A mis hermanas por compartir siempre mis triunfos

A mis amigos por los grandes momentos que hemos compartido

A mis maestros por su dedicación y empeño para hacer de nosotros mejores hombres

A la UNAM por permitirme tener una educación integral

Silvia Liliana Gómez Figueroa

---

Introducción.....	I
Capítulo 1 Marco Teórico.....	1
1.1 Sistemas de Telefonía.....	1
1.1.1 La Red de Telefonía .....	1
1.1.2 El Sistema de Telefonía Celular .....	2
1.1.3 IVR .....	8
1.2 Internet.....	10
1.2.1 Historia y desarrollo de Internet .....	10
1.2.2 Protocolos .....	11
1.2.3 Modelo OSI .....	12
1.2.4 Arquitectura del protocolo TCP/IP .....	13
1.2.5 Direcciones de Internet.....	14
1.2.6 Servidores de Internet.....	15
1.2.7 Servidores DNS .....	16
1.2.8 Seguridad en las redes .....	17
1.2.9 Red Privada Virtual (VPN).....	18
1.3 Microcontroladores.....	20
1.3.1 Microprocesadores y Microcontroladores .....	20
1.3.2 Antecedentes históricos .....	21
1.3.3 El mercado de los microcontroladores .....	23
1.3.4 Arquitectura básica .....	24
1.3.5 Fabricantes.....	26
1.3.6 Microcontrolador HC11 .....	27
1.3.6.1 Unidad central de procesamiento (CPU) .....	29
1.3.6.2 Periféricos on-chip.....	31
1.3.6.3 Sistema de interrupciones.....	34
1.4 Sensores y Transductores electrónicos.....	35
1.4.1 Características deseables de los transductores (sensores) .....	36
1.4.2 Tipos de sensores.....	37
1.5 Filtros.....	40
1.5.1 Filtros ideales.....	41
1.5.2 Filtros digitales .....	41
1.6 Lenguajes de Programación.....	46
1.6.1 Evolución de los lenguajes de programación. ....	47
1.6.2 Fases en la resolución de problemas.....	47
1.6.3 Programación modular .....	48
1.6.4 Programación Estructurada.....	49
1.6.5 Programación orientada a objetos.....	50
Capítulo 2. Detección de Tonos Multi-frecuencia.....	51
2.1 Señalización.....	51
2.2 Hardware .....	54
2.3 Software.....	55
2.4 Algoritmo de Goertzel .....	58

Capítulo 3 Diseño de hardware y software para detección de tonos con el Microcontrolador .....	62
3.1 Comunicación Serie Asíncrona Entre el MCU y la PC.....	62
3.2 Norma RS232 .....	63
3.3 Conector DB9.....	64
3.4 El chip MAX32 .....	65
3.5 Interacción del microcontrolador y la PC.....	67
3.6 Diseño de software de interfaz para el MCU .....	68
3.6.1 PUMMA 11 .....	68
3.7 Programas empleados en el microcontrolador.....	72
3.7.1 Interrupción del programa (Int_Prog.asm) .....	72
3.7.2 Transmisión de Datos (Trans_dato.asm).....	76
Capítulo 4. Diseño de Software.....	79
4.1 Introducción.....	79
4.2 Diseño de portal Web .....	79
4.2.1 Código Portal Web .....	82
4.2.2 Autenticación de usuarios.....	87
4.2.3 Adquisición de datos Web.....	89
4.3 Diseño de aplicación Voz .....	93
4.3.1 Circuito de acoplamiento.....	94
4.3.2 Algoritmo de la aplicación de Voz .....	95
4.3.3 Diseño de software y código fuente .....	95
4.3.4 Módulos para detector tonos duales de multifrecuencia .....	107
Capítulo 5 Integración de Bloques Funcionales .....	109
5.1 SMC (Sistema de Monitoreo y Control) .....	110
5.2 Servidor WEB (Local y Remoto) .....	111
5.3 Consulta telefónica .....	111
5.4 Consulta por Internet .....	114
5.5 Monitoreo por el HC11 .....	117
Capítulo 6. Resultados, aplicación y conclusiones.....	118
6.1. Resultados.....	118
6.2. Aplicación.....	120
6.3. Conclusiones.....	123
Bibliografía.....	125
Acrónimos .....	127
Glosario .....	128
Anexo 1 .....	129

## Introducción

Con todos los avances tecnológicos que se han venido presentando en las últimas tres décadas todos los seres humanos aspiramos a una mejor calidad de vida y parece lógico que esa aspiración se refleje en nuestro hogar, al que todos nos esforzamos en dotar de mayor confort, comodidad y seguridad. La “ciencia” encargada de integrar los avances tecnológicos en un hogar es la Domótica. En este entorno hay una palabra muy importante para concebir dicho término llamada “integración”, todas las necesidades se deben satisfacer de forma global y en conjunto. En otro caso no puede hablarse de Domótica, sino simplemente de la automatización de tal o cual actividad.

Resulta imposible precisar una fecha concreta para el nacimiento de la automatización de un espacio como tal, ya que no se trata de un hecho puntual, sino de todo un proceso evolutivo que comenzó con las redes de control de los edificios inteligentes y se ha ido adaptando a las necesidades propias de la vivienda.

Luego se implantan sistemas de control de las instalaciones. En un principio, a comienzos de los años 80, sólo se hacían cosas básicas como la gestión integral de calefacción y aire acondicionado, que hasta entonces se hacía de forma aislada. Posteriormente se ha ido sofisticando continuamente hasta llegar a una integración total de la gestión.

En este aspecto es importante realizar una aclaración en la construcción del presente trabajo ya que no se pretende aplicar Domótica a un cierto espacio, sino, simplemente automatizar cierta actividad que nos permita monitorear o controlar ciertos aspectos.

El objetivo principal de la automatización de ciertas actividades en un espacio radica en satisfacer todas las necesidades del hombre y su entorno. Podemos destacar en este campo la optimización de la energía, el aumento de la seguridad, ahorro de dinero y tiempo, así como también comunicarse, proteger el hogar, controlar a distancia los aparatos domésticos, en otras palabras, vivir mejor.

Por otro lado y derivado de la inseguridad que prevalece en las grandes ciudades de nuestro país, se hace entonces necesario tener un sistema para monitorear nuestro hogar. En el mercado actual existen sistemas que permiten monitorear una casa habitación, pero su costo aun es muy elevado, y partiendo de la premisa de que en un gran número de hogares en metrópolis ya se cuenta con una PC con acceso a Internet, el motivo de este proyecto es generar un sistema que permita, con una inversión menor, poder realizar dicha actividad vía una línea telefónica (celular o convencional) verificando con un menú preestablecido de voz, el estado de puertas, ventanas o luces, esto también se podrá verificar en una página WEB, y en caso de que se altere el estado físico de dichos elementos se almacena y se lee desde una base de datos conociendo así el estado actual

El presente trabajo por tanto está basado en lo siguiente:

En el capítulo I Marco Teórico, iniciamos hablando de la red telefónica y la telefonía celular sus inicios y evolución. En el caso de la telefonía Celular ahondamos en los diferentes estándares. Se comentan las características que diferencian las tres generaciones de telefonía celular.

Abordamos el tema del inicio y evolución de la Internet hasta llegar a ser la gran herramienta de comunicación de nuestros tiempos, la arquitectura del protocolo TCP/IP, los protocolos de seguridad utilizados, los diferentes servidores de Internet.

Hablamos también de los microcontroladores, su evolución, el mercado de los controladores, su arquitectura básica, los fabricantes más importantes, y explicamos los rasgos generales del microcontroladores HC11

También en este capítulo abordamos el tema de los sensores y transductores eléctricos, su clasificación en analógicos y digitales, su fiabilidad y ventajas de unos sobre otros, dependiendo de su aplicación.

Por último tratamos el tema de los filtros, su utilización para limitar la banda de señales a transmitir, los filtros paso-bajas, paso-altas, paso-banda, los filtros reales de primero y segundo orden y por último los filtros con fuente de voltaje controlada por voltaje.

En el capítulo 2 se explica la norma DTMF (Dual Tone Multi Frequency) la cual hace posible la interpretación de las teclas marcadas en un sistema de telefonía por tonos, así como, el Hardware y el Software encargados de la detección de dichos tonos.

En el capítulo 3 se describe el Hardware asociado con el microcontrolador y el diseño del software del mismo en lenguaje ensamblador para detectar los tonos y realizar las interrupciones que permitirán detectar cierto estado.

En el capítulo 4 se describe el software en Visual Basic con el cual son generados los menús de voz que el usuario escuchará para marcar cierta tecla de su aparato telefónico. De igual forma el software para el portal WEB realizado en ASP (Active Server Page) que se encarga de leer una base de datos que contiene los estados de los elementos a monitorear.

Por último en el capítulo 5 se describe la integración de todos los bloques que hacen posible la tarea antes descrita, señalando las entradas y salidas de los mismos para que cada bloque en conjunto sea contemplado como sistema.

## Capítulo 1 Marco Teórico

### 1.1 Sistemas de Telefonía

#### 1.1.1 La Red de Telefonía

El 10 de marzo de 1876 Alexander Graham Bell logró comunicar a dos personas utilizando un micrófono de carbón, una batería, un electroimán y un diafragma de hierro que en conjunto permitían el envío y recepción de la voz, había inventado el teléfono.

Dado que cada circuito de este tipo requería de un enlace físico y era imposible tener una conexión punto a punto para cada persona con la que se quisiera comunicar, se envió el cable de cada cliente a un switch para que éste conmutara la llamada; en un principio había personal que se encontraban detrás del switch preguntando por el destinatario para realizar la conexión en el momento, con la creciente demanda resultó imposible dar abasto a tantos clientes, para solucionarlo se crearon la llamada telefónica y el conmutador telefónico; esto es, marcar un identificador telefónico y enviar la voz a un switch automatizado que conmute la llamada a su destino, dando rapidez al sistema y permitiendo una fácil escalabilidad.

La red que nos permite realizar una llamada telefónica de un punto a otro del país es conocida como Red de Telefonía Pública Conmutada (PSTN) y se compone de cuatro partes principales:

El aparato, la transmisión, la conmutación y la señalización. El primero incluye también los periféricos como identificador de llamada y contestadora automática. La transmisión son los diversos modos de enviar la información dependiendo de si es voz, datos o ambos. La conmutación concierne a la manera en que están conectados los enlaces y la forma de encontrar el camino correcto de comunicación. Finalmente la señalización, que controla la red telefónica y administra las conexiones.

Inicialmente, para que el teléfono generara la marcación se usaban impulsos enviados a través de la línea telefónica, pero desde hace unas décadas la selección del número por medio de pulsos fue sustituida por una selección de distintas frecuencias donde a cada número le corresponde un par de frecuencias distinto. De esta forma al digitar un número, se envían sus dos frecuencias correspondientes sobre la línea telefónica. A esta técnica de marcación se le conoce como DTMF (Dual Tone MultiFrequency).

### **1.1.2 El Sistema de Telefonía Celular**

#### **Historia de la telefonía celular**

Con la invención del sistema de radio, Marconi inicia en 1896, la era de las telecomunicaciones por medio de ondas electromagnéticas.

El desarrollo de la radiocomunicación móvil es usado por primera vez en Detroit en el año de 1921 por el departamento de Policía de esta ciudad. Las patrullas se comunicaban con el puesto de central por medio de un mismo canal debiendo avisar cuando hacían uso de éste. A este sistema se le nombró "Push to Talk".

Posteriormente al inicio de los años cincuenta se desarrollaron sistemas que usan un canal separado para hablar; estos sistemas al tener doble banda evitan que el usuario pida cambio de canal para ceder el turno, a estos se les conoce como sistemas "Full Duplex" porque la comunicación es simultánea en ambos sentidos. La telefonía celular es ideada por los laboratorios Bell en el año de 1947.

En 1946 AT&T que se desarrolla el sistema IMTS (Improved Mobile Telephone System) que trabajaba en la frecuencia de 450 Mhz y le permitió a los clientes marcar directamente sus llamadas, sin la ayuda de una operadora, contando con una central automática y una antena que cubría toda el área de servicio.

En 1969 en Europa, los países Nórdicos generan el estándar de Sistemas Celulares NMT (Nordic Mobile Telephone) y en 1981 se instala en Arabia Saudita el primer sistema celular del mundo por el proveedor Ericsson en la frecuencia de 450 Mhz.

### **Estándares Celulares**

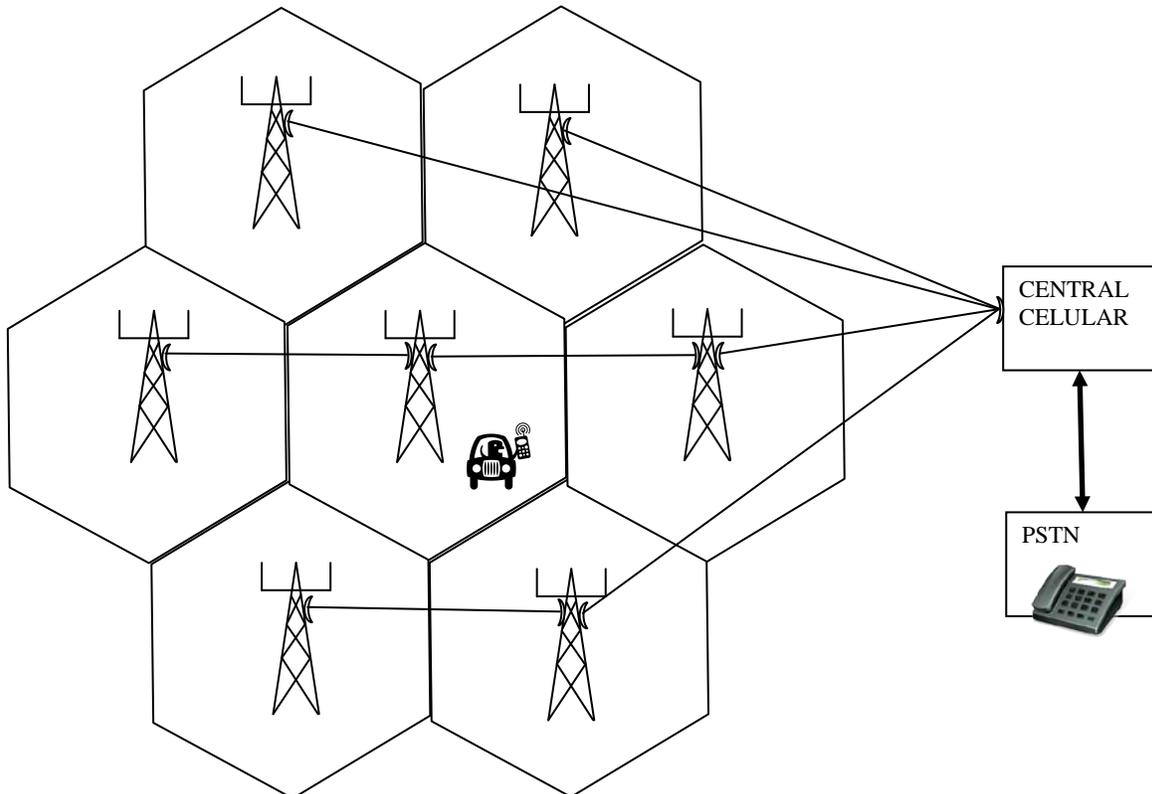
Desde el desarrollo del estándar NMT 450 en 1981 se han desarrollado varios estándares alrededor del mundo para las comunicaciones móviles. Cada estándar se desarrolló pensando en los requerimientos particulares de cada país o intereses de los grupos involucrados en las especificaciones. Por esta razón un estándar utilizado en un país puede no ser el mismo para otro. En la **tabla 1** se incluyen los principales estándares, así como los principales mercados en el mundo.

Año	Estándar	Sistema de Telefonía Móvil	Tecnología	Principales Mercados
1981	NTM 450	Nortic Mobile Telephony	Analógica	Europa, Medio Oriente
1983	AMPS	Advanced Mobile Phone System	Analógica	Norte y Sur América
1985	TACS	Total Access Communication System	Analógica	Europa y China
1986	NTM 900	Nordic Mobile Telephony	Analógica	Europa, Medio Oriente
1991	GSM	Global System for Mobile Communication	Digital	Todo el mundo
1991	TDMA  (D-AMPS)	Time Division Multiple Access (Digital-AMPS)	Digital	Norte y Sur América
1993	CdmaOne	Code Division Multiple Access One	Digital	Norte América, Korea
1992	GSM 1800	Global System for Mobile Communication	Digital	Europa
1994	PDC	Personal Digital Celular	Digital	Japón
1995	PCS 1900	Personal communication Services	Digital	Norte América
2001	GSM 800	Global System for Mobile Communications	Digital	Norte América
2001	WCDMA  UMTS	Code Division multiple Acces Universal Mobile Telephone System	Digital	Todo el mundo

**Tabla 1.1. Estándares de Telefonía Celular.**

## Radio Telefonía Móvil Celular

El sistema celular es un moderno sistema de telecomunicaciones que satisface las necesidades de comunicación tecnológica, ya que permite estar en contacto a toda hora y desde cualquier lugar dentro del Área de Servicio Celular.



**Figura 1.1. Diagrama básico de un Sistema Celular.**

El término “celular” se refiere a la manera en que están agrupadas las zonas de servicio que proporcionan el sistema por medio de las estaciones de radio (radiobases). Estas radiobases permiten un contacto bidireccional con la unidad móvil ó teléfono celular permitiendo el establecimiento de la conversación telefónica. Ver figura 1.1. Cada radiobase esta conectada a una Central Celular (MSC), dicha central a su vez también esta conectada a la Red de Telefonía

Publica Conmutada, para poder dar paso a las llamadas que entran o salen de la Red Celular.

También se pueden interconectar varios MSC para realizar la función de ROAMING, que consiste en poder hacer y recibir llamadas en diferentes áreas de Servicio Celular. Ver figura 1.1

Dentro de las características principales de la Telefonía Celular, están:

- Amplia capacidad de dar servicio a suscriptores
- Uso eficiente del espectro (reuso de frecuencias)
- Compatibilidad nacional e internacional
- Amplia disponibilidad
- Adaptabilidad a la densidad de tráfico
- Servicios de valor agregado

### **Concepto de Radiobase y la Célula**

La radiobase es la interfase entre el teléfono celular y la central celular, y es capaz de comunicarse con cualquier teléfono celular mientras se mueva dentro de un área cerca de esta, que es su área de cobertura.

Dependiendo del tipo de antenas de transmisión empleadas por la radiobase se puede cubrir una o más áreas a las que se les denomina células. Existen los siguientes dos tipos de células:

#### **Células Omnidireccionales**

En este tipo las antenas transmiten igualmente en todas direcciones, este tipo de configuración es más utilizada cuando se desea cubrir áreas rurales o suburbanas con baja densidad de tráfico.

#### **Células Sectoriales**

Para este caso, en la radiobase se instalan 3 antenas direccionales que cubren una célula sectorial de 120 grados a la que se le asigna cierta capacidad de tráfico a cada uno de los sectores. Esta configuración es más utilizada en zonas urbanas de alta densidad de tráfico para mejorar la penetración de la señal, así como distribuir la capacidad de tráfico en los diferentes sectores, pudiendo asignar mayor capacidad a cualquiera de ellos.

Es posible tener radiobases con dos sectores por ejemplo para cubrir tramos carreteros.

El desarrollo de la telefonía celular se engloba en 3 generaciones:

### **Telefonía Celular de 1ra Generación**

La primera generación de telefonía celular inicia con el desarrollo del estándar NTM 450 que se utilizó en Europa y Medio Oriente en 1981 así como los estándares AMPS y TACS. En estos inicios únicamente se podía realizar llamadas de voz, no se podían transmitir datos la comunicación entre la radiobase y el teléfono móvil es analógica

### **Telefonía Celular de 2da Generación**

Al inicio de la década de los 90's se liberan los estándares GSM, TDMA y CdmaOne con los cuales ya es posible además de la comunicación de voz el envío de datos, la comunicación entre la radiobase y el móvil es totalmente digital permitiendo hacer más eficiente el uso del espectro. Se inicia el envío de mensajes de texto, también llamado Short Message Service o SMS. En la transmisión de datos se pueden alcanzar velocidades de 114 Kbps

### **Telefonía Celular de 3ra Generación**

En algunos sistemas celulares de tercera generación que se encuentran implantados en México se está migrando de GSM a UMTS, la diferencia principal radica en los siguientes dos conceptos:

---

- Circuitos Conmutados (Circuit Switched)
- Paquetes Conmutados (Paquet Switched)

En los Circuitos Conmutados (Circuit Switched) se usa una conexión punto a punto (una conexión dedicada todo el tiempo) en donde no se está hablando todo el tiempo, ya que al momento de hablar se tienen pausas entre palabras dando una eficiencia del 35%. Una analogía de esto es como si en un restaurante se reserva una mesa para todo el día y a la hora de la comida se utiliza por 3 horas y el resto no se utiliza.

En los Paquetes Conmutados (Paquet Switched) se tiene el envío de paquetes de información por demanda que no necesariamente son en tiempo real y cuya eficiencia llega al 85% resultando en costos más bajos. En UMTS se incrementa la velocidad que tiene el sistema para enviar datos, pudiendo alcanzar velocidades de hasta 14 Mbps.

La diferencia entre uno y otro es que en circuitos conmutados se está utilizando el dispositivo todo el tiempo y si hay espacio de tiempo sin enviar datos, como el dispositivo está asignado a un suscriptor, ningún otro suscriptor puede enviar información por ese dispositivo. En el caso de paquetes conmutados solo se utiliza el medio cuando se tiene información por enviar, aprovechando de mejor manera el ancho de banda y optimizando recursos.

### 1.1.3 IVR

IVR son las siglas de Interactive Voice Response, que se traduce del inglés como *Respuesta de Voz Interactiva*. También se utiliza el término VRU (*Voice Response Unit*) *Unidad de Respuesta por Voz*.

Consiste en un sistema telefónico que es capaz de recibir una llamada e interactuar con el usuario a través de grabaciones de voz y el reconocimiento de respuestas simples, como "sí", "no" u otras. Lo podemos definir también como un sistema automatizado de respuesta interactiva, orientado a entregar y/o capturar información a través del teléfono, permitiendo el acceso a servicios de información u otras aplicaciones.

Concibiéndolo técnicamente el IVR consiste en una plataforma de hardware, más una conexión a red de área local, a la cual se conectan host de bases de datos y la aplicación propiamente dicha.

Los sistemas automatizados de reconocimiento de voz permiten a los llamantes usar directamente palabras o frases que estos sistemas reconocen y convierten en comandos que se comunican de forma interactiva con aplicaciones informáticas.

Cuando los usuarios del sistema llaman a un centro de servicios automatizados basado en un sistema IVR, una serie de menús grabados les van guiando sobre las diferentes opciones/servicios que se van presentando. Los clientes hacen su elección contestando desde el teclado de su teléfono, siempre y cuando dicho aparato contenga marcación por tonos multifrecuencia (DTMF), o por respuesta hablada, y en función de cada respuesta, la aplicación IVR realiza una serie de acciones sobre la base de datos. Entre éstas se incluye proporcionar determinado tipo de información recogida de la base de datos o archivos de documentos y su lectura, traducidos a voz mediante sistemas de conversión/síntesis texto-voz, que no abarcaremos en este apartado.

Por su definición, los sistemas IVR son de uso indispensable en los servicios de atención de llamadas, ya que proporcionan una descongestión necesaria en aquellos servicios de atención de llamadas saturados por consultas simples a bases de datos o información general.

Los beneficios que proporcionan los sistemas IVR para los centros de atención a clientes son:

- Reducción de costos: los sistemas IVR sustituyen a un gran número de agentes o material humano sin disminuir la productividad de alguna empresa.
- Actuar como receptor de llamadas en horarios fuera de atención al público.
- Aumento de la disponibilidad del servicio.
- Identificación y verificación de la identidad del usuario llamante.

En la actualidad existen una considerable cantidad de fabricantes y desarrolladores de estos sistemas con aplicaciones tan complejas como nos podamos imaginar, en el presente trabajo se contempla el uso de esta tecnología simplemente para que guie al usuario a elegir tal o cual acción recibiendo la respuesta de los tonos del aparato telefónico y ejecutando dicha actividad.

## **1.2 Internet**

### **1.2.1 Historia y desarrollo de Internet**

En 1969 el departamento de defensa de los Estados Unidos de América trabajaba en un proyecto denominado DARPA NET para cubrir la necesidad de contar con un sistema de comunicaciones fiable, capaz de comunicar diferentes estados del país en caso que los sistemas telefónicos existentes se viesen dañados por un ataque militar.

Las investigaciones realizadas dieron como fruto el nacimiento del protocolo TCP/IP (Transmisión Control Protocol/Internet Protocol), base de un sistema

robusto de comunicaciones cuyo envío de información en su unidad elemental es denominada *datagrama*.

Durante el desarrollo de dicho protocolo, se vio un incremento en las redes gubernamentales y de universidades para participar en el proyecto, dando origen a la red de redes mundial. Poco a poco las funciones militares se separaron del desarrollo, permitiendo el acceso a la red de todas aquellas instituciones que lo requirieran, siempre y cuando el acceso fuera con fines académicos y/o de investigación. A finales de 1972 ya había más de 40 nodos conectados a la red. En 1977 y 1979 ARPA era la pionera en la investigación de redes de paquetes conmutados con su red ARPANET.

Internet global se inició alrededor de 1980 cuando ARPA comenzó a convertir sus máquinas conectadas a su red al protocolo TCP/IP.

En el año de 1992 dejó de funcionar la red que dio origen a Internet, ARPANET, pero en ese mismo año un informático del CERN (Organización Europea de Investigación Nuclear) llamado, Tim Berners Lee creó el lenguaje HTML (Hypertext markup Language), con el que nació la World Wide Web.

### **1.2.2 Protocolos**

Al conjunto de convenios de comunicación entre dispositivos y aplicaciones se les denomina protocolos. Los protocolos son el conjunto de reglas que rigen el intercambio de datos entre dos entidades. Los puntos clave que caracterizan un protocolo son:

- **Sintaxis:** el formato de los datos y los niveles de señal

- **Semántica:** la información de control para la coordinación y el manejo de errores.
- **Temporización:** la sintonización de velocidades y secuenciación

### 1.2.3 Modelo OSI

El modelo OSI (Open System Interconnection) se desarrolló por la Organización Internacional de Estandarización ISO (International Organization for Standardization) como una arquitectura para comunicaciones entre computadoras, con el objetivo de ser el marco de referencia en el desarrollo de protocolos estándares. Las capas que considera son:

- **Aplicación.** Proporciona a los programas de aplicación un medio para que accedan al entorno.
- **Presentación.** Define el formato de los datos que se van a intercambiar entre las aplicaciones y ofrece a los programas de aplicación un conjunto de servicios de transmisión de datos.
- **Sesión.** Proporciona los mecanismos para controlar el dialogo entre las aplicaciones de los sistemas finales.
- **Transporte.** Proporciona los mecanismos para intercambiar datos entre los sistemas finales.
- **Red.** Realiza la transferencia de información entre sistemas finales a través de alguna red de comunicaciones. Libera a las capas superiores de tener conocimiento de la transmisión de datos subyacentes y las tecnologías de conmutación utilizadas para conectar los sistemas.
- **Enlace de datos.** Proporciona los medios para activar, mantener y desactivar el enlace. El principal servicio que ofrece a las capas superiores es la detección y control de errores.

- **Física.** Se encarga como su nombre lo indica, de la interfaz física entre los dispositivos, además define las reglas de la transmisión de los bits. Las características más importantes son: mecánicas, eléctricas, funcionales y de procedimiento.

#### 1.2.4 Arquitectura del protocolo TCP/IP

El modelo TCP/IP está organizado en cuatro capas conceptuales por la forma en que los datos pasan por ellas (**figura 1.2**):

- **Capa de aplicación.** Está en el nivel más alto, pasa los datos en la forma requerida por la capa de transporte para su entrega.
- **Capa de transporte.** Su principal tarea es proporcionar la comunicación entre un programa de aplicación y otro, regulando el flujo de información. Puede también proporcionar un transporte confiable, asegurando que los datos lleguen sin errores y en secuencia.
- **Capa de Internet.** Esta capa maneja la comunicación de una máquina a otra. Maneja la entrada de datagramas, verificando su validez y utiliza un algoritmo de ruteo para decidir si debe procesarlo de manera local o debe transmitirlo.
- **Capa Física.** Define las características del medio, señalización y codificación de las señales.

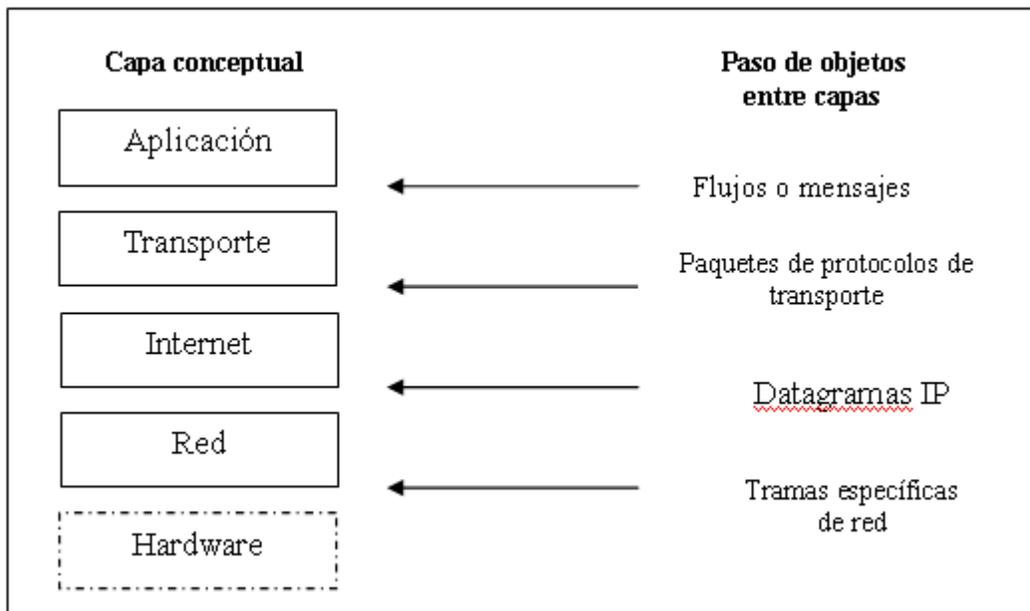


Figura 1.2 Diagrama del Protocolo TCP/IP.

### 1.2.6 Direcciones de Internet

La dirección que utiliza el protocolo IP para identificar los equipos en Internet, esta formada por 32 bits que se divide en cuatro grupos de ocho bits separados por punto. Como un número de 8 bits ofrece 256 combinaciones, tenemos que cada grupo es un número decimal entre 0 y 255. A cada grupo de 8 bits se le conoce como octeto.

Las direcciones a su vez, se dividen en 3 clases estandarizados de direcciones IP:

**Clase A:** Se emplea el primer octeto de la dirección para identificar la red en la que se encuentra el equipo, y los 3 restantes para identificar el equipo.

**Clase B:** Los 2 primeros octetos se usan para identificar la red y los 2 últimos para los ordenadores (hosts).

**Clase C:** Se utilizan 3 octetos para identificar la red y el último octeto para los ordenadores (hosts).

**Clase D:** Todos los segmentos son utilizados para identificar una red y son reservados para los llamados multicast.

Para controlar el uso de las direcciones IP y evitar desperdiciar direcciones, existe un organismo llamado IANA (Internet Assigned Numbers Authority) que además asigna los números de puertos para aplicaciones. En colaboración con IANA trabaja ISOC que entre otras funciones, gestiona las peticiones de direcciones, e INIC.

### 1.2.7 Servidores de Internet

A las computadoras que están conectadas a Internet se les nombran hosts. Estos tienen direcciones IP propias y únicas en la red, cuando los usuarios se conectan desde sus PC, lo que hacen es conectarse a ellos y utilizar los diferentes recursos y servicios que ofrecen; a esta arquitectura se le llama cliente-servidor por que el servidor ofrece un determinado servicio a uno o más clientes que lo utilizan temporalmente conectándose a el de modo remoto. Este tipo de arquitectura hace de Internet un sistema distribuido de información, que permite acceder a una cantidad casi inagotable de recursos en todo el mundo.

Se clasifican a los servidores que componen Internet de acuerdo con los servicios que ofrecen en:

- Servidores WEB (ofrecen páginas HTML)

- Servidores FTP
- Servidores E-MAIL
- Servidores IRC
- Servidores Gopher
- Servidores DNS, etc

Una computadora puede tener la función de uno o más de estos tipos de servidor, ofreciéndolos a través de diferentes puertos.

### **Conexión a Internet**

- Si todos los usuarios actuales de Internet intentaran conectarse a Internet como un nodo más, serían insuficientes las direcciones IP existentes. Para solucionar este problema existen los servidores o proveedores de acceso, que se encuentran conectados a la red. El usuario contrata el servicio, paga una cuota y se conecta al servicio.

#### **1.2.8 Servidores DNS**

El sistema de nombres por domino DNS (Domain Name System) se utiliza para identificar una maquina en lugar de buscarla con una dirección IP, es necesario un sistema que traduzca los nombres a dichas direcciones IP. Esta tarea es realizada por los servidores DNS, los cuales mantienen las bases de datos que vinculan la IP con el nombre de dominio asignado.

Cuando deseamos acceder a un determinado recurso de Internet, este puede localizarse gracias a la URL (Uniform Resource Locator) que es la dirección en cierto formato que nos permite identificar el recurso de modo unívoco en la red. El formato de las URL es el siguiente:

- Identificador del protocolo del servicio a través del cual vamos a acceder al recurso
- Dirección IP del servidor en el que se encuentra el recurso solicitado
- Por último el nombre del fichero concreto, incluyendo la ruta completa donde se encuentra dentro del servidor.

Dado que este formato es complejo y difícil de recordar para el usuario, la IP del servidor puede ser sustituida por un nombre que la identifique y sea más fácil de manejar. Este nombre está compuesto por el dominio y subdominio en los que se encuentra el servidor separados por puntos, por ejemplo, [www.nic.com](http://www.nic.com), donde la última parte identifica el tipo de dominio que se trata (.mx uno de México, .us uno de Estados Unidos, .com comercial, .gob del gobierno, etc).

Los nombres de dominio son asignados por los diferentes NIC (Network Information Center) que funcionan a nivel nacional en cada país. Estos garantizan que los nombres de dominio que se asignan sean únicos.

### **1.2.9 Seguridad en las redes**

Actualmente Internet se enfrenta a los ataques de seguridad. En sus inicios Internet tenía como principal objetivo crear una red mundial descentralizada y escalable para el intercambio de información, sin embargo, debido a las necesidades de confidencialidad de la información transferida por las nuevas aplicaciones, se tuvo la necesidad de implementar medidas de seguridad para evitar que se haga uso inadecuado de la información.

El método más seguro de impedir estas entradas no deseadas en el sistema consiste en establecer barreras fiables de control entre la información accesible al público y la confidencialidad, llamadas firewalls.

**Técnicas más comunes de ataque**

**IP spoofing.** Consiste en falsear la dirección IP de origen que va en los paquetes que le llegan al servidor haciéndolo creer que se está conectando de una máquina autorizada.

**Secuestro de sesiones.** Se trata de acceder a través de una sesión ya iniciada por otro usuario.

**Prueba y error.** Consiste en tratar de entrar al servidor mediante diferentes usuarios y contraseñas.

**Diccionario de encriptación.** Muchos servidores realizan comparaciones de contraseñas una vez codificadas, de modo que contiene las contraseñas encriptadas. Un modo muy utilizado es atacar los servidores, accediendo posteriormente a la tabla de resultados de encriptación de contraseñas, comparándolas para obtener un patrón que los lleve a descubrir el patrón de encriptación.

**Caballo de Troya.** Una vez que han accedido a un sistema, pueden instalar un programa que deje abierta alguna puerta para que le sea más sencillo entrar posteriormente.

**Sniffers.** Es un programa que una vez introducido de algún modo en el servidor recoge y almacena todas las operaciones que se realicen en el.

**1.2.10 Red Privada Virtual (VPN)**

Es una red virtual que permite utilizar Internet como una WAN (Wide Area Network) privada, lo cual implica que los datos transmitidos entre una fuente y un destino no son accesibles a usuarios no autorizados. Los datos transportados a través de una VPN son codificados de manera que solo los fuente y destino pueden decodificarlo.

### **Protocolos de seguridad VPN**

**PPTP.** Utiliza un algoritmo de encriptación punto a punto patentado por Microsoft que proporciona encriptación y autenticación para conexiones de enlace telefónico remoto y conexiones LAN a LAN.

**L2F.** Este protocolo proporciona tunelaje entre un servidor de enlace telefónico del ISP y la red. En esta aplicación, un usuario debe realizar una conexión de enlace telefónico de protocolo Punto a Punto (PPP) con el servidor del ISP. Este servidor se encapsula en PPP dentro de un bloque L2F, que es enviado a un dispositivo capa 3 para su transmisión a la red. El enrutador es el responsable de la autenticación de los usuarios y direccionamiento de la red.

**L2TP.** Define el método para el tunelaje de sesiones PPP a través de la red. Combina PPTF y L2F. Encapsula PPP, que luego encapsula en IP o IPX. Utiliza en puerto UDP 1701. No ofrece cifrado por si mismo.

**IPsec.** Es un grupo de protocolos desarrollados por el EITF. Opera en la capa 3 del modelo OSI, proporciona autenticación de dirección, codificación de datos, e intercambios automáticos de claves entre los nodos emisor y receptor.

## 1.3 Microcontroladores

### 1.3.1 Microprocesadores y Microcontroladores

La palabra microprocesador ( $\mu P$ ) procede de la contracción de *microcircuit processor*, y se aplica a los circuitos integrados que contienen un procesador o CPU. Una CPU es un circuito digital capaz de realizar operaciones aritméticas y lógicas a partir de un conjunto de instrucciones previamente almacenadas en forma de programa. La CPU contiene un bloque funcional llamado *unidad de control* (UC) que es capaz de encontrar la siguiente instrucción a ejecutar y activar los recursos necesarios para completar dicha ejecución.

Los microcontroladores son conceptualmente iguales a los microprocesadores, es decir, siguen el mismo esquema de funcionamiento y su diferencia reside en los elementos que se integran en el chip del  $\mu C$  ya que habitualmente estos circuitos incluyen memoria ROM, RAM, periféricos de entrada y salida, entre otros. El área destinada a estos elementos merma la capacidad computacional del circuito integrado y hace que los  $\mu C$  tengan un conjunto de instrucciones más reducido, menor capacidad de cálculo. Por esta razón, podemos decir que las áreas de aplicación de los  $\mu C$  y de los  $\mu P$  son complementarias.

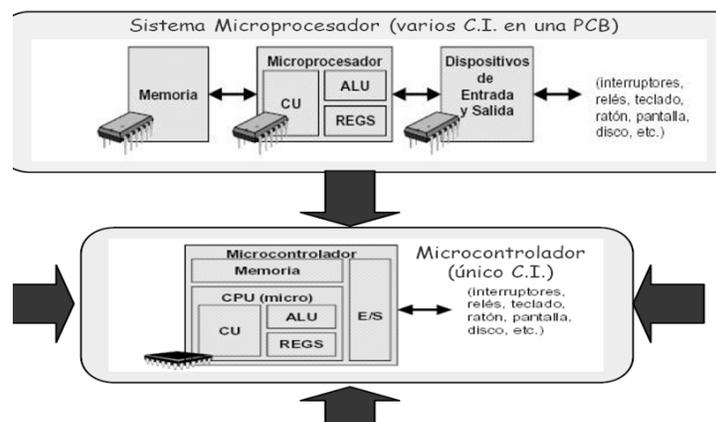


Figura 1.3 Microprocesador y microcontrolador

---

### 1.3.2 Antecedentes históricos

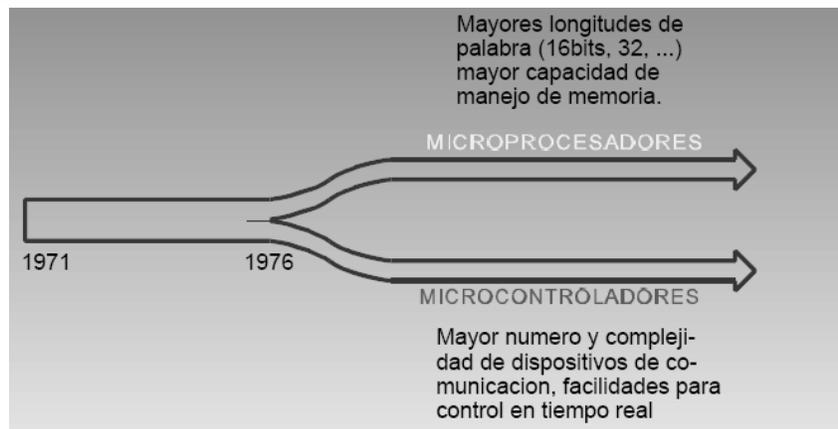
La siguiente es una lista cronológica de los eventos tecnológicos más recientes que han tenido impacto sobre la aparición y el desarrollo del campo de los microcontroladores en la electrónica digital.

- 1971 Intel fabrica el primer microprocesador (el 4004) de tecnología PMOS. Este era un microprocesador de 4 bits y fue fabricado por Intel a petición de Datapoint Corporation con el objeto de sustituir la CPU de terminales inteligentes fabricadas en esa fecha por Datapoint mediante circuitería discreta. El dispositivo fabricado por Intel resultó 10 veces más lento de lo requerido y Datapoint no lo compró, de esta manera Intel comenzó a comercializarlo. El 4004 podía direccionar sólo 4096 (4k) localidades de memoria de 4 bits, reconocía 45 instrucciones y podía ejecutar una instrucción en 20  $\mu$ seg en promedio.
- 1972 Las aplicaciones del 4004 estaban muy limitadas por su reducida capacidad y rápidamente Intel desarrolló una versión más poderosa (el 8008), el cual podía manipular bytes completos, por lo cual fue un microprocesador de 8 bits. La memoria que este podía manejar se incrementó a 16 kbytes, sin embargo, la velocidad de operación continuó siendo lo mismo.
- 1973 Intel lanza al mercado el 8080 el primer microprocesador de tecnología NMOS, lo cual permite superar la velocidad de su predecesor (el 8008) por un factor de diez, es decir, el 8080 puede realizar 500 000 operaciones por segundo, además se incrementó la capacidad de direccionamiento de memoria a 64 kbytes. A partir del 8080 de Intel se produjo una revolución en el diseño de microcomputadoras y varias compañías fabricantes de circuitos integrados comenzaron a producir microprocesadores.

---

Algunos ejemplos de los primeros microprocesadores son: el IMP-4 y el SC/MP de National Semiconductors, el PPS-4 y PPS-8 de Rockwell International, el MC6800 de Motorota y el F-8 de Fairchild.

- 1975 Zilog lanza al mercado el Z80, uno de los microprocesadores de 8 bits más poderosos. En ese mismo año, Motorola abate dramáticamente los costos con sus microprocesadores 6501 y 6502 (este último adoptado por APPLE para su primera microcomputadora personal) que se comercializan en \$20 y \$25 (dls. USA) respectivamente., provocando un auge en el mercado de microcomputadoras de uso doméstico y un caos en la proliferación de lenguajes, sistemas operativos y programas (ningún producto era compatible con el de otro fabricante).
- 1976 Surgen las primeras microcomputadoras de un solo chip, que más tarde se denominarán microcontroladores. Dos de los primeros microcontroladores, son el 8048 de Intel y el 6805R2 de Motorola.
- 198X En la década de los 80's comienza la ruptura entre la evolución tecnológica de los microprocesadores y la de los microcontroladores ya que los primeros siguieron incorporando cada vez más y mejores capacidades para las aplicaciones en donde se requiere el manejo de grandes volúmenes de información mientras los segundos han incorporado más capacidades que les permiten la interacción con el mundo físico en tiempo real, así como mejores desempeños en ambientes de tipo industrial. Esta ruptura se representa esquemáticamente en la figura 1.3.2.



**Figura 1.6 Ruptura entre el microprocesador y el microcontrolador.**

### 1.3.3 El mercado de los microcontroladores

Existe una gran diversidad de microcontroladores cuya clasificación más importante se da por el número de bits: 4, 8, 16 ó 32 bits en los que las prestaciones de los microcontroladores de 16 y 32 bits son superiores a los de 4 y 8 bits. Los microcontroladores de 8 bits dominan el mercado y los de 4 bits se resisten a desaparecer. Por que los microcontroladores de 4 y 8 bits son apropiados para la gran mayoría de las aplicaciones.

Uno de los sectores que más ocupa y determina el tipo de microcontroladores a usara es el mercado automovilístico. De hecho, algunas de las familias de microcontroladores actuales se desarrollaron pensando en este sector, siendo modificadas posteriormente para adaptarse a sistemas más genéricos. El mercado del automóvil es además uno de los más exigentes: los componentes electrónicos deben operar bajo condiciones extremas de vibraciones, choques, ruido, etc. y seguir siendo fiables.

En cuanto a las técnicas de fabricación, prácticamente la totalidad de los microcontroladores actuales se fabrican con tecnología CMOS 4 (Complementary

Metal Oxide Semiconductor) que supera a las técnicas anteriores por su bajo consumo y alta inmunidad al ruido.

La distribución de las ventas según su aplicación es la siguiente:

- Un 30% se absorbe en las aplicaciones relacionadas con los computadores y sus periféricos.
- Otro 25% se utiliza en las aplicaciones de consumo (electrodomésticos, juegos, TV, vídeo, etc.)
- El 20% de las ventas mundiales es para el área de las comunicaciones.
- Un 15% se emplea en aplicaciones industriales.
- El resto de los microcontroladores vendidos en el mundo, aproximadamente un 10% son adquiridos por las industrias de automoción.

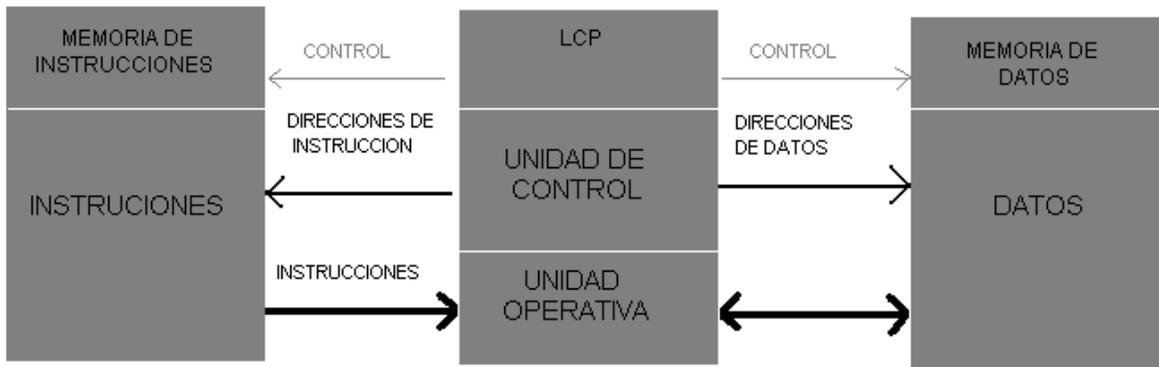
Los microcontroladores de 32 bits van afianzando sus posiciones en el mercado, siendo las áreas de más interés el procesamiento de imágenes, las comunicaciones, las aplicaciones militares, los procesos industriales y el control de los dispositivos de almacenamiento masivo de datos.

#### **1.3.4 Arquitectura básica**

Aunque inicialmente todos los microcontroladores adoptaron la arquitectura clásica de Von Neumann, en la actualidad se impone la arquitectura Harvard.

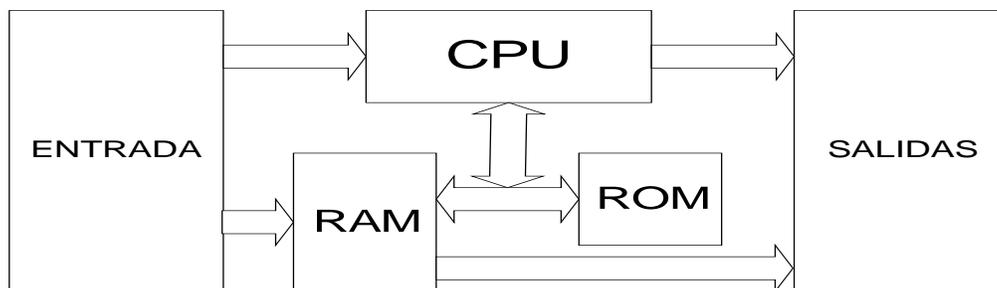
La arquitectura Von Neumann se caracteriza por disponer de una sola memoria principal donde se almacenan datos e instrucciones de forma indistinta. Se accede a dicha memoria a través de un sistema de buses único (direcciones, datos y control).

La arquitectura Harvard dispone de dos memorias independientes; una que contiene sólo instrucciones y otra, sólo datos. Ambas disponen de sus respectivos sistemas de buses y es posible realizar operaciones de acceso (lectura o escritura) simultáneamente en ambas memorias.



**Figura 1.7 Arquitectura de Harvard**

Todos los microcontroladores tienen en común: la unidad central de proceso (CPU), memoria de programa (ROM), memoria de datos (RAM) así como las entradas – salidas.



**Figura 1.8 Arquitectura básica de un microcontrolador.**

El CPU o procesador está compuesto esencialmente por una unidad lógica aritmética (ALU), registro/s de trabajo, contador de programa, decodificador de instrucciones y algunos otros registros especiales.

---

### 1.3.5 Fabricantes

Varias empresas se dedican a la fabricación, investigación y soporte de microcontroladores, la que sigue es una lista de fabricantes y sus modelos más conocidos.

#### Intel

**8048** Es el padre de los microcontroladores actuales, el primero de todos. Su precio, disponibilidad y herramientas de desarrollo hacen que todavía sea muy popular.

**8051 (Intel y otros)** Es sin duda el microcontrolador más popular pues es fácil de programar y potente además de que está bien documentado y posee cientos de variantes e incontables herramientas de desarrollo.

**80186, 80188 y 80386 EX** Son las versiones en microcontrolador de los populares microprocesadores 8086 y 8088. Su principal ventaja es que permiten aprovechar las herramientas de desarrollo para PC.

Otros microcontroladores de Intel son el **8049** y el **80C196** entre otros.

#### Motorota

**68HC11 (Motorola y Toshiba)** Es un microcontrolador de 8 bits potente y popular con gran cantidad de variantes.

**683xx** Surgido a partir de la popular familia 68k, a la que se incorporan algunos periféricos. Son microcontroladores de

---

altísimas prestaciones.

Otros microcontroladores de Motorola son: **6805, 68HC12, 68HC16**.

### **MicroChip**

**PIC (MicroChip)** Familia de microcontroladores que gana popularidad día a día. Fueron los primeros microcontroladores RISC.

### **Otros Fabricantes**

Zilog, Texas Instruments, Infineon, Dallas, NEC, National Semiconductor, Hitachi, Phillips, SGS-Thomson...

#### **1.3.6 Microcontrolador HC11**

Para el desarrollo de la presente tesis se optó por el microcontrolador HC11, tomando en cuenta características como: costos de operación, disponibilidad, seguridad, diseño y consumo de energía.

Gracias a sus recursos internos y la facilidad de su programación, el microcontrolador 68HC11 es muy popular tanto en la industria como en los centros de educación e investigación.

El microcontrolador (MCU) 68HC11 es un circuito integrado que incorpora una unidad central de proceso (CPU) y una serie de recursos internos: memoria RAM, memoria ROM, memoria EEPROM, puerto serie, puertos de entrada/salida, temporizadores, comparadores, capturadores, etc., además, tiene un CPU que le permite ejecutar instrucciones almacenadas en una memoria.

El microcontrolador F1 es un derivado de la familia M68HC11 de 8 bits, caracterizada por una gran velocidad de procesamiento, bajo consumo de corriente y un diseño completamente estático. En la memoria EEPROM on-chip pueden guardarse programas y datos.

Entre sus principales especificaciones están:

- CPU de 8 bits (68HC11).
- Modos de operación para ahorro de energía (STOP, WAIT).
- 512 bytes EEPROM.
- 1024 bytes RAM.
- Buses de datos y direccionamiento no multiplexado.
- Temporizador de 16 bits.
- Canales de captura.
- Comparadores de salida.
- Acumulador de pulsos de 8 bits.
- Interrupciones en tiempo real.
- Convertidor analógico – digital de 8 canales y 8 bits de resolución.
- Puerto serie asíncrono no retorno a cero (NRZ).
- Puerto serie asíncrono.

El microcontrolador puede usarse básicamente en 4 modos diferentes de Operación:

1. Modo de chip único (single-chip modus)
2. Modo de varios chips (expanded modus)
3. Modo de prueba, por ejemplo, para la prueba de módulos en factoría (test modus)
4. Modo de arranque (bootstrap modus)

La selección del modo de operación deseado se realiza según la conexión de los terminales MODA y MODB. El modo de prueba está cableado de forma permanente en el módulo y permite direccionar memoria externa en un espacio de direcciones de hasta 64 KB. Los bits de dirección de mayor valor salen por el puerto B. Los bits de dirección de menor valor y los datos salen por el puerto C multiplexados por división de tiempo.

### **1.3.6.1 Unidad central de procesamiento (CPU)**

La CPU de la familia M68HC11 es una versión perfeccionada de la CPU 6801 y compatible con ésta.

#### **Registros**

Los siguientes registros se encuentran a disposición para el programador:

- Acumulador A (8 bits)
- Acumulador B (8 bits)
- Acumulador D (16 bits, formado por acum. A + acum. B)
- Registro índice X (IX) (16 bits)
- Registro índice Y (IY) (16 bits)
- Puntero de pila SP (16 bits)
- Contador de programa PC (16 bits)
- Registro de códigos de condición CCR (8 bits)

Estos registros forman parte de la CPU y por lo tanto no están localizados en el espacio de direcciones del microcontrolador.

Acumuladores (A, B, D) funcionan como memorias de operandos y resultados en operaciones aritméticas y lógicas. A y B son iguales con muy pocas excepciones.

Registros índice (X, Y) los registros índice de 16 bits se usan para instrucciones con direccionamiento indexado. También pueden emplearse para guardar operandos.

Puntero de pila (SP) funciona como indicador de la siguiente dirección libre de la pila (memoria en pila). Los datos son almacenados en la pila según el principio "last in - first out". Una vez que los datos han sido almacenados, el valor indicado por el puntero se reduce.

Contador de programa (PC) contiene la dirección de la próxima instrucción a ser ejecutada.

Registro de códigos de condición (CCR) contiene 5 banderas de estado, 2 bits para enmascarar interrupciones y un bit des habilitador de la instrucción STOP (stop disable bit).

### **Instrucciones**

La CPU no sólo puede ejecutar todas las instrucciones de la 6800 y la 6801, sino que dispone además de 91 códigos de instrucción nuevos. A disposición se encuentran los siguientes grupos de instrucciones:

- Instrucciones de carga
- Instrucciones de almacenamiento
- Instrucciones de desplazamiento
- Instrucciones de rotación
- Instrucciones lógicas
- Instrucciones de testeo de bits
- Instrucciones de colocación y borrado de bits
- Instrucciones aritméticas

- Instrucciones de ramificación de programa
- Instrucciones de ejecución de subrutinas, interrupciones y memoria en pila (stack).

El número de ciclos de reloj que se necesitan para la ejecución de las instrucciones y otros detalles pueden consultarse en su lista de instrucciones. La CPU del microcontrolador del módulo PSD2 necesita para un ciclo de reloj 500  $\mu$ s.

### **Tipos de direccionamiento**

- Direccionamiento inmediato (immediate addressing)
- Direccionamiento extendido (extended addressing)
- Direccionamiento directo (direct addressing)
- Direccionamiento indexado (indexed addressing)
- Direccionamiento inherente (inherent addressing)

### **Memorias**

En el espacio de direcciones lineal (64 KB), el microcontrolador dispone de ROM, RAM y EEPROM. Desde cada uno de dichos sectores puede ejecutarse código de programa.

#### **1.3.6.2 Periféricos on-chip**

##### **Puertos de entrada y salida**

Dependiendo del modo de operación, el microcontrolador dispone de un máximo de 38 líneas de entrada/salida (puertos PA, PB, PC, PD y PE). Las funciones de cada puerto varían según el modo de operación del microcontrolador. En el modo especial de prueba PB y PC se emplean para la conexión del chip PSD. Los pines

PD6 y PD7 se usan además como señales de control del bus externo, y los pines PD0 y PD1 para la comunicación serie con el PC anfitrión empleado.

Función de los pines de puerto:

Input	PA0 - PA2, PE0 - PE7
Output	PA3 - PA6, PB0 - PB7 (A8 - A15)
Input/Output	PA7, PD0 - PD5, PC0- PC7 (AD0 – AD7)

Los registros respectivos se encuentran localizados en el área de direcciones a partir de \$1802 como se indica a continuación.

P0	Input	\$1802
	Direction	\$1804
	Output	\$1806
P1	Input	\$1803
	Direction	\$1805
	Input	\$1807

### Serial Communication Interface (SCI)

La interfaz de comunicación serie es una interfaz serie asíncrona full dúplex. La transferencia de datos se realiza por caracteres, estando formado un carácter por un bit de arranque, 8 ó 9 bits de datos, un bit de paridad y un bit de parada. De forma alternativa puede emplearse en vez del bit de paridad un segundo bit de parada. Los registros de funciones especiales SCCR1 y SCCR2 se usan para configurar el modo de operación. La velocidad de transferencia es ajustada a

través del SFR BAUD. Para los datos recibidos y los datos a transmitir está el registro SCDR.

### **Sistema de temporización**

Además del temporizador del modo de operación estándar (generación de interrupciones cíclicas), el sistema dispone de la posibilidad de ejecutar funciones de comparación y captura a través de varios canales. Para ello se emplea un contador de 16 bits autónomo alimentado por un divisor previo (predivisor) con el impulso de reloj dividido para realizar la temporización. Mediante la función de comparación pueden generarse fácilmente señales PWM (pulse-width modulation).

El sistema de temporización consta además de un acumulador de pulsos con una capacidad de 8 bits que puede ser usado como temporizador o contador.

Serial Peripheral Interface (SPI).

A través de la interfaz de periféricos serie se puede conectar al 68HC11 diversos circuitos integrados tales como memorias, convertidores A/D u otros microcontroladores. En este caso, el microcontrolador puede programarse como maestro o esclavo. La interfaz puede configurarse o emplearse para el intercambio de datos a través de 3 registros de funciones especiales.

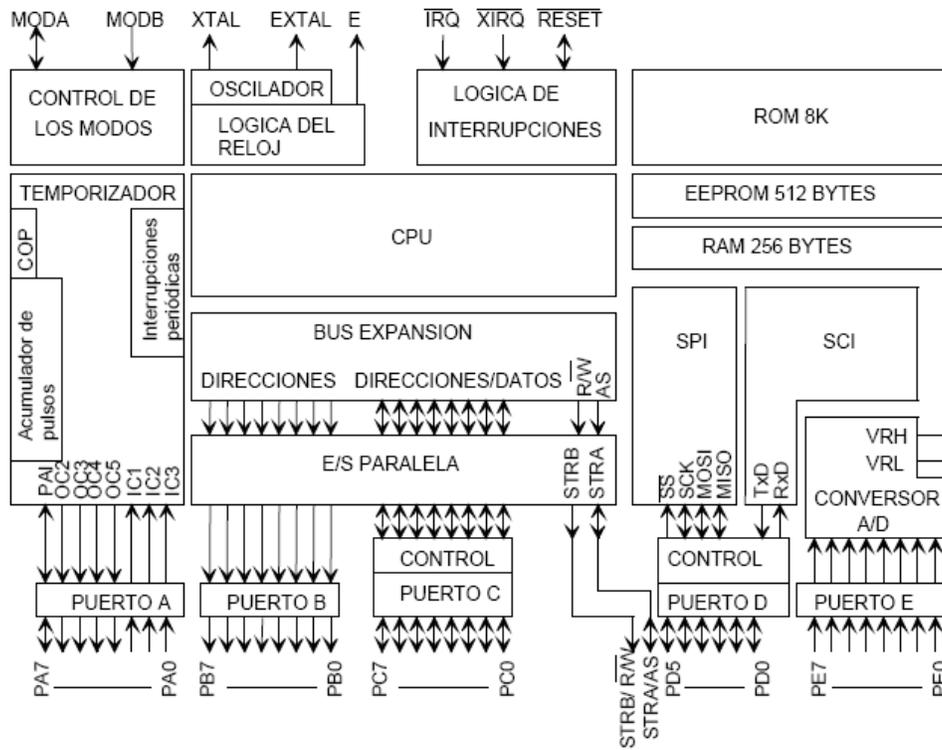
### **Convertidor A/D**

El convertidor analógico-digital de la familia 68HC11 funciona conforme el principio de la aproximación sucesiva, tiene una resolución de 8 bits y consta de un circuito de muestreo y retención (sample and hold). Gracias al multiplexador analógico en la entrada, es posible conectar hasta 8 señales de entrada. Las entradas analógicas no utilizadas pueden emplearse como entradas digitales.

El tiempo de conversión de las señales de entrada a una frecuencia de reloj de 8 MHz incluyendo el tiempo de muestreo y retención, es de 16  $\mu$ s (32 ciclos de máquina). A través del SFR ADCTL pueden configurarse diferentes modos de operación. Así, por ejemplo, puede seleccionarse entre la conversión única o continua, de uno o cuatro canales de entrada. Para el convertidor A/D se requiere una tensión de referencia externa.

**1.3.6.3 Sistema de interrupciones**

El microcontrolador dispone de 17 interrupciones de hardware y 1 interrupción de software. Con un bit del SFR CCR y otros bits de control adicionales pueden enmascarse 15 interrupciones individualmente. La instrucción de interrupción de software (SWI) no puede enmascarse.



**Figura 1.9 Aspectos de Hardware del 68HC11F1.**

## 1.4 Sensores y Transductores electrónicos

Los sensores son componentes críticos en todos los sistemas de medición y control. Debido a la disminución de su tamaño y precio, actualmente se encuentran en gran variedad de productos como juguetes, automóviles, lavadoras, entre otros.

Los sensores reales obedecen a infinidad de variantes, tecnologías e implementaciones de fabricantes por lo que resultaría imposible el estudio y conocimiento de todos dado la continua evolución del mercado y la aparición de nuevos modelos.

El sensor es un transductor eléctrico que provee una señal de salida eléctrica como respuesta a una medición específica. En términos generales, un transductor es un dispositivo que convierte de una energía a otra. Las mediciones pueden ser propiedades físicas, químicas o biológicas.

Los transductores pueden clasificarse en dos tipos básicos, dependiendo de la forma de la señal convertida:

- Transductores analógicos
- Transductores digitales

Los transductores analógicos proporcionan una señal analógica continua, por ejemplo voltaje o corriente eléctrica. Esta señal puede ser tomada como el valor de la variable física que se mide.

Los transductores digitales producen una señal de salida digital, en forma de un conjunto de bits de estado en paralelo o formando una serie de pulsaciones que pueden ser contadas. En una u otra forma, las señales digitales representan el

---

valor de la variable medida. Los transductores digitales, debido a su compatibilidad con las computadoras digitales, ofrecen ventajas sobre los sensores analógicos en la automatización y control de procesos.

### **1.4.1 Características deseables de los transductores (sensores)**

#### **Exactitud**

La exactitud de la medición debe ser tan alta como sea posible. Se entiende por exactitud que el valor verdadero de la variable se detecte sin errores sistemáticos positivos o negativos en la medición. Al tener varias mediciones de la variable, el promedio de error entre el valor real y el valor detectado debe tender a ser cero.

#### **Precisión**

La precisión de la medición debe ser tan alta como fuese posible. La precisión significa que puede existir una pequeña variación aleatoria en la medición de la variable. La dispersión en los valores de una serie de mediciones debe ser mínima.

#### **Rango de funcionamiento**

El sensor debe tener un amplio rango de funcionamiento y debe ser exacto y preciso en todo el rango.

#### **Velocidad de respuesta**

El transductor debe ser capaz de responder a los cambios de la variable detectada en un tiempo mínimo. Lo ideal es una respuesta instantánea.

#### **Calibración**

El sensor debe ser fácil de calibrar. El tiempo y los procedimientos necesarios para llevar a cabo el proceso de calibración deben ser mínimos. Además, el

sensor no debe necesitar una re calibración frecuente. El término desviación se aplica con frecuencia para indicar la pérdida gradual de exactitud del sensor que se produce con el tiempo y el uso, lo cual hace necesaria su re calibración.

### **Fiabilidad**

El sensor debe tener una alta fiabilidad, es decir, no debe estar sujeto a fallos frecuentes durante el funcionamiento.

### **1.4.2 Tipos de sensores**

Dentro de las mediciones físicas que realizan los sensores están: temperatura, fuerza, presión, desplazamiento, posición, velocidad, aceleración, sonido, viscosidad, flujo, y campo electromagnético.

#### **Sensores de temperatura**

Los sensores de temperatura se catalogan en dos series diferentes: TD y HEL/HRTS. Los sensores constan de una fina película de resistencia variable con la temperatura (RTD) y están calibrados por láser para una mayor precisión e intercambiabilidad. Las salidas lineales son estables y rápidas.

El termopar es la unión de dos metales diferentes en el que se genera una diferencia de potencial en la unión al incrementar la temperatura, debido a las diferentes distribuciones de energía en cada metal, por la excitación térmica de los electrones. En la figura 1.9 se muestra su principio de funcionamiento.

#### **Sensores magnéticos**

Los sensores magnéticos se basan en la tecnología magnetoresistiva SSEC y ofrecen una alta sensibilidad. Se usan en brújulas, control remoto de vehículos,

detección de vehículos, realidad virtual, sensores de posición, sistemas de seguridad e instrumentación médica.

### **Sensores de presión**

Los sensores de presión están basados en tecnología piezoresistiva, combinada con microcontroladores que proporcionan una alta precisión, independiente de la temperatura y capacidad de comunicación digital directa a una PC. Las aplicaciones en las que se usan estos productos incluyen instrumentos para aviación, laboratorios, controles de quemadores y calderas, comprobación de motores, tratamiento de aguas residuales y sistemas de frenado.

### **Sensores de posición**

Los sensores de posición de estado sólido son detectores de proximidad de metales y de corriente que se consiguen disponibles en varios tamaños y terminaciones. Estos sensores combinan fiabilidad, velocidad, durabilidad y compatibilidad con diversos circuitos electrónicos para aportar soluciones a las necesidades de aplicación.

### **Sensores de caudal de aire**

Los sensores de caudal de aire contienen una estructura de película fina aislada térmicamente, que contiene elementos sensibles de temperatura y calor. La estructura de puente suministra una respuesta rápida al caudal de aire u otro gas que pase sobre el chip.

### **Sensores de corriente**

Los sensores de corriente monitorizan corriente continua o alterna. Existen sensores de corriente lineales ajustables, de balance nulo, digitales y lineales. Los sensores de corriente digitales pueden hacer sonar una alarma, arrancar un motor, abrir una válvula o desconectar una bomba. La señal lineal duplica la forma de la onda de la corriente captada, y puede ser utilizada como un elemento de

---

---

respuesta para controlar un motor o regular la cantidad de trabajo que realiza una máquina.

### **Sensores de efecto Hall**

Son semiconductores y por su costo no están muy difundidos pero se emplean mucho como codificadores ("encoders") de servomecanismos.

### **Sensores de humedad**

Los sensores de humedad relativa / temperatura y humedad relativa están configurados con circuitos integrados que proporcionan una señal acondicionada. Estos sensores contienen un elemento sensible capacitivo en base de polímeros que interacciona con electrodos de platino. Están calibrados por láser y tienen una intercambiabilidad de +5% HR, con un rendimiento estable y baja desviación.

### **Detectores de ultrasonidos**

Los detectores de ultrasonidos resuelven los problemas de detección de objetos de prácticamente cualquier material. Trabajan en ambientes secos y polvorientos. Se usan para control de presencia/ausencia, distancia o rastreo.

### **Interruptores básicos**

Se consiguen interruptores de tamaño estándar, miniatura, subminiatura, herméticamente sellados y de alta temperatura. Los mecanismos de precisión se ofrecen con una amplia variedad de actuadores y características operativas. Estos interruptores son idóneos para aplicaciones que requieran tamaño reducido, poco peso, repetitividad y larga vida.

### **Interruptores final de carrera**

Es un microswitch que consta de un conmutador de 2 posiciones con retorno a la posición de reposo y con un botón o palanca de accionamiento que en ocasiones también puede traer una ruedita.

Cuando el interruptor está en reposo la patita común (COM) y la de contacto normal cerrado (NC), están en contacto permanente hasta que la presión aplicada a la palanca del microswitch hace saltar la pequeña platina acerada interior logrando que el contacto pase de la posición de normal cerrado a la de normal abierto (NO), se puede escuchar un pequeño clic que sucede casi al final del recorrido de la palanca.

### **Interruptores manuales**

Son los sensores más básicos, incluye pulsadores, llaves, selectores rotativos y conmutadores de enclavamiento. Estos productos tienen una amplia variedad de opciones en técnicas de actuación y disposición de componentes.

### **Productos encapsulados**

Tienen diseños robustos, de altas prestaciones y resistentes al entorno o herméticamente sellados. Esta selección incluye finales de carrera miniatura, interruptores básicos estándar y miniatura, interruptores de palanca y pulsadores luminosos.

### **Productos infrarrojos**

La optoelectrónica es la integración de los principios ópticos y la electrónica de semiconductores. Los componentes optoelectrónicos son sensores fiables y económicos. Entre ellos están los diodos emisores de infrarrojos (IREDs), sensores y montajes.

## **1.5 Filtros**

Dentro de un sistema de comunicaciones, en la etapa de transmisión, los filtros se emplean principalmente para *limitar en banda* a las señales a transmitir. En la etapa de recepción los filtros son empleados para *seleccionar* las señales deseadas y para *reducir* o *eliminar* los efectos de ruido e interferencias.

### 1.5.1 Filtros ideales

Un filtro de frecuencia selectiva es un circuito que permite el paso de señales con frecuencias comprendidas en ciertos rangos o bandas llamadas *pasabandas* y bloquea o atenúa señales cuyas frecuencias están en otros rangos, llamadas *bandas de rechazo* o de *supresión*. En cada caso, las *frecuencias de corte* son las frecuencias que definen los límites entre las frecuencias que pasan y las que se eliminan, es decir, las frecuencias en la *banda de paso* y en la *de supresión*.

La forma de  $|H(j\omega)| = M(\omega)$  se puede usar para clasificar diferentes tipos de filtros de acuerdo a la localización de su banda de paso o su banda de rechazo.

Un *filtro ideal* es un filtro que tiene una respuesta en fase lineal en su banda de paso, ninguna atenuación en dicha banda y una atenuación infinita ( $M(\omega) = 0$ ) en su banda de supresión.

### 1.5.2 Filtros digitales

La finalidad de un filtro es procesar una señal presente a su entrada, de forma que la señal de salida presente unas características frecuenciales cambiadas conforme a ciertas especificaciones. Este objetivo de todo filtro es independiente de su realización, sea ésta digital o analógica, y su comportamiento selectivo en frecuencias puede manifestarse en el módulo de la señal de salida, en la fase, o en ambos.

Los filtros analógicos están constituidos por elementos reactivos, fáciles de conseguir en el mercado y cuya impedancia varía con la frecuencia (condensadores e inductores); pudiendo haber, además, elementos resistivos u

otros componentes activos. Según las especificaciones del filtro, su realización puede ser más o menos compleja.

Otro de los atractivos de los filtros analógicos es su capacidad para manejar niveles de potencia importantes, aspecto difícil o, en muchos casos, imposible de conseguir solamente con filtros digitales. Y, finalmente, los filtros analógicos permiten trabajar con bandas frecuenciales muy altas, aspecto que los filtros digitales se ve limitado por la velocidad de procesamiento.

Un filtro digital es un algoritmo matemático, expresable como una ecuación en diferencias e implementado en hardware y/o software, cuyo objetivo es el mismo que el de los filtros analógicos: ofrecer un procesamiento selectivo en frecuencias de la señal de entrada. Al ser digital, ya se puede intuir algunas de sus ventajas, como por ejemplo la capacidad de memorizar o de ejecución de decisiones basadas en reglas lógicas, según valores observados en la señal de entrada o de salida.

Algunas ventajas de los filtros digitales son: filtros con auto-aprendizaje basado en la memorización de resultados previos, ajuste automático de las especificaciones del filtro, listados y estadísticas de resultados, etc.

Un filtro digital puede ser implementado por software utilizando una computadora o un microcontrolador, es decir, un procesador genérico no dedicado al tratamiento de señales. Este método de implementación posee como ventaja la versatilidad que ofrece el manejo habitual de los procesadores o microcontroladores, con los cuales, la mayoría de los diseñadores está familiarizado, tanto con los programas usuales de tratamiento de señales como con los lenguajes de bajo nivel.

Asimismo, permite una implementación de bajo costo por el reducido precio que poseen los microcontroladores o porque una computadora es un equipo de uso común que no necesariamente debe quedar dedicado a la realización del filtro sino

---

que puede realizar una gran cantidad de tareas. Es muy útil para la realización de prototipos y permite alcanzar un corto tiempo entre el diseño y la implementación.

La principal desventaja es la velocidad debido al procesamiento secuencial que sufren las señales, a diferencia de las implementaciones por hardware (PLDs o ASICs dedicados) o DSPs que realizan operaciones en paralelo.

La velocidad en la implementación por software queda íntimamente relacionada con la capacidad de procesamiento, memoria y también con la velocidad de conversión. Lo mismo ocurre con las limitaciones relativas al orden del filtro.

El tratamiento de la señal puede hacerse en tiempo real, o bien, desde archivo. Si se tratan señales de audio, la entrada que alimenta al filtro puede tomarse de un archivo con formato de audio, de la entrada de línea a la tarjeta de sonido de la PC, puede generarse con un algoritmo en Visual Basic u otro programa.

### **Filtros IIR y FIR**

Un filtro FIR con longitud M, entrada  $x(n)$  y salida  $y(n)$  se describe por la ecuación de diferencias:

$$y(n) = \sum_{k=0}^{M-1} b_k x(n-k) \quad (1)$$

donde k y b son los coeficientes del filtro [1].

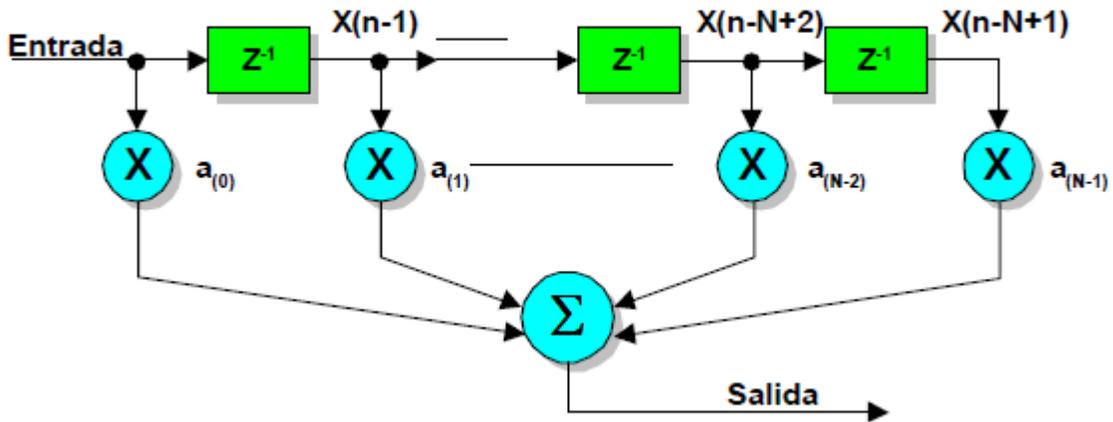


Figura 1.10 Estructura FIR

Los filtros IIR (Infinite Impulse Response) son sistemas causales, lineales, recursivos, con respuesta impulsiva de duración infinita y normalmente no ofrecen respuesta lineal en fase [2]. Los filtros IIR obedecen a la siguiente ecuación en diferencias:

$$\sum_{i=0}^M a_i y(n-i) = \sum_{j=0}^N b_j x(n-j) \quad (2)$$

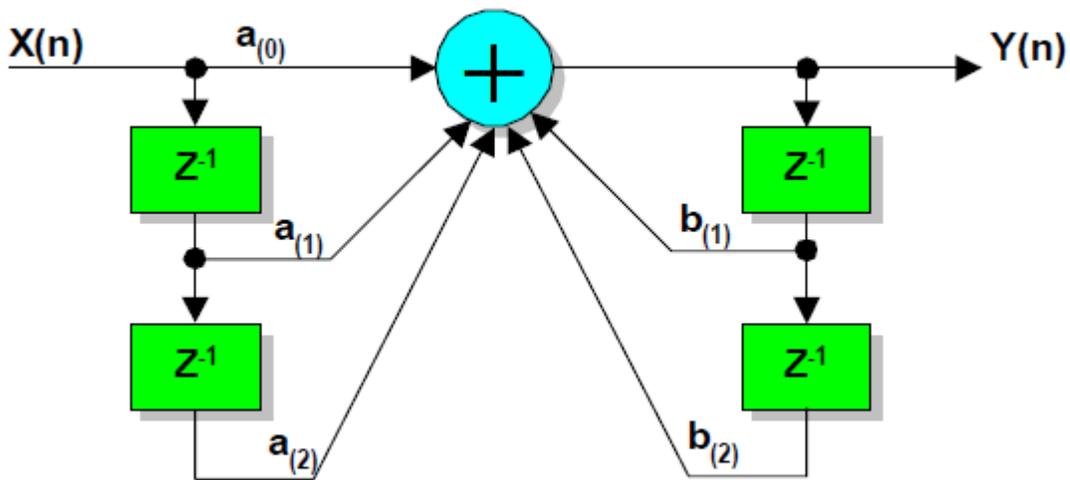


Fig. 1.11 Estructura IIR

En realidad cada tipo de filtro tiene sus ventajas y desventajas:

FIR:

- Son siempre estables
- Pueden tener fase lineal
- Métodos de diseño generalmente lineales
- Pueden ser montados con eficiencia en hardware
- Un Filtro FIR no tiene polos, solo ceros.

IIR:

- IIR es llamado infinito porque está formado por una función recursiva: su salida es una suma pesada de entradas y salidas.
- Debido a que éste es recursivo, su respuesta puede continuar indefinidamente.
- La respuesta de frecuencia de un filtro IIR tiene polos y ceros.

### **Ventajas de los filtros digitales:**

- Alta inmunidad al ruido.
- Alta precisión (limitada por los errores de redondeo en la aritmética empleada).
- Fácil modificación de las características del filtro.
- Muy bajo coste.

Por estas razones, los filtros digitales están reemplazando rápidamente a los filtros analógicos.

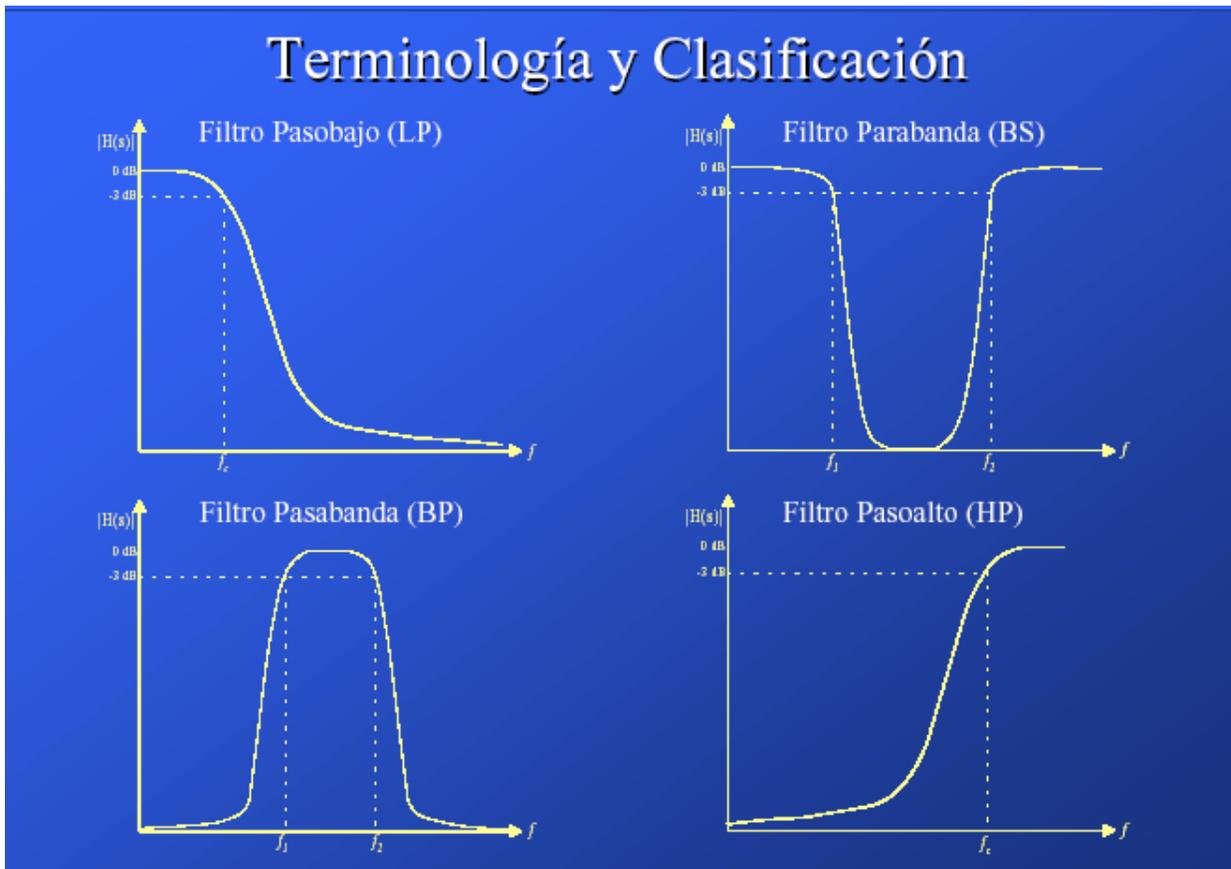


Figura 1.12, Clasificación de filtros

**El proceso de diseño de un filtro digital requiere tres pasos:**

- Establecer las especificaciones del filtro para unas determinadas prestaciones. Estas especificaciones son las mismas que las requeridas por un filtro analógico: frecuencias de operación, atenuaciones, ganancia dc, etc.
- Determinar la función de transferencia que cumpla las especificaciones.
- Realizar la función de transferencia en hardware o software.

## 1.6 Lenguajes de Programación

### **1.6.1 Evolución de los lenguajes de programación.**

En la década de los cuarenta cuando nacían las primeras computadoras digitales, el lenguaje que utilizaban para programar era un lenguaje máquina que traducía directamente el código máquina (código binario) comprensible para las computadoras. Las instrucciones en lenguaje máquina dependían de cada computadora y debido a la dificultad de su escritura, los investigadores de la época simplificaron el proceso de programación desarrollando sistemas de notación en los cuales las instrucciones se presentaban en formatos nemóticos (nemotécnicos) en vez de en formatos numéricos que eran más difíciles de recordar. Para convertir los programas escritos en código nemotécnico a lenguaje máquina se desarrollaron programas ensambladores para convertirlos en instrucciones numéricas en lenguaje máquina que son compatibles y legibles por la máquina.

En las décadas de los cincuenta y sesenta se desarrollaron lenguajes (de tercera generación) que diferían de las generaciones anteriores en que sus instrucciones eran de alto nivel, es decir sus instrucciones eran comprensibles por el programador como si fueran lenguajes naturales, e independientes de la máquina. Los ejemplos más conocidos de estos lenguajes son FORTRAN, COBOL, PASCAL, BASIC.

### **1.6.2 Fases en la resolución de problemas**

El proceso de resolución de un problema con una computadora conduce a la escritura de un programa ya a su ejecución. Aunque el proceso de diseñar programas es, esencialmente un proceso creativo, se puede considerar una serie de fases o pasos comunes:

- Análisis del problema

- Diseño del algoritmo
- Codificación
- Compilación y ejecución
- Verificación
- Depuración
- Mantenimiento
- Documentación

Las dos primeras fases se conducen a un diseño detallado escrito de algoritmo. Durante la tercera fase (codificación) se implementa el algoritmo en un código escrito en un lenguaje de programación, reflejando la idea desarrollada en las fases de análisis y diseño.

Las fases de compilación y ejecución traducen y ejecutan el programa. En las fases de Verificación y depuración el programador busca errores de las etapas anteriores y los elimina. Comprobar que mientras más tiempo se gaste en la fase de análisis y diseño, menos gastará en la etapa de depuración del programa.

Antes de conocer las tareas a realizar en cada fase, se considera el concepto y significado de la palabra algoritmo, el cual es un método para resolver un problema mediante una serie de pasos precisos, definidos y finitos.

### **1.6.3 Programación modular**

La programación modular es uno de los métodos de diseño más flexible y potente para mejorar la productividad de un programa. El programa modular se divide en módulos (partes independientes) cada uno de los cuales ejecuta una única actividad o tarea y se codifican independientemente de otros módulos. Cada uno de estos módulos se analiza, codifica y pone a punto por separado. Cada programa conviene un módulo denominado programa principal que controla todo

lo que sucede; se trasfiere el control a submodulos, de modo que ellos puedan realizar sus funciones; sin embargo, cada submodulo devuelve el control al modulo principal cuando se haya completado su tarea.

Dado que los módulos son independientes, diferentes programadores pueden trabajar simultáneamente en las diferentes partes del mismo programa. Esto reducirá el tiempo de diseño del algoritmo y posterior codificación del programa. Además, un modulo se puede modificar radicalmente sin afectar a otros módulos.

### **1.6.4 Programación Estructurada**

Cuando los programas se vuelven más grandes, cosa que lógicamente sucede cuando aumenta la complejidad del problema a resolver, la lista de instrucciones aumenta considerablemente, de modo tal que el programador tiene muchas dificultades para controlar ese gran número de instrucciones. Los programadores puede controlar de modo normal, un centenar de de líneas de instrucciones. Para resolver este problema los programas se descomponen en unidades, mas pequeñas que adoptaron el nombre de funciones (procedimientos, subprogramas o subrutinas). De este modo en un programa orientado a procedimientos se divide en funciones, de modo que cada función tiene un propósito bien definido y se resuelve una tarea concreta, y se diseña una interfaz claramente definida para su comunicación con otras funciones.

Con el paso de los años, la idea de romper el programa en funciones fue evolucionando y se llevo al agrupamiento de las funciones en otras unidades más grandes llamadas módulos; sin embargo el principio seguía siendo el mismo: agrupar componentes que ejecutan listas de instrucciones (sentencias). Esta característica hace que a medida que los programas se hacen más grandes y complejos y resultando muy difícil terminar los programas de modo eficiente.

### 1.6.5 Programación orientada a objetos

La programación orientada a objetos, tal vez el paradigma de programación más utilizado en el mundo de desarrollo de software y de la ingeniería de software del siglo XXI, trae un nuevo enfoque a los retos que se plantean en la programación estructurada cuando los problemas a resolver son complejos. Al contrario que la programación procedimental que enfatiza en los algoritmos la programación orientada a objetos enfatiza en los datos. En lugar de intentar ajustar un problema al enfoque procedimental de un lenguaje, la programación orientada a objetos intenta ajustar el lenguaje al problema. La idea de diseñar formatos de datos que se correspondan con las características esenciales de un problema.

La idea fundamental de los lenguajes orientados a objetos es combinar en una única unidad o módulo, tanto los datos como las funciones que operan sobre esos datos. Tal unidad se le llama objeto. Los datos de un objeto se le conoce también como atributos o variables de instancia. Si se desea leer datos de un objeto se llama a una función miembro del objeto. Se accede a los datos y se devuelve un valor. No se puede acceder a los datos directamente. Los datos están ocultos, de modo que están protegidos de alteraciones accidentales. Los datos y las funciones se dice que están encapsulados en una única entidad. El encapsulamiento de datos y la ocultación de los datos son términos clave en la descripción de lenguajes orientados a objetos.

Si se desea modificar los datos de un objeto, se conoce exactamente cuáles son las funciones que interactúan con los miembros del objeto. Ninguna otra función puede acceder a los datos. Esto simplifica la escritura, depuración y mantenimiento del programa.

## Capítulo 2. Detección de Tonos Multi-frecuencia

### 2.1 Señalización

La Señalización DTMF está basado en el envío de tonos, que llevan consigo la suma algebraica de dos señales senoidales. La primera de ellas compuesta por frecuencias bajas y la segunda por frecuencias altas, es un arreglo matricial de (4 X 4) que nos permite tener 16 combinaciones únicas. Estas son los números del 0 al 9, además de las teclas #, \*, A, B, C y D, las cuales nos sirven para señalizaciones especiales.

Con este tipo de señalización podemos construir un sistema capaz de decodificar dichos tonos y poder emplearlos con tal fin de que, cada tono o conjunto de ellos nos puedan servir para el control o mando a distancia de un entorno, en este caso de una casa habitación.

El decodificador puede ser diseñado por hardware y software, la decisión de cual usar depende mucho de las características del sistema a diseñar. Veamos algunas características de estos decodificadores.

El esquema de codificación DTMF asegura que cada señal contienen uno y solo un componente de cada uno de los grupos de tonos alto y bajo. Esto simplifica de manera significativa la decodificación por que la señal compuesta DTMF puede ser separada con filtros pasa banda en sus dos componentes de frecuencia simples cada uno de los cuales puede ser manipulado de forma individual.

La división de frecuencias en este aspecto juega un papel muy importante ya que los grupos alto y bajo simplifica el diseño de receptores DTMF como se muestra en la *figura 2.1*. Este diseño particular incluye una aproximación estándar. Cuando se encuentra conectado a una línea telefónica, receptor de radio o cualquier otra

fuentes de señal DTMF, el receptor filtra el ruido del tono, separa la señal en los componentes de grupos de alta y baja frecuencia para luego medir el cruce por cero promediando los periodos para producir la decodificación de un dígito.

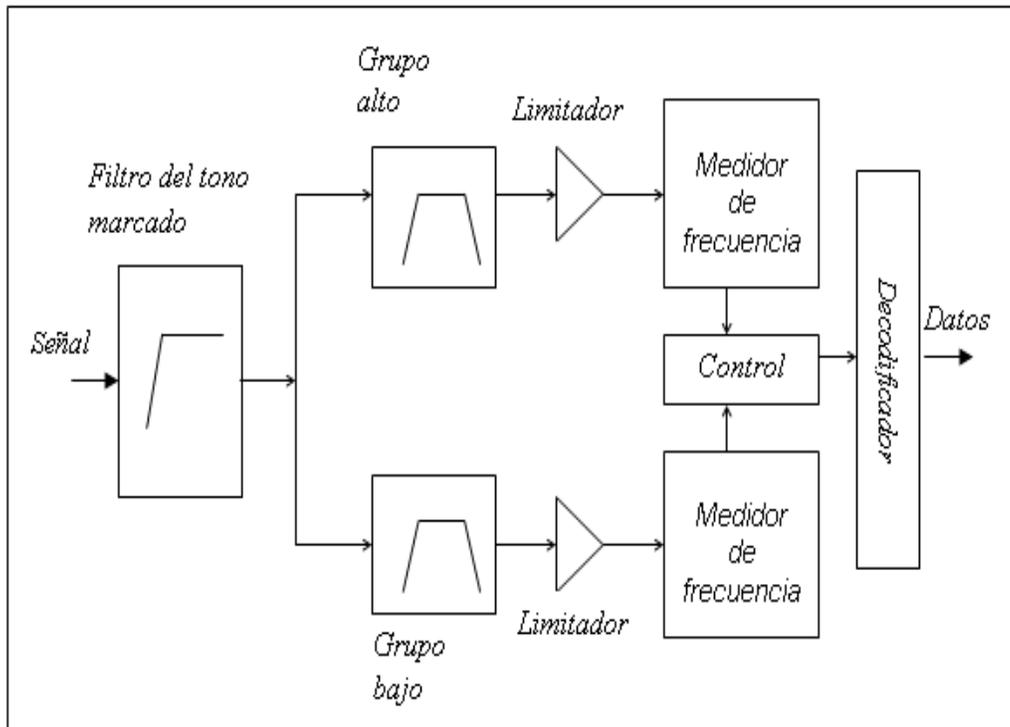


Figura 2.1 Arquitectura típica de un decodificador DTMF.

Para tener un resultado óptimo de cómo vamos a captar dichos tonos, depende de la selección de un buen circuito o componente capaz de detectarlas.

Para tal fin se señalan algunas de las características que debe poseer un buen decodificador de señales o tonos DTMF.

- Atenuación de la señal: El detector debe trabajar con una relación señal-ruido (SNR) de 15dB y en el peor caso con una atenuación de 26dB.

- Frecuencia: los símbolos con una desviación en frecuencia mayor al 3.5% deberán ser rechazados.
- Fase: El detector debe operar con un máximo de 8dB en fase normal y 4dB en fase invertida.
- Rechazo al habla: El detector debe operar en la presencia del habla rechazando la voz como un símbolo DTMF no válido

Como se muestra en la *Figura 2.2*, la detección DTMF se puede ver complicada por la presencia de ruido de línea de 50/60 Hz, tonos de varias frecuencias, ruido aleatorio y otras fuentes de interferencia. Tratar con estos problemas mientras permanece inmune a la simulación de tonos por voz presenta el más grande reto para los diseñadores de receptores DTMF.

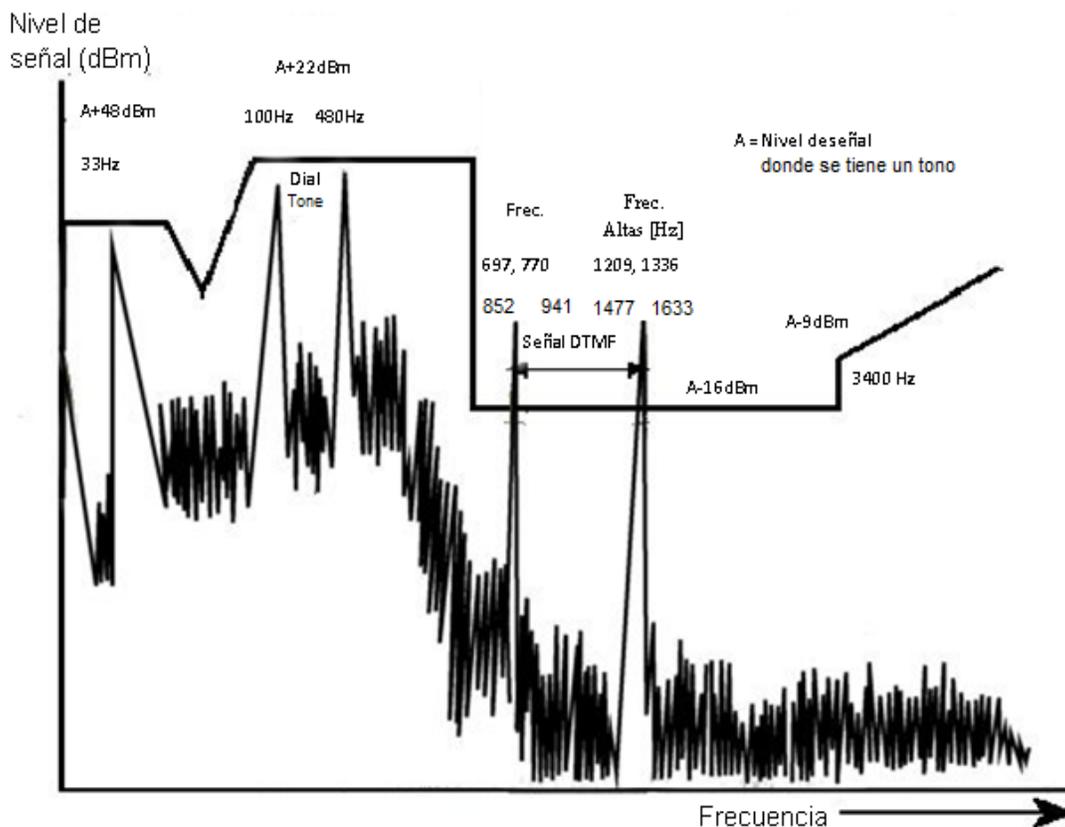


Figura 2.2 Entorno de la señalización DTMF.

La interferencia de línea tolerable mostrada en la figura es la recomendada por CEPT (Conferencia Europea de Administraciones de Correos y Telecomunicaciones) y es considerada la meta de diseño por los fabricantes de receptores DTMF de calidad.

Actualmente existe una gran variedad de circuitos integrados, tanto generadores, como detectores DTMF, así mismo ya empiezan a aparecer en el mercado circuitos microcontroladores que incluyen el detector y generador de DTMF como parte interna de los mismos y con capacidad de control del programa.

Es decir, el software viene a complementar las funciones de los circuitos integrados para la detección DTMF y así poder diseñar aplicaciones más completas basadas en DTMF.

En los accesorios telefónicos se utiliza frecuentemente la señalización DTMF, para programar alguna función, para seleccionar opciones al llamar a un banco, para activar o desactivar alguna característica y muchas otras aplicaciones.

Una observación importante es utilizar un aparato telefónico de tonos.

## **2.2 Hardware**

En el mercado comercial hay varios tipos de circuitos decodificadores, entre ellos el MT88L70 de la compañía MITEL, el NTE1690 de la compañía NTE Electronics, el MC14LC5436 de Motorola Semiconductor. Los primeros dos no ofrecen tecnología CMOS (Complementary Metal Oxide Semiconductor, "Semiconductor Complementario de Óxido Metálico") ni tampoco tolerancia al ruido que como vimos es un factor importante para una buena señalización.

En el Anexo 1 podemos observar el diagrama a bloques del decodificador, el receptor filtra el ruido del tono, separa la señal en los componentes de grupos de alta y baja frecuencia para luego determinar la frecuencia de cada grupo para producir al final la decodificación de un dígito.

### 2.3 Software

Es evidente que para la detección de tonos DTMF por software se requiere de ciertos algoritmos aplicados a filtros digitales. Uno de ellos es el algoritmo de Goertzel que proporciona un método eficaz para medir y comparar los niveles de frecuencia de un conjunto de muestras de datos de entrada.

Las frecuencias son elegidas de forma que ninguno de los armónicos ni la intermodulación producida por productos de ellas coincidan con alguna de estas frecuencias. La separación entre tonos es típicamente del 10%.

Los tonos están divididos en dos grupos y cada dígito queda determinado por uno de ellos.

El grupo bajo contiene las siguientes frecuencias:

697 Hz	770 Hz	852 Hz	941 Hz
--------	--------	--------	--------

El grupo alto contiene las siguientes frecuencias:

1209 Hz	1336 Hz	1477 Hz	1633 Hz
---------	---------	---------	---------

En la figura 2.4 se muestra la correspondencia entre las frecuencias y el teclado, se aprecia claramente que a cada número y/o carácter lo tocan dos frecuencias únicas.

	697	770	852	941
1209	1	2	3	A
1336	4	5	6	B
1477	7	8	9	C
1633	*	0	#	D

Figura 2.4 Arreglo matricial

El proceso de Decodificación DTMF se base en la identificación de la banda de frecuencias bajas y altas presentes en la señal de entrada y así identificar la tecla presionada. Por ejemplo, para representar el número 1 se deben emitir sonidos combinados de frecuencias de 1209 Hz y 967 Hz.

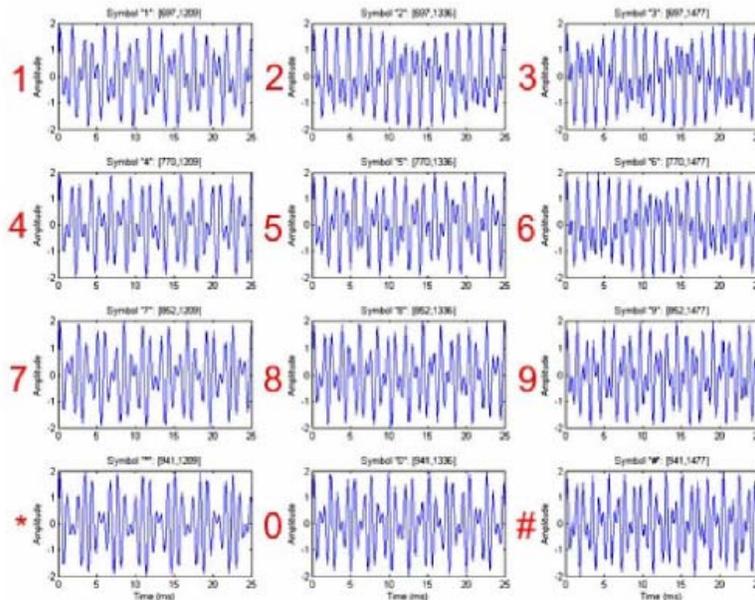


Figura 2.5 Señales DTMF representadas en amplitud y tiempo

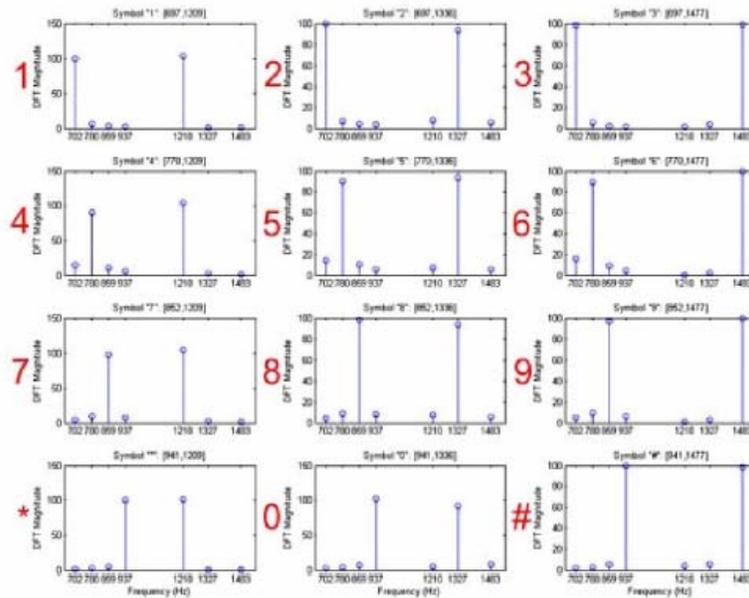


Figura 2.6 Señales DTMF representadas por Amplitud/Frecuencia

Para lograr la detección de cualquier tono DTMF se utiliza un filtro digital paso bandas, que nos deje solamente el espectro de frecuencias bajas y altas en donde pueden existir posibles tonos DTMF, tal como se muestra en la Figura 2.7.

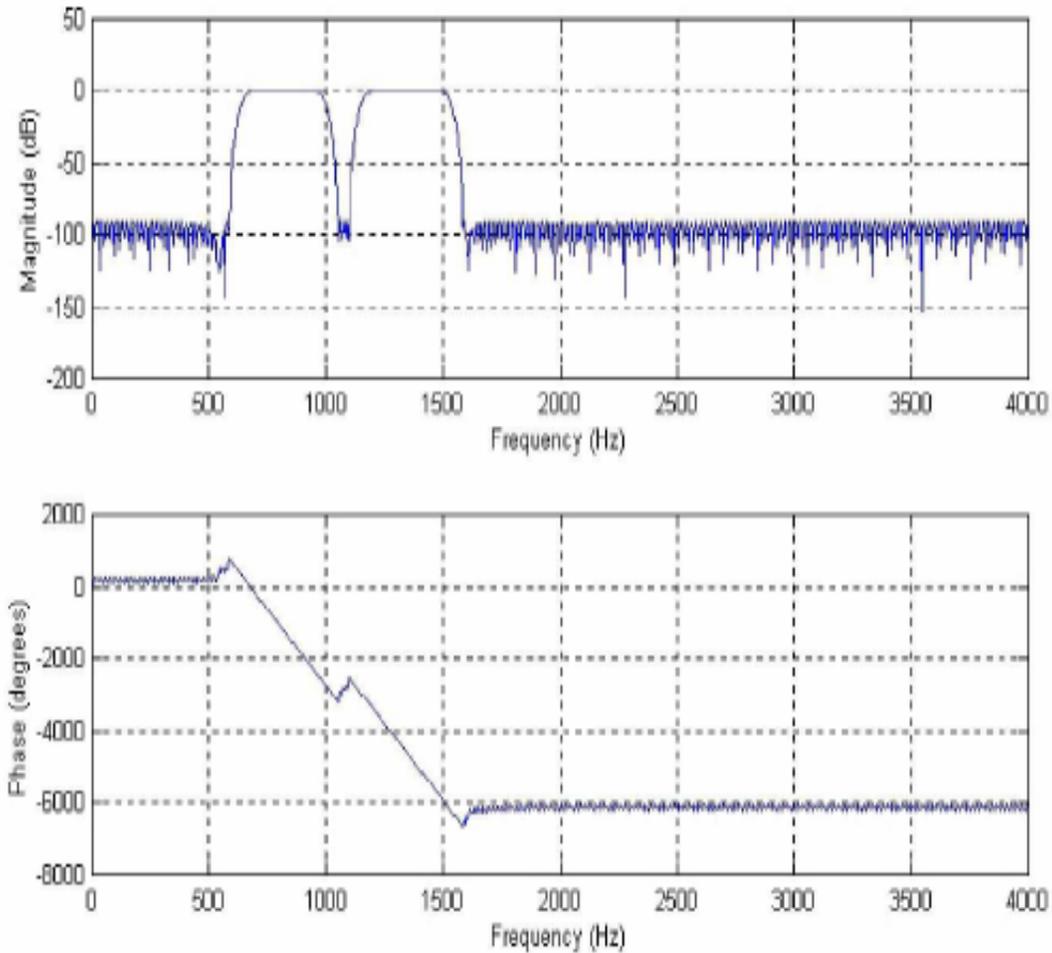


Figura 2.7 Filtro paso banda para dejar sólo frecuencias DTMF

La señal de salida es la convolución de la señal de entrada con el filtro digital. Se utiliza una variación de un Filtro IIR.

En el área de las señales, el filtrado es un proceso mediante el cual se modifica el contenido espectral de una señal.

### 2.4 Algoritmo de Goertzel

La Transformada Rápida de Fourier (FFT) se usa a menudo para detectar diferentes niveles de frecuencia en una señal o conjunto de valores de datos, pero requiere de una importante sobrecarga de procesamiento.

El algoritmo de Goertzel puede ser utilizado para decodificar señales DTMF y minimizar la sobrecarga de procesamiento.

Este algoritmo fue publicado en 1958 por el Dr. Gerald Goertzel y consiste en una técnica de procesamiento de señales digitales (DSP), que se utiliza para identificar frecuencias dentro de una señal.

Dos puntos importantes del algoritmo de Goertzel son: 1) que reduce la cantidad de operaciones en números reales en casi la mitad en relación con el cálculo directo de la DFT. Y 2) mientras la FFT analiza señales en todo el espectro de frecuencias, el algoritmo de Goertzel solo lo hace en puntos específicos y predeterminados. Es ideal para la decodificación DTMF, y más cuando solo tiene que encontrar la señal en ocho niveles de frecuencias específicas.

El algoritmo de Goertzel se obtiene como una adaptación de la ecuación de la DFT, equivalente a una convolución que puede ser implementada mediante un filtro digital.

La función de transferencias del algoritmo de Goertzel representa a un filtro IIR recursivo con un par de polos conjugados.

$$H_k(z) = \frac{1 - \theta^{j2\pi k / N} z^{-1}}{1 - 2 \cos \left( \frac{2\pi k}{N} \right) z^{-1} + z^{-2}}$$

donde:

$$k = N * \frac{f_{dtmf}}{f_{sample}}$$

$f_{dtmf}$  es la frecuencia correspondientes a cada tono,  $f_{sample}$  es la frecuencia de muestreo y  $N$  son el número de muestras.

De esta manera, y entendiendo que cada valor de  $k$  se encuentra relacionado con una banda del espectro de frecuencias de una señal  $x(n)$ , podemos obtener la energía únicamente en las bandas que nos interesan y ahorrar cálculos en bandas no requeridas.

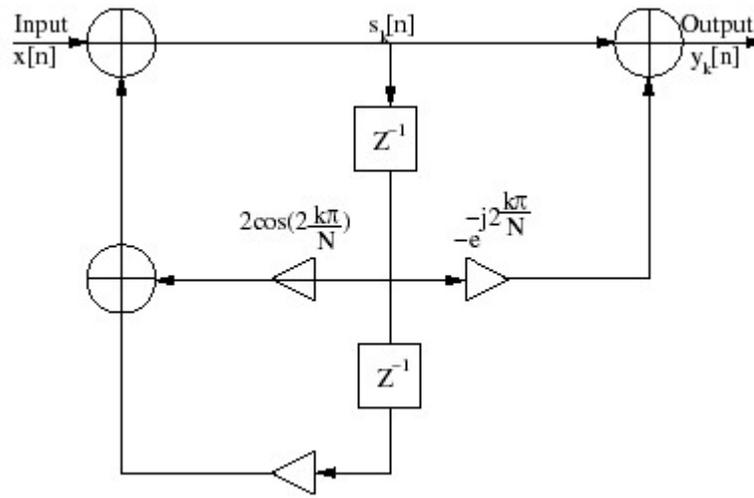


Figura 2.8 Diagrama del filtro

Este algoritmo cumple bastante bien las especificaciones requeridas y debido a la implementación del algoritmo de Goertzel en forma.

La programación del filtro queda fuera del alcance de este trabajo, en consecuencia se recurrió a utilizar una aplicación desarrollada por un tercero.

Para poder lograr que Visual Basic pueda detectar los tonos DTMF se encontraron dos métodos:

1. Utilizar un control diseñado por una empresa llamada TAPIEx.
2. Utilizar los módulos de clase de un proyecto ya diseñado.

TAPIEx es una empresa dedicada al desarrollo de software aplicado a telefonía. Su portafolio incluye aplicaciones para: CTI e IVR. Con características como: transferencia, conferencia, SR, TTS, detección e identificación de llamadas digitales, grabación y/o reproducción, detectar tonos DTMF, generar tonos DTMF.

TAPIEx rediseña la TAPI de Microsoft a simples objetos de telefonía que pueden ser fácilmente manipulados para crea aplicaciones tan robustas con un CTI fácilmente.

PhoneTone Decoder es un software que puede detectar todos DTMF de una señal analógica de audio. Su funcionamiento se basa en al Transformada Rápida de Fourier (FFT) utilizando el algoritmo de Goertzel, convirtiendo una muestra de audio analógico en un espectro de datos.

Está escrito en C++ usando ATL (Active Template Library), corre desde Windows 98 hasta Windows XP y trabaja en cualquier software de desarrollo que soporte controles ActiveX.

El control cuenta con Propiedades, Métodos y Eventos para facilitar su uso y esta orientado a objetos.

Su uso requiere de una Licencia y al momento de la descarga otorga un periodo de 15 días para su evaluación.

El Control PhoneTone Decoder necesita de una fuente audio y esta función la realiza el control WaveExCtrl, este se encargara de alimentar al Control PhoneTone Decoder.

En contra parte, se tiene la opción de usar código libre en diversos portales como:

- <http://www.canalvisualbasic.net>
- <http://vbasic.astalaweb.com>
- <http://www.lawebdelprogramador.com>
- <http://www.recursosvisualbasic.com.ar>
- <http://www.planet-source-code.com>

---

## Capítulo 3 Diseño de hardware y software para detección de tonos con el Microcontrolador

Se describió en el marco teórico el microcontrolador MC68HC11 de Motorola empleado en el presente trabajo, sin embargo, en este capítulo se describe la interacción del hardware que convive con el microcontrolador y el software que otorga las órdenes necesarias para obtener el control requerido para el sistema de señalización DTMF.

### 3.1 Comunicación Serie Asíncrona Entre el MCU (Micro Controller Unit) y la PC.

Existen diversas maneras para conectar dos dispositivos digitales, una de ellas es mediante las comunicaciones serie asíncronas. En dichas comunicaciones, los bits de datos se transmiten "en serie" (uno de tras de otro) y cada dispositivo tiene su propio reloj. Es importante tomar en cuenta que ambos dispositivos deben transmitir datos a la misma velocidad.

Los datos serie se encuentran encapsulados en tramas de la forma mostrada en la Figura 3.1.



Figura 3.1 Encapsulación de datos.

Primero se envía un bit de start, a continuación los bits de datos (primero el bit de mayor peso) y finalmente los bits de STOP.

Los parámetros que pueden ser configurables son: el número de bits de datos, el número de bits de Stop y el criterio de paridad (par o impar) para la detección de

errores. Normalmente, las comunicaciones serie tienen los siguientes parámetros: 1 bit de Start, 8 bits de Datos, 1 bit de Stop y sin paridad.

En esta Figura 3.2 se puede ver un ejemplo de la transmisión del dato binario 10011010. La línea en reposo está a nivel alto:

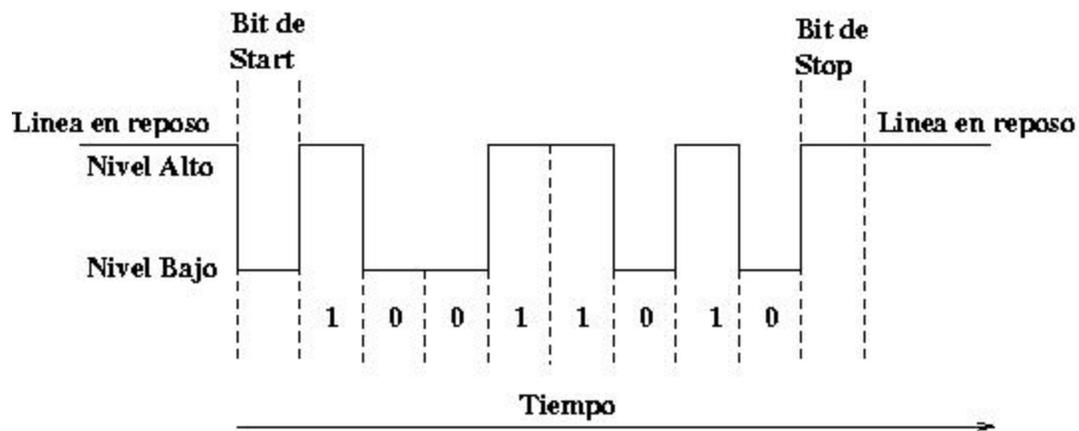


Figura 3.2 Transmisión de un dato binario.

### 3.2 Norma RS232

La Norma RS-232 fue definida para conectar una computadora a un Modem e indicar además de la manera en que se transmiten los datos (de forma serie asíncrona), las señales adicionales necesarias. Las tensiones empleadas están comprendidas entre +15 y -15 voltios.

Para conectar la PC a un microcontrolador por el puerto serie se utilizan las señales Tx (Transmisión), Rx (Recepción) y GND (Tierra). La PC utiliza la norma RS232 por lo que los niveles de tensión de los pines están comprendidos entre +15 y -15 voltios. Los microcontroladores normalmente trabajan con niveles TTL

(0-5v) por lo que es necesario intercalar un circuito que adapte los niveles, como el mostrado en la Figura 3.3.

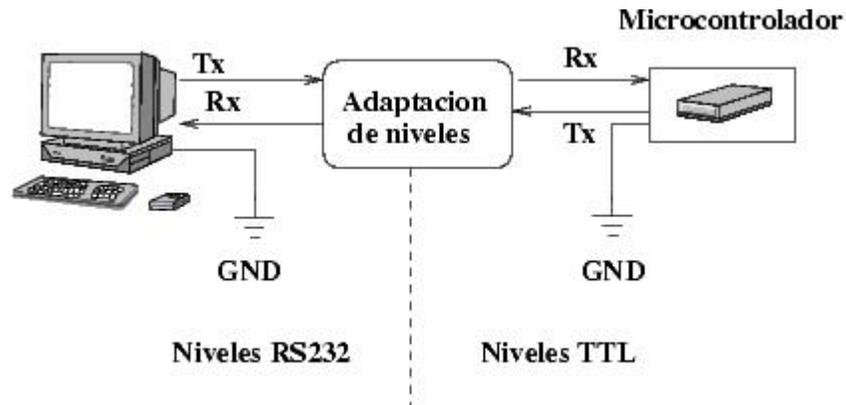


Figura 3.3 Conexión de un Microcontrolador y una PC

### 3.3 Conector DB9

Las PCs cuentan con conectores DB9 macho de 9 pines por los cuales se conectan los dispositivos al puerto serie. Los conectores hembra que se enchufan tienen una colocación de pines correspondiente, es decir, se conecta el pin 1 del macho con el pin 1 del hembra, el pin2 con el 2, etc. La Figura 3.4 describe lo anterior.

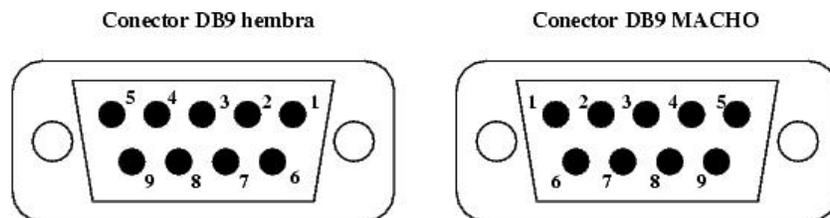


Figura 3.4 Conectores DB9.

La información asociada a cada uno de los pines es la siguiente:

1	<b>DCD</b> (Data Carrier Detect)
2	<b>RX</b> Recepción
3	<b>TX</b> Transmisión
4	<b>DTR</b> (Data Terminal Ready)
5	<b>GND</b>
6	<b>DSR</b> (Data Sheet Ready)
7	<b>RTS</b> (Request To Send)
8	<b>CTS</b> (Clear To Send)
9	<b>RI</b> (Ring Indicator)

Tabla 3.1 Descripción de señales por pin del conector DB9

### 3.4 El chip MAX32

Este chip adapta los niveles RS232 y TTL permitiendo conectar la PC con el microcontrolador. Para hacerlo, solo es necesario este chip y 4 condensadores electrolíticos de 22 micro-faradios. A continuación se muestra en la **figura 3.5** el esquema y los pines TX, RX y GND usados.

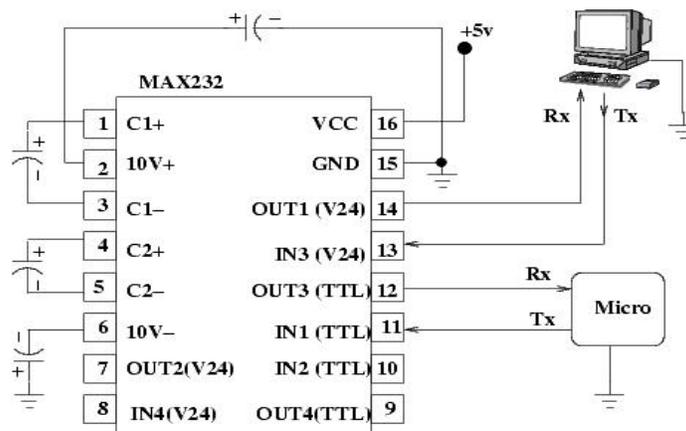


Figura 3.5 Chip MAX32 para conectar niveles TTL y RS232.

Para realizar la conexión entre la PC y nuestro circuito se pueden usar diferentes alternativas. Una manera es utilizar un cable serie macho-hembra no cruzado, y en el circuito un conector hembra DB9 para circuito impreso como se muestra en la Figura 3.6.



**Figura 3.6 Cables serie y entrada DB9 hembra.**

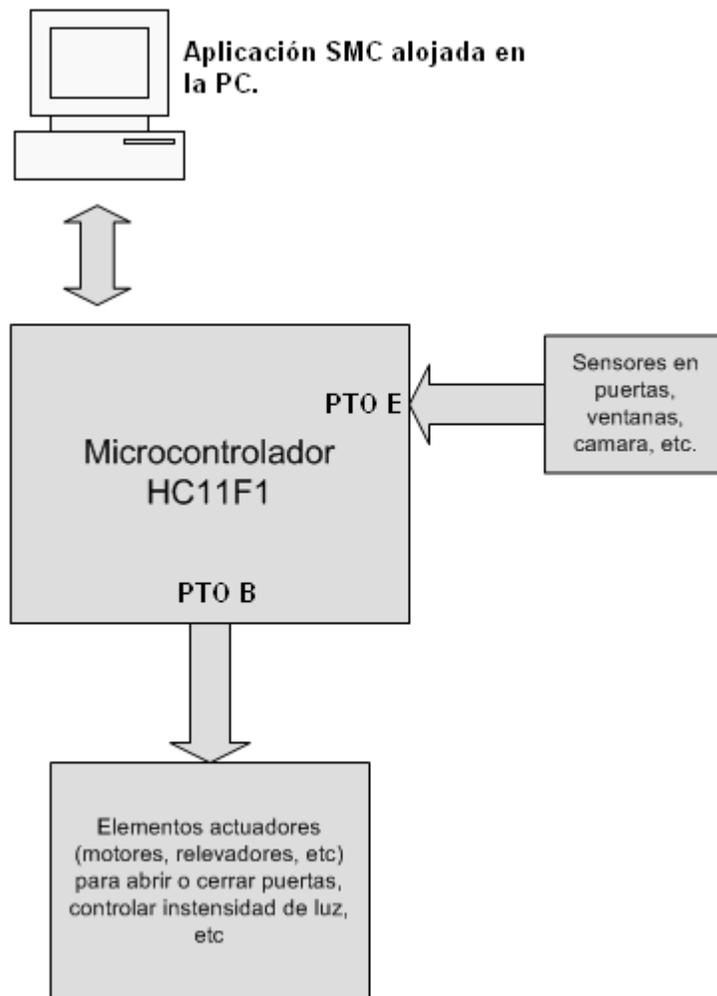
También se puede fabricar un cable serie utilizando cable plano de bus y conectando un conector DB9 hembra para bus como el mostrado en la Figura 3.7.



**Figura 3.7 Cable serie Bus.**

### 3.5 Interacción del microcontrolador y la PC

En la Figura 3.8 se muestra cómo el usuario escanea a través del puerto E los diferentes estados que guardan los sensores, después envía a través del puerto serial esta información a la aplicación SMC.



**Figura 3.8 Conexión microcontrolador - PC.**

Cuando el usuario desea cambiar el estado de un elemento (ventana, puerta, luz, etc.), el SMC recibe la indicación y este a través de la PC transfiere la indicación al  $\mu$ C, este la interpreta y manda la orden de ejecución a través de su puerto B. El sensor asignado monitorea el cambio de estatus y manda la información al  $\mu$ C para que este a la vez envíe la información a la aplicación SMC e informe al usuario del cambio solicitado.

### **3.6 Diseño de software de interfaz para el MCU**

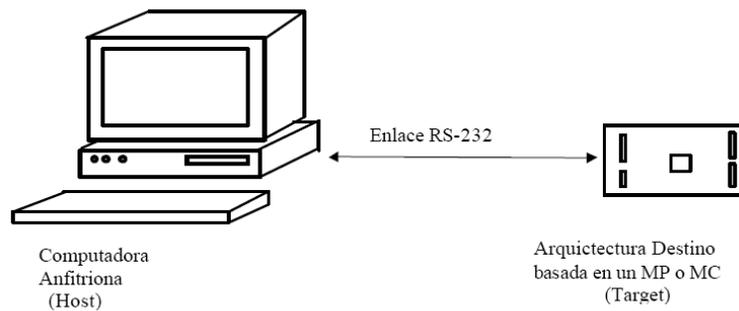
Se entiende por software de interfaz, aquel programa escrito en lenguaje ensamblador y cargado en el microcontrolador que permite que se interpreten los comandos DTMF provenientes del decodificador y su interfaz con la línea telefónica. Este software es el responsable de realizar también las tareas correspondientes a los comandos previamente establecidos.

A continuación se describe el software que permite leer un comando cualquiera y realizar una tarea determinada tomando en cuenta que puede variar dependiendo de la aplicación a implementar y que cuenta con la posibilidad de realizar control remotamente, es decir, a distancia.

#### **3.6.1 PUMMA 11**

Para el desarrollo del software necesario en este trabajo, se utilizó el software PUMA\_11 cortesía del M.I. Antonio Salva Calleja. Este software opera en modo visual desde una microcomputadora anfitriona y permite cargar, editar y ejecutar el programa en el microcontrolador.

PUMMA\_11 es un sistema que corre bajo ambiente Windows y que funciona como un sistema anfitrión de la arquitectura destino (AD) basada en alguna versión del HC11 (A1, E1, E2, E9, F1) como se muestra en la Figura 3.9. Al ejecutarse PUMMA\_11 en la computadora anfitriona se crea un ambiente que permite al usuario desarrollar y/o depurar programas en la AD, esto podría ser para fines de aprendizaje o diseño profesional de una aplicación.



**Figura 3.9 Sistema anfitrión – destino.**

Las principales facilidades actuales del PUMMA\_11 son:

- A) Capacidad para cargar, en la AD, archivos S19 que pudieran contener datos o programas en lenguaje de máquina.
- B) Capacidad para ejecutar un programa previamente cargado en la AD.
- C) Capacidad para escribir datos a memoria o puerto en la AD.
- D) Capacidad para examinar memoria o puertos en la AD.
- E) Capacidad para programar la EEPROM contenida en el microcontrolador de la AD.
- F) Capacidad para desensamblar código desde memoria del microcontrolador, o bien desde un archivo objeto S19.
- G) Capacidad para ejecutar paso a paso programas en el microcontrolador.
- H) Capacidad para colocar y capturar puntos de ruptura como técnica auxiliar para depuración y prueba de programas.
- I) Ensamblador cruzado integrado al sistema desarrollado especialmente para PUMMA\_11 y denominado ENS11, el cual cuenta con su propio editor y capacidad para ejecutar de inmediato el código generado a partir de un determinado programa fuente.

- J) Panel de observación del estado de cada uno de los ocho canales de conversión analógica a digital, con las que cuenta el microcontrolador.
- K) Capacidad para pasar de modo *boot-strap* a los modos *single-chip* y *expandidos*.

La Figura 3.10 muestra una ventana de un programa escrito en PUMMA\_11:

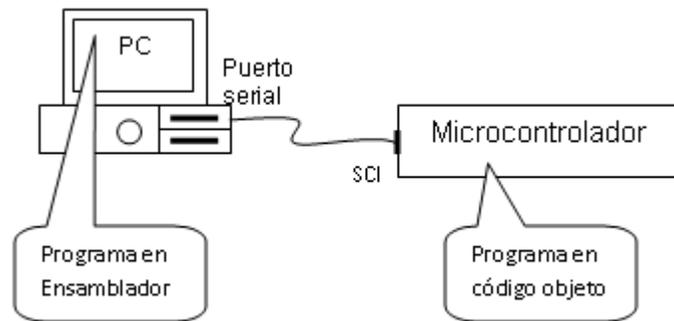
```

*****Constantes simbólicas*****
AP_BUF    equ  $0300    ;Apuntador del bufer
BUF_RAM   equ  $0302    ;Inicio del bufer
FIN_BUF   equ  $030F    ;Fin del bufer
CONTD     equ  $0320    ;Contador de digitos capturados
PORTE     equ  $100A    ;Dirección del puerto E
SCCR2     equ  $102D    ;DIRECCIÓN DEL REGISTRO DE CONTROL DEL SCI
SCCR1     equ  $102C    ;DIRECCIÓN DEL REGISTRO DE BAUDAJE DEL SCI
BAUD      equ  $102B    ;DIRECCIÓN DEL REGISTRO DE DATOS DEL SCI
SCIDR     equ  $102F    ;DIRECCIÓN DEL REGISTRO DE ESTADO DEL SCI
SCISR     equ  $102E    ;Dirección de la bandera de salida del programa
BANDERA   equ  $0322
NUMD      equ  #$04
*****Programa principal*****
        org  $0100
        ldx  #BUF_RAM
Limp:   clr  $00,x      ;Limpiamos el bufer
        inx  ;Limpiamos el bufer
        cpx  #FIN_BUF  ;Limpiamos el bufer
        bne  Limp     ;Limpiamos el bufer
        ldx  #BUF_RAM  ;Cargamos en x el inicio del BUFFER
        stx  AP_BUF    ;Transferimos el inicio del BUFFER a la dirección del
*       ;apuntador del BUFFER
        clr  BANDERA   ;Bandera de salida
        clr  CONTD    ;Pone a cero el contenido de la dirección en memoria
        ldaa #$33
        staa BAUD
        ldaa #500

```

Figura 3.10 Ventana típica de un programa escrito en PUMMA\_11

El software de la PC pasa el programa escrito en ensamblador a código de máquina y lo manda, a través del puerto SCI (Serial Communication Interface), al microcontrolador; el microcontrolador lo escribe en memoria y lo ejecuta. La Figura 3.11 muestra lo anteriormente descrito.



**Figura 3.11 Comunicación PC - Microcontrolador**

El programa en ensamblador que permite el control del sistema, se desarrollo en bloques, es decir se dividió en pequeños programas que posteriormente se integraron en uno solo. Estos programas se fueron desarrollando uno a la vez para dar oportunidad de resolver los inconvenientes que fueron surgiendo.

El proceso de dividir un problema en partes ayuda en el desarrollo tanto de software como de hardware puesto que permite identificar más fácilmente los problemas e inconvenientes que van surgiendo durante el proceso de su elaboración, si se hubiera intentado realizar este programa como una sola unidad, el proceso hubiera llevado más tiempo y su dificultad hubiera sido mayor porque al realizar pruebas no se sabe con exactitud en qué parte del código se tiene el error o cuantos son los errores que se tienen debido a la gran cantidad de código y a la cantidad de periféricos del microcontrolador que se están utilizando. Por ejemplo, si se está utilizando el puerto SCI, el puerto A y el G, es difícil determinar si existe error al leer el puerto A o al escribir en el puerto SCI o en ambas cosas.

A continuación se muestra la división del programa en la Figura 3.12 y la descripción de cada una de sus partes.

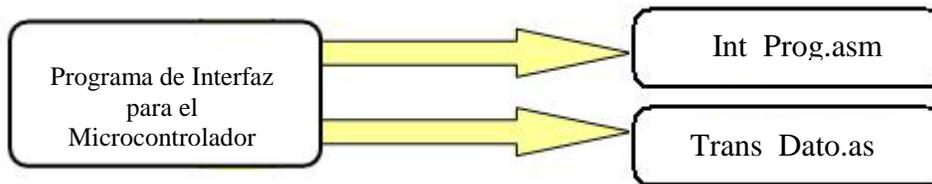


Figura 3.12 Diagrama de bloques de los programas.

### 3.7 Programas empleados en el microcontrolador

#### 3.7.1 Interrupción del programa (Int\_Prog.asm)

Como su nombre lo indica, este bloque de programa permite interrumpir cualquier proceso o tarea que esté realizando el microcontrolador. Para ello, cuando el microcontrolador esté en espera de un nuevo tono decodificado se transmite un carácter (definido previamente) de la PC al microcontrolador que está configurado para generar una interrupción al recibir dicho carácter y terminar inmediatamente el proceso que esté realizando.

Este bloque se incorpora debido a que la interfaz remota de voz (IRVDTMF) no puede depender de que el usuario lleve a cabo completamente las indicaciones que se le den y el microcontrolador debe interrumpir los programas que dependan de la respuesta del usuario, es decir, los programas que realizan poleo de la recepción de un nuevo tono. Un caso que ejemplifica esto, es el siguiente: cuando el sistema está poleando la recepción de un tono, puede ser que el usuario remoto ya no digite ningún número o puede ser que hasta cuelgue y dado que el microcontrolador no se da cuenta, continua verificando la recepción indefinidamente; para resolver esto, el software residente en la PC lleva el control del tiempo, en que el usuario remoto digita un nuevo número y si este tiempo es

demasiado, el programa se interrumpe, para después ejecutar otro programa que depende del número de interrupciones que se han realizado y del tiempo que ha transcurrido sin respuesta del usuario.

## **Descripción paso a paso del programa**

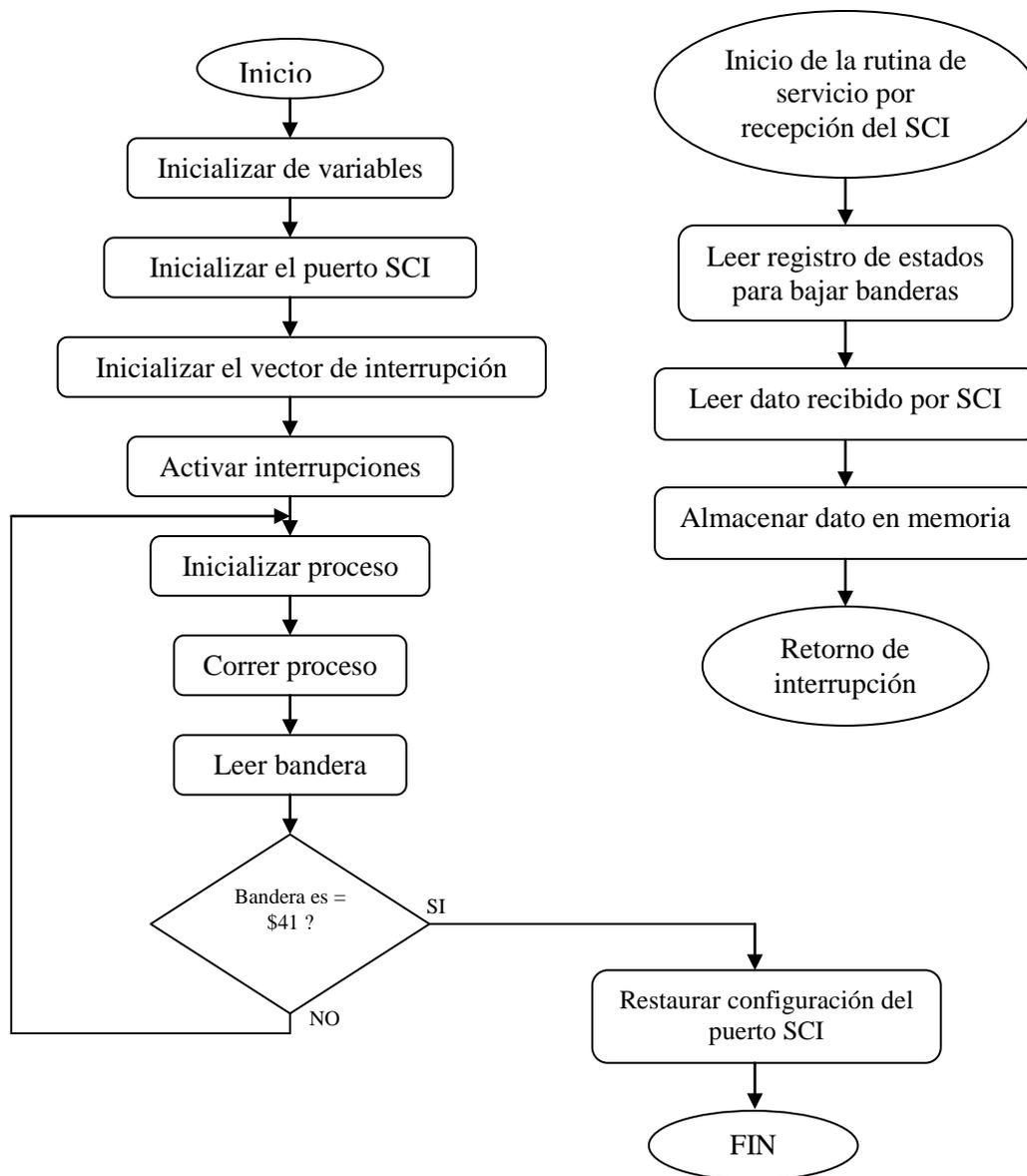
### **Programa principal**

1. Primero se limpia la dirección en memoria que sirve como bandera para determinar en qué momento se interrumpirá el programa.
2. Se configura el puerto SCI (serial Communication interface) para la recepción del carácter que indica cuando termina el programa.
3. Se configura el seudo-vector de interrupción dado que se debe generar una interrupción por recepción en el SCI.
4. Deben activarse las interrupciones poniendo a 1 el bit I del registro CCR.
5. Se inicializa proceso, es decir, se ponen en 0 los contadores, se limpian las variables, etc. Cabe aclarar que este proceso puede ser cualquiera, sin embargo, para la elaboración de este bloque se usó el siguiente proceso: poner a 0 todos los pines del puerto B e ir incrementando desde el 1 hasta el FF en hexadecimal. En el diseño final del software el proceso que será interrumpido es el de poleo de recepción de un nuevo tono.
6. Correr el proceso, es decir ejecutar las instrucciones para realizar una tarea dada.
7. Leer la bandera que informa si el proceso debe terminar.
8. Efectuar la comparación para determinar si el carácter leído es el que indica que el proceso debe terminar. Si el carácter leído es el definido para terminar el proceso, el flujo continúa al siguiente paso, en caso contrario, se repite el paso 5.
9. Se restablece la configuración del puerto SCI a la original para mantener la configuración adecuada de comunicación con la PC.

## Rutina de servicio de interrupción

1. Leer el registro de estado del puerto SCI para bajar bandera.
2. Leer dato recibido por el SCI.
3. Almacenar el dato en memoria.

A continuación se muestra el diagrama de flujo del proceso antes descrito:



## CAPÍTULO 3

---

El código del programa en lenguaje ensamblador es el siguiente:

```

baud equ $102b
ccrg equ $0300

****INICIO DE PROGRAMA*****

    org $0100
    clr $0301
    ldaa #$33 ; SE CONFIGURA EL PUERTO SERIE A 1200 BPS
    staa baud
    ldaa #$2c ; GENERACIÓN DE INTERUPCIÓN POR RECEPCIÓN
    staa sscr2
    ldaa #$7e ; SE CARGA EL PSEUDOVECTOR ASOCIADO POR SCI
    staa $c4
    ldx #serv_rx
    stx $c5
    cli
    clra
otro: staa $1004
    inca
    bsr ret
    ldab $0301
    cmpb #$41
    beq salida
    bra otro

salida: ldaa #$0c
    staa sscr2
    jmp $0000

****RUTINA DE RETARDO*****

ret:  pshx
    ldx #$ffff
vuelta: nop
    dex
    bne vuelta
    pulx
    rts
****RUTINA DE SERVICIO DEL SCI POR RECEPCION*****

serv_rx: ldaa $102e
    ldaa $102f
    staa $0301
    rti

```

### **3.7.2 Transmisión de Datos (Trans\_dato.asm)**

Este bloque de programa transmite datos hacia la computadora; los datos de los sensores. Para la comunicación se utiliza el puerto SCI, que ya fue configurado para la transmisión de los programas de la PC hacia el microcontrolador.

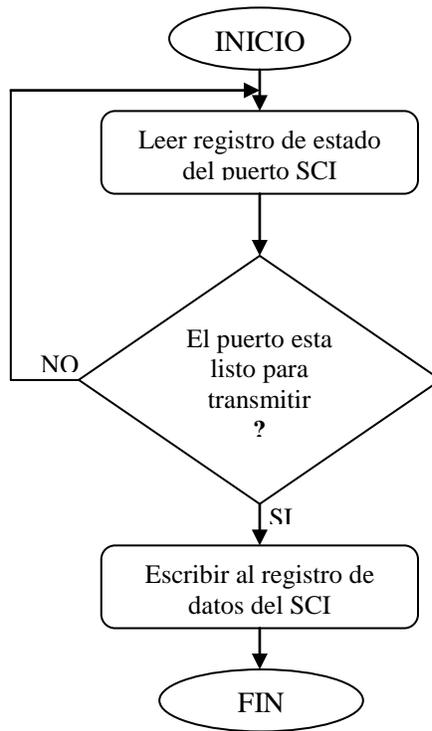
#### **Descripción paso a paso del programa**

1. Se lee el registro de estado del puerto de comunicación SCI.
2. Mediante la lectura realizada anteriormente, se determina si el puerto está listo para transmitir o aún está en proceso una transmisión anterior. Si el puerto ya está listo para la transmisión, se ejecuta el siguiente paso, en caso contrario, se ejecuta nuevamente el paso 1.
3. Se escribe el dato a transmitir en el registro de datos del puerto y el módulo SCI se encarga de transmitirlo.

Se describe a continuación el diagrama de flujo y su código en lenguaje ensamblador.

```

*****Rutina de transmisión*****
trans: psha
chec: ldaa $102E
      anda #$40
      beq chec
      stab $102F
      pula
      rts
*****
    
```

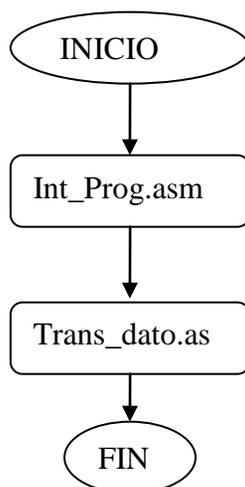


## Integración

El flujo del programa por bloques es el siguiente:

1. Primero se ejecuta el bloque de programa “Int\_Prog.asm”.
2. Una vez leídos los puertos con la información de los sensores, éstos serán transmitidos hacia la PC para que su software determine el comando que debe llevarse a cabo.

Diagrama de flujo del programa ya integrado en bloques.



## Capítulo 4. Diseño de Software.

### 4.1 Introducción

En este Capítulo se procede al desarrollo del software que nos permitirá interactuar con los dispositivos manejados por el microcontrolador.

El proyecto consta de tres partes a nivel software:

1. Portal Web.
2. Aplicación SMC (Sistema de Monitoreo y Control).
3. Aplicación Voz.

El objetivo del portal Web además de mostrar el estado actual de los sensores de la casa, es permitir modificar su estatus. El portal Web estará alojado de forma local en una PC dentro de la casa y tendrá una imagen en un servidor Web remoto, incluyendo el estatus de los sensores.

La aplicación SMC tendrá una comunicación bidireccional con el HC11 para leer el estado actual de los sensores y enviar información para modificarlos, esta aplicación se muestra en el capítulo 3.

La aplicación Voz se encargara de contestar llamadas entrantes, enviar mensaje de bienvenida y solicitar contraseña de acceso. Si la clave es correcta el sistema envía una serie de opciones a elegir.

La aplicación de Voz utilizara el modem para aceptar llamadas entrantes y a través de la tarjeta de sonido interactuara con el usuario.

El portal Web se desarrollo utilizando ASP, y la base de datos en Access.

La aplicación de voz se desarrollo con Visual Basic 6.0.

### 4.2 Diseño de portal Web

El portal Web permitirá interactuar en tiempo real con el HC11, revisando el estatus que tienen los dispositivos controlados, cambiando su estatus, ver fotografías, ver video en tiempo real, configurar el portal, crear permisos de acceso, etc.

El portal Web estará regido por una base de datos, el cual contendrá toda la información necesaria.

El portal Web se encargara de leer un archivo de texto proporcionado por la aplicación SMC, almacenara los datos en la Base de Datos y al mismo tiempo los mostrara al usuario, este proceso se estará refrescando cada cierto tiempo (cada 45 segundos por ejemplo), el cual podrá ser programado para el intervalo que se desee.

El objetivo final del Portal Web es presentar la información adquirida por el HC11, de vincular al usuario con el estatus que tiene su casa de un modo agradable y eficiente.

La estructura del Portal Web se muestra a continuación:

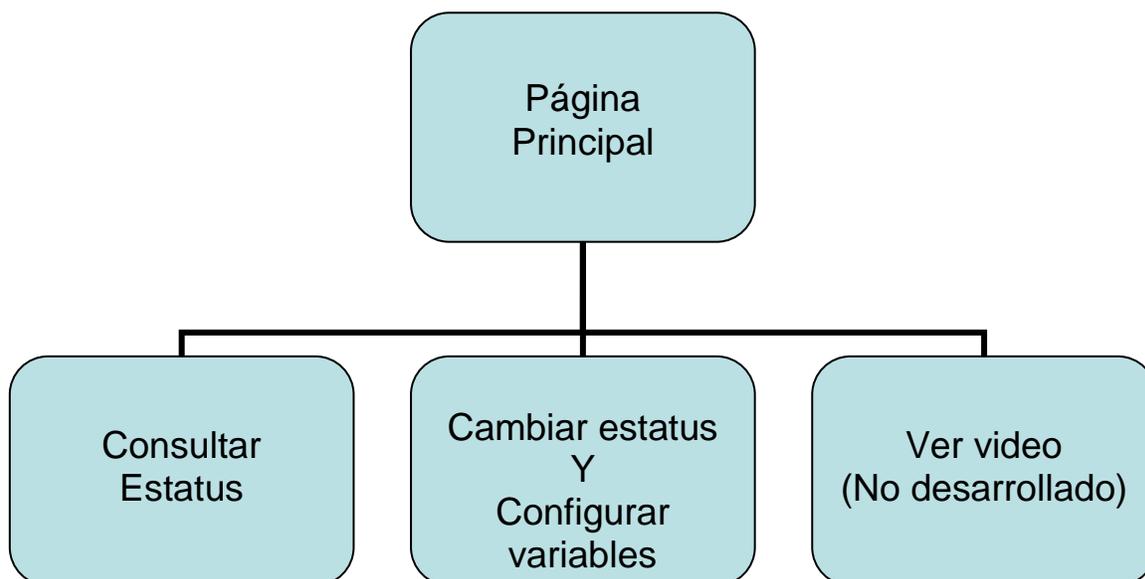


Figura 4.1 Estructura Portal Web

### **Página principal**

No muestra las opciones de consultar estatus, cambiar estatus, ver video y configurar.

### **Consultar estatus**

Nos muestra el estatus actual de los dispositivos de vigilancia, tal como sensores infrarrojos en ventanas, puertas y/o sitios predeterminados por el usuario. También

nos indica si algún dispositivo se alertó y se está refrescando cada cierto tiempo, por ejemplo cada 30 seg.

### **Cambiar estatus y configurar variables**

Nos permite cambiar el estatus de los dispositivos de control, por ejemplo prender un foco, abrir una ventana, cerrar una puerta, etc. También, se podrá configurar el tiempo de refresco y de toma de fotos.

Para poder entrar a esta parte necesita tener permisos de administrador.

### **Ver video**

El sistema puede ser escalable hasta el término de observar un video de una cámara colocada estratégicamente, para poder tener un panorama general. De igual forma, el sistema guardará fotos cada cierto tiempo y las almacenará en el servidor Web.

La configuración ideal del sistema debería permitir la transmisión de video en tiempo real. Para lograr esto se requieren enlaces de alta velocidad, pues las secuencias de video utilizan volúmenes de información muy elevados. Actualmente existen sistemas de compresión de video, pero con estos se sacrifica en gran medida la calidad de video; incluso siendo de tamaño muy reducido (320 X 240 píxeles de resolución).

Una opción para poder realizar esta parte del proyecto, es el empleo de la tecnología que utilizan las llamadas webcam. Existen multitud de programas que gestionan el funcionamiento de estas webcam, transmitiendo las imágenes a páginas web.

Otra opción para este apartado son las cámaras IP con servidor web integrado para servir video que se puede visualizar desde cualquier parte del mundo. Se pueden conectar a una red de forma alámbrica o inalámbrica y colocarse casi en cualquier lugar permitiendo la vigilancia a distancia de la casa u oficina.

Estas cámaras tienen su propia dirección IP para actuar como dispositivo independiente.

Existen otras opciones, como un servidor de video, para esta parte del proyecto. Sin embargo, es necesario hacer un análisis amplio para elegir la opción correcta, razón por la cual queda fuera del alcance del proyecto.

Cualquier opción que se tome debe permitir la integración del video dentro del portal web, de esta forma enana sola dirección se tendrán los elementos necesarios para un mejor control.

### 4.2.1 Código Portal Web

A continuación se muestra el código completo de las páginas que integran el Portal Web, en su mayoría escrito en ASP (Active Server Pages).

#### Consola.asp

CODIGO DE BIENVENIDA AL PORTAL Y SELECCIÓN A LA OPCIÓN DESEADA

```

<html>
<head>
<title>Sistema de Control</title>
</head>
<body>
<center>
<h1>Bienvenido al Sistema de Monitoreo y Control de su Casa</h1>
<p></p>
<h1></h1>
<br />
<br />
<br />
<br />

<%

Dim oConn
Dim oRS
Dim sSQL
Dim sColor
Dim Usuario1, Passww, f
Dim n

Usuario1=Request.Form("Usuario")
Passww=Request.Form("Clave")

f = 0

Set oConn = Server.CreateObject("ADODB.Connection")
oConn.Open("Provider=Microsoft.Jet.OLEDB.4.0; Data Source=" & Server.MapPath("\db\bd1.mdb"))

sSQL = "SELECT * FROM Tabla_Claves"
Set oRS = oConn.Execute(sSQL)

Do While NOT oRS.EOF

        set objuser = oRS("Usuario")
        set objpas = oRS("Clave")
        set objname = oRS("Nombre")
        If objuser = Usuario1 AND objpas=Passww Then
            Response.Write "Hola "& objname & ", bienvenido a su consola de control."
            Response.Write("")
            f = 1
        End If
        oRS.MoveNext

Loop

oConn.Close
Set oRS = Nothing
Set oConn = Nothing

```

## CAPÍTULO 4

```

if f = 0 then
    Session("Var1")=True
    Response.Redirect("login.asp")

End if
%>
<br>
<%
Response.Write("De clic en la opción deseada")
%>
<p>
<table>
<tr>
<td>
<FORM METHOD="POST" ACTION="mostrar.asp">
<td align="center"><INPUT TYPE="SUBMIT" VALUE="Consultar Estatus"></td>
</table>
</FORM>
</td>
<td>
<FORM METHOD="POST" ACTION="modificar.asp">
<td align="center"><INPUT TYPE="SUBMIT" VALUE="Cambiar Estatus"></td>
</table>
</FORM>
</td>
<td>
<FORM METHOD="POST" ACTION="mostrar.asp">
<td align="center"><INPUT TYPE="SUBMIT" VALUE="Ver Video"></td>
</table>
</FORM>
</td>
<td>
<FORM METHOD="POST" ACTION="mostrar.asp">
<td align="center"><INPUT TYPE="SUBMIT" VALUE="Configurar Portal"></td>
</table>
</FORM>
</td>

</body>
</html>

```

Estatus.asp

CODIGO QUE LEE EL ESTADO INICIAL DE LOS ELEMENTOS EN LAS HABITACIONES

```

<html>
<head>
<title>Sistema de Control</title>
</head>
<body>
<center>
<h1>Bienvenido al Sistema de Monitoreo y Control de su Casa</h1>
<p></p>
<h1></h1>
<br />
<br />
<br />
<br />
<%

Dim oConn
Dim oRS
Dim sSQL
Dim sColor

```

## CAPÍTULO 4

Dim Usuario1, Passww, f  
Dim n

Usuario1=Request.Form("Usuario")  
Passww=Request.Form("Clave")

f = 0

Set oConn = Server.CreateObject("ADODB.Connection")  
oConn.Open("Provider=Microsoft.Jet.OLEDB.4.0; Data Source=" & Server.MapPath("db\bd1.mdb"))

sSQL = "SELECT \* FROM Tabla\_Claves"  
Set oRS = oConn.Execute(sSQL)

Do While NOT oRS.EOF

```

set objuser = oRS("Usuario")
set objpas = oRS("Clave")
set objname = oRS("Nombre")
If objuser = Usuario1 AND objpas=Passww Then
    Response.Write "Hola "& objname & ", bienvenido a su consola de control."
    Response.Write("")
    f = 1
End If
oRS.MoveNext

```

Loop

oConn.Close  
Set oRS = Nothing  
Set oConn = Nothing

if f = 0 then

```

Session("Var1")=True
Response.Redirect("login.asp")

```

End if

```

%>
<br>
<%
Response.Write("De clic en la opción deseada")
%>
<p>
<table>
<tr>
<td>
<FORM METHOD="POST" ACTION="mostrar.asp">
<td align="center"><INPUT TYPE="SUBMIT" VALUE="Consultar Estatus"></td>
</table>
</FORM>
</td>
<td>
<FORM METHOD="POST" ACTION="modificar.asp">
<td align="center"><INPUT TYPE="SUBMIT" VALUE="Cambiar Estatus"></td>
</table>
</FORM>
</td>
<td>
<FORM METHOD="POST" ACTION="mostrar.asp">
<td align="center"><INPUT TYPE="SUBMIT" VALUE="Ver Video"></td>
</table>
</FORM>
</td>
<td>
<FORM METHOD="POST" ACTION="mostrar.asp">
<td align="center"><INPUT TYPE="SUBMIT" VALUE="Configurar Portal"></td>
</table>
</FORM>
</td>

```

```

</body>
</html>

```

## Modificar.asp

CODIGO QUE ACTUALIZA EL ESTADO DE UN ELEMENTO (PUERTA, VENTANA O LUZ)

```

<html>
<head>
<title>Sistema de Control</title>
</head>
<body>
<center>
<h1>Bienvenido al Sistema de Monitoreo y Control de su Casa</h1>

<p></p>
<h1></h1>
<br />
<br />
<br />
<br />

<%
Dim oConn
Dim oRS, arddarray
Dim sSQL, ji

'Ide=Request.Form("Cont")
Ide=""
Element=""
Estat=""
Alert=""

Ide = Session("Ide1")
Element = Session("Element1")
Estat = Session("Estat1")
Alert = Session("Alert1")

Set oConn = Server.CreateObject("ADODB.Connection")
oConn.Open("Provider=Microsoft.Jet.OLEDB.4.0; Data Source=" & Server.MapPath("\db\bd1.mdb"))
set oRS=Server.CreateObject("ADODB.Recordset")
sSQL = "SELECT * FROM Tabla_Estatus"
Set oRS = oConn.Execute(sSQL)
arddarray = oRS.getrows

%>

<table border="2">
<tr>
<td width="32">
Id
</td>
<td width="100">
Elemento
</td>
<td width="60">
Estatus
</td>
<td width="60">
Alertado
</td>
</tr>
</table>
<table border="2">

```

## CAPÍTULO 4

```

<%for l = 0 to UBound(ardarray,2)%>
    <tr>
        <td width="32">
            <%=ardarray(4, l)%>
        </td>
        <td width="100">
            <%=ardarray(1, l)%>
        </td>
        <td width="60">
            <%=ardarray(2, l)%>
        </td>
        <td width="60">
            <%=ardarray(3, l)%>
        </td>
    </tr>
<%Next%>
</table>

<p>
Registro a modificar:
<FORM METHOD="POST" ACTION="hola.asp">
    <INPUT TYPE="text" NAME="Val_mod" size="1" maxlength="1">
    <INPUT TYPE="SUBMIT" VALUE="Enter">
</FORM>

<FORM METHOD="POST" ACTION="holax.asp">
<table border="2">
    <tr>
        <td width="32">
            <input type="text" name="Id" size="1" maxlength="1" value="<%=Ide%>">
        </td>
        <td width="100">
            <input type="text" name="Elemento" size="12" maxlength="12"
value="<%=Element%>">
        </td>
        <td width="60">
            <SELECT NAME="Estatus">
                <OPTION><%=Estat%>
                <OPTION>no
                <OPTION>si
            </SELECT>
        </td>
        <td width="60">
            <SELECT NAME="Alertado">
                <OPTION><%=Alert%>
                <OPTION>no
                <OPTION>si
            </SELECT>
        </td>
    </tr>
</table>
<INPUT TYPE="SUBMIT" VALUE="Modificar">
</FORM>

<%
oConn.Close
Set oRS = Nothing
Set oConn = Nothing
%>

&#169; Jorge Medina

</body>
</html>

```

### 4.2.2 Autenticación de usuarios

Un requerimiento importante en un sistema informático es su mecanismo de seguridad, el tema de seguridad informática requiere de un estudio aparte; por lo tanto solo nos limitaremos a usar el modelo de autenticidad más básico que consiste en decir si un usuario es quien dice ser simplemente basándonos en una prueba de conocimiento que “a priori” sólo ese usuario debe saber, esa prueba de conocimiento no es más que una contraseña.

Este modelo es evidentemente el mas vulnerable pero también el mas fácil de implementar.

Al ingresar al portal, este mostrara una página de inicio la cual solicitará nombre de usuario y contraseña para poder dar acceso (fig. 4.2).



Figura 4.2 Pantalla de autenticación

Los datos ingresados por el usuario serán validados en una base de datos, una vez aceptado muestra el menú principal, el cual se muestra en la Figura 4.3.

En caso de que el usuario o contraseña sean incorrectos la página se refrescara y enviara el mensaje de “Usuario y/o contraseña no validos”. Borra las cajas de texto y espera a que sean introducidos nuevamente los datos.

El diagrama de flujo de lo anteriormente mencionado se muestra en la figura 4.4.

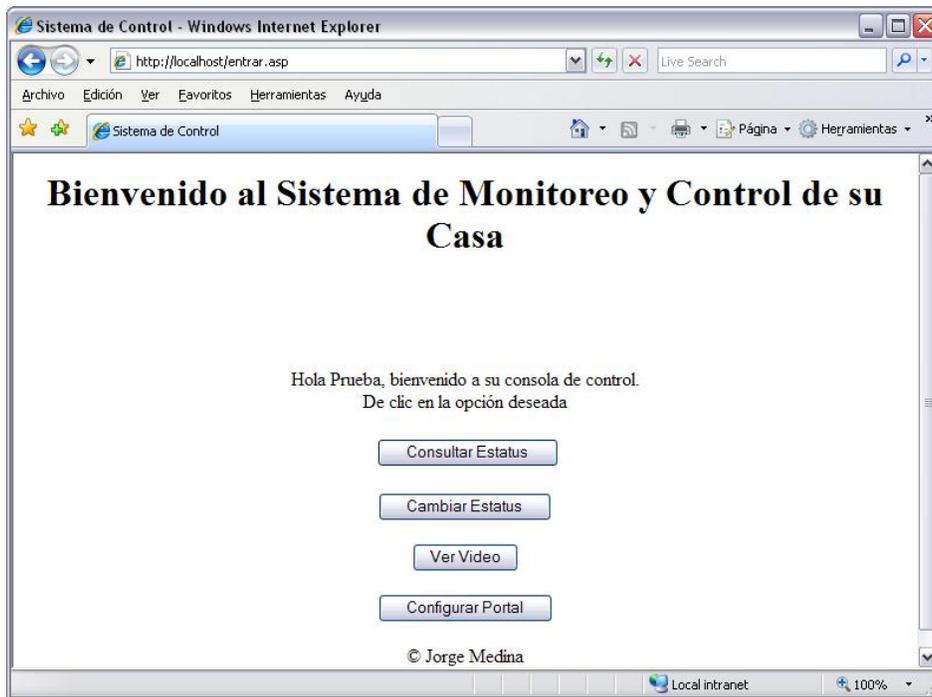


Figura 4.3 Pantalla de inicio

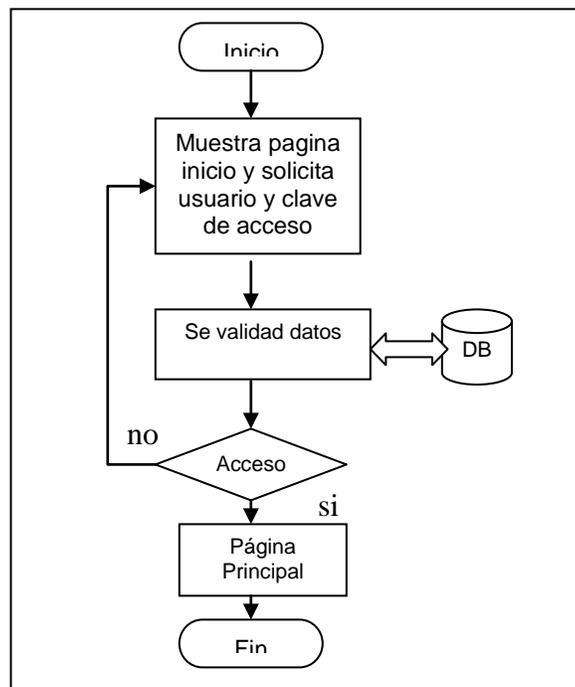


Figura 4.4 Diagrama de flujo de autenticación

A continuación se muestra el código final:

### Login.asp

*PROGRAMA QUE VALIDA EL ACCESO AL PORTAL WEB*

```
<html>
<head>
<title>Sistema de Control</title>
</head>
<body>
<center>
<h1>Identificación de Usuario</h1>
<p></p>
<h1>Teclee sus datos para acceder:</h1>
<br />
<br />
<br />
<br />

<%
Session.Abandon
Var2 = False
Var2 = Session("Var1")
If Var2 = True Then
    Response.Write("Usuario y/o contraseña incorrectos, favor de verificar")
    Session.Abandon
End If
%>

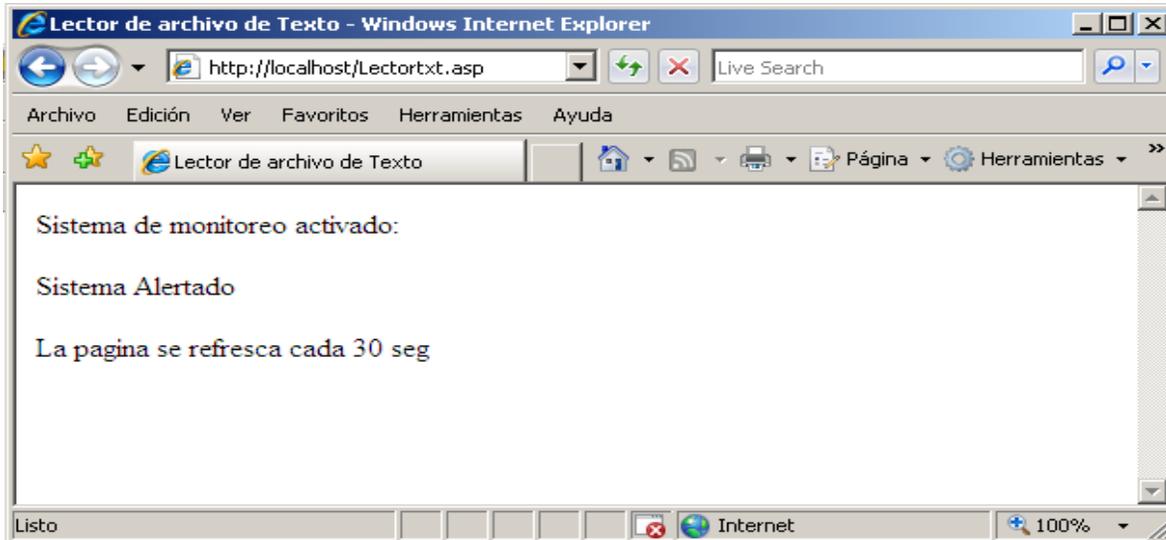
<FORM METHOD="POST" ACTION="entrar.asp">
<table>
<tr>
<td align="right">Usuario: </td>
<td align="center"> <input name = "Usuario"></td>
</tr>
<tr>
<td align="right">Password:</td>
<td align="center"> <input type = "password" name = "Clave"></td>
</tr>
<tr>
<td></td>
<td align="right"><INPUT TYPE="SUBMIT" VALUE="Entrar"></td>
</tr>
</table>
</FORM>
</body>
</html>
```

#### **4.2.3 Adquisición de datos Web**

Una vez que la aplicación SMC ha generado el archivo de texto, una página Web diseñada en ASP se encargara de estar monitoreando los cambios en este archivo. Mientras no existan cambios la página no generara ningún estado de alerta. Una vez que se genera actualiza la información en la base de datos y muestra el mensaje de "Sistema alertado" (figura 4.5).

La página se refresca cada 30 seg. y tiene la capacidad de enviar un mensaje de texto a un teléfono celular.

Esta página solo tiene la función de monitoreo y adquisición de datos; no se puede acceder desde Internet, solo se ejecuta en el servidor local IIS.



**Figura 4.5 Pantalla de Monitoreo y Adquisición de Datos**

El algoritmo de funcionamiento es el siguiente:

1. Inicio de la Página Lectortxt.asp.
2. Verifica la variable “Alerta” (Session) si es Falso o Verdadero.
3. Si la variable “Alerta” es Verdadera muestra el texto “Sistema Alertado”.
4. Carga los valores de la base de datos “Tabla\_Estatus” en un Recordset. Estos valores son los que programo el usuario.
5. Lee el archivo .txt generado por la aplicación SMC al leer los puertos del HC11F1.
6. Convierte la cadena de texto a un vector.
7. Compara los valores del vector con los valores del campo “Estatus” previamente cargado en el Recordset.
8. Si no hay cambios regresa al punto 2.
9. Redirecciona a la pagina “actualizar.asp” y actualiza los valores del campo “estatus” y “Alertado” en “Tabla\_Estatus”.
10. Cambia el valor de “Alerta” (Session) a Verdadero (True).
11. Redirecciona a Lectortxt.asp
12. Fin del proceso.

A continuación se muestran los códigos de las páginas Lectortxt.asp y actualizar.asp.

## CAPÍTULO 4

Lectortxt.asp

PROGRAMA QUE LEE EL ESTADO DE LAS HABITACIONES DESDE UN ARCHIVO EN TEXTO PLANO (TXT)

```

<html>
<head>
<title>Lector de archivo de Texto</title>
</head>

<body>
<%
Dim MiCadena, MiMatriz
Dim txt1, txt2, txt3
Dim fs, f, i
Dim oConn, sSQL, oRS, sSQLUP, valcomp, valcomp1, alerta, alerta2, valcomp2
'Const adOpenKeyset = 1
'Const adLockOptimistic = 3

Response.Write("Sistema de monitoreo activado:")
Response.Write "<p>"
Alerta2 = Session("Alerta")
If Alerta2 = True Then
    Response.Write("Sistema Alertado")
    Response.Write "<p>"
End If

***** Conecto a la base de datos *****

Set oConn = Server.CreateObject("ADODB.Connection")
oConn.Open("Provider=Microsoft.Jet.OLEDB.4.0; Data Source=" & Server.MapPath("db\bd1.mdb"))
set oRS=Server.CreateObject("ADODB.Recordset")
sSQL = "SELECT * FROM Tabla_Estatus"
Set oRS = oConn.Execute(sSQL)

***** Leo archivo de texto *****
Set fs=Server.CreateObject("Scripting.FileSystemObject")
Set f=fs.OpenTextFile(Server.MapPath("testread.txt"))
MiCadena = f.ReadAll
MiMatriz = Split(MiCadena, ",")

***** Rutina de comparacion *****
i=0
Do While NOT oRS.EOF
set valcomp = oRS("Estatus")
set valcomp2 = oRS("Alertado")
valcomp1 = MiMatriz(i)

if valcomp1 <> valcomp then
    actualizar="http://localhost/actualizar.asp?Estatus=" & valcomp1 & "&" & "Cont=" & i
    Response.Redirect(actualizar)
end if
if valcomp2 = "si" then
    Session("Alerta")=True
end if
oRS.MoveNext
i=i+1

Loop

***** Cierro base de datos *****
oRS.Close
Set oRS = Nothing
oConn.Close
Set oConn = Nothing

Response.Buffer = True

```

## CAPÍTULO 4

```

Response.Expires = 0
Response.Write("<p>")
Response.Write("La pagina se refresca cada 30 seg")
Response.AddHeader "Refresh", "30" ' tiempo
%>
</body>
</html>

```

Actualizar.asp

PROGRAMA PARA ACTUALIZAR EL ESTADO DE PUERTAS, VENTANAS Y/O LUZ ACTUALIZANDO EL ARCHIVO .TXT

```

<%@LANGUAGE="VBSCRIPT" CODEPAGE="1252"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
</head>
<%
Dim Alerta, oConn, oRS, si
'Recogemos los valores del formulario
Estatus=Request.QueryString("Estatus")
Cont= Request.QueryString("Cont")
response.Write("<p>")

'Instanciamos y abrimos nuestro objeto conexion

Set oConn = Server.CreateObject("ADODB.Connection")
oConn.Open("Provider=Microsoft.Jet.OLEDB.4.0; Data Source=" & Server.MapPath("db\bd1.mdb"))
set oRS=Server.CreateObject("ADODB.Recordset")

'Ahora creamos la sentencia SQL
si=si
sSQLUP="Update Tabla_Estatus Set Estatus="" & Estatus & "", Alertado='si' Where Cont="" & Cont & ""

oRS = oConn.Execute(sSQLUP)
oConn.Close
Session("Alerta")=True
Response.Redirect("Lectortxt.asp")

%>
<body>
</body>
</html>

```

La Figura 4.6, muestra la tabla “Tabla\_Estatus” la cual es parte de la base de datos que maneja el sistema. Esta compuesta por cinco columnas:

**Id**, es valor numérico que ayuda a identificar cada registro.

**Elemento**, contiene los elementos que a monitorear y controlar.

**Estatus**, para el objetivo de este trabajo “si” significa “abierto” y “no” significa “cerrado”.

**Alertado**, “si” significa “sistema alertado” y “no” significa “sistema normal”.

**Cont**, es de valor numérico y nos ayuda en el manejo de los vectores debido a que estos inician en “0”.

	Id	Elemento	Estatus	Alertado	Cont
▶ ⊕	1	Ventana	si	no	0
⊕	2	Luz_Principal	si	no	1
⊕	3	Infrarojo_Pta	si	si	2
⊕	4	Sist_Gen	si	no	3
*	(Autonumérico)				

Registro: 1 de 4

**Figura 4.6 Base de datos asociada a la adquisición de datos**

### 4.3 Diseño de aplicación Voz

La aplicación de Voz tendrá dos funciones importantes a saber:

1. Monitorea la línea telefónica en espera de llamadas entrantes.
2. Interactuar con el usuario a través de un sencillo IVR.

En el primer caso, el sistema ocupara el modem para poder monitorear una llamada entrante y a través de comandos AT descolgar la línea telefónica.

Ya que se ha establecido la conexión, el IVR se encargara de interactuar con el usuario.

Para interactuar con el usuario se utiliza el control Microsoft Multimedia Control 6.0 (MMControl) de visual basic, el control tiene los elementos necesarios para la reproducción de audio y video (ver figura 4.7).

Una gran funcionalidad de este control es que podemos acceder a todas sus funciones con código sin necesidad de utilizar la interfaz del control.

El control puede reproducir:

- Waveaudio, para extensiones wav.
- Claudio, reproducción de pistas de cd.
- Sequencer, para archivo midi.
- Avivideo, para archivos de video avi.



Figura 4.7 Interfaz del control MMControl

### 4.3.1 Circuito de acoplamiento

La forma de acoplar a la PC con la línea telefónica es a través del circuito mostrado en la figura 4.7.

El circuito tiene dos funciones importantes: a) proteger a la tarjeta de sonido contra transitorios de AC y acoplar los voltajes de DC y, b) mantener la llamada una vez que el programa acepta una llamada entrante.

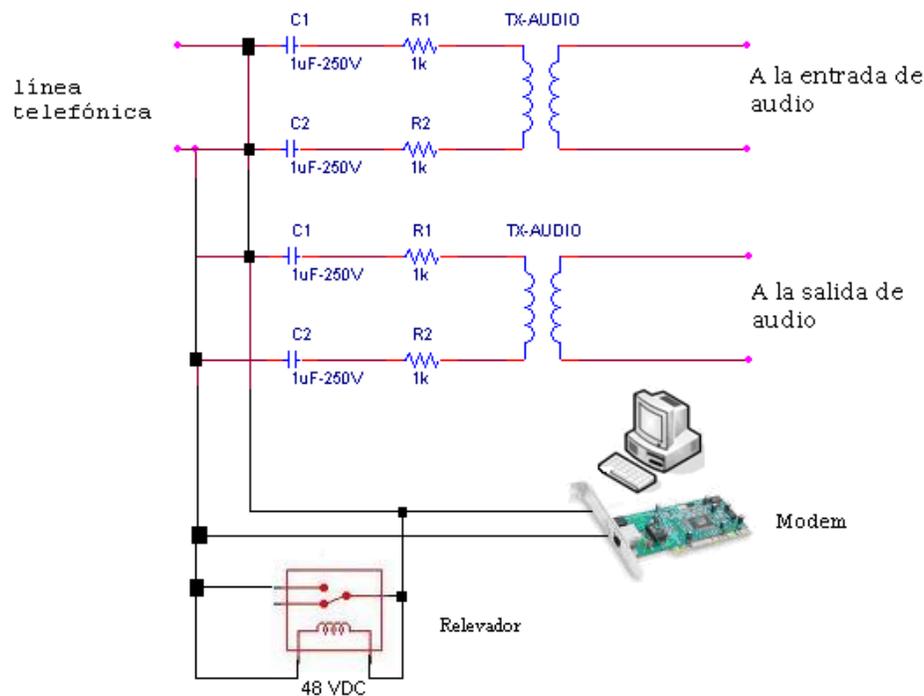


Figura 4.8 Circuito de acoplamiento

### 4.3.2 Algoritmo de la aplicación de Voz

El algoritmo de la aplicación de Voz es como sigue:

1. Modem envía señal de llamada entrante.
2. Aplicación de Voz acepta llamada después de 3 ring's.
3. Relevador cierra circuito para mantener llamada activa.
4. Aplicación de Voz libera al modem.
5. El IVR envía mensaje de bienvenida y solicita contraseña de acceso.
6. Módulo DTMF decodifica contraseña.
7. La aplicación busca contraseña tecleada con base de datos.
8. Si la contraseña es incorrecta la aplicación vuelve a solicitar contraseña, después de 3 intentos termina llamada.
9. Si la contraseña es correcta el sistema revisa si hay sensores alarmados, si los hay envía mensaje de sistemas alarmado y cual sensor se alarmo
10. Envía menú de opciones.
11. Módulo DTMF decodifica tecla marcada vía la entrada de micrófono de la tarjeta de audio.
12. La aplicación detecta si es tecla valida, si es tecla valida envía mensaje, si no es valida envía mensaje de "Opción no valida" y vuelve a enviar menú de opciones.
13. Espera nuevo tono y regresa al punto 9.
14. DTMF detecta tono de fin de llamada.
15. Aplicación termina llamada.

### 4.3.3 Diseño de software y código fuente

La aplicación de Voz esta compuesta por 3 módulos de formulario, 1 estándar y 2 de clase.

Al iniciar el programa aparecerá esta pantalla.



Figura 4.9 Monitor de llamadas

Al presionar el botón "Iniciar" el sistema se pone en el estado de espera de llamada.

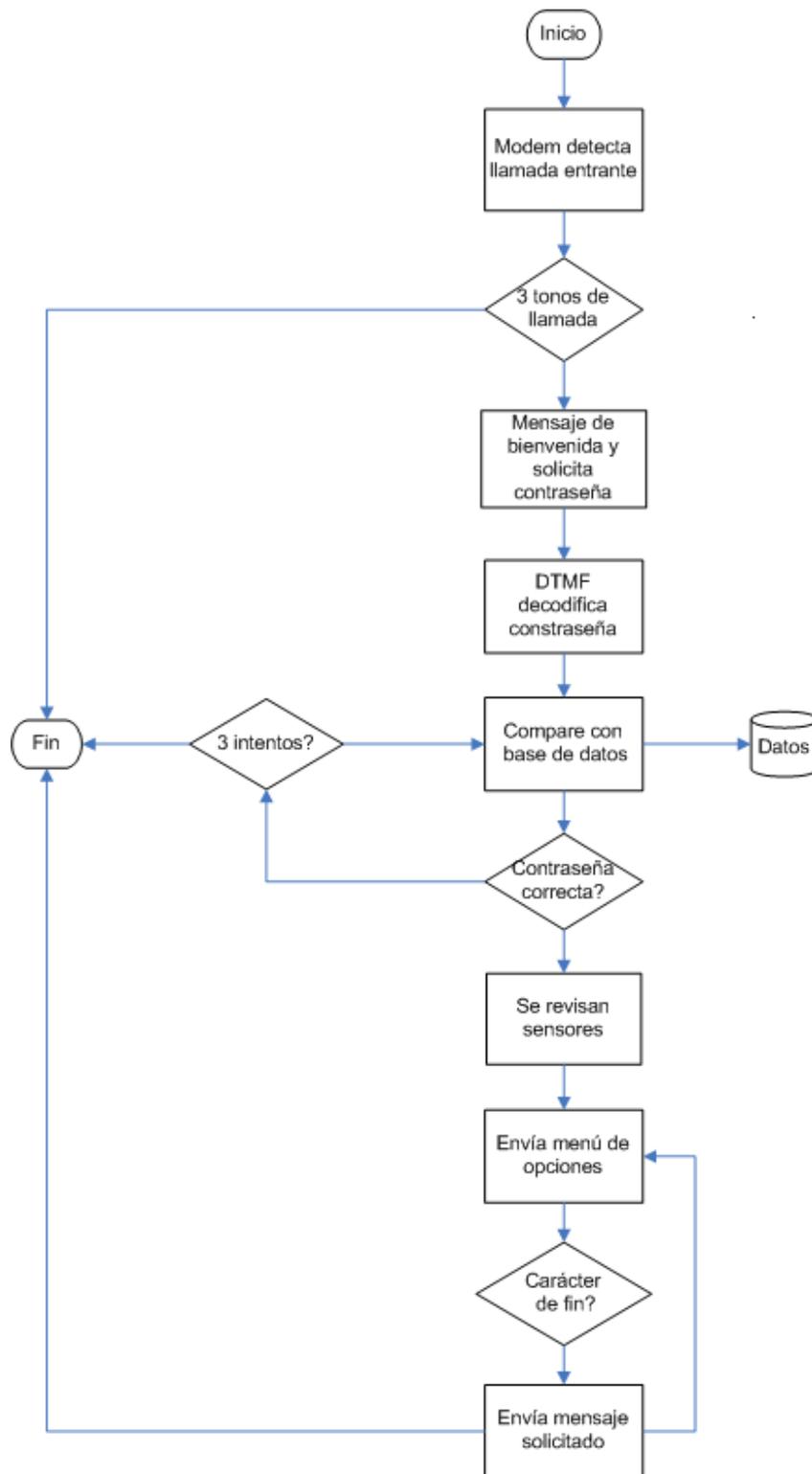


Figura 4.10 Diagrama de flujo de aplicación de Voz

Después de 3 tonos de llamada, el sistema acepta la llamada entrante y envía mensaje de bienvenida solicitando clave acceso.



Figura 4.11 Pantalla de detección de clave

El sistema decodifica la clave a través del módulo DTMF. Si la clave es correcta envía el menú de opciones, a partir de este punto el usuario comienza a interactuar con el sistema.

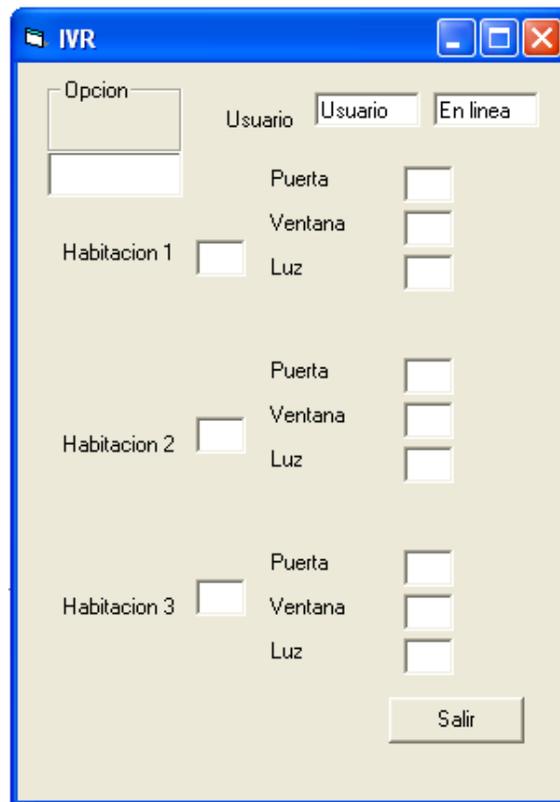


Figura 4.12 Menú de opciones

## Código fuente

### Formulario: Monitor de llamadas (Figura 4.x)

```

Dim cont As Integer

Private Sub CbDetener_Click()
    CbDetener.Enabled = False
    CbIniciar.Enabled = True
End Sub

Private Sub CbIniciar_Click()
    CbDetener.Enabled = True
    CbIniciar.Enabled = False
    Text1.Text = "Esperando llamada"

End Sub

Private Sub Form_Load()
    CbDetener.Enabled = False
    CbIniciar.Enabled = True

If Not MSComm1.PortOpen Then
    MSComm1.PortOpen = True
    If Err Then Exit Sub
End If

Text1.Text = "Sistema inactivo"
cont = 0

End Sub

Private Sub MSComm1_OnComm()
    If MSComm1.CommEvent = comEvRing Then
        cont = cont + 1
        Text1.Text = "Llamada entrante"
        If cont > 3 Then
            MSComm1.Output = "ATA" & vbCrLf
            SMC.Show
            Text1.Text = "Llamada activa"
        End If
    End If
End Sub

```

### Formulario: Detección de contraseña (figura 4.x)

```

Option Explicit
Dim m As Integer
Dim dgt As String
Dim passw, passw01 As String
Private WithEvents clsRecorder As WaveInRecorder
Private clsDSP As clsDSP
Private intSamples() As Integer
Private blnLoaded As Boolean

Private Sub clsRecorder_GotData(intBuffer() As Integer, lngLen As Long)
    intSamples = intBuffer
    If Not clsRecorder.IsRecording Then Exit Sub
End Sub

Private Sub Record_init()

```

## CAPÍTULO 4

---

```

If clsRecorder.IsRecording Then
    If Not clsRecorder.StopRecord Then
        MsgBox "Could not stop recording!", vbExclamation
    End If
Else
    clsDSP.samplerate = CLng(8000)
    clsDSP.Channels = 1 + 1

    If Not clsRecorder.StartRecord(8000, 1 + 1) Then
        MsgBox "Could not start recording!", vbExclamation
    End If
End If

End Sub

Private Sub Form_Load()
    exec = False
    freqs(0) = 697
    freqs(1) = 770
    freqs(2) = 852
    freqs(3) = 941
    freqs(4) = 1209
    freqs(5) = 1336
    freqs(6) = 1477
    freqs(7) = 1633
    MAX_BINS = 8
    GOERTZEL_N = 92
    SAMPLING_RATE = 8000

    MMControl1.Visible = False

    Text1.Text = ""
    Text1.Alignment = 2
    Set clsRecorder = New WaveInRecorder
    Set clsDSP = New clsDSP
    ReDim intSamples(FFT_SAMPLES - 1) As Integer
    blnLoaded = True
    clsDSP.samplerate = CLng(8000)
    clsDSP.Channels = 1 ' para estereo
    Me.Show
    Record_init

    MMControl1.DeviceType = "WaveAudio"
    MMControl1.FileName = "C:\VB\Menu_voz_raiz\Bienvenido.wav"
    MMControl1.Command = "open"
    MMControl1.Command = "play"

End Sub

Private Sub text1_change()
    passw = Text1.Text
    passw01 = InStr(passw, "#")
    If passw01 <> 0 Then
        Call validauser(passw)
        Text1 = ""
    End If

End Sub

Private Sub Form_Unload(Cancel As Integer)

    If clsRecorder.IsRecording Then
        Record_init
    End If

```

---

## CAPÍTULO 4

```

    Set clsRecorder = Nothing
End Sub

Private Sub tmrVis_Timer()
Dim i As Long
Dim Curnum As String

For i = 0 To UBound(intSamples)
    goertzel intSamples(i), dgt
    intSamples(i) = 0
    If Len(dgt) >= 2 Then Exit For
Next i

If Len(dgt) >= 2 Then
Text1.Text = Text1.Text & Left$(dgt, 1)
Curnum = Left$(dgt, 1)

DoEvents
End If
dgt = ""

End Sub

Sub validauser(ByVal Userval As String)

'Private Sub cmdEntrar_Click()
' Cadena de conexión ( INDICAR EL PATH DE LA BASE DE DATOS )
Const C_CADENA = "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=" & "C:\VB\Menu_voz_raiz_v1.0\Usuarios.mdb" & ";"

' Variable para el recordset
Dim Rst_Login As Recordset

' crea el recordset
Set Rst_Login = New Recordset

Dim SQL As String

SQL = "SELECT Passwd FROM Usuarios WHERE Passwd = " & Userval & ""

With Rst_Login
' Abre el recordset
.Open SQL, C_CADENA

' Si el recordset está vacío es por que es incorrecto
If .EOF Then
MMControl1.Command = "prev"
MMControl1.FileName = "C:\VB\Menu_voz_raiz\ContraseñaIncorrecta.wav"
MMControl1.Command = "open"
MMControl1.Command = "play"

Rst_Login.Close
Set Rst_Login = Nothing
Exit Sub
Else:
IVR.Show
SMC.Visible = False
End If
End With

' Cierra y descarga el Recordset
Rst_Login.Close
Set Rst_Login = Nothing

```

## CAPÍTULO 4

```

End Sub
Private Sub cmdSalir_Click()
    OK = False
    Unload Me
End Sub

Private Sub Form_KeyPress(KeyAscii As Integer)
If KeyAscii = 13 Then Call cmdEntrar_Click
End Sub

```

## Formulario: Menú de opciones (figura 4.x)

```

Public cadena01 As String

Private Sub Command1_Click()
IVR.Command = "close"
End
End Sub

Private Sub Form_Load()
IVR.Visible = False
IVR.DeviceType = "resume"
IVR.DeviceType = "WaveAudio"
IVR.FileName = "C:\IVR\Sonidos\ContraseñaOK.wav"
IVR.Command = "open"
IVR.Command = "play"
TxtHab1.Enabled = False
TxtHab1Pta.Enabled = False
TxtHab1Vta.Enabled = False
TxtHab1Luz.Enabled = False
End Sub

Private Sub CallData(valor As Integer)
Do While Not MiRecordset.EOF
    i = i + 1
    Set TxtHab1Pta.DataSource = MiRecordset
    MiRecordset.MoveNext
Loop
End Sub

Private Sub text1_change()
If Text1 <> "" Then
Dim i As Integer
Dim LogicaH1() As String
'Dim cadena01 As String
ReDim LogicaH1(0 To 9)

cadena01 = cadena01 & Text1
Label9.Caption = cadena01

'*** Cargo la base de datos
'Declarar MiConexion y MiRecordset
Dim MiConexion As ADODB.connection
Dim MiRecordset As ADODB.Recordset
'Dim i As Integer

'Crear la conexión y ponerle la cadena de conexión en su propiedad ConnectionString
Set MiConexion = New ADODB.connection
MiConexion.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\IVR\Estados_Casa.mdb;"

MiConexion.Open
Set MiRecordset = New ADODB.Recordset
If InStr(cadena01, "1") = 1 Then

```

## CAPÍTULO 4

```

'Abrir el objeto Recordset
MiRecordset.Open "SELECT Estado FROM HabitacionA", MiConexion, adOpenDynamic, adLockOptimistic

i = 0
Do While Not MiRecordset.EOF
    LogicaH1(i) = MiRecordset.Fields("Estado").Value
    i = i + 1
    MiRecordset.MoveNext
Loop
'Habitacion 1
TxtHab1.Enabled = True
TxtHab1.BackColor = vbGreen
TxtHab1.Text = "On"
TxtHab1Pta.Enabled = True
TxtHab1Vta.Enabled = True
TxtHab1Luz.Enabled = True

TxtHab1Pta.Text = LogicaH1(0)
TxtHab1Vta.Text = LogicaH1(1)
TxtHab1Luz.Text = LogicaH1(2)

If InStr(cadena01, "1") = 1 And InStr(cadena01, "11") = 0 And InStr(cadena01, "12") = 0 And InStr(cadena01, "13") = 0
And InStr(cadena01, "14") = 0 And InStr(cadena01, "15") = 0 Then
    IVR.DeviceType = "WaveAudio"
    IVR.FileName = "C:\IVR\Sonidos\HabitacionA.wav"
    IVR.Command = "open"
    IVR.Command = "play"
Elseif InStr(cadena01, "11") = 1 Then

    If LogicaH1(0) = "On" Then
        'Puerta abierta
        IVR.DeviceType = "WaveAudio"
        IVR.FileName = "C:\IVR\Sonidos\Puertaabierta.wav"
        IVR.Command = "open"
        IVR.Command = "play"
        cadena01 = "1"
    Else
        'Puerta cerrada
        IVR.DeviceType = "WaveAudio"
        IVR.FileName = "C:\IVR\Sonidos\Puertacerrada.wav"
        IVR.Command = "open"
        IVR.Command = "play"
        cadena01 = "1"
    End If

Elseif InStr(cadena01, "12") = 1 Then

    If LogicaH1(1) = "On" Then
        'Puerta abierta
        IVR.DeviceType = "WaveAudio"
        IVR.FileName = "C:\IVR\Sonidos\ventanaabierta.wav"
        IVR.Command = "open"
        IVR.Command = "play"
        cadena01 = "1"
    Else
        'Puerta cerrada
        IVR.DeviceType = "WaveAudio"
        IVR.FileName = "C:\IVR\Sonidos\ventanacerrada.wav"
        IVR.Command = "open"
        IVR.Command = "play"
        cadena01 = "1"
    End If

Elseif InStr(cadena01, "13") = 1 Then

    If LogicaH1(2) = "On" Then

```

## CAPÍTULO 4

```

'Puerta abierta
IVR.DeviceType = "WaveAudio"
IVR.FileName = "C:\IVR\Sonidos\luzencendida.wav"
IVR.Command = "open"
IVR.Command = "play"
cadena01 = "1"
Else
'Puerta cerrada
IVR.DeviceType = "WaveAudio"
IVR.FileName = "C:\IVR\Sonidos\luzapagada.wav"
IVR.Command = "open"
IVR.Command = "play"
cadena01 = "1"
End If

Elseif InStr(cadena01, "14") = 1 Then
IVR.DeviceType = "WaveAudio"
IVR.FileName = "C:\IVR\Sonidos\HabitacionA.wav"
IVR.Command = "open"
IVR.Command = "play"
cadena01 = "1"

Elseif InStr(cadena01, "15") = 1 Then
cadena01 = ""
IVR.DeviceType = "resume"
IVR.DeviceType = "WaveAudio"
IVR.FileName = "C:\IVR\Sonidos\ContraseñaOK.wav"
IVR.Command = "open"
IVR.Command = "play"
TxtHab1.Enabled = False
TxtHab1.BackColor = vbWhite
TxtHab1.Text = ""
TxtHab1Pta.Enabled = False
TxtHab1Vta.Enabled = False
TxtHab1Luz.Enabled = False

End If
MiRecordset.Close
MiConexion.Close

Set MiRecordset = Nothing
Set MiConexion = Nothing

Elseif InStr(cadena01, "2") = 1 Then
'Habitacion 2
'Abrir el objeto Recordset
'Abrir el objeto Recordset
MiRecordset.Open "SELECT Estado FROM HabitacionB", MiConexion, adOpenDynamic, adLockOptimistic

i = 0
Do While Not MiRecordset.EOF
LogicaH1(i) = MiRecordset.Fields("Estado").Value
i = i + 1
MiRecordset.MoveNext
Loop
'Habitacion 2
TxtHab2.Enabled = True
TxtHab2.BackColor = vbGreen
TxtHab2.Text = "On"
TxtHab2Pta.Enabled = True
TxtHab2Vta.Enabled = True
TxtHab2Luz.Enabled = True

TxtHab2Pta.Text = LogicaH1(0)
TxtHab2Vta.Text = LogicaH1(1)
TxtHab2Luz.Text = LogicaH1(2)

```

## CAPÍTULO 4

```

If InStr(cadena01, "2") = 1 And InStr(cadena01, "21") = 0 And InStr(cadena01, "22") = 0 And InStr(cadena01, "23") = 0
And InStr(cadena01, "24") = 0 And InStr(cadena01, "25") = 0 Then
  IVR.DeviceType = "WaveAudio"
  IVR.FileName = "C:\IVR\Sonidos\HabitacionB.wav"
  IVR.Command = "open"
  IVR.Command = "play"
Elseif InStr(cadena01, "21") = 1 Then

  If LogicaH1(0) = "On" Then
    'Puerta abierta
    IVR.DeviceType = "WaveAudio"
    IVR.FileName = "C:\IVR\Sonidos\Puertaabierta.wav"
    IVR.Command = "open"
    IVR.Command = "play"
    cadena01 = "2"
  Else
    'Puerta cerrada
    IVR.DeviceType = "WaveAudio"
    IVR.FileName = "C:\IVR\Sonidos\Puertacerrada.wav"
    IVR.Command = "open"
    IVR.Command = "play"
    cadena01 = "2"
  End If

Elseif InStr(cadena01, "22") = 1 Then

  If LogicaH1(1) = "On" Then
    'Puerta abierta
    IVR.DeviceType = "WaveAudio"
    IVR.FileName = "C:\IVR\Sonidos\ventanaabierta.wav"
    IVR.Command = "open"
    IVR.Command = "play"
    cadena01 = "2"
  Else
    'Puerta cerrada
    IVR.DeviceType = "WaveAudio"
    IVR.FileName = "C:\IVR\Sonidos\ventanacerrada.wav"
    IVR.Command = "open"
    IVR.Command = "play"
    cadena01 = "2"
  End If

Elseif InStr(cadena01, "23") = 1 Then

  If LogicaH1(2) = "On" Then
    'Puerta abierta
    IVR.DeviceType = "WaveAudio"
    IVR.FileName = "C:\IVR\Sonidos\luzencendida.wav"
    IVR.Command = "open"
    IVR.Command = "play"
    cadena01 = "2"
  Else
    'Puerta cerrada
    IVR.DeviceType = "WaveAudio"
    IVR.FileName = "C:\IVR\Sonidos\luzapagada.wav"
    IVR.Command = "open"
    IVR.Command = "play"
    cadena01 = "2"
  End If

Elseif InStr(cadena01, "24") = 1 Then
  IVR.DeviceType = "WaveAudio"
  IVR.FileName = "C:\IVR\Sonidos\HabitacionA.wav"
  IVR.Command = "open"
  IVR.Command = "play"
  cadena01 = "2"

```

## CAPÍTULO 4

```

ElseIf InStr(cadena01, "25") = 1 Then
    cadena01 = ""
    IVR.DeviceType = "resume"
    IVR.DeviceType = "WaveAudio"
    IVR.FileName = "C:\IVR\Sonidos\ContraseñaOK.wav"
    IVR.Command = "open"
    IVR.Command = "play"
    TxtHab2.Enabled = False
    TxtHab2.BackColor = vbWhite
    TxtHab2.Text = ""
    TxtHab2Pta.Enabled = False
    TxtHab2Vta.Enabled = False
    TxtHab2Luz.Enabled = False
End If
MiRecordset.Close
MiConexion.Close
Set MiRecordset = Nothing
Set MiConexion = Nothing
ElseIf InStr(cadena01, "3") = 1 Then
'Habitacion 3
'Abrir el objeto Recordset
MiRecordset.Open "SELECT Estado FROM HabitacionC", MiConexion, adOpenDynamic, adLockOptimistic

i = 0
Do While Not MiRecordset.EOF
    LogicaH1(i) = MiRecordset.Fields("Estado").Value
    i = i + 1
    MiRecordset.MoveNext
Loop
'Habitacion 3
TxtHab3.Enabled = True
TxtHab3.BackColor = vbGreen
TxtHab3.Text = "On"
TxtHab3Pta.Enabled = True
TxtHab3Vta.Enabled = True
TxtHab3Luz.Enabled = True

TxtHab3Pta.Text = LogicaH1(0)
TxtHab3Vta.Text = LogicaH1(1)
TxtHab3Luz.Text = LogicaH1(2)

If InStr(cadena01, "3") = 1 And InStr(cadena01, "31") = 0 And InStr(cadena01, "32") = 0 And InStr(cadena01, "33") = 0
And InStr(cadena01, "34") = 0 And InStr(cadena01, "35") = 0 Then
    IVR.DeviceType = "WaveAudio"
    IVR.FileName = "C:\IVR\Sonidos\HabitacionA.wav"
    IVR.Command = "open"
    IVR.Command = "play"
ElseIf InStr(cadena01, "31") = 1 Then

    If LogicaH1(0) = "On" Then
        'Puerta abierta
        IVR.DeviceType = "WaveAudio"
        IVR.FileName = "C:\IVR\Sonidos\Puertaabierta.wav"
        IVR.Command = "open"
        IVR.Command = "play"
        cadena01 = "3"
    Else
        'Puerta cerrada
        IVR.DeviceType = "WaveAudio"
        IVR.FileName = "C:\IVR\Sonidos\Puertacerrada.wav"
        IVR.Command = "open"
        IVR.Command = "play"
        cadena01 = "3"
    End If

ElseIf InStr(cadena01, "32") = 1 Then

```

```

If LogicaH1(1) = "On" Then
  'Puerta abierta
  IVR.DeviceType = "WaveAudio"
  IVR.FileName = "C:\IVR\Sonidos\ventanaabierta.wav"
  IVR.Command = "open"
  IVR.Command = "play"
  'IVR.Command = "close"
  cadena01 = "3"
Else
  'Puerta cerrada
  IVR.DeviceType = "WaveAudio"
  IVR.FileName = "C:\IVR\Sonidos\ventanacerrada.wav"
  IVR.Command = "open"
  IVR.Command = "play"
  cadena01 = "3"
End If

Elseif InStr(cadena01, "33") = 1 Then

  If LogicaH1(2) = "On" Then
    'Puerta abierta
    IVR.DeviceType = "WaveAudio"
    IVR.FileName = "C:\IVR\Sonidos\luzencendida.wav"
    IVR.Command = "open"
    IVR.Command = "play"
    cadena01 = "3"
  Else
    'Puerta cerrada
    IVR.DeviceType = "WaveAudio"
    IVR.FileName = "C:\IVR\Sonidos\luzapagada.wav"
    IVR.Command = "open"
    IVR.Command = "play"
    cadena01 = "3"
  End If

Elseif InStr(cadena01, "34") = 1 Then
  IVR.DeviceType = "WaveAudio"
  IVR.FileName = "C:\IVR\Sonidos\HabitacionA.wav"
  IVR.Command = "open"
  IVR.Command = "play"
  cadena01 = "3"

Elseif InStr(cadena01, "35") = 1 Then

  cadena01 = ""
  IVR.DeviceType = "resume"
  IVR.DeviceType = "WaveAudio"
  IVR.FileName = "C:\IVR\Sonidos\ContraseñaOK.wav"
  IVR.Command = "open"
  IVR.Command = "play"
  TxtHab3.Enabled = False
  TxtHab3.BackColor = vbWhite
  TxtHab3.Text = ""
  TxtHab3Pta.Enabled = False
  TxtHab3Vta.Enabled = False
  TxtHab3Luz.Enabled = False
End If
MiRecordset.Close
MiConexion.Close
Set MiRecordset = Nothing
Set MiConexion = Nothing
Elseif InStr(cadena01, "4") = 1 Then
'Repetir el menu
  IVR.DeviceType = "WaveAudio"
  IVR.FileName = "C:\IVR\Sonidos\ContraseñaOK.wav"

```

```

IVR.Command = "open"
IVR.Command = "play"
IVR.Command = "close"
Elseif InStr(cadena01, "5") = 1 Then
'Terminar llamada, colgar.
MsgBox "Cuelga llamada"
End
Else
'Opción no valida
End If
End If
Text1 = ""
End Sub

```

### 4.3.4 Módulos para detector tonos duales de multifrecuencia

Para poder detectar los tonos DTMF por visual basic 6.0 se encontraron muy pocas opciones, entre las que se pueden destacar:

- a) DTMF Decoder, <http://www.polar-electric.com>, costo \$19.90 US.
- b) Tone Decoder ActiveX Control, <http://www.tapiex.com>, costo \$79 Euros.
- c) DTMF Decoder V2.0, <http://www.planet-source-code.com/vb/scripts/ShowCode.asp?txtCodeId=67331&lngWid=1>, el código se puede usar libremente y sin cargos.

Las dos primeras opciones solo permiten un periodo de prueba de 15 días, posterior a esto los proyectos donde se usen solicitaran el registro y no permitirán su ejecución.

La tercera opción es más viable para uso didáctico, aunque es inestable ya que se cierran de forma inesperada los programas. El autor (Sagar Devkota) permite el uso del código, compilarlo y distribuirlo (ya compilado) libremente y sin cargos.

DTMF Decoder V2.0 detecta tonos a través de la tarjeta de sonido (entrada de micrófono) en tiempo real. El código entrega el valor detectado a una variable que se puede manipular libremente.

El código esta incluido en un módulo y dos módulos de clase. El código esta basado en el algoritmo de Goertzel.

En la figura 4.12 se muestran los módulos para detección de tonos DTMF.

En el módulo esta incluido el algoritmo de Goertzel, en los módulos de clase esta un DSP y un grabador de sonido.

No se entra más en detalles ya que no se tiene manual ni datos adicionales.

El código se muestra en el anexo B.

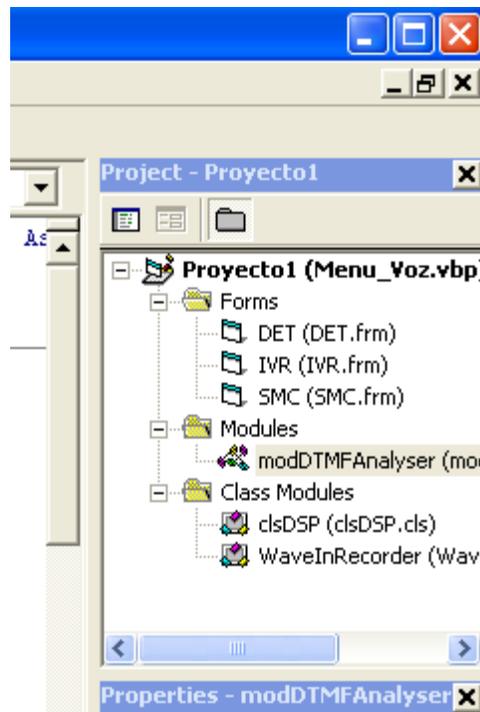


Figura 4.13 Elementos de aplicación de Voz.

## Capítulo 5 Integración de Bloques Funcionales

En los capítulos anteriores, revisamos la teoría de los componentes que integran el sistema de forma aislada. En este capítulo, se hablará del funcionamiento de los bloques del sistema y su interacción con el usuario.

Tenemos los siguientes:

- SMC (Sistema de Monitoreo y Control)
- Servidor WEB (Local y Remoto)
- Modulo de consulta
- Monitoreo por el HC11

El la figura 5.1, se aprecia este sistema a bloques

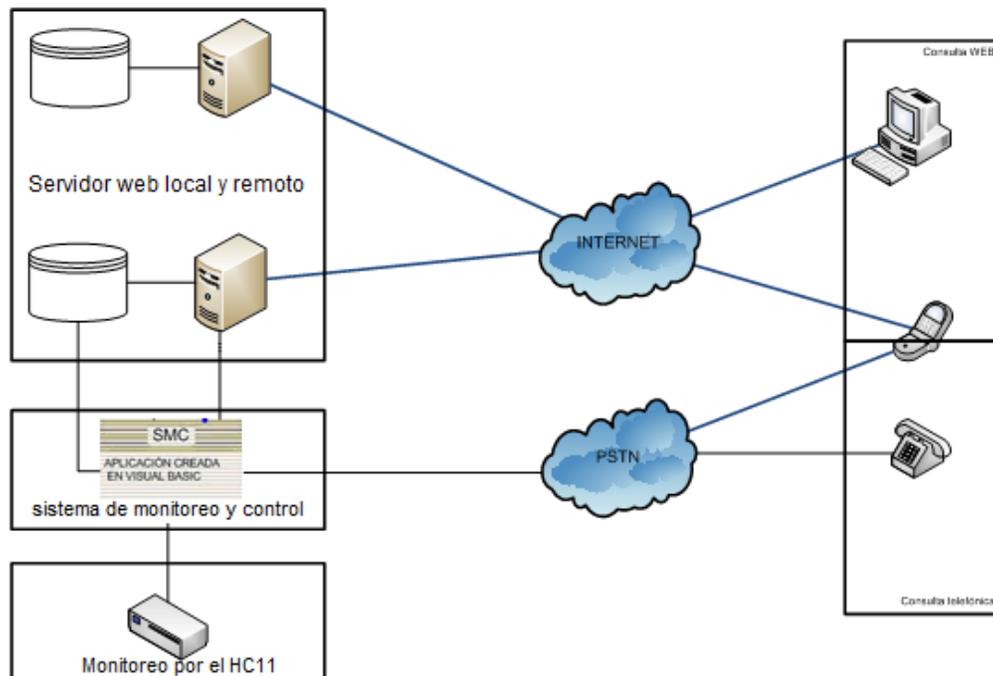


Figura 5.1 Sistema a bloques

## 5.1 SMC (Sistema de Monitoreo y Control)

La aplicación SMC (Sistema de Monitoreo y Control) es la espina dorsal del sistema, es quien se encargara de controlar la comunicación entre el HC11 y los medios de consulta, ya sean por teléfono o portal Web. Esta aplicación está desarrollada en Visual Basic 6.

El SMC está integrado por los siguientes módulos:

<b>Módulo</b>	<b>Descripción</b>
Adquisición	Se encarga de adquirir los datos del HC11 a través de una comunicación bidireccional.
Decodificador	Se encarga de contestar las llamadas una vez que el sistema se a programado para esta función y detectará los números marcados para asignarlos a las funciones correspondientes.
IVR	El IVR se encarga de reproducir los menús, estados consultados, y cambios solicitados.
Control TXT	Generar un archivo txt para ser leído por la aplicación Web. También, leerá el archivo txt generado por la aplicación Web con las modificaciones solicitadas por el usuario.

A grandes rasgos, las funciones del SMC son:

1. Tener una comunicación bidireccional con el HC11, leyendo los estatus de los puertos y enviando modificaciones que los usuarios deseen hacer, ya sea por medio de teléfono o vía web. La información recolectada del HC11 será enviada a una base de datos.
2. Generar un archivo .txt con el estatus actual de los sensores para que pueda ser leído por la aplicación Web.
3. Contestar llamadas telefónicas después de cierta cantidad de rings (programable en la aplicación) y entre el IVR reproduciendo las opciones programadas.

## **5.2 Servidor WEB (Local y Remoto)**

El servidor Web tendrá alojado una imagen del Portal Web en forma local y tendrá acceso tanto a la Base de Datos como al archivo de texto. El Portal Web Local enviará cada cierto intervalo de tiempo información al Servidor Web Remoto, alojado con un proveedor de Hosting. La ventaja de este punto es mantener la información resguardada en un sitio lejano a la casa para que en caso de generarse algún percance la información no sea destruida.

## **5.3 Consulta telefónica**

Cuando la consulta se realiza vía telefónica, el usuario genera una llamada por la PSTN a la casa habitación. Se establece la llamada con un modem y se lleva a cabo la comunicación con el software detector de tonos (SMC). Una vez establecida la llamada y detectada por el software, este dará inicio al menú de voz

para que el usuario pueda tomar las decisiones respecto a las consultas del estado de la casa mediante un árbol de decisiones. El software SMC es el encargado en interpretar los tonos DTMF para reproducir la opción elegida por el usuario del menú. En dicho menú, se tienen distintas opciones para las habitaciones que conforman la casa habitación de acuerdo a los elementos que se estén sensando.

Los campos del menú son los que se muestran en la figura 5.3.1.

NIVEL 0	NIVEL 1	NIVEL 2	NIVEL 3
Raíz	Habitación	Elemento_de_habitación	Estado_del_Elemento

**Figura 5.3.1 Niveles del menú de opciones**

Como se aprecia en la figura 5.3.1, dependiendo del Nivel donde se encuentre el usuario en el menú, será la grabación que el software reproducirá.

- En el Nivel 0 (raíz) el usuario escuchará la bienvenida al sistema y se le solicitará al usuario su contraseña de acceso.
- En el Nivel 1 se escucharán las opciones para seleccionar la habitación deseada de la casa habitación, así como la opción de *Repetir Menú* y *Salir del Sistema*.
- En el nivel 2 se podrá seleccionar algún elemento de la habitación seleccionada anteriormente como podría ser el estado de la puerta, el estado de la ventana, el estado de la luz, etc, dependiendo de los elementos que se estén sensando en la habitación. También se tiene la opción *Repetir Menú*, *Regresar al menú Anterior* y *Salir del Sistema*.

- Finalmente en el nivel 3 se informará al usuario el estado que guarda al momento de realizar la consulta el elemento seleccionado. Una vez que ha notificado el estado del elemento, el IVR preguntará si requiere realizar otra consulta o solicitarle al usuario que cuelgue para salir del sistema. Si el usuario no realiza ninguna acción después de un tiempo definido (timer), el sistema lo desconectará para evitar dejar la llamada establecida.

En la figura 5.3.2 muestra como quedaría el árbol del IVR que compone el sistema de consulta telefónico para una casa habitación de tres habitaciones con tres elementos sensados cada una. Cada elemento cuenta con dos posibles estados (ON/OFF).

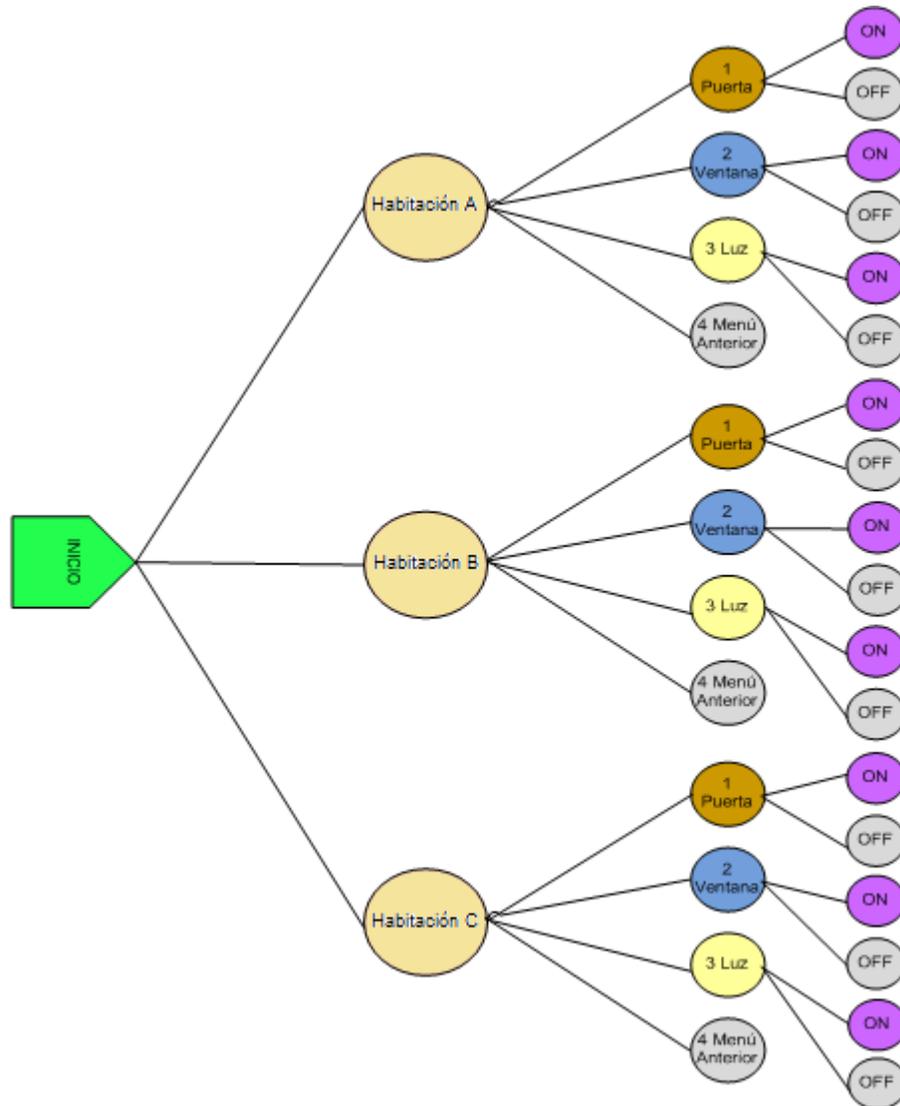


Figura 5.3.2 Árbol del IVR

## 5.4 Consulta por Internet

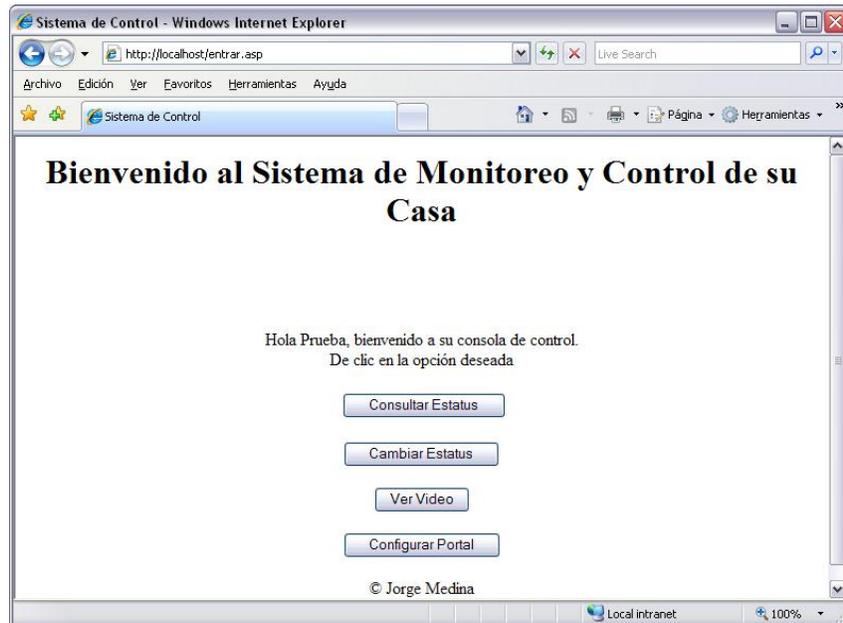
Si la consulta es mediante Internet, se tiene la interfaz de usuario donde se pueden realizar consultas del estado actual de los elementos de la casa habitación navegando gráficamente en la página de la aplicación. Esta interfaz, realizará la consulta a una base de datos donde se guardarán los cambios de estado de la casa habitación ubicada localmente. En este servidor local se tendrá un histórico

de los cambios de estado de la casa habitación, así como un servidor remoto con la imagen de este.

En la interfaz WEB tendremos un menú gráfico, donde podremos seleccionar que deseamos consultar dentro de la casa.

En la figura 5.4.1 se muestra la pantalla principal del sistema WEB. Como se aprecia, podemos seleccionar las siguientes opciones:

- Consultar Estatus
- Configurar Estatus
- Ver videos



**Tabla 5.4.1** pantalla principal sistema web

La interfaz WEB será mucho más amigable gracias a su ambiente gráfico, permitiendo una fácil interacción con el usuario. En la figura 5.4.2 se muestra la pantalla que el usuario estaría consultando en Internet para cambiar el estatus de un elemento de la casa habitación.



**Figura 5.4** Modificación de registros

### **5.5 Monitoreo por el HC11**

El Sistema de Monitoreo y Control es el encargado de leer los puertos del HC11 y generar la información necesaria en un archivo de texto a su vez el HC11 lee a los dispositivos electrónicos involucrados (sensores infrarrojos, sensores de luz, etc.).

El microcontrolador HC11 de Motorola, estará poleando cada cierto tiempo los sensores de la casa habitación para poder consultar el estado de los elementos medidos. Esta es la interfaz entre el software de control y la casa habitación, es decir, envía la información necesaria al software para que este sepa interpretar al usuario el estado de los elementos.

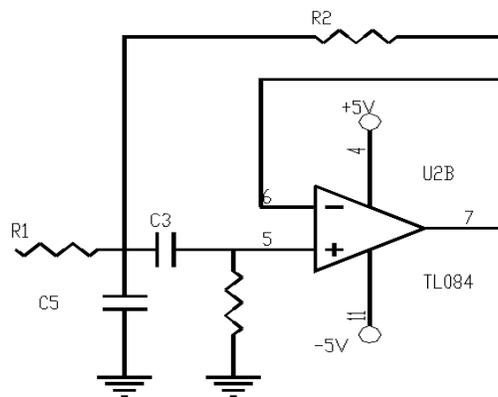
## Capítulo 6. Resultados, aplicación y conclusiones.

### 6.1. Resultados

Al principio de este trabajo se planteo y se desarrollo el sistema con el fin de realizar la detección de tonos DTMF acoplando la línea telefónica al circuito integrado MC14LC5436P de Motorola semiconductor y así mediante transformadores de audio y filtros analógicos suprimir señales indeseables de ruido.

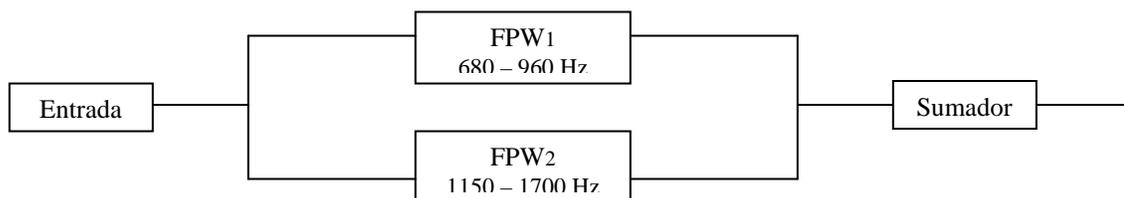
Para nuestro caso en particular se requirió del diseño de dos filtros paso-banda, los cuales se tenían que adaptar al intervalo de frecuencias requerido para la voz. El circuito general empleado en la etapa de filtrado se muestra en la Figura 6.1.1.

*FPBW1 (680 - 960 Hz) y FPBW2 (1150 – 1700 Hz) obteniendo.*



**Figura 6.1.1. Filtro paso-banda**

Una representación por medio de bloques de esta etapa de filtrado es la siguiente:



De igual forma la detección de una llamada entrante la realizaría el Microcontrolador MC68HC11 donde básicamente se realizaba una comparación para determinar si hay una señal de ring, es decir si el pin 7 del puerto E del microcontrolador se pone en 0 lógico quiere decir que está presente un aviso de llamada, donde la entrada provenía de la señal de la línea telefónica ya filtrada y se continuaba con la siguiente instrucción en el programa en lenguaje ensamblador, pero si el valor leído es un 1 lógico se regresaba para volver a tomar otra lectura definiendo así a los cuantos rings se debía contestar.

El microcontrolador también realizaba las funciones de detectar un tono válido una vez que el circuito DTMF captaba una entrada del aparato telefónico y por último un programa capaz de transmitir el dato hacia una computadora. Finalmente en la computadora se programarían los menús de voz y una interfaz que regresaba los datos leídos al usuario para tomar alguna decisión. El resultado de todo lo anterior no fue satisfactorio ya que los valores de los filtros fueron seleccionados de acuerdo a técnicas obtenidas en programas de emulación y no se consiguió obtener el rango de frecuencia deseado para permitir la correcta decodificación de la tecla marcada, es por eso que el trabajo se basó en una tecnología apoyada en software y los beneficios que proveen una computadora personal con el filtrado digital que proporcionan los modem.

De esta forma se obtuvieron resultados de detección óptimos visualizados en la pantalla de la PC y mostrados mediante una interfaz WEB y una interfaz de voz con una línea telefónica convencional o celular como ya se describió en los capítulos anteriores.

## 6.2. Aplicación

La aplicación para este sistema de detección DTMF puede ser implementada de forma sencilla, ya que solo es necesario contar con una línea telefónica y una PC conectada a Internet, o bien un teléfono celular. El uso más frecuente para este tipo de sistemas y para lo cual está enfocado el presente trabajo es la doméstica, la cual incorpora el monitoreo en el hogar como se muestra en la figura 6.2.1.

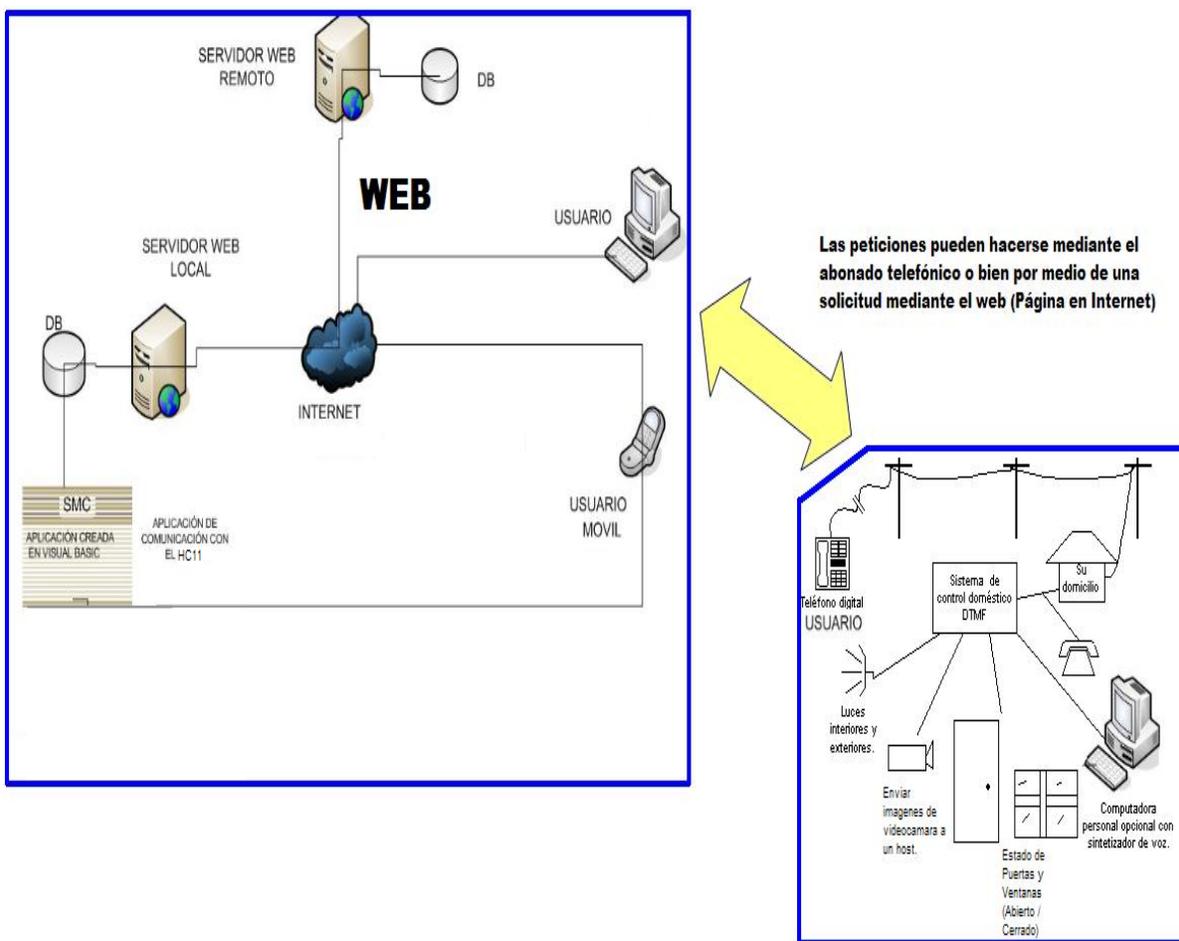


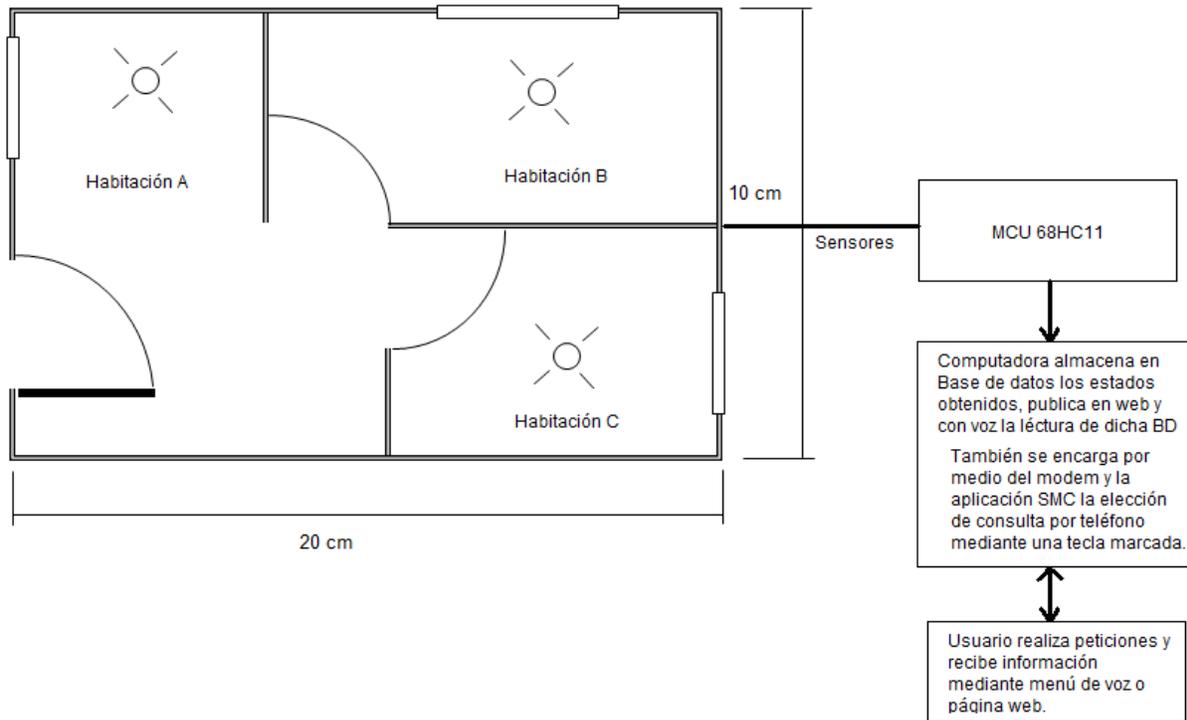
Figura 6.2.1 Diagrama de Uso

En la figura se observa la manera de realizar las peticiones al sistema de control usando DTMF con lo cual podemos enviar señales lógicas para monitorear ventanas, puertas y luces.

En la Interfaz basada en DTMF y el teléfono convencional, se observa que mediante la decodificación de tonos se puede enviar una señal de tono capaz de ser interpretada por un dispositivo microcontrolador para verificar su estado, apagado/encendido en el caso de luces y abierto ó cerrado para el caso de puertas y ventanas y la computadora mediante un menú de voz arrojando los resultados de la consulta.

La Interfaz web quedó compuesta por una PC que conectada a Internet realiza una petición por medio de dos host, uno local conectado también a Internet que realiza la función de almacenar en una BD los datos enviados del Microcontrolador que está recibiendo los datos de los sensores, este a su vez, está comunicado a otro host remoto, el cual por medio de IIS (Servicios de Información en Internet) es el que se encarga de publicar la información en una página y desplegar la información solicitada.

El modelo resultante es una Casa habitación la cual se implementó en una pequeña maqueta con tres habitaciones: "A", "B" y "C", las cuales cuentan con sensores en puertas, ventanas y focos como se muestra en la figura 6.2.2.

**Figura 6.2.2 Maqueta de la casa**

### **6.3. Conclusiones**

Como pudimos observar existen diferentes formas de obtener un resultado o cumplir con un objetivo. Cuando iniciamos este proyecto teníamos planteado su desarrollo con otra configuración y diferentes elementos, pero a medida que íbamos realizando la implementación y el armado del hardware surgieron algunas dificultades en la integración por ende ciertos módulos no funcionaban correctamente y nos vimos en la necesidad de buscar otras alternativas para llegar a la solución.

Probablemente el mayor cambio realizado en este proyecto fue el caso de utilizar el modem con el que ya cuenta la PC en lugar de utilizar un circuito integrado para poder identificar y decodificar los tonos DTMF, de esta forma se resolvió la problemática del ruido en la transmisión de tonos.

Con esto se puede ver que muchas soluciones que anteriormente requerían forzosamente de la implementación de hardware, con el desarrollo y evolución de software encontramos una alternativa al problema planteado originalmente.

Retomando el objetivo del proyecto este fue diseñar un sistema que permitiera monitorear de forma remota una casa habitación vía una línea telefónica o un portal web y dado el avance de las telecomunicaciones y la disminución de sus costos, no es necesario invertir en sistemas de seguridad que impliquen gran infraestructura (partiendo de que el usuario ya cuenta con una PC con conexión a Internet en casa).

Adicionalmente con la conexión a Internet, se pueden integrar aplicaciones en un portal web siendo el proyecto escalable hasta el envío de imágenes por una cámara web

Algo que le da un valor agregado a nuestro proyecto es el servidor web remoto (host), ya que en el caso de que un intruso dañe la PC o corte el suministro de energía eléctrica, esta información fue previamente ya enviada y por lo tanto puede ser consultada en el mismo portal WEB.

### **6.3. Conclusiones**

Como pudimos observar existen diferentes formas de obtener un resultado o cumplir con un objetivo. Cuando iniciamos este proyecto teníamos planteado su desarrollo con otra configuración y diferentes elementos, pero a medida que íbamos realizando la implementación y el armado del hardware surgieron algunas dificultades en la integración por ende ciertos módulos no funcionaban correctamente y nos vimos en la necesidad de buscar otras alternativas para llegar a la solución.

Probablemente el mayor cambio realizado en este proyecto fue el caso de utilizar el modem con el que ya cuenta la PC en lugar de utilizar un circuito integrado para poder identificar y decodificar los tonos DTMF, de esta forma se resolvió la problemática del ruido en la transmisión de tonos.

Con esto se puede ver que muchas soluciones que anteriormente requerían forzosamente de la implementación de hardware, con el desarrollo y evolución de software encontramos una alternativa al problema planteado originalmente.

Retomando el objetivo del proyecto este fue diseñar un sistema que permitiera monitorear de forma remota una casa habitación vía una línea telefónica o un portal web y dado el avance de las telecomunicaciones y la disminución de sus costos, no es necesario invertir en sistemas de seguridad que impliquen gran infraestructura (partiendo de que el usuario ya cuenta con una PC con conexión a Internet en casa).

Conclusiones

---

Adicionalmente con la conexión a Internet, se pueden integrar aplicaciones en un portal web siendo el proyecto escalable hasta el envío de imágenes por una cámara web

Algo que le da un valor agregado a nuestro proyecto es el servidor web remoto (host), ya que en el caso de que un intruso dañe la PC o corte el suministro de energía eléctrica, esta información fue previamente ya enviada y por lo tanto puede ser consultada en el mismo portal WEB.

## Glosario

**Ancho de Banda** Diferencia entre las frecuencias límite de un espectro de frecuencia continuo.

**Bus** Uno o más conductores que sirven para conexión común para un grupo de dispositivos relacionados.

**Ensamblador** Lenguaje de programación que traduce cada sentencia del lenguaje a una instrucción de máquina.

**Microcontrolador** Es dispositivos que dentro del mismo circuito contiene un microprocesador así como periférico comunes (ADC, PWM, o puerto de comunicación serie)

**Microprocesador** Dispositivo Digital que permite manipular datos. Cuentan con una unidad de control, unidad aritmético-lógica y bus de datos, direcciones y control.

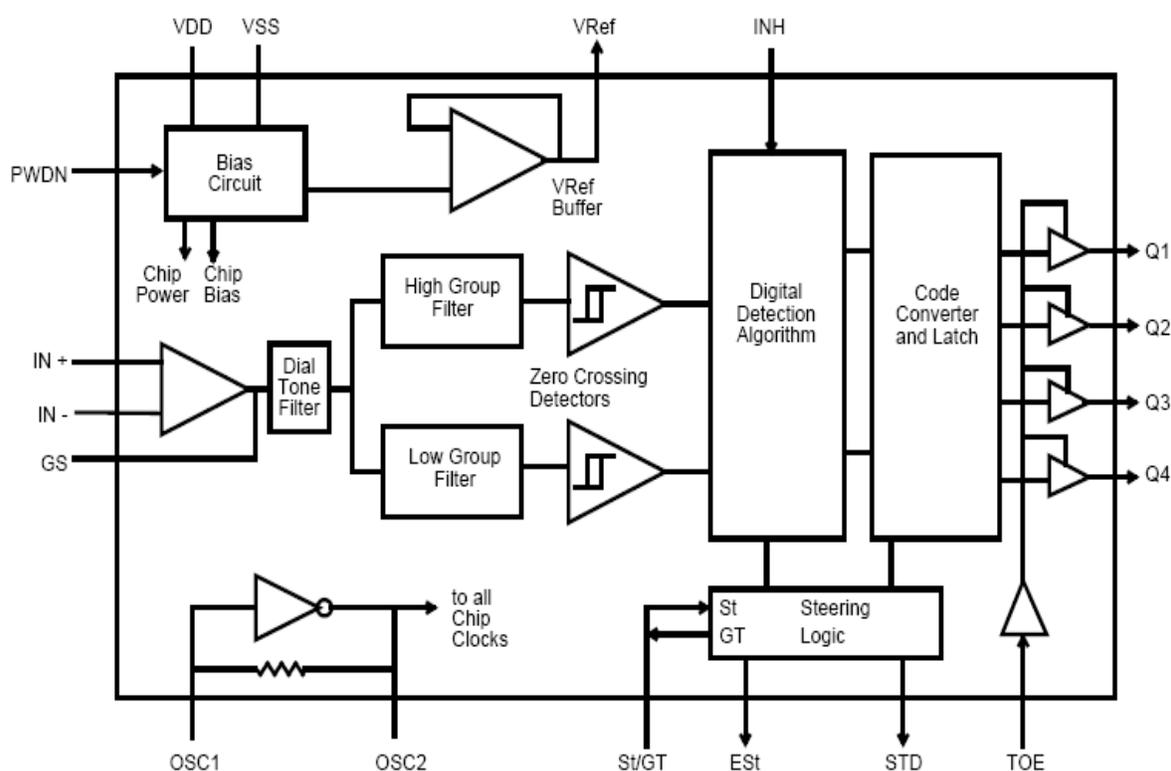
**Modem** (Modulador/Demodulador) Transforma un flujo de bits digitales en una señal analógica (modulador) y viceversa (demodulador)

**Protocolo** Conjunto de reglas que gobiernan la operación de unidades funcionales para llevar a cabo la comunicación.

**Sensor** Es un transductor eléctrico que provee una señal de salida eléctrica como respuesta de una medición específica

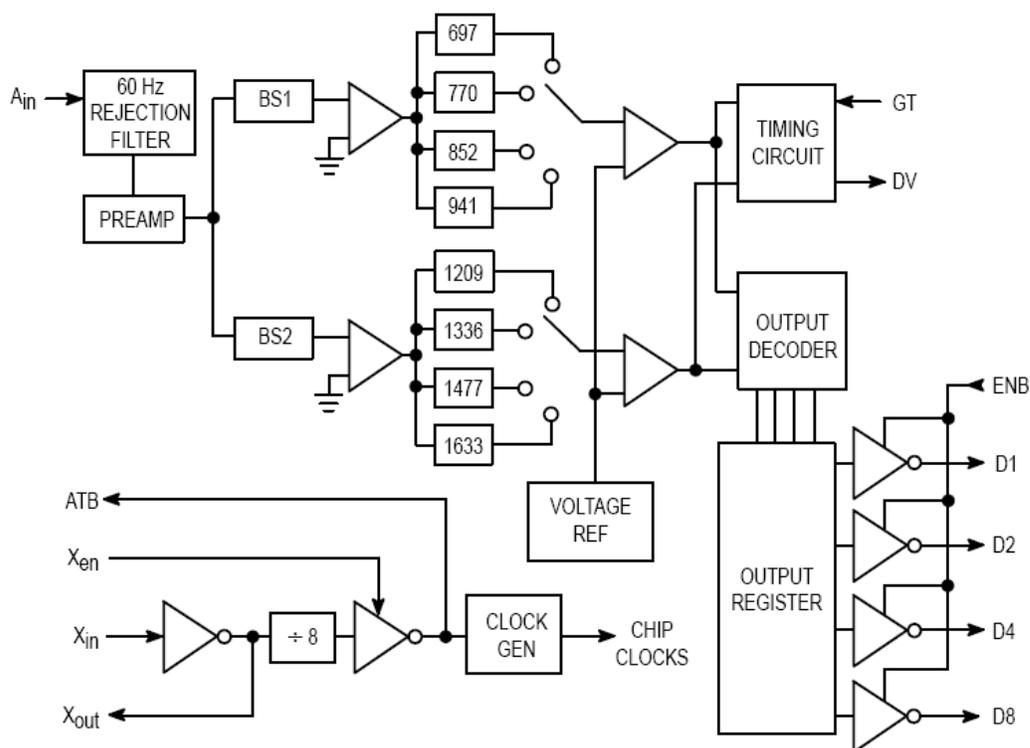
**MC14LC5436P**

El DTMF *MC14LC5436P* por ejemplo (fabricado por Motorola Semiconductor), es un dispositivo con tecnología CMOS LSI, puede trabajar con una polarización de 5V y un cristal comercial de 3.579545 MHz tiene un bajo consumo de energía, una entrada analógica para los tonos DTMF y una salida de cuatro bits, en código binario; presenta una buena tolerancia al ruido y filtrado a la entrada de señales de 60Hz.



**Diagrama de bloques funcionales del MT88L70**

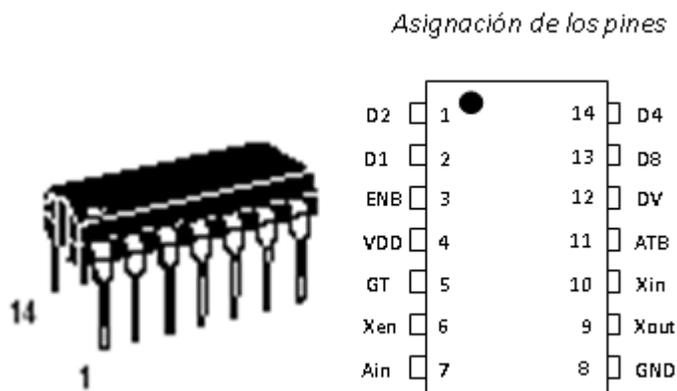
## Diseño de un sistema de Monitoreo y Control para casa habitación basado en DTMF y WEB



**Diagrama de bloques del circuito decodificador. (Cortesía de Motorola Semiconductor Technical Data)**

Las cuatro salidas de este circuito pasan por un latch, lo cual nos permite, mediante el pin ENB habilitar los pines D1, D2, D4 y D8 como salidas, en los cual pueden presentarse estado bajo (+0v), estado alto (+5V) y también estar deshabilitarlos, es decir, tendría un estado de alta impedancia, a esta característica del circuito se le conoce como salida de tres estados (three state).

Primero se realizara una descripción de cada uno de los pines de este circuito, para poder entender la polarización y su conexión.



*Circuito decodificador (Cortesía de Motorola Semiconductor Technical Data)*

### **VDD**

#### **Fuente de alimentación positiva (Pin 4)**

Este pin es conectado a la fuente de corriente directa positiva del sistema, el potencial que puede tener es de 4.5 V a 5.5 V y su valor típico es de 5 V .

### **VSS**

#### **Tierra (Pin 8)**

Se conecta a la tierra (0V) del sistema

### **D1, D2, D4, D8**

#### **Datos de salida (Pines 2, 1, 14, 13 respectivamente)**

Estas salidas digitales proporcionan el código hexadecimal correspondiente al dígito detectado. Las salidas digitales serán validas después de haberse detectado un par de tonos y limpiadas después de un cierto tiempo de ausencia de

detección. Estos pines de salida presentan características de tres estados, cuando el ENABLE está en cero presentan alta impedancia.

## **ATB**

### ***Señal de reloj alternativa (Pin 11)***

Este pin funciona como una referencia de frecuencia cuando más de un MC145436 está siendo usado, así que solamente es necesario un cristal. Cuando este sea el caso, deberán estar conectados entre si por el pin ATB y cuando solo se trabaje con uno, el pin ATB deberá mantenerse desconectado.

## **$A_{in}$**

### ***Entrada analógica (Pin 7 )***

Por este pin se recibe la señal analógica, la cual puede estar acoplada en AC o en DC siempre y cuando no exceda el voltaje de alimentación positivo.

## **$X_{in}/X_{out}$**

### ***Entrada y salida del oscilador (Pines 10 y 9 respectivamente)***

Estos pines son conectados a un oscilador a cristal interno. Para su operación es necesario conectar un cristal en paralelo de  $X_{in}$  a  $X_{out}$ , como también una resistencia de  $1M\Omega$  en paralelo con el cristal. Cuando se utilice la fuente de reloj alternativa del pin ATB , el pin  $X_{in}$  deberá ser puesto a  $V_{DD}$ .

## **ENB**

### ***ENABLE (Pin 3)***

Las salidas D1, D2, D4, D8 son habilitadas cuando ENB está en 1 lógico (5V) y deshabilitadas (alta impedancia) cuando ENB está en 0 lógico (0V).

***X<sub>en</sub> Habilitación del oscilador (Pin 6)***

Un 1 lógico (es decir un voltaje de 5V) en este pin habilita el funcionamiento del circuito con un cristal. Cuando se utilice la fuente de reloj alternativa del pin ATB, el pin X<sub>en</sub> deberá ser puesto a V<sub>SS</sub>.

***DV Dato valido (Pin 12)***

DV señala una detección de un par de tonos DTMF validos, DV se pone en alto en una decodificación valida y permanece así hasta que ocurre la perdida de la señal.

***Diagrama de tiempos***

Otra parte importante de los decodificadores DTMF es su diagrama de tiempos, este nos permite tener una idea más clara de la forma en que trabaja y bajo qué condiciones de tiempo.

En la figura se muestra el mencionado diagrama de tiempos, donde dependiendo de las diferentes entradas analógicas tenemos diferentes intervalos de tiempos en On y Off. Su explicación se muestra en la tabla.

Diseño de un sistema de Monitoreo y Control para casa habitación basado en DTMF y WEB

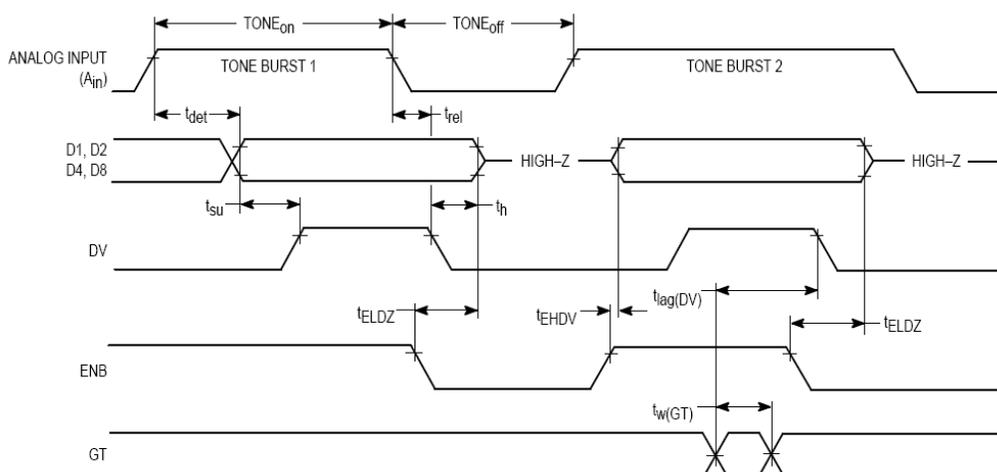


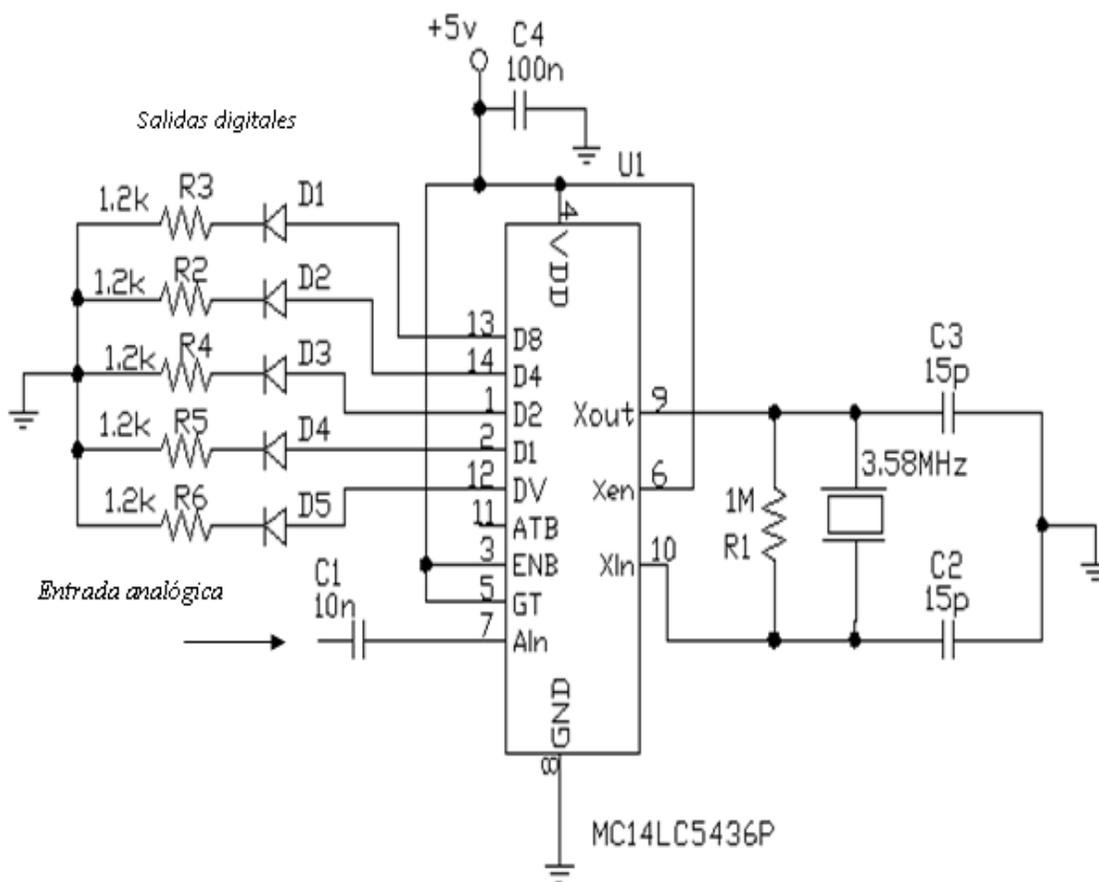
Diagrama de tiempos (Cortesía de Motorola Semiconductor Technical Data)

Característica	Descripción	Mínimo	Típico	Máximo
Tono On	Tiempo del tono presente (Para detección) (Para rechazo)	40ms --	-- --	-- 20ms
Tono Off	Tiempo sin tono (Para detección) (Para rechazo)	40ms --	-- --	-- 20ms
t <sub>su</sub>	Tiempo de establecimiento del dato	7ms	--	--
t <sub>h</sub>	Tiempo de sujeción de dato	4.2ms	4.6ms	5ms
t <sub>ELDZ</sub>	Tiempo en pasar a alta impedancia	--	110ns	300ns
t <sub>det</sub>	Tiempo de detección	32ms	--	50ms
T <sub>rel</sub>	Tiempo de liberación			

**Descripción de los tiempos presentes en el diagrama.**

Diseño de un sistema de Monitoreo y Control para casa habitación basado en DTMF y WEB

En la figura se muestra un circuito con el arreglo necesario para detectar señales DTMF. Está polarizado con un voltaje de +5V, los diodos de señalización para las salidas digitales, la entrada analógica de la línea telefónica y un cristal de 3.58 MHz como oscilador.



Circuito polarizado y sus elementos.

**Bibliografía**

1. Ranz Abad Jesús. *Breve Historia de Internet*. Anaya Multimedia.1997
2. López González Ángel. *Protocolos de Internet*. Alfaomega. México, 2000.
3. Abaurca Velarde Jorge . *Guía Visual de Internet 2008*. Anaya Multimedia. Madrid. 2008.
4. Vicente López Camacho. *Linux Guía de Instalación y Administración. Configuración y Programación de Servidores de Internet*. Mc. Graw Hill. 2001.
5. Valdés Perez Fernando. *Microcontroladores Fundamentos y Aplicaciones con PLC*. Alfaomega, 2007.
6. Orduñas Huerta Juan Manuel. *Arquitectura y Programación de Microcontroladores*. Universidad de Valencia, 1996.
7. Peter Spasow. *Microcontroller Technology The 68HC11*. Prentice Hall
8. Ramón Pallas Areny. *Sensores y Acondicionamiento de Señal*. Marcombo. Barcelona Alfaomega, México. 2007.

9. Harry N. Norton . *Sensores y Analizadores*. Gustavo Gili
  
10. B. A. Gregory. *Instrumentación Eléctrica y Sistemas de Medida*. Gustavo Gili.
  
11. Barrios Romano Jesús. *Introducción a los filtros Digitales*. UAM. México 1992.
  
12. Psenicka, Bohumil. *Procesamiento Digital de Señales, Transformada de Fourier y Filtros Digitales*. UNAM. F.I. 1991.
  
13. Edward W. Kamen. *Introducción a Señales y sistemas*. CECSA.
  
14. Luis Joyanes Aguilar. *Fundamentos de Programación, algoritmos, estructura y Objetos*. Mc Graw Hil

**Acrónimos**

<b>AMPS</b>	Time Division Multiple Access
<b>ASP</b>	Active Server Pages
<b>CDMA</b>	Code Division multiple Acces
<b>CEPT</b>	Conferencia Europea de Administración de Correos y Telecomunicaciones
<b>CMOS</b>	Complementary Metal Oxide Semiconductor
<b>CPU</b>	Unidad central de procesamiento
<b>DNS</b>	Domain Name System
<b>DTMF</b>	Dual Tone MultiFrecuency
<b>HTML</b>	Hypertext markup Language
<b>IANA</b>	Internet Assigned Numbers Authority
<b>IMTS</b>	Improved Mobile Telephone System
<b>ISO</b>	International Organization for Standarization
<b>IVR</b>	Interactive Voice Response <i>Respuesta de Voz Interactiva</i>
<b>NIC</b>	Network Information Center
<b>NMT</b>	Nordic Mobile Telephone
<b>OSI</b>	Open System Interconection
<b>PSTN</b>	Red de Telefonía Pública Conmutada
<b>PWM</b>	Pulse-Width Modulation.
<b>SCI</b>	Serial Communication Interface
<b>SPI</b>	Serial Peripheral Interface
<b>TCP/IP</b>	Transmisión Control Protocol/Internet Protocol
<b>UC</b>	Unidad de Control
<b>UMTS</b>	Universal Mobile Telehone System
<b>URL</b>	Uniform Resource Locator
<b>VPN</b>	Red Privada Virtual
<b>WAN</b>	Wide Area Network