

ESCUELA POLITÉCNICA NACIONAL  
FACULTAD DE INGENIERÍA ELÉCTRICA

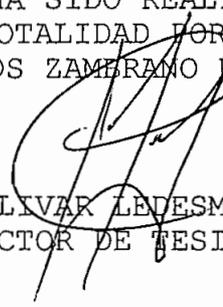
DISEÑO Y CONSTRUCCIÓN DE UN INTERFAZ  
DTMF PARA UN COMPUTADOR PERSONAL

CARLOS EDUARDO ZAMBRANO RHEA

TESIS PREVIA A LA OBTENCIÓN DEL TÍTULO DE  
INGENIERO ELÉCTRICO EN LA ESPECIALIZACIÓN DE  
ELECTRÓNICA Y CONTROL

NOVIEMBRE DE 1996

CERTIFICO QUE EL PRESENTE  
TRABAJO HA SIDO REALIZADO  
EN SU TOTALIDAD POR EL  
SR. CARLOS ZAMBRANO RHEA



ING. BOLIVAR LEDESMA G.  
DIRECTOR DE TESIS

AGRADECIMIENTO

AL ING. BOLIVAR LEDESMA  
POR SU AYUDA EN LA REALIZACION  
DE ESTA TESIS

DEDICATORIA

A MI MADRE

# DISEÑO Y CONSTRUCCIÓN DE UN INTERFAZ

## DTMF PARA UN COMPUTADOR PERSONAL

### ÍNDICE

	PÁGINA
INTRODUCCIÓN	1
CAPÍTULO 1	
DISEÑO DEL HARDWARE	
1.1 REQUERIMIENTOS	3
1.2 DIAGRAMA DE BLOQUES DEL SISTEMA PROPUESTO	4
1.2.1 RECEPTOR DE DOBLE TONO MULTI-FRECUENCIAL	6
1.2.1.1 DESCRIPCIÓN GENERAL DE OPERACIÓN DEL CI 273-1303	6
1.2.1.2 DESCRIPCIÓN DE PINES	10
1.2.2 CHIP GRABADOR/REPRODUCTOR DE VOZ	13
1.2.2.1 DESCRIPCIÓN INTERNA DEL CI ISD1000A	15
1.2.2.2 DIRECCIONAMIENTO BÁSICO	18
1.2.2.3 MODOS DE OPERACIÓN	19
1.2.2.4 UTILIZACIÓN DEL ISD1000A	22
1.2.2.5 DIAGRAMAS DE TIEMPO PARA LA GRABACIÓN Y REPRODUCCIÓN	23
1.2.3 CIRCUITOS DE INTERFAZ CON LA LÍNEA TELEFÓNICA	25
1.2.3.1 INTERFAZ DETECTORA DE DATOS DTMF	25
1.2.3.2 INTERFAZ DETECTORA DE TIMBRADAS	28
1.2.3.3 INTERFAZ DEL CI GRABADOR /REPRODUCTOR DE VOZ	32
1.2.4 CIRCUITO DE INTERFAZ CON EL COMPUTADOR	33
1.2.5 CIRCUITO COLGADOR/DESCOLGADOR DE LA LÍNEA TELEFÓNICA	36
1.3 DIAGRAMA CIRCUITAL COMPLETO	37

## CAPÍTULO 2

### DISEÑO DEL SOFTWARE

2.1	REQUERIMIENTOS	39
2.2	DESCRIPCIÓN DEL PROGRAMA DEL MICROCONTROLADOR	42
2.2.1	FUNCIONES BÁSICAS DEL PROGRAMA	42
2.2.2	DIAGRAMA DE BLOQUES DEL PROGRAMA DEL MICROCONTROLADOR	48
2.3	PROGRAMA PARA EL COMPUTADOR PERSONAL	55
2.3.1	PANEL DEL PROGRAMA DEL COMPUTADOR PERSONAL	56
2.3.1.1	CONTROLES PARA LA GRABACIÓN	57
2.3.1.2	CONTROLES PARA LA REPRODUCCIÓN MANUAL	57
2.3.1.3	CONTROLES PARA EL MANEJO DEL ARCHIVO DE DIRECCIONES DE MENSAJE	58
2.3.1.4	CONTROLES PARA EL MANEJO DEL ARCHIVO DE CLAVES VÁLIDAS	58
2.3.1.5	ARCHIVO DE ÓRDENES RECIBIDAS	58
2.3.1.6	SELECTOR ENTRE MODO AUTOMÁTICO Y MANUAL DE REPRODUCCIÓN	58
2.3.1.7	INTERRUPTOR DEL PARLANTE	59
2.3.1.8	SELECTOR DEL NÚMERO DE TIMBRADAS	59
2.3.1.9	BOTÓN PARA LA INICIALIZACIÓN DE LAS DIRECCIONES DE MENSAJE	59
2.3.1.10	CONTROLES PARA LA SELECCIÓN DEL PÓRTICO	59
2.3.2	DIAGRAMA DEL PROGRAMA DEL COMPUTADOR PERSONAL	59
2.3.2.1	INICIALIZACIÓN DEL PÓRTICO SERIAL	60
2.3.2.2	REPRODUCCIÓN MANUAL	61
2.3.2.3	GUARDAR LA INFORMACIÓN DE UN "ARRAY" DE "CLUSTERS" EN UN ARCHIVO	62
2.3.2.4	RECUPERAR LA INFORMACIÓN DE UN ARCHIVO A UN "ARRAY" DE "CLUSTERS"	64

2.3.2.5 TRANSFERENCIA DE DIRECCIONES AL MICROCONTROLADOR	65
2.3.2.6 INICIALIZACIÓN DEL NÚMERO DE TIMBRADAS Y CONTROL DEL PARLANTE	66
2.3.2.7 GRABACIÓN DE MENSAJES EN EL CIRCUITO GRABADOR/REPRODUCTOR DE VOZ	67
2.3.2.8 REPRODUCCIÓN AUTOMÁTICA	68
2.3.3 DIAGRAMA DE JERARQUÍAS	71

## CAPÍTULO 3

### PRUEBAS Y RESULTADOS

3.1 RESULTADOS	73
3.1.1 CARACTERÍSTICAS CONSTRUCTIVAS	73
3.1.2 PRESUPUESTO	74
3.2 PRUEBAS	80
3.2.1 GRABACIÓN DE MENSAJES	80
3.2.2 CONTROLES PARA LA REPRODUCCIÓN MANUAL	82
3.2.3 CONTROLES PARA LA INICIALIZACIÓN DE DIRECCIONES	83
3.2.4 ARCHIVO DE CLAVES Y PERSONAS AUTORIZADAS	83
3.2.5 ARCHIVO DE ÓRDENES RECIBIDAS	83
3.2.6 MODO AUTOMÁTICO DE REPRODUCCIÓN	84
3.2.7 INTERRUPTOR DEL PARLANTE	84
3.2.8 SELECTOR DEL NÚMERO DE TIMBRADAS	85
3.2.9 EJEMPLO DE FUNCIONAMIENTO DEL SISTEMA	85

## CAPÍTULO 4

### CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES	87
4.2 RECOMENDACIONES	90

ANEXOS	93
--------	----

# INTRODUCCIÓN

Esta tesis trata acerca del diseño y construcción de un equipo que permitirá al computador obtener instrucciones desde una línea telefónica a través de el teclado de un teléfono de tonos. Se ha enfocado el funcionamiento del equipo de tal manera que sea interactivo con el usuario, es decir, el equipo le indicará verbalmente a través de la línea telefónica las acciones que debe seguir el usuario para la correcta comunicación con el computador.

El primer capítulo trata sobre el diseño del hardware que conforma el Interfaz DTMF. En primer lugar se plantean los requerimientos y características que deberá cumplir el sistema y en base a ésto se plantea un diagrama de bloques que expresa las relaciones que deberían existir entre los distintos elementos que conformarán el equipo a diseñarse y construirse. Luego de haber definido este diagrama de bloques se explica el esquema de funcionamiento que seguirá el sistema y se describe cada uno de los elementos que lo conforman , abarcando su estructura interna, sus modos de operación , principio de funcionamiento y formas de conexión. En la parte final de este capítulo se realizan los cálculos que permiten determinar los elementos que conforman los circuitos de interfaz con la línea telefónica.

El segundo capítulo trata acerca del diseño del software que permite operar al equipo. El capítulo se divide en dos partes, la primera contempla la programación del software para el microcontrolador, mientras que la segunda trata acerca de la programación del computador con el programa Labview. En cada una de las partes se empieza analizando las funciones que

debería cumplir el programa. A continuación se explica cada uno de los pasos que se siguen para cumplir con estas funciones. Para explicar el programa en Assembler se utilizan bloques que explican el funcionamiento de las subrutinas. El programa en Labview no se puede expresar en un flujograma debido a que casi todas las instrucciones se ejecutan en paralelo. En lugar del flujograma se describirán cada una de las partes principales que conforman el diagrama de programación.

El tercer capítulo trata acerca de las pruebas y resultados realizadas con el equipo. En la parte de resultados se analizan las características físicas del equipo y el costo de fabricación del mismo. En la parte de pruebas se analizan los procedimientos que deben seguirse para utilizar el programa desde el computador y desde el teclado de un teléfono de tonos. Por último se muestra un ejemplo de utilización del equipo en el que se describen las acciones que este tomará, en base a los datos ingresados desde el teclado de un teléfono de tonos.

En el cuarto capítulo se plantean las conclusiones acerca del trabajo realizado y por último se dan recomendaciones para el uso y mejoramiento de este equipo.

En los anexos se incluyen las hojas de datos de los elementos especiales dentro del equipo, como el receptor DTMF y el grabador/reproductor de voz. También se incluye el listado del programa en ensamblador para el microcontrolador y el listado de un programa en Qbasic que fue realizado inicialmente para probar el equipo y que pudiera ser de utilidad en el caso de querer usar el Interfaz DTMF con un computador que no permita correr el programa realizado en LabView .

# CAPÍTULO 1. DISEÑO DEL HARDWARE

## 1.1 REQUERIMIENTOS

A continuación se describen las características de funcionamiento que se quieren conseguir:

Para acceder al servicio del computador, el usuario deberá marcar el número telefónico al cual esté conectado el equipo, el cual contestará luego de que detecte un número de timbradas definido previamente desde el computador. A continuación el equipo le indicará verbalmente los datos que deberá ingresar desde el teclado del teléfono, y le repetirá las instrucciones recibidas para que el usuario las confirme o las cancele. A continuación el equipo le dará la opción de ingresar más instrucciones o terminar la comunicación; en este caso el equipo colgará. El computador permite cumplir con los siguientes objetivos: Ejecutar las órdenes recibidas desde el teclado telefónico, instruir al equipo algunas de las respuestas que debe dar, permitir grabar los mensajes bases y efectuar la programación de los mismos, permitir programar el número de timbradas antes de descolgar la línea telefónica y conectar o desconectar el parlante externo. Además debe llevar un registro de las órdenes recibidas con hora, fecha, y clave.

El equipo poseerá una entrada de 110V, otra para la línea telefónica y otra para el pórtico serial. También tendrá LEDs indicadores que permitirán monitorear visualmente que se ha descolgado la línea para atender una llamada y también

monitorear la entrada de datos DTMF ("Dual Tone Multi Frequency").

## 1.2 DIAGRAMA DE BLOQUES DEL SISTEMA PROPUESTO:

Para cumplir con las características de este interfaz DTMF, se ha planteado la creación de un sistema que base su funcionamiento en el diagrama de bloques de la figura 1 que se muestra en la siguiente hoja.

En base a este diagrama de bloques se deberá poseer los siguientes elementos principales para implementar el sistema:

- receptor DTMF
- microcontrolador
- computador
- circuito Integrado grabador/reproductor de voz
- circuitos de interfaz con la línea telefónica
- micrófono y parlante

En el diagrama de bloques se muestra que el interfaz con la línea telefónica está constituido de cuatro partes: la que detecta el número de timbradas, la que permite ingresar los datos DTMF, la que conecta la línea telefónica con el grabador/reproductor de voz y la que cuelga y descuelga la línea telefónica. El microcontrolador tiene como entradas los siguientes elementos:

- el interfaz detector de timbradas
- la línea de EOM que sale desde el circuito grabador /reproductor de voz, que se utiliza para indicar al

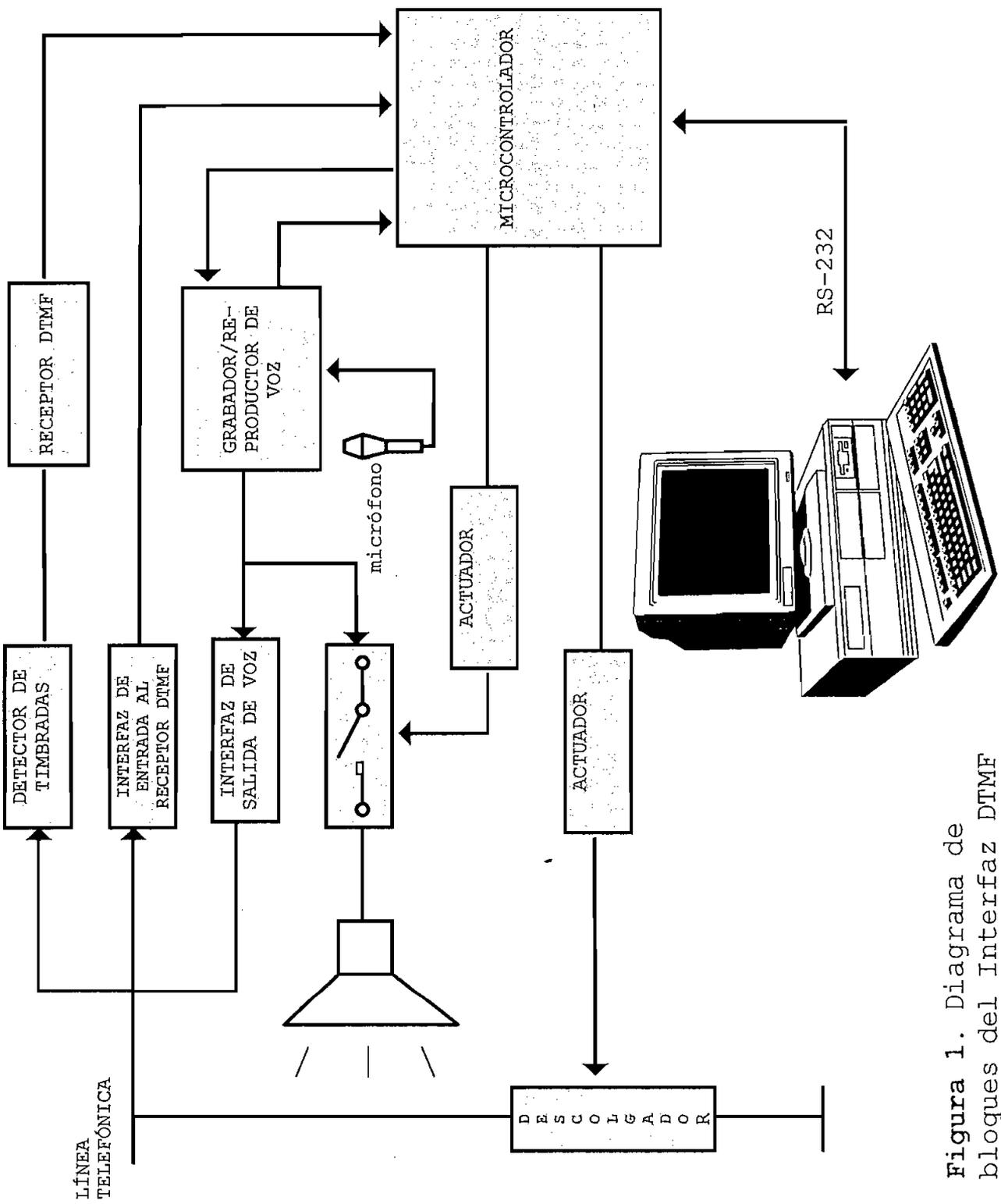


Figura 1. Diagrama de bloques del Interfaz DTMF

microcontrolador cuando el circuito grabador/reproductor de voz ha terminado de pronunciar un mensaje

- las líneas de datos del receptor DTMF
- la línea de transmisión serial del computador

Las líneas de salida del microcontrolador serán las siguientes:

- líneas de control y direccionamiento del grabador/reproductor de voz
- una línea que le permite colgar y descolgar la línea telefónica
- una línea que le permite conectar y desconectar el parlante
- la línea de recepción serial del computador.

A continuación se describen cada uno de los elementos constitutivos del hardware del sistema:

### **1.2.1 RECEPTOR DE DOBLE TONO MULTI-FRECUENCIAL (DTMF)**

El objetivo de un receptor DTMF es detectar la presencia de un par de tonos válidos en una línea telefónica u otro medio de transmisión.

La presencia de un par de tonos válidos indica un dígito presionado. Para generar una secuencia válida del dígito, cada par de tonos debe estar separado por una pausa válida.

#### **1.2.1.1 DESCRIPCIÓN GENERAL DE OPERACIÓN DEL CI 273-1303:**

La siguiente tabla indica el sistema Bell estandarizado para un par de tonos y pausa válidos.

TONOS DEL GRUPO BAJO	697 o 770 o 852 o 941 Hz
TONOS DEL GRUPO ALTO	1209 o 1336 o 1477 o 1633 Hz
TOLERANCIA DE FRECUENCIA	$f_o(1.5\%+2\text{Hz})$
RANGO DE AMPLITUD	$-24 \text{ dBm} < A < 6\text{dBm} @ 600 \text{ Ohm}$
AMPLITUD RELATIVA	$-8 \text{ dB} < \text{Tono del grupo ALTO/BAJO} < +4\text{dB}$
DURACIÓN	mayor o igual a 40 ms
PAUSAS ENTRE TONOS	mayor o igual a 40 ms

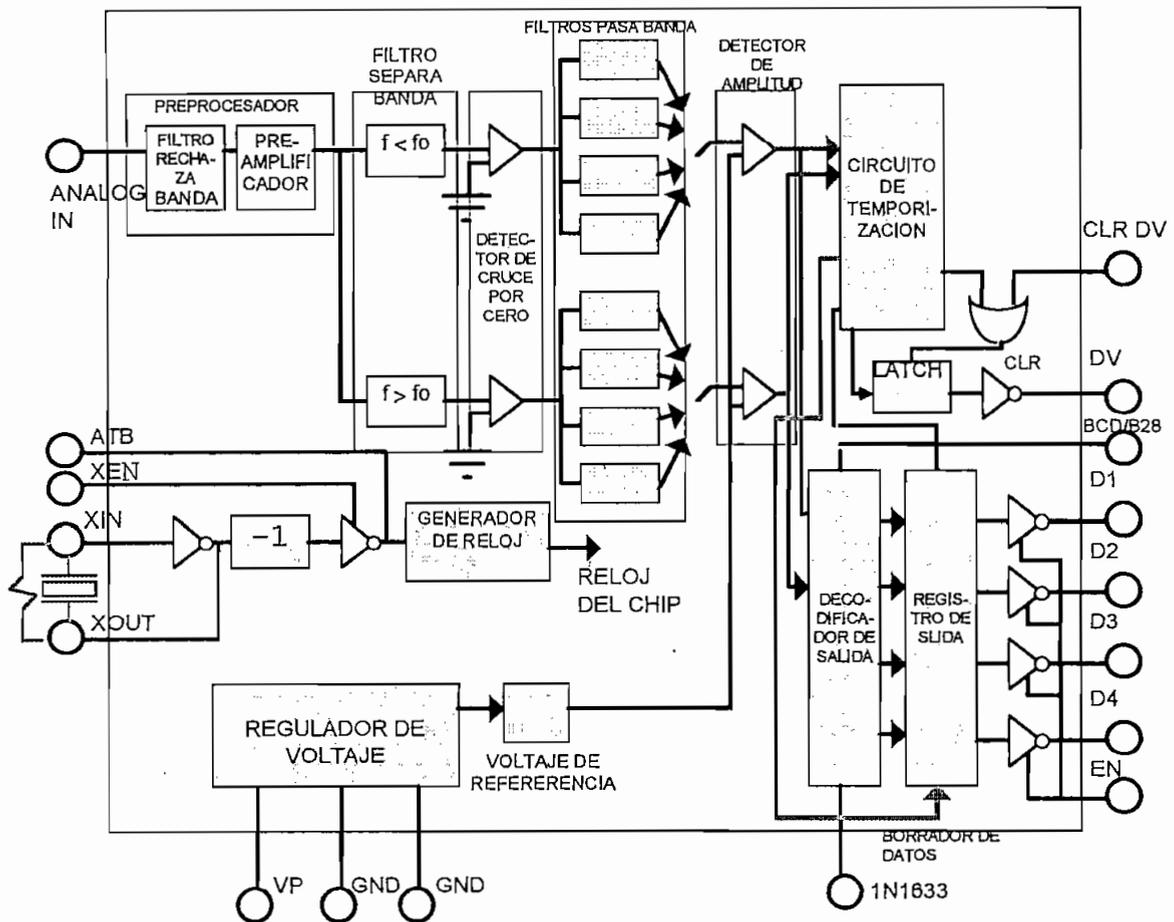
**Tabla 1.** Sistema Bell para la determinación de un par de tonos válidos

Los dos problemas más comunes que suele tener un receptor DTMF son los siguientes:

- a) Distinguir entre señales válidas que posean dos tonos válidos y señales de voz por ejemplo.
- b) Detectar un par de tonos válidos en la presencia de ruido.

El circuito integrado 273-1303 utiliza algunas técnicas para poder distinguir entre pares de tonos válidos, las cuales se describen a continuación:

- Prefiltrado de la señal de audio. Remueve el ruido de la fuente y el tono de marcar de la señal de audio de entrada y enfatiza el dominio de frecuencia de voz.
- Detección de cruce por cero. Limita el nivel de ruido aceptable en la detección de un par de tonos. Importante para el rechazo de voz.
- Muestreado de un par de tonos o pausa válido. Muestra el filtro detector de tonos y chequea por consistencia antes de determinar que el par de tonos o pausa recibidos son válidos.
- Este circuito permite detectar un grupo de 12 o 16 dígitos estándar, sin la necesidad de filtrado previo. Los únicos elementos externos que se necesitan para que el circuito opere son los siguientes: un cristal de 3.58 MHz, y una resistencia de polarización. El circuito integrado ARCHER 276-1313 requiere únicamente una fuente de polarización de baja tolerancia, posee 18 pines y combina funciones analógicas y digitales dentro del mismo encapsulado. La entrada analógica está inicialmente integrada por un filtro rechaza banda de 60 Hz, a continuación se encuentra un control automático de ganancia (AGC) y por último ocho filtros pasa banda para detectar los tonos individuales. La parte digital mide la duración de los tonos y determina la salida digital correcta. A continuación se muestra un diagrama de bloques de este circuito integrado.



**Figura 2.** Diagrama de bloques del receptor DTMF 276-1303

El esquema básico de detección que se muestra en la figura 2 se puede describir como sigue:

La señal de entrada es prefiltrada y luego separada en dos bandas, cada una de las cuales contiene un solo grupo de tonos DTMF. La salida de cada filtro es amplificada y limitada por un detector de cruce por cero. Las señales limitadas cuadradas pasan a través de filtros pasa banda. Una lógica digital es utilizada a continuación para proveer un muestreo del detector para determinar la validez de la detección, también para presentar la salida en el formato correcto y por último para sincronismo y control.

### 1.2.1.2 DESCRIPCIÓN DE PINES:

El circuito integrado 276-1303 posee los siguientes pines con las funciones que se describen a continuación:

#### **ANALOG IN (PIN 9):**

Este pin permite ingresar la señal a analizarse que puede poseer componentes de continua mientras no sobrepase el voltaje de polarización. Este circuito integrado está diseñado para aceptar señales sinusoidales, sin embargo acepta señales distorsionadas con armónicos de hasta 20 dB menos que la fundamental.

#### **OSCILADOR DE CRISTAL (PINES 8,11,12 Y 13):**

El CI receptor DTMF utiliza un cristal o un oscilador externo para su funcionamiento. La utilización del oscilador de cristal se habilita con el pin 8 (XEN) en alto. El cristal es conectado entre XIN y XOUT en paralelo con una resistencia de 1Mohm. Conectado como se indica, el pin 13 (ATB) se convierte en una señal auxiliar de reloj que puede servir a otros circuitos que se configuren para utilizar en vez de un cristal, una entrada externa de reloj. Si se utilizan otros circuitos receptores DTMF esta salida de reloj (ATB) del primer circuito receptor DTMF se puede conectar a las entradas ATB de los otros circuitos receptores DTMF que se hayan adicionado. Los pines 13 de los CI receptores DTMF adicionales, ya no se configuran como salidas auxiliares de reloj, sino como entradas para un oscilador externo. Las entradas XIN y XOUT deben ser conectadas a VCC y GND

respectivamente en los circuitos DTMF que no posean un cristal propio. Hasta 10 de estos circuitos pueden ser conectados en cascada sin requerirse cristales adicionales.

**HEX/B28 (pin2) :**

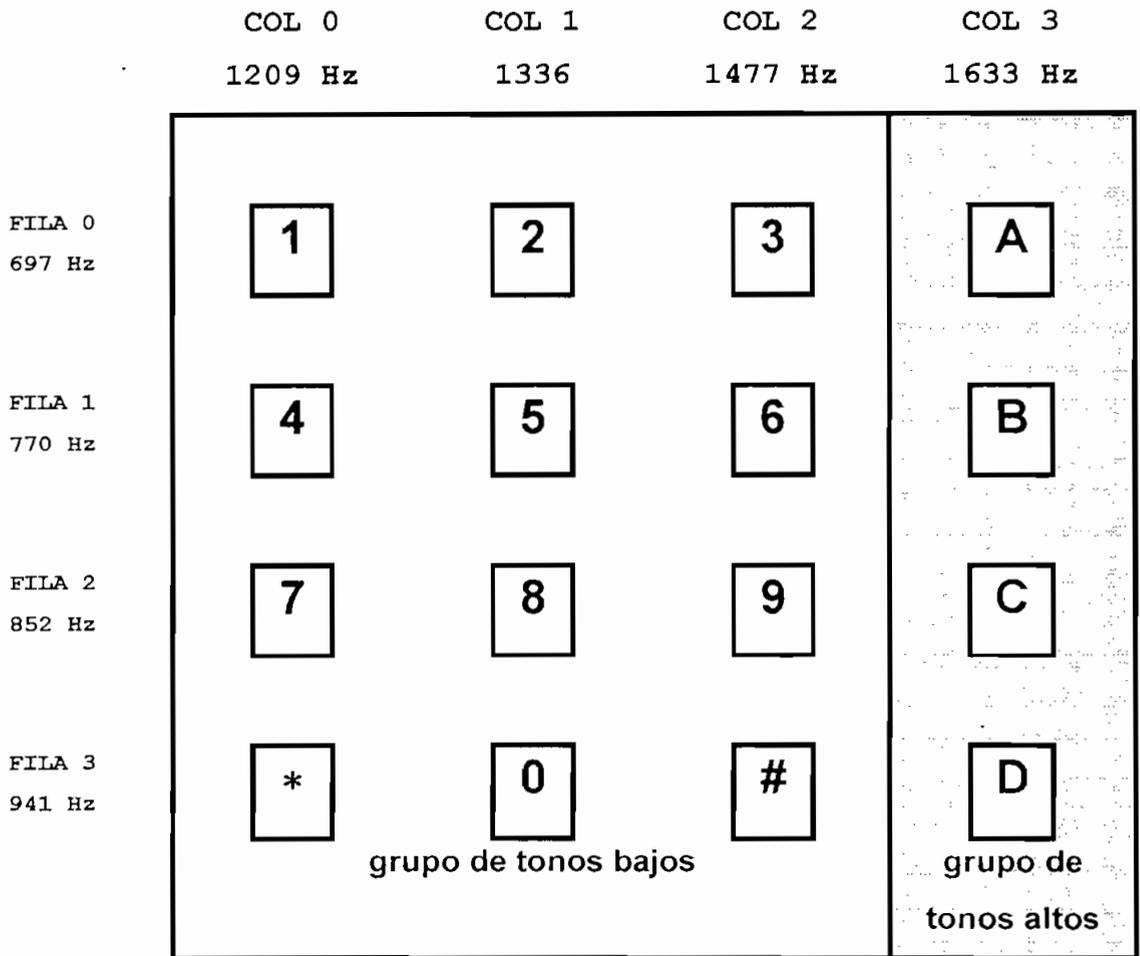
Este pin selecciona el formato del código de salida digital. Cuando HEX/B28 está en alto la salida es hexadecimal. Cuando está atado a bajo, la salida es código "2 de 8", la siguiente tabla indica los dos formatos mencionados.

Dígito	Hexadecimal				Código binario 2 de 8			
	D8	D4	D2	D1	D8	D4	D2	D1
1	0	0	0	1	0	0	0	0
0	0	0	1	0	0	0	0	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	0	0
5	0	1	0	1	0	1	0	1
6	0	1	1	0	0	1	1	0
7	0	1	1	1	1	0	0	0
8	1	0	0	0	1	0	0	1
9	1	0	0	1	1	0	1	0
0	1	0	1	0	1	1	0	1
*	1	0	1	1	1	1	0	0
#	1	1	0	0	1	1	1	0
A	1	1	0	1	0	0	1	1
B	1	1	1	0	0	1	1	1
C	1	1	1	1	1	0	1	1
D	0	0	0	0	1	1	1	1

**Tabla 2.** Códigos utilizados para representar la detección de tonos dobles mediante el CI 273-1303:

**1N1633 (PIN 4) :**

Cuando este pin está atado a VCC, se inhibe la detección de los componentes de 1633Hz, es decir, los tonos adicionales para la detección de una columna extra de 4 teclas como se muestra en la figura 3. Para la detección de 16 dígitos estándar 1N1633 debe estar conectado a GND.



**figura 3.** Distribución del teclado matricial y su relación con su grupo DTMF.

**SALIDAS D1,D2,D4,D8,EN (PINES 1,18,17,16,13) :**

Cuando el pin EN se encuentra en alto las salidas D1-D8 son del tipo push-pull compatibles con CMOS, mientras que cuando EN es bajo, las salidas se ponen en alta impedancia. Estas salidas cambian su formato en base al pin HEX/B28 ya descrito anteriormente. Las salidas se vuelven válidas luego que el par de tonos son detectados o luego de un tiempo válido de pausa.

#### **DV Y CLRDV (PINES 14 Y 15) :**

La salida DV se pone en alto cuando se detecta un par de tonos válidos y las salidas de datos se activan con el código correspondiente. DV permanece en alto hasta que una pausa válida ocurra o que el pin de CLRDV se ponga en alto, teniendo prioridad, lo que ocurra primero.

#### **N/C (PIN 6) :**

Este pin no posee conexiones internas y debe permanecer flotante.

### **1.2.2 CHIP GRABADOR/REPRODUCTOR DE VOZ**

El ISD es un sistema muestreador de datos analógicos con un preamplificador del micrófono incluido en el chip, además posee un control automático de ganancia AGC, filtros pasa bajos y "anti-aliasing", arreglo de almacenamiento, manejador del parlante, interfaz de control y reloj de referencia interno. La diferencia entre un ISD1000A y otras tecnologías de almacenamiento de voz es que éste no requiere conversores A/D o D/A. A medida que el ISD1000A toma cada muestra éste la almacena temporalmente en un circuito "sample and hold" que

eventualmente graba este dato analógico en una sola celda EEPROM . Ya que la grabación está realizada a un equivalente de precisión de 8 bits, ésta requeriría una memoria de 1 Mbit de memoria digital así como un conversor A/D y uno D/A para igualar las 128000 celdas en un arreglo de almacenamiento del ISD1000A. El ISD1000A guarda 20 segundos de sonido a una frecuencia de muestreo de 6.4 kHz. Esto se puede comprobar en base a los siguientes cálculos:

bits de memoria digital = tiempo total de grabado x  
frecuencia de muestreo x bits por muestra

bits de memoria digital = 20s x 6400 muestras/s x 8  
bits/muestra

bits de memoria digital = 1024000 bits

celdas de memoria analógica = segundos de grabación x número  
de muestras por segundo x  
celdas por muestra

celdas de memoria analógica = 20s x 6400 muestras/s x 1  
celda/muestra

celdas de memoria analógica = 128000 celdas

Al muestrear una onda sinusoidal, se requieren tomar solo dos puntos por período, para permitir identificarla. En base a estos puntos se podrá reconstruirla nuevamente en amplitud y frecuencia. Sin embargo, por estos dos puntos en que se tomó el muestreo, pudo haber pasado una sinusoidal con una frecuencia múltiplo de la muestra tomada y se podría confundirla con la original, a esto se llama efecto "aliasing". Al muestrear a 6.4kHz máximo se podría identificar ondas sinusoidales de 3.2 kHz a dos muestras por período, pero se podría confundir muestras de 6400 Hz con

muestras de 3200 Hz. Para evitar esto se debería filtrar el sonido a grabarse, idealmente a frecuencias menores a 3.2 kHz

Los filtros internos realizan el "anti-aliasing" durante la grabación y suavizan durante la reproducción, la cual es necesaria para un buen sistema de sonido muestreado. Estos filtros están fijados a 2700 Hz, lo cual define la máxima frecuencia de corte del sistema, es decir, 3dB menos que 3200Hz.

#### **1.2.2.1 DESCRIPCIÓN INTERNA DEL C.I. ISD1000A**

Este circuito integrado utiliza la tecnología de grabado analógico directo "DAST" que provee información analógica en su memoria de semiconductor. Esta EEPROM escribe información analógica en una celda individual sin conversión A/D o D/A . Esto resulta en un incremento de densidad con respecto a métodos digitales equivalentes, y almacenamiento no volátil de información analógica.

Se requieren únicamente unos pocos elementos externos como, micrófono, parlante, interruptores y unas pocas resistencias y capacitores además de la fuente, para poder grabar y reproducir la información. Los demás elementos tales como preamplificador, filtros, AGC, amplificador de potencia, lógica de control y almacenamiento analógico se encuentran dentro del chip.

En la siguiente figura se muestra el diagrama de bloques del CI ISD1000A.

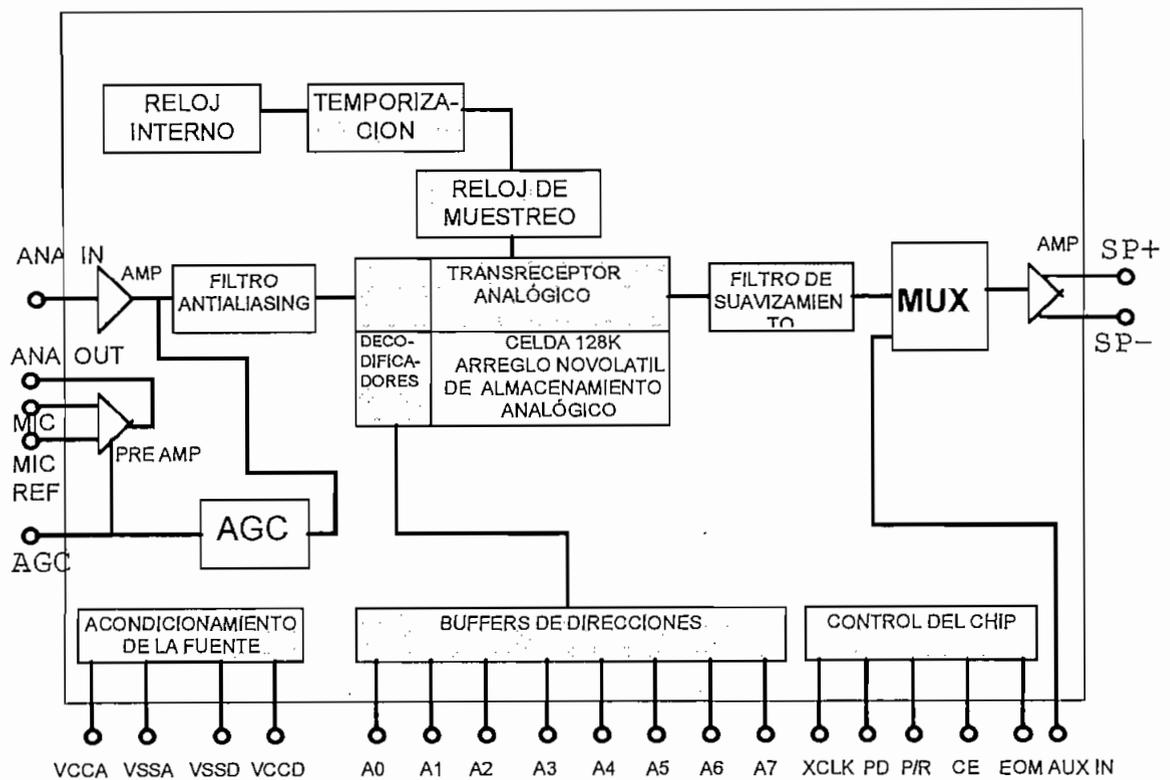


Figura 4. Diagrama funcional del CI ISD1000A

A continuación se presenta la descripción de los pines de el CI ISD1000A:

pin No.	Nombre	Función
1, 2, 3, 4, 5, 6, 9, 10	A0-A7	DEFINE LA DIRECCIÓN BASE PARA MENSAJE O ESTABLECE EL CHIP PARA UN MODO DE OPERACIÓN
7, 8, 22	NC	SIN CONEXIÓN
11	AUX IN	CUANDO ESTÁ EN EL MODO DE REPRODUCCIÓN Y SIN REPRODUCIR UN MENSAJE, ESTA ENTRADA MANEJA LA SALIDA DEL PARLANTE . ESTO PERMITE A OTROS CIRCUITOS USAR EL MANEJADOR DEL PARLANTE. TAMBIÉN SE LA UTILIZA PARA APLICACIONES EN CASCADA

12, 13	V <sub>SSD</sub> , V <sub>SS</sub> A	TIERRAS ANALÓGICAS Y DIGITALES
14, 15	SP+, SP -	ESTOS PINES SE CONECTAN DIRECTAMENTE A UN PARLANTE DE 16 OHM SIN CAPACITORES DE ACOPLAMIENTO
16, 28	V <sub>CCD</sub> , V <sub>CC</sub> A	FUENTES ANALÓGICAS Y DIGITALES. EL EQUIPO UTILIZA BUSES SEPARADOS PARA LA PARTE DIGITAL Y ANALÓGICA PARA MINIMIZAR RUIDO
17	MIC	EL PREAMPLIFICADOR DEL MICRÓFONO PUEDE AMPLIFICAR SEÑALES ENTRE 1 Y 20 mV, POSEE UNA IMPEDANCIA DE ENTRADA DE 10kOHM 24 dB DE GANANCIA MÁXIMA
18	MIC REF	ESTA ES LA ENTRADA INVERSORA DEL PREAMPLIFICADOR DEL MICRÓFONO. ESTE PROVEE UNA ENTRADA QUE RECHAZA EN MODO COMÚN EL INGRESO DE RUIDO. NORMALMENTE SE DEBE CONECTAR UN CAPACITOR ENTRE ESTE PIN A LA TIERRA ANALÓGICA. ESTE CAPACITOR DEBE SER DEL MISMO VALOR QUE EL CAPACITOR DE ACOPLAMIENTO DE ENTRADA.
19	AGC	SE DEBE CONECTAR UNA RESISTENCIA DE 470 OHM Y UN CAPACITOR DE 4,7 uF ENTRE ESTE PIN Y TIERRA. ESTOS COMPONENTES REGULAN LAS CONSTANTES DE TIEMPO DEL CIRCUITO DE GANANCIA AUTOMÁTICA DEL PREAMPLIFICADOR DEL MICRÓFONO
20	ANA IN	ESTE PIN VA A UN AMPLIFICADOR DE ENTRADA QUE MANEJA EL ARREGLO DE ALMACENAMIENTO INTERNO
21	ANA OUT	ESTE PIN ES LA SALIDA DIRECTA DEL PREAMPLIFICADOR DEL MICRÓFONO
23	CE	ESTE PIN EMPIEZA LA REPRODUCCIÓN Y CONTROLA LAS OPERACIONES DE GRABADO.
24	PD	ESTE PIN CONTROLA EL MODO DE APAGADO Y RESETEA

		LA MEMORIA LUEGO QUE EL DISPOSITIVO HA REPRODUCIDO O GRABADO HASTA EL FINAL DE LA MEMORIA. CHIP SE APAGA CUANDO PD ESTÁ EN ALTO.
25	EOM	ESTE PIN SACA UN PULSO BAJO DE 16 mS PARA INDICAR QUE UN MENSAJE HA TERMINADO Y SE MANTIENE EN BAJO EN EL CASO QUE LA MEMORIA ESTÉ LLENA.
26	XCLK	NORMALMENTE CONECTADO A TIERRA. ESTO HABILITA AL RELOJ INTERNO. SE PUEDE MANEJAR ESTA ENTRADA CON UN RELOJ DE NIVELES TTL DE 819.1 MHz
27	P/R	EL ESTADO DE ESTE PIN CUANDO "CE" TIENE UN FLANCO NEGATIVO DETERMINA QUE EL CHIP SE ENCUENTRE EN EL MODO DE GRABACIÓN O DE REPRODUCCIÓN.

**Tabla 3.** Descripción de pines del CI ISD1000A

#### 1.2.2.2 DIRECCIONAMIENTO BÁSICO:

Se puede direccionar múltiples mensajes en un ISD1000A. Las entradas de dirección A0-A7 permiten dividir el área de mensaje en 160 segmentos y permite acceder a los modos de operación . Las entradas de direcciones tienen lógica positiva. Cada incremento de una dirección es equivalente a 0.125 segundos.

Las direcciones externas controlan un registro llamado MSP (puntero de inicio de mensaje). El MSP apunta a la localidad donde la siguiente operación de reproducción o grabación empiece.

Para determinar el valor en el MSP, se debe multiplicar la dirección por 0.125 segundos. Al direccionar al ISD1000A, la dirección indica que tan dentro del espacio de mensajes uno se encuentra y no indica el espacio restante de la memoria de mensajes.

Se puede pensar en el ISD1000A como una grabadora con la habilidad de posicionar la cabeza grabadora/reproductora en cualquier lugar de la cinta, limitada solo por la resolución de las direcciones.

### 1.2.2.3 MODOS DE OPERACIÓN:

Hasta ahora se ha dicho que al escribir una dirección entre 0 y 159 se localiza un puntero de inicio de mensajes, sin embargo, no se ha dicho que sucede si la dirección sobrepasa estos valores. Existe una zona de direcciones que se utiliza ya no para indicar el posicionamiento del puntero de inicio de mensaje, sino para indicar algunas instrucciones para la grabación y reproducción a las que llamamos "modos de operación".

El direccionamiento de mensajes y modos de operación son mutuamente excluyentes. Cuando se utiliza un modo de operación, el MSP se inicializa en cero a menos que se seleccione un modo de "overriding".

Entre las direcciones 0 y 159 se puede direccionar al ISD1000A para la grabación, entre las direcciones 160 y 191 se estaría en direcciones no utilizadas que darían como

resultado un sobre-flujo y a partir de las direcciones 192 hasta la 255 se tiene la selección de los modos de operación.

El ISD1000 utiliza las últimas 64 direcciones para los modos de operación. Esta parte del mapa de direcciones está definido por las direcciones A6 y A7 en alto. Todas las acciones en los modos de operación empiezan en la dirección A0. Operaciones posteriores pueden empezar en otras direcciones si los bits apropiados del modo de operaciones son elegidos. Cuando el "chip" varía su modo entre grabación y reproducción o viceversa y cuando el ciclo de apagado es ejecutado, el puntero interno de la dirección queda reseteado en cero.

Los modos de operación no se memorizan. Una acción en el modo de operación se ejecuta cuando CE se pone en bajo y los bits A6 y A7 de las direcciones se encuentran en alto. Si la siguiente operación del CE encuentra a cualquiera de estos bits en bajo, una operación de dirección de mensaje es ejecutada y el estado de cualquier modo de operación previo se pierde.

**BIT DE DIRECCIÓN A4 (PIN 5).- RESETEO DEL PUNTERO DE INICIO DE MENSAJE:**

Cuando este bit se encuentra en alto, y nos encontramos en modo de operación, entonces el puntero de inicio de mensaje se inicializará en cero, solo si el modo grabación /reproducción cambia o si un ciclo de apagado se ejecuta. Esta opción permite apilar mensajes grabados o revisar

secuencialmente mensajes durante la reproducción sin tener que saber la dirección origen de cada mensaje. Esto se debe a que la última dirección grabada o reproducida fija la dirección de origen de la siguiente operación.

**BIT DE DIRECCIÓN A2 (PIN 3).-CONTROL DEL EOM:**

La salida EOM comúnmente realiza dos funciones. Cuando se está reproduciendo un mensaje que no termina con la totalidad de la memoria, el EOM manda un pulso en bajo de 16ms para indicar que ha terminado el mensaje. El EOM permanece en OL solo si se ha terminado la memoria y no solo el mensaje. Si CE se encuentra en alto, la reproducción termina en este punto. Si A2 está en alto, en el modo de operación, el indicador EOM de final de mensaje queda deshabilitado.

**BIT DE DIRECCIÓN A3 (PIN 4).- CONTROL DE REPRODUCCIÓN CONTINUA:**

Este bit en modo de operación permite repetir automáticamente el mensaje localizado en el inicio del espacio de mensajes. Esto continúa hasta que CE tenga un flanco negativo. La única excepción a esta operación es que el mensaje a ser repetido no ocupe toda la memoria del ISD1000A.

**BIT DE DIRECCIÓN A5 (PIN 6).- CE ACTIVADO POR NIVEL:**

El modo preestablecido es que CE sea activado por flanco en el modo de reproducción y por nivel en el modo de grabación.

Con el bit A5 en alto en el modo de operación, CE es activado por nivel en el modo de reproducción. Si los bits de dirección A7, A6 y A5 están en alto, un flanco negativo de CE inicia la reproducción en el inicio del mensaje y la reproducción continúa mientras CE se encuentre en bajo. Si CE se pone en alto nuevamente, la reproducción termina y un segundo flanco negativo de CE reinicializa la reproducción al principio del mensaje. Si el bit A4 también es puesto en alto la activación de CE inicia la reproducción del mensaje al principio y si CE es puesto en alto, el mensaje vuelve a terminar. Si CE se vuelve a OL, la reproducción continúa en el punto en que terminó anteriormente.

#### 1.2.2.4 UTILIZACIÓN DEL ISD1000A:

Para grabar un mensaje se deben seguir los siguientes pasos:

1. direccionar el inicio del mensaje
2. pin P/R en bajo
3. pin PD en bajo
4. pin CE en bajo
5. la grabación continúa mientras se mantenga CE en bajo
6. la grabación termina cuando se cambia CE a alto o el fin de memoria se alcanza

Si se termina de grabar antes de terminar el fin de memoria, se marcará una señal de fin de mensaje (EOM).

Para reproducir un mensaje se deben seguir los siguientes pasos:

1. direccionar el inicio del mensaje que se quiere reproducir
2. pin P/R en alto
3. pin PD en bajo
4. pin CE en bajo
5. la reproducción continúa mientras no se llegue a una señalización interna de EOM, un breve pulso de 0L en el CE es suficiente para iniciar la reproducción del mensaje. Al final de la memoria EOM permanece bajo hasta poner a PD en alto.

#### 1.2.2.5 DIAGRAMAS DE TIEMPO PARA LA GRABACIÓN Y REPRODUCCIÓN:

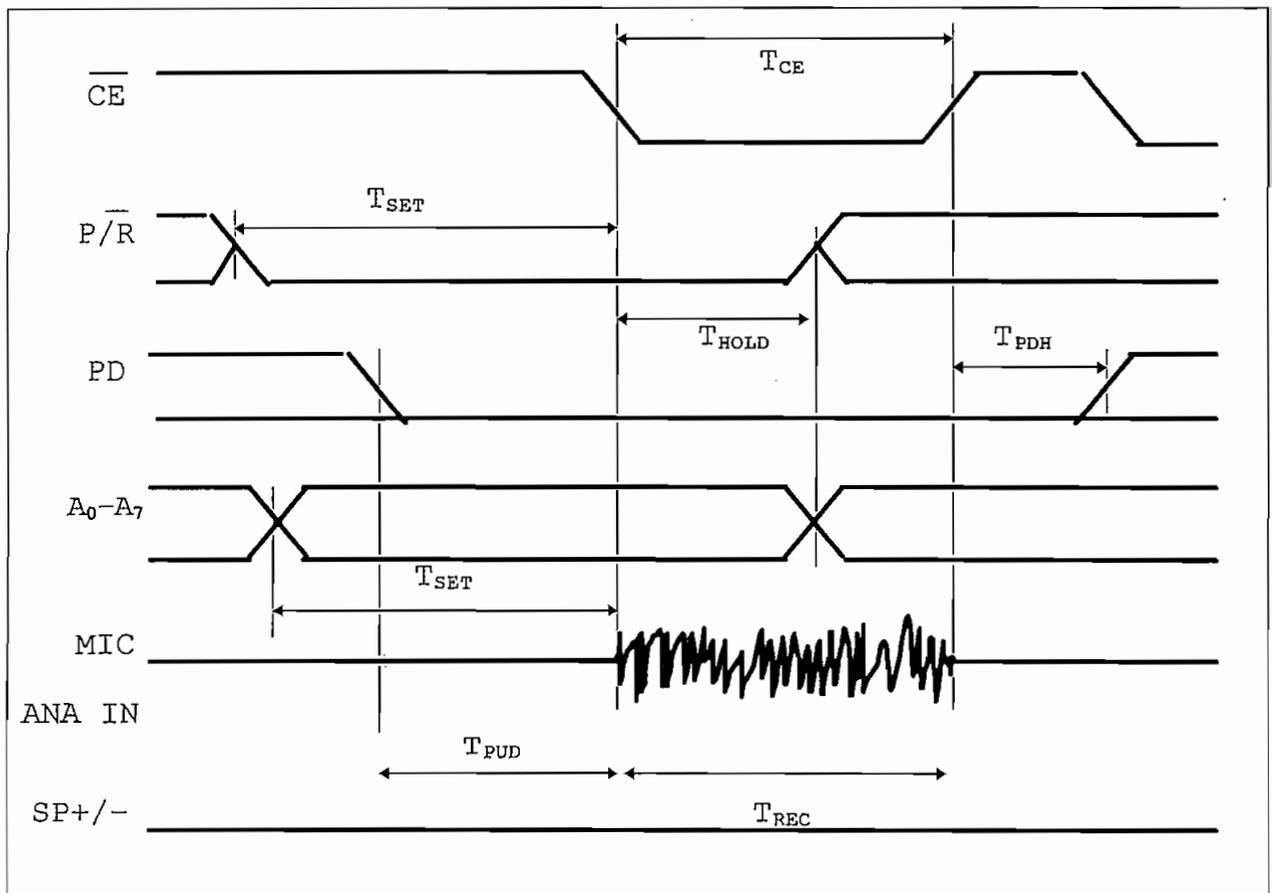


FIGURA 5. Diagramas de tiempo para la grabación

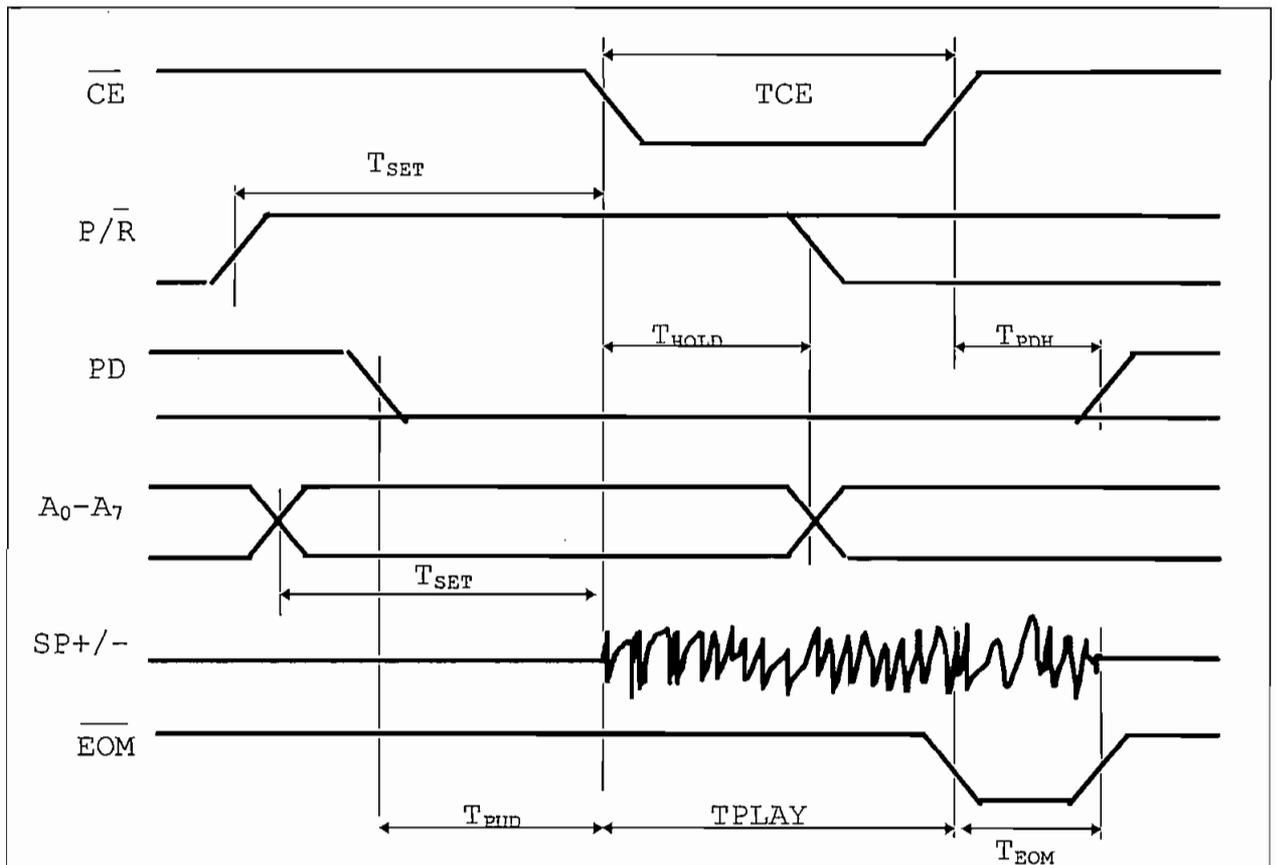


FIGURA 6. Diagramas de tiempo para la reproducción

Los tiempos son los siguientes:

$$T_{SET} = 300 \text{ ns min.}$$

$$T_{HOLD} = 0 \text{ ns min.}$$

$$T_{CE} = 100 \text{ ns min.}$$

$$T_{PUD} = 31.25 \text{ ms min.}$$

$$T_{PDH} = 0 \text{ ns min.}$$

$$T_{REC} = 19.5 \text{ seg min.} \quad 20 \text{ seg typ.} \quad 20.5 \text{ seg máx.}$$

$$T_{EOM} = 15.6 \text{ ms typ.}$$

## 1.2.3 CIRCUITOS DE INTERFAZ CON LA LÍNEA TELEFÓNICA

### 1.2.3.1 INTERFAZ DETECTORA DE DATOS DTMF

Para permitir que los circuitos internos de este equipo, como el receptor DTMF, el microcontrolador y el CI grabador reproductor de voz, interactúen con la línea telefónica, se requieren algunos circuitos auxiliares para lograr la interconexión.

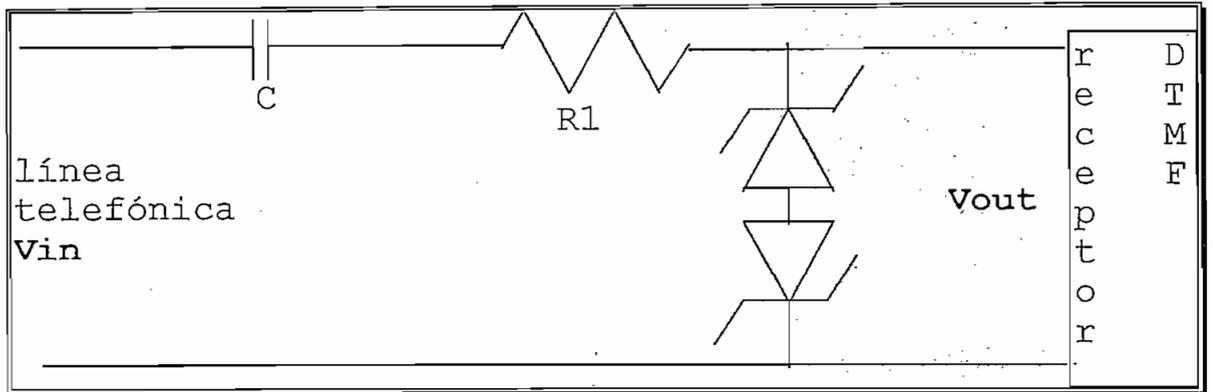
En primer lugar se requiere detectar el par de tonos correspondientes a un dígito presionado, para esto se debe poseer un interfaz entre el receptor DTMF y la línea telefónica. Para el diseño de este circuito debe considerarse que la línea telefónica puede darnos una variedad de señales en base al estado en que se encuentre como se describe a continuación:

Si la línea se encuentra colgada, entonces el teléfono se comporta como una alta impedancia y el voltaje a los terminales de la línea telefónica es de corriente continua de aproximadamente 50V.

En el momento de descolgar el teléfono, éste conecta una carga a la línea telefónica con lo que el voltaje de continua decrece a aproximadamente 8V, pero al sonar el tono de marcar, aparece un voltaje de corriente alterna sinusoidal de aproximadamente 0.6 Vpico sobre la componente de continua. Así mismo las señales sonoras de voz o de los tonos, por ejemplo, poseen una amplitud que puede variar en general entre 1 a 4V pico de corriente alterna.

Cuando el teléfono está colgado y timbra, la línea posee además de la componente de continua un voltaje sinusoidal de aproximadamente 90V pico.

En base a lo expuesto, se utilizará el circuito que se muestra a continuación:



**figura 7.** Interfaz del receptor DTMF con la línea telefónica.

Las funciones que poseen cada uno de los elementos que se muestran en la figura son las siguientes:

El capacitor C se utiliza para filtrar la componente de continua que no se requiere para el receptor DTMF. Los 2 diodos zener en contraposición se utilizan para limitar el voltaje en el caso de que el teléfono timbre, esto para proteger al receptor DTMF de voltajes excesivos. La resistencia R1 sirve para limitar la corriente a través de los diodos zener en el caso que entren en conducción debido a voltajes mayores al voltaje zener.

A continuación se determinan los valores adecuados de los elementos que constituyen el circuito:

El voltaje de los diodos zener debe ser tal que no sobrepase el voltaje máximo permitido en la entrada analógica del receptor DTMF. En base a las hojas de datos de este circuito integrado se puede ver que este voltaje no debe superar en 0.5 voltios al voltaje de la fuente. como la fuente es de 5.1 voltios, se ha escogido por seguridad dos diodos zener de 5.1 V cada uno, para limitar el voltaje positivo y negativo como se indicó en la figura correspondiente a este circuito.

El valor de R1 debe ser mucho menor al de la impedancia en la entrada "ANALOG IN" del receptor DTMF, para que se logre aprovechar la mayor parte del voltaje de los tonos de entrada desde la línea telefónica. Pero la resistencia R1 debe ser lo suficientemente grande para limitar la corriente en el zener. En base a esto se ha escogido R1=5.1kOhm.

A continuación se calcula el capacitor de entrada C:

$$G = \frac{V_{out}}{V_{in}} = \frac{R_{in}}{\sqrt{(R_{in} + R1)^2 + (1/(2\pi fC))^2}}$$

Rin es la resistencia de entrada del receptor DTMF, el cual en base a las hojas de datos del fabricante es de 100kΩ

Se quiere que  $\frac{V_{out}}{V_{in}}$  sea mayor a 0.9.

El tono de menor frecuencia que debe detectar el receptor DTMF es de 700Hz. En base a estos datos se calcula el valor de C.

$$C = \frac{1}{2\pi f} \sqrt{\frac{1}{(\frac{R_{in}}{0.9})^2 - (R_{in} + R1)^2}}$$

$$C = \frac{1}{2\pi \times 700} \sqrt{\frac{1}{(\frac{100k}{0.9})^2 - (100k + 5.1k)^2}}$$

$$C = 6.29nF$$

En base a este valor mínimo de C se escoge C=0.01uF

El receptor DTMF ha sido configurado para entregar el código binario del dígito presionado, este número binario se ingresa al microcontrolador por el pórtilo 1 a los pines p1.0 hasta p1.3. Además de conectar los pines del receptor DTMF que poseen el código binario, se conecta un pin adicional llamado DV, el cual se vuelve positivo cuando se ha detectado un par de tonos válidos.

El pin DV del ISD1000A indica que existe un dato DTMF válido en las salidas de datos DTMF del ISD1000A. La conexión del DV hacia el microcontrolador se hace a través del pin P3.3 de la interrupción externa 1. El microcontrolador ingresa a la interrupción en caso de detectar un flanco negativo en el pin P3.3. Sin embargo, el pin DV indica con un flanco positivo, la existencia de un dato válido. Debido a que se requiere detectar flancos positivos del pin DV, se deberá colocar una compuerta inversora entre el pin DV del ISD1000A y el pin p3.3 del microcontrolador.

#### **1.2.3.2 INTERFAZ DETECTORA DE TIMBRADAS**

La línea telefónica, además de estar conectada al receptor DTMF, debe conectarse a un circuito de interfaz que permita conectar al microcontrolador para que detecte el número de timbradas, y posteriormente descuelgue la línea telefónica. Este circuito debe dejar pasar solamente señales de corriente alterna que posean un nivel de voltaje que pueda provenir de una timbrada telefónica, es decir, de aproximadamente 90 voltios pico. El circuito que puede hacer esto es el que se muestra a continuación:

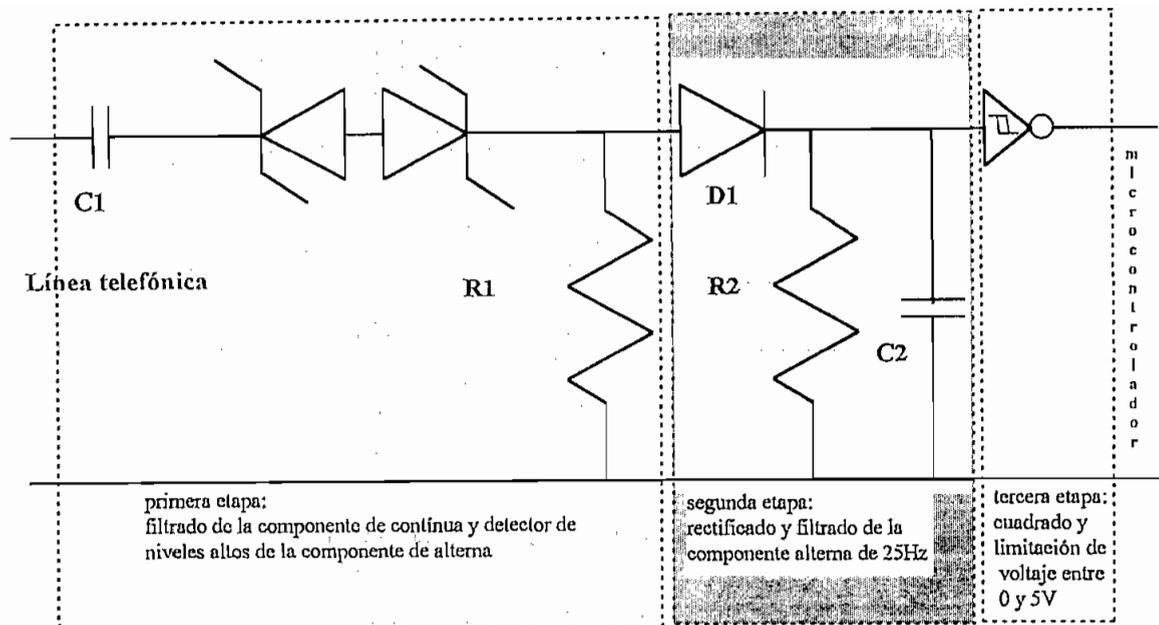


figura 8. Interfaz detectora de timbradas

En el momento de una timbrada, se tiene en la línea telefónica un voltaje de alterna de aproximadamente 90 V pico sobre una componente de continua de aproximadamente 50V. La duración de una timbrada es de aproximadamente 1.3 segundos, mientras que las pausas entre cada timbrada son de aproximadamente 4.7 segundos.

La primera etapa del circuito que detecta las timbradas, utiliza a la entrada un capacitor C1, que solo permite el paso de corriente alterna. Esta etapa utiliza los dos diodos zener de entrada para no permitir pasar corriente a la resistencia R1 a menos que la diferencia de potencial pico de la componente de alterna de la señal de entrada, sea superior en valor absoluto al voltaje zener. El filtro de entrada de esta etapa, no deberá dejar pasar a la componente de continua, pero deberá dejar pasar a las frecuencias iguales o superiores a la de 25 Hz de la timbrada

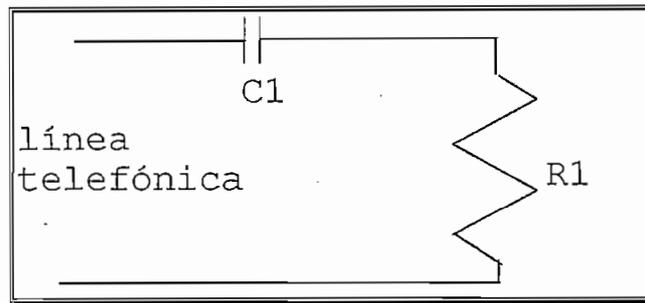
En la segunda etapa, el voltaje aplicado a la resistencia R1 se lo rectifica con el diodo D1 y luego se filtra la componente de alterna de 25Hz, dando como resultado una forma de onda aproximadamente cuadrada con la frecuencia entre timbradas que es aproximadamente de 0.25Hz. En la tercera etapa, este voltaje se aplica a una compuerta Schmitt trigger para cuadrar el voltaje y limitarlo entre 0 y 5 voltios. El valor del filtro de la segunda etapa debe ser tal que permita el paso de la componente de 0.2Hz del timbre, pero debe filtrar casi totalmente la componente de 25Hz de la señal del timbre.

La tercera etapa permite que el voltaje obtenido de la segunda etapa se cuadre, es decir, que posea flancos bien definidos y que se obtengan a la salida niveles de voltaje TTL. De acuerdo a esto, mientras el teléfono timbre se tendrán a la salida de la compuerta inversora del tipo Schmitt trigger, un nivel de voltaje de cero lógico, mientras que en las pausas entre timbradas se tendrá un voltaje equivalente a 1L.

A continuación se tienen los cálculos realizados para la determinación de los valores de los componentes del circuito:

El voltaje de los diodos zener debe ser mayor al voltaje máximo pico de corriente alterna que se pueda producir en la línea telefónica debido al sonido en una comunicación o por ruido. El voltaje zener no debe ser excesivo, para que el voltaje de la línea telefónica no se vea muy disminuido en la resistencia R1. En base a estos parámetros, se escoge diodos zenner de 15V.

Para el cálculo del capacitor C1 se utiliza un circuito equivalente aproximado, considerando que, el voltaje que ingresa desde la línea telefónica es mayor al voltaje zener, y que R1 es mucho menor a la impedancia de entrada de la segunda etapa de este interfaz. A continuación se muestra este circuito equivalente:



**figura 9.** Circuito equivalente aproximado del filtro de entrada del interfaz detector de timbradas

En base al anterior circuito equivalente aproximado, Se escoge R1 de un valor igual a  $1k\Omega$  y se quiere que el voltaje en la resistencia R1 sea aproximadamente un tercio del voltaje de entrada desde la línea telefónica, por lo tanto,

$$\frac{V_{R1}}{V_{líneaTelefónica}} = 0.3 = \frac{R1}{\sqrt{R1^2 + X_{C1}^2}}$$

$$X_{C1} = \frac{\sqrt{(1 - 0.3^2)}}{0.3} \times R1$$

$$X_{C1} = 3.18k\Omega$$

$$X_{C1} = \frac{1}{\omega \times C1}$$

$$f = 25Hz$$

$$C1 = \frac{1}{2\pi \times 25 \times 3.18k}$$

$$C1 = 2\mu F$$

Para el cálculo del filtro de la segunda etapa, se considera un factor de rizado del 1% para la componente de 25Hz. Se

escoge R2 igual a 20kΩ debido a que R2 debe ser mucho mayor a R1. Debido a que la segunda etapa posee un rectificador de media onda, se tiene la siguiente ecuación para el factor de rizado en la resistencia R2:

$$\tau = \frac{1}{2\sqrt{3}fC_2R_2} \quad [1]$$

$$C_2 = \frac{1}{\tau 2\sqrt{3}fC_2R_2}$$

$$C_2 = \frac{1}{0.01 \times 2\sqrt{3} \times 25 \times 20k}$$

$$C_2 = 104\mu F \approx 100\mu F$$

### 1.2.3.3 INTERFAZ DEL C.I. GRABADOR/REPRODUCTOR DE VOZ

El siguiente circuito de interfaz utilizado es el que permite conectar el circuito integrado grabador/reproductor de voz hacia la línea telefónica. Este circuito requería que los niveles de voltaje conectados a los terminales del parlante del mismo no sean superiores a las señales que éste era capaz de entregar. La impedancia que se debía conectar a este circuito integrado con la línea telefónica tampoco debía ser muy grande para que no se atenuara la voz demasiado, sin embargo tampoco podía poseer una impedancia hacia la línea telefónica demasiado baja para que ésta no sienta como que se ha descolgado la línea. El circuito que cumple con estos requerimientos es el siguiente:

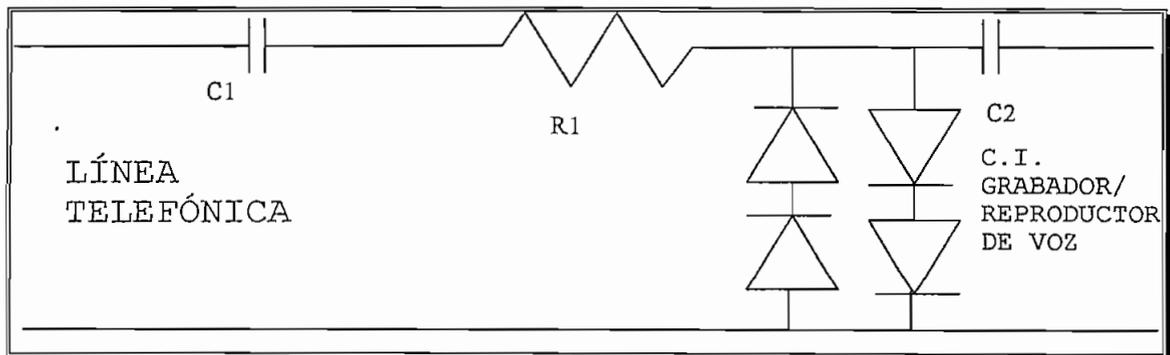


figura 10. Interfaz del grabador/reproductor de voz con la línea telefónica

[1] Sánchez Tarquino, Electrónica Básica, Escuela Politécnica Nacional, 1993. p.79

La primera etapa filtra la componente de continua para que cuando el teléfono esté colgado, la impedancia del interfaz hacia la línea telefónica sea prácticamente infinita. En el caso de que el teléfono timbre, el voltaje se recortará a  $\pm 1.4V$  gracias a los diodos que limitan el voltaje. La última parte del interfaz utiliza un capacitor para permitir pasar únicamente a la componente de alterna desde el chip grabador/reproductor de voz hacia la línea telefónica, este se utiliza debido a que cuando no existe salida de señal de voz, los terminales del parlante están conectados a tierra. En base a las recomendaciones del fabricante se escoge  $C1=C2=2.2\mu F$  y  $R1=1k\Omega$ .

#### 1.2.4 CIRCUITO DE INTERFAZ CON EL COMPUTADOR

El circuito que está conectado a la línea telefónica deberá estar aislado, ya que de lo contrario se podría introducir ruido hacia la línea. Para lograr el aislamiento, la fuente de voltaje que alimenta al circuito utiliza un transformador a la entrada y el pórtico serial de computador utiliza optoacopladores para el aislamiento eléctrico.

A continuación se muestra el circuito utilizado para el aislamiento eléctrico entre el pórtico serial del computador y el microcontrolador.

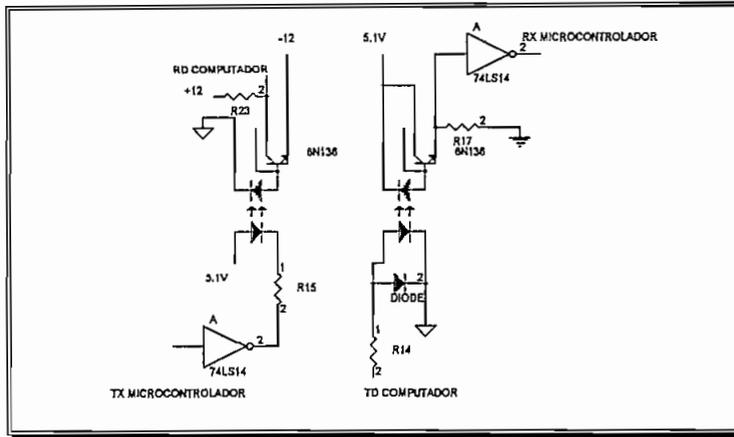


figura 11. Circuito de aislamiento de la comunicación serial.

El circuito anterior utiliza como polarización del lado del computador a las líneas de control DTR y RTS, mientras que del lado del microcontrolador, la misma fuente de polarización de éste.

A continuación se tiene el cálculo de los elementos utilizados en el circuito anterior.

Se han escogido optoacopladores 6N136, los cuales son especialmente apropiados para esta aplicación ya que permiten transmitir datos hasta a una velocidad de 1Mbit/s. Estos optoacopladores poseen las siguientes características:

$$I_f = 25\text{mA}$$

$$V_{R_{led}} = 5\text{V}$$

$$V_{CE_{m\acute{a}x}} = 15\text{V}$$

$$I_{colector} / I_{diodo} = 15\%$$

$$I_{colector_{m\acute{a}x}} = 8\text{mA}$$

La compuerta 74LS14 que alimenta a uno de los optoacopladores, posee una capacidad máxima de corriente de salida de 400uA en 1L, y 8mA en 0L. De acuerdo al circuito

anterior, se debe utilizar como dato la corriente máxima de 8mA, por lo que se escoge como corriente de trabajo 7mA. A continuación se calcula la resistencia  $R_{15}$ :

$$R_{15} = \frac{V_{cc} - V_{diodo}}{I_{diodo}}$$

$$R_{15} = \frac{5.1V - 1.3V}{7mA}$$

$$R_{15} = 543\Omega$$

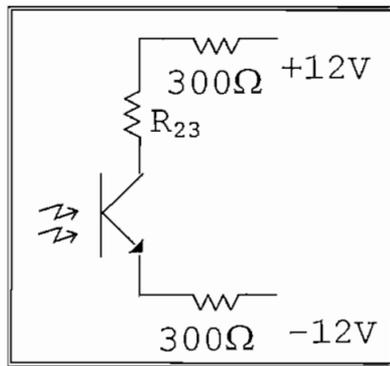
$$\Rightarrow R_{15} \approx 560\Omega$$

$$I_{R_{15}} = 7mA$$

$$I_{R_{23}} = 0.15 \times I_{R_{15}}$$

$$I_{R_{23}} = 1.05mA$$

La impedancia de salida de las líneas de control del pórtico serial es de  $300\Omega$ , por lo que se tendrá el siguiente circuito equivalente en el lado del computador:



**Figura 12.** Circuito equivalente del optoacoplador conectado a las líneas de control del computador

En base al circuito equivalente, se tiene el cálculo de  $R_{23}$ :

$$I_C = \frac{V_{positivo} - V_{negativo} - V_{ce_{sat}}}{300\Omega + 300\Omega + R_{23}}$$

$$I_C = \frac{24 - 0.2}{300\Omega + 300\Omega + R_{23}}$$

$$R_{23} = \frac{23.37V}{1.05mA}$$

$$R_{23} = 22k\Omega$$

La línea de transmisión serial del computador (TD) que está conectada al diodo del optoacoplador, posee una impedancia de salida de  $300\Omega$ . El voltaje cae en el diodo emisor de luz es de  $1.3V$ . La corriente que se escoge que pase por el led es  $7mA$ . En base a estos datos se calcula  $R_{14}$ :

$$R_{14} = \frac{V_{TD} - V_{led} - 300\Omega \times I_{led}}{I_{led}}$$

$$R_{14} = \frac{12V - 1.3V - 300\Omega \times 7mA}{7mA}$$

$$R_{14} = 1228\Omega$$

$$\Rightarrow R_{14} = 1.2k\Omega$$

$$I_{R17} = 0.15 \times I_{R14}$$

$$I_{r17} = 0.15 \times 7mA$$

$$I_{R17} = 1.05mA$$

$$R_{17} = \frac{V_{CC} - V_{CE_{sat}}}{I_{R17}}$$

$$R_{17} = \frac{5.1V - 0.2V}{1.05mA}$$

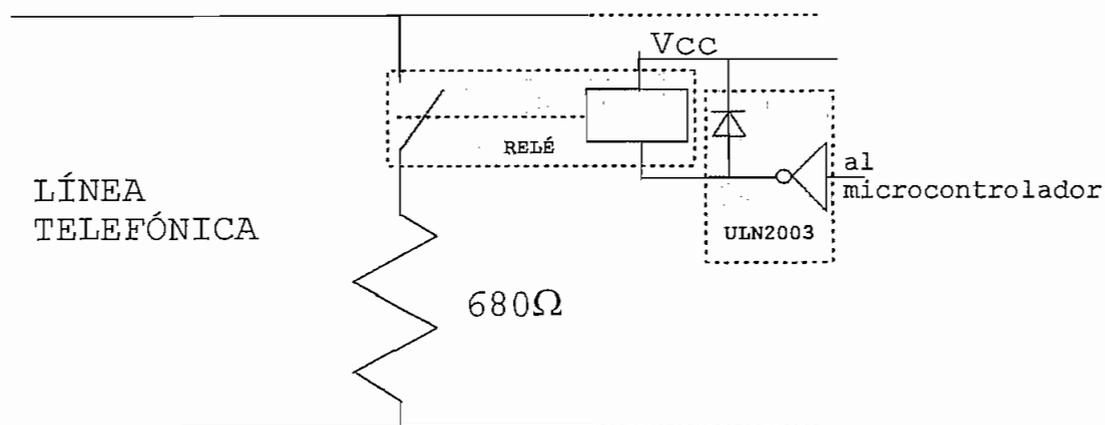
$$= 4600\Omega$$

$$\Rightarrow R_{17} \approx 4.7k\Omega$$

### 1.2.5 CIRCUITO COLGADOR/DESCOLGADOR DE LA LÍNEA TELEFÓNICA

Cuando un teléfono está descolgado, éste posee una resistencia de entrada muy alta, vista desde la línea telefónica. En el caso que se descuelga el teléfono, su

resistencia disminuye a un valor aproximado de  $700\Omega$ . En base a esta característica se plantea el circuito colgador/descolgador que se muestra a continuación:



**Figura13.** Circuito que permite colgar y descolgar la línea telefónica

En este circuito se utiliza un relé para conectar y desconectar la resistencia a la línea telefónica. Este relé será manejado por un "driver" llamado ULN2003, el cual permite manejar hasta 600mA, y posee internamente un diodo de conmutación, el cual es necesario cuando se manejan relés, debido a que éstos son una carga altamente inductiva, que introduciría ruido en el momento de la conmutación y podría crear problemas en los demás elementos electrónicos.

### 1.3 DIAGRAMA CIRCUITAL COMPLETO

En base a los circuitos y elementos anteriormente descritos se los ha combinado para formar el circuito completo que se muestra a continuación:

## CAPÍTULO 2. DISEÑO DEL SOFTWARE

### 2.1 REQUERIMIENTOS

El software realizado tanto para el microcontrolador, como para el computador, están concebidos para ser utilizados en un edificio que esté controlado por un computador. Este computador tendrá funciones tales como programar los horarios de encendido y apagado de luces, así como del resto de circuitos eléctricos del edificio, tales como los de aire acondicionado, tomacorrientes, etc.

El control de los circuitos eléctricos en base a la programación de horarios de utilización de los mismos, permite obtener un substancial ahorro de energía que dependiendo del caso puede estar en alrededor del 30% mensual. El centralizar el control de los horarios desde un solo computador, permite realizar cambios en los horarios con mucha facilidad, sin tener que estar yendo por todo el edificio programando o revisando los horarios en cada temporizador.

Estos horarios programados desde el computador suelen permanecer prácticamente invariantes en los edificios que pertenecen a una sola institución, debido a que existen horarios de trabajo bien definidos dependiendo de cada departamento u oficina. Estos edificios, suele tener una distribución de los circuitos eléctricos bien distribuida en relación con el tipo de oficina, es decir, las oficinas que pertenecen a un mismo departamento poseen alimentación tanto

para tomacorrientes como para iluminación, que proviene de un mismo grupo de "breakers". Por lo tanto, las oficinas que están alimentadas por un mismo circuito, serán consideradas de un mismo sector. Cada sector consta de oficinas, gradas, o pasillos que poseen alguna relación entre ellos, tal que se pueden compartir los circuitos de luces, tomacorriente o aire acondicionado sin ningún problema. El computador central que maneja el edificio controla los horarios de encendido o apagado de luces, aire acondicionado o tomacorrientes por sectores, es decir, habilita o deshabilita la utilización de sectores del edificio mediante el control de contactores en serie con los "breakers" de cada sector. En general, existen PLCs en cada piso que son los que controlan directamente a los contactores de cada sector. Estos PLCs reciben la información desde el computador de los horarios que deben regir el encendido o apagado de los circuitos. Por lo tanto, una vez programados los horarios en el PLC, no se requiere que el computador mande información del momento en el que se deben prender o apagar los circuitos, ya que estos horarios ya quedan registrados en la memoria del PLC. El computador solo tendrá que volver a mandar información en el caso que algún horario haya cambiado.

Pese a todas las ventajas ya nombradas acerca del control de los circuitos de un edificio mediante horarios programados, existe el problema de la flexibilidad del cambio de horarios. Por ejemplo, si alguien desea quedarse trabajando pasado las seis de la tarde en una oficina en la cual su circuito de luces se desconecta automáticamente a ésta hora, tendría que llamar por teléfono al cuarto de control del edificio donde se encuentra el computador de control, para que el encargado del manejo del computador re programe el horario. En un edificio donde se tengan muchas oficinas, este procedimiento

puede ser un poco problemático, ya que cualquier persona que llame al centro de control podría influir en el cambio de horarios. Esto se podría solucionar si el encargado del computador de control pide que se identifique la persona que llama a pedir la orden, y compruebe en una lista de personas si está autorizada para realizar cambios en los horarios. Luego de reprogramar el horario para ese día de acuerdo a la petición recibida, el encargado debería llevar un registro de las peticiones de cambio de horario que ha recibido. En base a este registro se podrá determinar posteriormente, si algún horario está mal programado, mal escogido o si existe alguna persona que está pidiendo cambios de horario sin ninguna razón.

El software utilizado por este Interfaz DTMF permite que una persona que posea una clave válida de 5 dígitos se comunice con el computador de control del edificio, sin que el encargado del computador tenga que intervenir en ningún momento. Ingresando la clave correcta a través del teclado de cualquier teléfono de tonos, que puede estar ubicado dentro o fuera del edificio, podrá ordenar al computador prender o apagar algún sector del edificio. El computador ejecutará la orden, de ser aceptada y llevará un registro de las órdenes recibidas.

La secuencia de operación del sistema será la siguiente:

El circuito estará conectado al pórtico serial de computador, además de estar conectado a un tomacorriente de 115V y a la línea telefónica. El computador deberá estar conectado y corriendo el programa correspondiente bajo Windows, aunque éste puede estar minimizado. Cuando se detecta una llamada telefónica, el microcontrolador espera el número de timbradas

previamente programadas desde el computador y luego descuelga la línea y se envía un mensaje de voz que le indica ingresar las instrucciones. A continuación el equipo espera a que se haya terminado de ingresar una secuencia de números con el formato de ingreso correcto, el cual en este caso es una clave de 5 números, seguido de asterisco y otra secuencia de 4 números que representa la orden codificada, por último se ingresa el signo de número o el asterisco dependiendo si se quiere prender o apagar el sector seleccionado del edificio. Cuando se ha ingresado la secuencia correcta, el computador recibe las instrucciones y luego envía al microcontrolador los comandos que permitan al equipo repetir verbalmente las instrucciones recibidas por el computador, para luego preguntar al usuario si quiere aceptar o cancelar la orden. Si el usuario acepta, la orden es ejecutada y registrada por el computador, caso contrario es cancelada y se pregunta si se desea ingresar una nueva instrucción. En caso de ser positiva la respuesta, se esperará por una nueva instrucción, caso contrario se colgará el teléfono. Se debe notar que si se supera un tiempo determinado en el ingreso de la clave, se pedirá verbalmente ingresar nuevamente las instrucciones. En caso de no recibir respuesta en un tiempo prudencial, se colgara la línea telefónica.

## **2.2 DESCRIPCIÓN DEL PROGRAMA DEL MICROCONTROLADOR**

### **2.2.1 FUNCIONES BÁSICAS DEL PROGRAMA**

Existen dos tipos de funciones que realizan tanto el programa del microcontrolador, como el del computador. Las primeras funciones son las que permitirán inicializar las condiciones

adecuadas que permitan posteriormente correr el programa del computador en modo de reproducción automática.

La ejecución del programa en el modo de reproducción automática, permite cumplir con todas las acciones que se especificaron en el numeral anterior. Si no se tuviera que inicializar al equipo con algunas condiciones iniciales de operación ni se tuviera que cambiar estas condiciones nunca, el programa en modo de reproducción automática sería el único que se utilizaría, y por lo tanto solo existirían las funciones de modo automático.

Las funciones de inicialización permiten cumplir con los siguientes objetivos:

- grabar mensajes de voz en el circuito integrado grabador/reproductor de voz
- guardar en la memoria RAM del microcontrolador las direcciones de origen de los mensajes grabados en el circuito integrado grabador/reproductor de voz.
- Mandar a pronunciar mensajes al circuito integrado grabador/reproductor de voz.
- Inicializar en la memoria RAM del microcontrolador el número de timbradas a esperar antes de contestar una llamada.
- Conectar o desconectar el parlante auxiliar del grabador/reproductor de voz.

A continuación se describe la necesidad de cumplir con cada una de las funciones de inicialización:

**Grabar mensajes de voz en el circuito integrado grabador/reproductor de voz.-** El circuito integrado grabador reproductor tendrá grabados en su memoria mensajes genéricos que al ser combinados formarán frases. Para poder hacer que este circuito integrado diga las frases que se requieren en el modo automático, los mensajes genéricos deberán estar previamente grabados en su memoria.

**Guardar en la memoria RAM del microcontrolador las direcciones de origen de los mensajes grabados en el circuito integrado grabador/reproductor de voz.-** Aunque ya tengamos grabados los mensajes en el grabador/reproductor de voz, si no conocemos donde empieza cada uno, no se podrá utilizar los mensajes grabados cuando estemos en el modo automático de reproducción.

**Mandar a pronunciar mensajes al circuito integrado grabador/reproductor de voz.-** Para poder guardar en la memoria RAM del microcontrolador las direcciones de origen de los mensajes grabados en el circuito integrado grabador/reproductor de voz, es necesario conocer las direcciones que queremos guardar en la memoria RAM. Para conocer estas direcciones, se debe manualmente hacerle pronunciar al circuito integrado grabador/reproductor de voz los mensajes grabados, para de esta manera darnos cuenta de las direcciones de origen de cada mensaje. El procedimiento para hacerle pronunciar los mensajes manualmente será escribir una dirección y enviarla al circuito integrado grabador/reproductor de voz para que éste pronuncie el

mensaje que se encuentra en dicha dirección. De esta manera probando con varias direcciones se busca la que permita pronunciar cada mensaje desde su inicio.

**Inicializar en la memoria RAM del microcontrolador el número de timbradas a esperar antes de contestar una llamada.-** Como el número de timbradas recibidas es contado por el microcontrolador, y éste por si mismo decide cuando debe descolgar la línea telefónica, entonces hay que indicarle cuantas llamadas se quiere que espere antes de contestar. Como el número de timbradas a esperar es escogido por el usuario, este valor variable deberá ser almacenado en la memoria RAM del microcontrolador.

**Conectar o desconectar el parlante auxiliar del grabador/reproductor de voz.-** Cuando se está haciendo que el circuito integrado grabador/reproductor de voz pronuncie palabras en forma manual, estos mensajes deberán ser escuchados por la persona que está inicializando el sistema por lo que se requiere que el parlante auxiliar esté conectado. Cuando los mensajes están dirigidos a la persona que se encuentra en la línea telefónica, el parlante auxiliar deberá estar desconectado. El microcontrolador es el que se encargará de realizar esta acción.

A continuación se explica la secuencia de operación que sigue el programa del microcontrolador:

El microcontrolador continuamente se encuentra monitoreando la entrada del detector de timbradas de la línea telefónica, y de la salida DV del receptor DTMF mediante el programa principal y la interrupción externa 1. En el caso de que detecte un número de timbradas en un tiempo menor al

determinado, se mandará un 1L al pin p1.5 del mismo para que el teléfono se descuelgue, en este mismo momento inicializa un lazo que dura aproximadamente 25 segundos y luego manda las órdenes al CI grabador/reproductor de voz para que éste pronuncie los mensajes "ingrese" e "instrucciones". Si pasados los 25 segundos, éste no detecta la presencia de un par de tonos válidos, es decir, si no entra a la rutina de interrupción externa 1, el microcontrolador mandará un 0L al pin p1.5 para que el teléfono se vuelva a colgar. Si se recibe una señal de entrada del DV, entrará en la rutina de interrupción externa 1 y empezará a contar con el temporizador 0, 10 segundos, si en este tiempo no se ha vuelto a ingresar ningún dígito a través de la línea telefónica, entonces, el microcontrolador mandará a pronunciar la siguiente frase al CI grabador/reproductor de voz: "repita las" e "instrucciones", esta orden se la dá en la rutina de interrupción del temporizador0. En caso de que se ingrese un formato de dígitos por los 4 pines menos significativos del pórtico 1, que corresponda al formato preestablecido, se enviará el conjunto de datos de las últimas instrucciones al computador. El formato preestablecido es el siguiente: 5 números binarios del 1 al 10 correspondientes a los dígitos 1 al 9 y 0 seguidos del número binario 11 que representa el asterisco, y luego 4 números entre el 0 y 9 seguidos por número binario 12 correspondiente a "#" o el binario del asterisco. El envío de los datos al computador se produce en la misma rutina de interrupción externa, luego de haber recibido el último dato que completa la secuencia de dígitos correcta. El computador luego de haber recibido el paquete de las últimas instrucciones, enviará al microcontrolador a través del pórtico serial, las direcciones de los mensajes correspondientes' al último paquete de instrucciones

recibidas. El microcontrolador luego transmitirá estas direcciones al CI grabador/reproductor en una forma secuencial, dentro de la rutina de interrupción serial en la que entró al recibir el primer dato proveniente del computador. Se debe considerar que el microcontrolador diferencia el tipo de orden en la interrupción serial en base al primer dato recibido desde el computador, como se muestra a continuación:

<i>caracter</i>	<i>ORDEN</i>
G	iniciar grabación
R	reiniciar grabación
P	pausa en la grabación
H	reproducción manual
X	conectar el parlante
Y	desconectar el parlante
T	cargar las direcciones de origen de los mensajes que se pronuncian sin previa orden del computador
A	inicializar el modo automático de reproducción
B	inicializar el número de timbradas

**tabla 4.** caracteres de control utilizados en la comunicación serial

Luego de recibir el primer caracter desde el pórtico serial del computador, se debe saltar a donde corresponda dentro de esta misma subrutina, para utilizar los datos que vengán a continuación o para ejecutar la orden correspondiente a dicho caracter.

En el caso de encontrarse en el modo automático, se realizarán las siguientes funciones:

-Luego de recibir el grupo completo de datos que forman una instrucción, y enviarlas mediante el p rtico serial al computador,  ste le enviar  de regreso las direcciones del grabador/reproductor de voz de los mensajes correspondientes a dichas instrucciones. El microcontrolador enviar  secuencialmente estas direcciones controlando el tiempo entre env os usando el bit EOM.

-Despu s de terminar de enviar estas direcciones mandar  las correspondientes a la pronunciaci n del mensaje compuesto "presione 1 para aceptar, `presione 2 para cancelar". A continuaci n se tendr  un lazo que esperar  una respuesta por un tiempo prudencial luego del cual se enviar  al computador la se al de "cancelar" o de ser la respuesta afirmativa, "aceptar". A continuaci n se enviar n los comandos y direcciones al grabador/reproductor de voz, para que pronuncie el mensaje compuesto "presione 1 para continuar presione 2 para terminar". A continuaci n se tendr  un lazo que esperar  una respuesta por un tiempo prudencial luego del cual se colgar . En caso de obtener una respuesta afirmativa, se volver  a repetir el proceso completo o de ser negativa, se colgar  la l nea telef nica.

### **2.2.2 DIAGRAMA DE BLOQUES DEL PROGRAMA DEL MICROCONTROLADOR**

El programa del microcontrolador estar  formado por una inicializaci n, un programa principal que forma un lazo que se ejecuta continuamente, subrutinas, y rutinas de interrupci n.

A continuaci n se describen cada una de las partes del programa del microcontrolador.

## INICIALIZACIÓN

INICIALIZACIÓN DE SALTOS DE LAS INTERRUPCIONES EXTERNA1, TIMER0, SERIAL

INICIALIZACIÓN DE PÓRTICOS PARA COLGAR EL TELÉFONO Y ESTAR EN ALTO PARA RECIBIR LOS DATOS DE DTMF Y EOM

INICIALIZACIÓN APROXIMADA DE LAS DIRECCIONES DE MENSAJES

INICIALIZACIÓN DEL MODO DE COMUNICACIÓN SERIAL PARA 9600 BAUDIOS, 8 BITS ASINCRÓNICO

INICIALIZACIÓN DE NÚMERO DE TIMBRADAS A ESPERAR EN 2

HABILITACIÓN DE INTERRUPCIONES

## PROGRAMA PRINCIPAL

LAZO DE 4 SEGUNDOS DE ESPERA DE ESTADO 0L DE LA ENTRADA DEL PIN SENSOR DE TIMBRADA, SI EN ESTE TIEMPO NO LLEGA, VUELVO A REPETIR EL LAZO, SIN HABER REGISTRADO NADA,

A CONTINUACIÓN SE ESPERA QUE LLEGUE UN ESTADO DE 1L DEBIDO A LA PAUSA ENTRE TIMBRADAS, ENSEGUIDA DECREMENTO UN REGISTRO CONTADOR DE TIMBRADAS, EN CASO DE LLEGAR A CERO SE DESCUELGA EL TELÉFONO CON EL BIT P1.5 Y SE LE MANDA LAS INSTRUCCIONES CORRESPONDIENTES AL GRABADOR/REPRODUCTOR DE VOZ QUE PRONUNCIE "INGRESE", LUEGO LLAMA A SUBROUTINA "FLANCO" LA CUAL ESPERA EL BIT EOM, Y LUEGO SE PRONUNCIA "INSTRUCCIONES" Y SE VUELVE A LLAMAR A LA SUBROUTINA FLANCO.

A CONTINUACIÓN SE EJECUTA UN LAZO DE 25 SEGUNDOS DE ESPERA, TIEMPO DESPUÉS DEL CUAL SE CUELGA EL TELÉFONO.

ESTE TIEMPO SE PUEDE AUMENTAR O DISMINUIR EN EL TRANSCURSO DEL PROGRAMA DE ACUERDO A LAS CONDICIONES DE LA LLAMADA TELEFÓNICA.

## SUBROUTINA DE INTERRUPCIÓN EXTERNA1 (EXTERNA)

CADA VEZ QUE SE DETECTA UN DATO DTMF SE INGRESA A ESTA SUBROUTINA.

ESTA SUBROUTINA ES LA ENCARGADA DE LEER LA INFORMACIÓN RECIBIDA DESDE EL RECEPTOR DTMF Y ORDENARLA Y ALMACENARLA DE TAL MANERA QUE SE CUMPLAN LAS SIGUIENTES CONDICIONES:

CADA VEZ QUE SE RECIBE UN DATO DTMF, SE LO ALMACENA A CONTINUACION DEL ANTERIOR.

EN CASO DE RECIBIR EL DATO DE ASTERISCO Y HABERSE INGRESADO ANTERIORMENTE 5 DATOS, LOS SIGUIENTES DATOS SE SEGUIRÁN ALMACENANDO A CONTINUACIÓN, PERO SERÁN CONSIDERADOS COMO DATOS DE FUNCIÓN Y NO COMO CLAVE COMO EN EL CASO DE LOS 5 PRIMEROS.

EN CASO DE RECIBIR EL DATO DE ASTERISCO Y HABERSE INGRESADO ANTERIORMENTE MÁS DE 5 DATOS, LOS ÚLTIMOS DATOS SERÁN DESCARTADOS QUEDANDO ALMACENADOS SOLO LOS ÚLTIMOS 5. LOS SIGUIENTES DATOS SE SEGUIRÁN ALMACENANDO A CONTINUACIÓN, PERO SERÁN CONSIDERADOS COMO DATOS DE FUNCIÓN Y NO COMO CLAVE COMO EN EL CASO DE LOS 5 PRIMEROS.

EN CASO DE RECIBIR EL DATO DE ASTERISCO Y HABERSE INGRESADO ANTERIORMENTE MENOS DE 5 DATOS, SE INICIALIZARÁ EL REGISTRO DE DATOS PARA QUE LOS SIGUIENTES 5 DATOS SEAN LOS CORRESPONDIENTES A LA CLAVE.

PARA EL CASO DE LOS DATOS CORRESPONDIENTES A FUNCIÓN Y NO CLAVE SE CUMPLEN IGUALES REGLAS CON EXCEPCIÓN DE QUE LUEGO DEL ÚLTIMO DATO DE FUNCIÓN SE PUEDE ACEPTAR ASTERISCO O SIGNO DE NÚMERO.

EN CASO DE QUE A PARTIR DEL PRIMER DATO INGRESADO, TRANSCURRAN 10 SEGUNDOS Y NO SE TERMINE DE INGRESAR LA CLAVE, MEDIANTE LA RUTINA DE INTERRUPCIÓN DEL TIMER 0, SE INICIALIZARÁ EL REGISTRO DE DATOS Y SE DEBERÁ INGRESAR LAS INSTRUCCIONES NUEVAMENTE.

AL COMPLETAR EL REGISTRO DE DATOS, SE LLAMARÁ A LA SUBROUTINA SERIAL.

## SUBROUTINA SERIAL

ESTA SUBROUTINA PERMITE TRANSMITIR AL COMPUTADOR EL PAQUETE COMPLETO DE INSTRUCCIONES PERO EXPRESADOS EN CÓDIGO ASCII.

LUEGO DE TRANSMITIR EL DATO, SE VOLVERÁ A INICIALIZAR EL PUNTERO DEL REGISTRO DE DATOS,, DE TAL MANERA QUE EL SIGUIENTE DATO DTMF CORRESPONDA A UNA NUEVA INSTRUCCIÓN.

## SUBROUTINA DE INTERRUPCIÓN DEL TIMER 0 (FINTIEMPO)

ESTA SUBROUTINA PERMITE LUEGO DE SER LLAMADA UN NÚMERO DETERMINADO DE VECES, CONTAR 10 SEGUNDOS A PARTIR DEL INGRESO DEL PRIMER DATO DTMF. SI EN ESTOS 10 SEGUNDOS, NO SE HA TERMINADO DE INGRESAR UNA ISTRUCCIÓN DTMF COMPLETA, ESTA SUBROUTINA REINICIALIZA EL REGISTRO DE DATOS AL INICIO. A CONTINUACIÓN SE MANDA LOS COMANDOS Y DIRECCIONES CORRECTAS AL GRABADOR/REPRODUCTOR DE VOZ PARA QUE SE PRONUCIE LOS MENSAJES PREGRABADOS "REPITA LAS" E "INSTRUCCIONES". PARA QUE EXISTA UNA CONTINUIDAD EN LA PRONUNCIACIÓN DE AMBOS MENSAJES, SE UTILIZA EL BIT DE EOM DEL GRABADOR/REPRODUCTOR DE VOZ , EL CUAL INDICARÁ AL MICROCONTROLADOR EL MOMENTO EN EL CUAL SE DEBE VOLVER A ENVIAR LAS DIRECCIONES Y COMANDOS PARA PRONUNCIAR LA SIGUIENTE FRASE.

## SUBROUTINA r50ms

ESTA SUBROUTINA CREA UN RETARDO DE 50MS LA CUAL ES UTILIZADA PARA LA SINCRONIZACIÓN ENTRE COMANDOS PARA LA PRONUNCIACIÓN DE MENSAJES

## SUBROUTINA DE INTERRUPCIÓN SERIAL (COMUNICACIÓN)

ESTA SUBROUTINA SE DIVIDE EN VARIOS RAMALES PARA REALIZAR DISTINTAS FUNCIONES. EN BASE AL PRIMER CARACTER ENVIADO POR EL COMPUTADOR A CONTINUACIÓN SE DESCRIBE LO QUE DEBERÁ HACERSE LUEGO DE HABER DETECTADO EL CARACTER DE CONTROL RECIBIDO.

EN CASO DE RECIBIR EL COMANDO DE GRABAR, ENTONCES INMEDIATAMENTE ENVIARÁ AL GRABADOR/REPRODUCTOR DE VOZ LAS INSTRUCCIONES A SUS PINES DE CONTROL Y DIRECCIONAMIENTO PARA QUE INICIALICE LA MEMORIA EN CERO Y PROCEDA A GRABAR.

EN CASO DE RECIBIR EL COMANDO PARA REINICIALIZAR, ENTONCES, VOLVERÁ A ORDENAR AL GRABADOR/REPRODUCTOR DE VOZ PARA QUE CONTINÚE GRABANDO, PERO YA NO DESDE CERO SI NO A CONTINUACIÓN DEL ÚLTIMO MENSAJE.

EN CASO DE RECIBIR DESDE EL COMPUTADOR, LA ORDEN DE PAUSA, EL MICROCONTROLADOR MANDARÁ AL GRABADOR/REPRODUCTOR DE VOZ LA ORDEN DE PARAR DE GRABAR.

EN CASO DE RECIBIR LA ORDEN DESDE EL COMPUTADOR DE CONECTAR EL PARLANTE, EL MICROCONTROLADOR MANDARÁ UN 1L AL PIN P1.6 QUE ES EL QUE COMANDA EL RELÉ QUE CONECTA Y DESCONECTA AL PARLANTE, ASÍ MISMO SI SE RECIBE LA ORDEN DESDE EL COMPUTADOR DE DESCONECTAR EL PARLANTE, EL MICROCONTROLADOR MANDARÁ UN 0L AL PIN P1.6 QUE ES EL QUE COMANDA EL RELÉ QUE CONECTA Y DESCONECTA AL PARLANTE.

EN CASO DE RECIBIR DESDE EL COMPUTADOR, EL COMANDO PARA SELECCIONAR EL NÚMERO DE TIMBRADAS A ESPERAR ANTES DE CONTESTAR, SE ESPERARÁ UN NUEVO DATO DESDE EL COMPUTADOR QUE ESPECIFIQUE EL NÚMERO DE LLAMADAS, Y ESTE NÚMERO SE MANDARÁ AL REGISTRO QUE CONTROLA EL NÚMERO DE TIMBRADAS A ESPERAR.

EN CASO DE RECIBIR DESDE EL COMPUTADOR EL COMANDO DE MODO AUTOMÁTICO, EL MICROCONTROLADOR ESPERARÁ QUE SE ENVÍEN DESDE EL COMPUTADOR LAS DIRECCIONES DE LOS MENSAJES QUE EL GRABADOR/REPRODUCTOR DE VOZ DEBERÁ PRONUNCIAR LUEGO QUE EL USUARIO INGRESÓ MEDIANTE LA LÍNEA TELEFÓNICA LAS INSTRUCCIONES.

EN EL CASO DE RECIBIR EL COMANDO DE REPRODUCCIÓN MANUAL, SE ESPERARÁ POR 3 DATOS QUE INGRESEN POR EL PÓRTICO SERIAL Y CON ESTOS DATOS SE DIRECCIONARÁ AL GRABADOR/REPRODUCTOR DE VOZ PARA QUE PRONUNCIE EL MENSAJE QUE SE ENCUENTRE EN DICHA DIRECCIÓN. AQUÍ SE ESPERARÁ HASTA QUE SE TERMINE DE PRONUNCIAR EL MENSAJE

EN CASO DE RECIBIR EL COMANDO DE INICIALIZACIÓN DE DIRECCIONES, SE MANDARÁ A ALMACENAR EN LOS REGISTROS CORRESPONDIENTES, LAS DIRECCIONES QUE A CONTINUACIÓN ENTRARÁN DESDE EL PÓRTICO SERIAL.

EN CASO DE RECIBIR EL COMANDO DE MODO AUTOMÁTICO, EL MICROCONTROLADOR ESPERARÁ POR EL INGRESO DE LAS DIRECCIONES DE LOS MENSAJES RECIBIDOS EN UN LAZO REPETITIVO, A CONTINUACIÓN MANDARÁ A PRONUNCIARSE LOS MENSAJES CON ESTAS

DIRECCIONES DE ORIGEN A MÁS DE LOS MENSAJES DE "PRESIONE1 PARA ACEPTAR PRESIONE 2 PARA CANCELAR". LUEGO ESPERARÁ DENTRO DE UN LAZO DE 5 SEGUNDOS, UNA RESPUESTA, SI NO LLEGAN INSTRUCCIONES, SE CONSIDERARÁ QUE SE HA CANCELADO. EN EL CASO DE CANCELAR, INMEDIATAMENTE SE MANDARÁ AL COMPUTADOR EL CÓDIGO ASCII DE "2" . EN CASO DE ACPTAR, SE MANDARÁ AL COMPUTADOR EL CÓDIGO ASCII DE "2" A CONTINUACIÓN MANDARÁ A PRONUCNIARSE LOS MENSAJES DE "PRESIONE1 PARA CONTINUAR PRESIONE 2 PARA TERMINAR". LUEGO ESPERARÁ DENTRO DE UN LAZO DE 5 SEGUNDOS, UNA RESPUESTA, SI NO LLEGAN INSTRUCCIONES, SE CONSIDERARÁ QUE SE HA TERMINADO. EN EL CASO DE TERMINAR, INMEDIATAMENTE SE MANDARÁ UN 0L AL PIN P1.5 QUE DESCOLGARÁ LA LÍNEA TELEFÓNICA. EN EL CASO DE ESCOGER CONTINUAR, SE AMPLIARÁ EL TIEMPO DE ESPERA PARA COLGAR , SE HARÁ PRONUNCIAR EL MENSAJE "INGRESE INSTRUCCIONES" Y LUEGO SE SALDRÁ DE LA RUTINA DE INTERRUPCIÓN SERIAL.

#### **SUBROUTINA DETECTORA DE FLANCO NEGATIVO (FLANCO)**

ESTA RUTINA SE UTILIZA PARA DETECTAR FLANCOS NEGATIVOS DEL BIT P0.3 CORRESPONDIENTE AL EOM DEL GRABADOR/REPRODUCTOR DE VOZ, DOS LAZOS CONSECUTIZOS QUE DETECTAN EL ESTADO DEL BIT .

#### **SUBROUTINA LECTORA DEL PÓRTICO SERIAL (LECTCOMP)**

ESTA RUTINA SE UTILIZA PARA LEER 3 CARACTERES DESDE EL PÓRTICO SERIAL Y ASIGNARLAS A 3 LOCALIDADES NUM1, NUM2, NUM3

## SUBROUTINA QUE UNE LAS CENTENAS, DECENAS Y UNIDADES (TRANSNUM)

ESTA RUTINA SE UTILIZA PARA CONVERTIR LAS 3 LOCALIDADES NUM1, NUM2, NUM3 EN UN SOLO NÚMERO QUE SE LE ASIGNA A LA LOCALIDAD AUXILIAR

### 2.3 PROGRAMA PARA EL COMPUTADOR PERSONAL.

Mediante este programa, cualquier persona que posea un teléfono de tonos y una clave válida de 5 dígitos, ingresada a través del teclado del teléfono, podrá acceder al computador de control de un edificio que posea el sistema de hardware y software propuesto en esta tesis.

El programa del computador cumple con tres objetivos básicos. En primer lugar, permite la comunicación entre el usuario del computador y el circuito grabador/reproductor de voz. En segundo lugar, permite la comunicación directa entre el computador y el usuario que se encuentra conectado a través de la línea telefónica mediante un teléfono de tonos. En tercer lugar, permite llevar un registro de las comunicaciones realizadas entre el computador y los usuarios de éste, conectados a la línea telefónica a través de un teléfono de tonos.

El computador deberá permitir grabar mensajes de voz en el circuito integrado grabador/reproductor de voz.

El computador deberá permitir llevar una comunicación interactiva entre el circuito grabador/reproductor de voz y

el usuario que se encuentre conectado al computador mediante el teclado de un teléfono de tonos.

El usuario del computador deberá poder controlar la conexión de un parlante auxiliar que le permita escuchar los mensajes de voz grabados en el circuito grabador/reproductor de voz.

El computador deberá permitir indicar al circuito grabador /reproductor de voz, los mensajes que quiere escuchar de entre una selección, en base al ingreso de una dirección desde el teclado del computador. Esta dirección corresponderá al inicio del mensaje.

El usuario del computador deberá poder programar a través el computador el número de timbradas que se debe esperar antes de descolgar la línea telefónica.

El computador debe ser capaz de llevar un registro de las órdenes recibidas desde el teclado de un teléfono de tonos. Este registro podrá ser consultado por el usuario en cualquier momento.

El programa utilizado en el computador está realizado en LabView para Windows. Este programa permite una programación gráfica, mediante la utilización de una pantalla de presentación del programa llamada Panel y otra pantalla posterior utilizada para la programación, llamada diagrama.

### **2.3.1 PANEL DEL PROGRAMA DEL COMPUTADOR PERSONAL**

A continuación se muestra el panel con todos sus elementos y funciones.

ENVIAR ORIGENES DE MENSAJES	
PARLANTE ON OFF	REPRODUCCIÓN MANUAL AUTO
INICIALIZAR	
DIRECCION DE ORIGEN	0
REC	PAUSE PLAY
NUMERO DE MENSAJES	
0	MENSAJE ACTUAL
TIEMPO RESTANTE	0.0
PORTICO SERIAL No.	1
INICIALIZAR	
NUMERO DE TIMBRADAS	2
INICIALIZAR	

GRABACION DE MENSAJES				
tiempo	MENSAJE			
0.7	CERO			
RECUPERAR	%C:\CARLOS\GRAB.LOG	GUARDAR		

ARCHIVO DE CLAVES VALIDAS				
CLAVE:	PERSONA			
12345	CARLOS			
RECUPERAR	%C:\CARLOS\CLAV.LOG	GUARDAR		

ARCHIVO DE DIRECCIONES DE MENSAJES				
DIRECCION	MENSAJE			
0	CERO			
RECUPERAR	%C:\CARLOS\DIREC.LOG	GUARDAR		

ARCHIVO DE ORDENES RECIBIDAS				
FECHA	HORA	CLAVE:	SECTOR	ORDEN:
12/02/96	08:58:59 PM	12345	1234	APAGAR
RECUPERAR	%C:\CARLOS\ORDER.LOG			TERMINAR

figura 15. Panel de control del computador

El panel anterior consta de las siguientes partes:

**2.3.1.1 CONTROLES PARA LA GRABACIÓN:**

Estos controles están conformados por el archivo para la grabación de mensajes, el botón de "rec" , el indicador de número de mensaje actual con su correspondiente frase, el control numérico del total de mensajes a grabar y el contador de tiempo restante para el mensaje actual.

**2.3.1.2 CONTROLES PARA LA REPRODUCCIÓN MANUAL:**

Los controles para la reproducción manual están constituidos por el selector de tipo de reproducción, el botón de "play", y el control numérico de dirección de origen.

#### **2.3.1.3 CONTROLES PARA EL MANEJO DEL ARCHIVO DE DIRECCIONES DE MENSAJES:**

Los elementos del panel que permiten el manejo de este archivo están constituidos por el control numérico de direcciones, el control alfanumérico para los mensajes, el selector de número de mensaje, los botones para guardar y recuperar el archivo y el control para ingreso del "path".

#### **2.3.1.4 CONTROLES PARA EL MANEJO DEL ARCHIVO DE CLAVES VÁLIDAS:**

Los controles para el manejo de este archivo son similares a los que manejan el Archivo de direcciones de mensajes.

#### **2.3.1.5 ARCHIVO DE ÓRDENES RECIBIDAS:**

Este archivo permite almacenar la fecha, hora, clave, sector y orden de cada una de las instrucciones recibidas. En este caso los dos únicos controles son el selector de orden y el botón para recuperar las órdenes actualizadas. No se posee el botón para grabación debido a que las órdenes serán grabadas por el programa y no por el usuario desde el computador.

#### **2.3.1.6 SELECTOR ENTRE MODO AUTOMÁTICO Y MANUAL DE REPRODUCCIÓN:**

El modo automático permite almacenar continuamente las órdenes almacenadas, mientras que el modo manual permite pronunciar los mensajes en forma individual e instantánea.

#### **2.3.1.7 INTERRUPTOR DEL PARLANTE:**

Este interruptor permite que la salida del grabador/reproductor de voz esté o no conectada a un parlante auxiliar.

#### **2.3.1.8 SELECTOR DEL NÚMERO DE TIMBRADAS:**

Este control permite escoger el número de timbradas a esperar antes de descolgar la línea telefónica. el rango está limitado entre 1 y 9 timbradas.

#### **2.3.1.9 BOTÓN PARA LA INICIALIZACIÓN DE LAS DIRECCIONES DE MENSAJES:**

Este botón permite transferir el archivo de direcciones de mensajes a la memoria del microcontrolador.

#### **2.3.1.10 CONTROLES PARA LA SELECCIÓN DEL PÓRTICO:**

Se posee dos controles para este objetivo, el primero es un control numérico que permite escoger entre COMM1 y COMM2, mientras que el otro es un botón que ejecuta la orden de cambio de pórtico.

### **2.3.2 DIAGRAMA DEL PROGRAMA DEL COMPUTADOR PERSONAL**

Este diagrama permite dividir a la ejecución del programa en tres partes:

- Instrucciones que se ejecutarán al principio del programa
- Instrucciones que se ejecutarán repetitivamente durante la ejecución del programa
- Instrucciones que se ejecutarán de acuerdo a los controles modificados en el panel.

Las instrucciones que se ejecutarán una sola vez al inicio del programa serán las de inicialización del pórtico serial con sus líneas de control RTS y DTR.

Todas las instrucciones restantes se ejecutarán dentro del lazo repetitivo externo "while-loop" del cual solo se podrá salir al presionar al control binario TERMINAR.

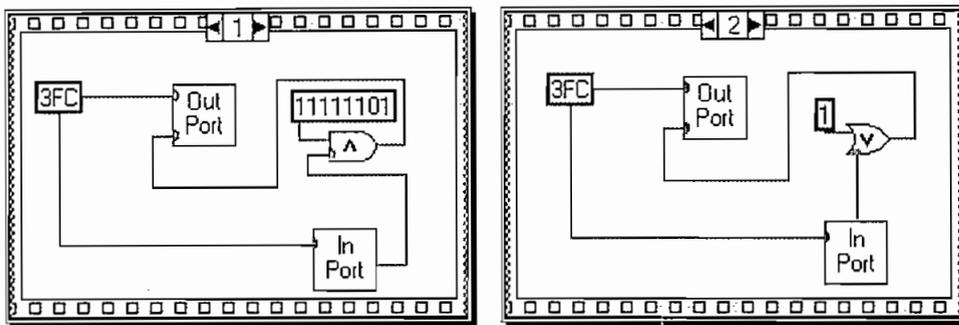
A continuación se describe cada una de las funciones que se realizan dentro del diagrama.

#### **2.3.2.1 INICIALIZACIÓN DEL PÓRTICO SERIAL:**

Esta es la única parte del diagrama que se realiza fuera del lazo "while-loop".

Debido a que se utiliza aislamiento para la comunicación serial, además de manejar las líneas TX y RX del pórtico serial, se van utilizar dos líneas de control adicionales. Estas dos líneas deberán poseer voltajes de +12V y -12V para polarizar uno de los optoacopladores. Se han escogido las líneas DTR y RTS para este fin. Para el manejo de estas líneas con el LabView se utilizan los VIs OUT PORT e IN PORT. el primero permite escribir en una dirección un valor determinado, mientras que el segundo permite leer el valor de una dirección. Se han combinado los dos VIs para poder

escribir solo en los bits correspondientes al DTR y RTS sin afectar al resto. Para lograr esto, se debe leer previamente al byte y luego escribir lo mismo cambiado solo el bit que se requiera. Además de la consideración anterior, se debe escribir en el DTR y RTS luego de la inicialización del pórtico, para garantizar esta secuencia de operación, la inicialización de los valores y del pórtico se los hace dentro de una estructura de secuencia. A continuación se muestran las escenas 1 y 2 de la secuencia, considerando que la escena 0 posee la inicialización del pórtico serial.



**figura 16.** Diagrama de la inicialización del RTS y DTR

Las siguientes funciones se realizan dentro del lazo repetitivo "while-loop" externo.

### 2.3.2.2 REPRODUCCIÓN MANUAL:

Para la ejecución de esta función se tiene una estructura "case" operada en base al selector de reproducción manual o automática, de tal manera que cuando se esté en el modo manual, se ejecute todo lo que se encuentra dentro del caso verdadero. Así mismo dentro de este "case" se tiene anidado otro que permite que se ejecute la reproducción manual solo en el caso que se presione el botón "play". Debe considerarse

que este botón posee la acción mecánica que le permite comportarse como un pulso, lo cual le da la propiedad de ejecutar una sola vez la estructura "case" que se controla con este botón. Dentro de esta estructura "case" se encuentra el diagrama que se muestra a continuación. En este diagrama se convierte al dato de la dirección de mensaje en un "string", además a este "string" se le aumenta la letra H que es el caracter que indica al microcontrolador que se le enviará una dirección para pronunciarse en el modo manual, esto se logra hacer mediante un VI creado con este propósito. Este VI posee como entradas un dato numérico a convertirse en "string" y el caracter a unirse al "string" de salida, la salida es el "string" resultante. Lo último por hacerse es transmitir el dato a través del pórtico serial especificado.

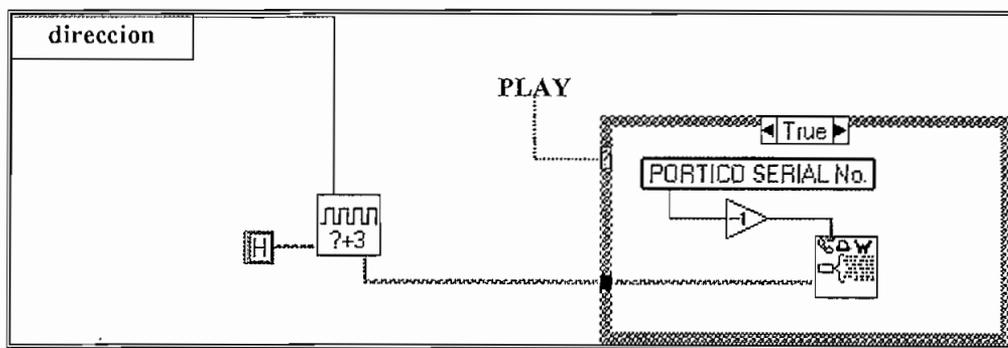


Figura 17. Diagrama para la reproducción manual

### 2.3.2.3 GUARDAR LA INFORMACIÓN DE UN "ARRAY" DE "CLUSTERS" EN UN ARCHIVO:

Esta función se aplica en el caso que se quiera guardar la información de una estructura similar a la que se tiene en el PANEL referida como archivo de direcciones de mensajes, de claves correctas o de mensajes para grabar.

A continuación se tratará el caso del archivo de direcciones de mensaje, sin embargo los demás casos son muy similares. En primer lugar se tiene una estructura "case" que se ejecutará en el caso de presionar el botón "guardar", el cual se observa en la figura como "guardarm". La variable local llamada archivo de direcciones de mensaje es la que contiene la información del PANEL expresada en un "array" de "clusters". A continuación se determina la longitud de este "array" y se ejecuta un lazo "for-loop" con un número de iteraciones igual a la longitud del "array". En este "array" se va descomponiendo los elementos del "array" de fila en fila. En el caso de la dirección que se tiene un dato numérico se lo convierte en "string", y luego se lo une al mensaje con lo que se forma un "array" de dos columnas por N filas y se lo almacena como un archivo de dos dimensiones que se identifica con el "file path".

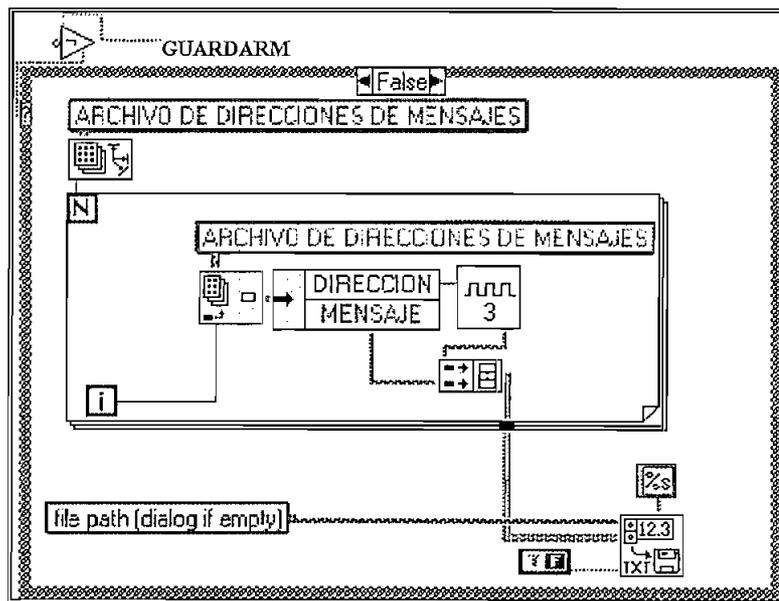


figura 18. Diagrama que permite guardar la información de un "array" de "clusters" en un archivo

#### 2.3.2.4 RECUPERAR LA INFORMACIÓN DE UN ARCHIVO A UN "ARRAY" DE "CLUSTERS":

Esta función se aplica en el caso que se quiera recuperar la información de un archivo y pasarla a una estructura similar a la que se tiene en el PANEL referida como archivo de direcciones de mensajes, de claves correctas, de mensajes para grabar o de órdenes recibidas.

A continuación se tratará el caso del archivo de direcciones de mensaje, sin embargo los demás casos son muy similares.

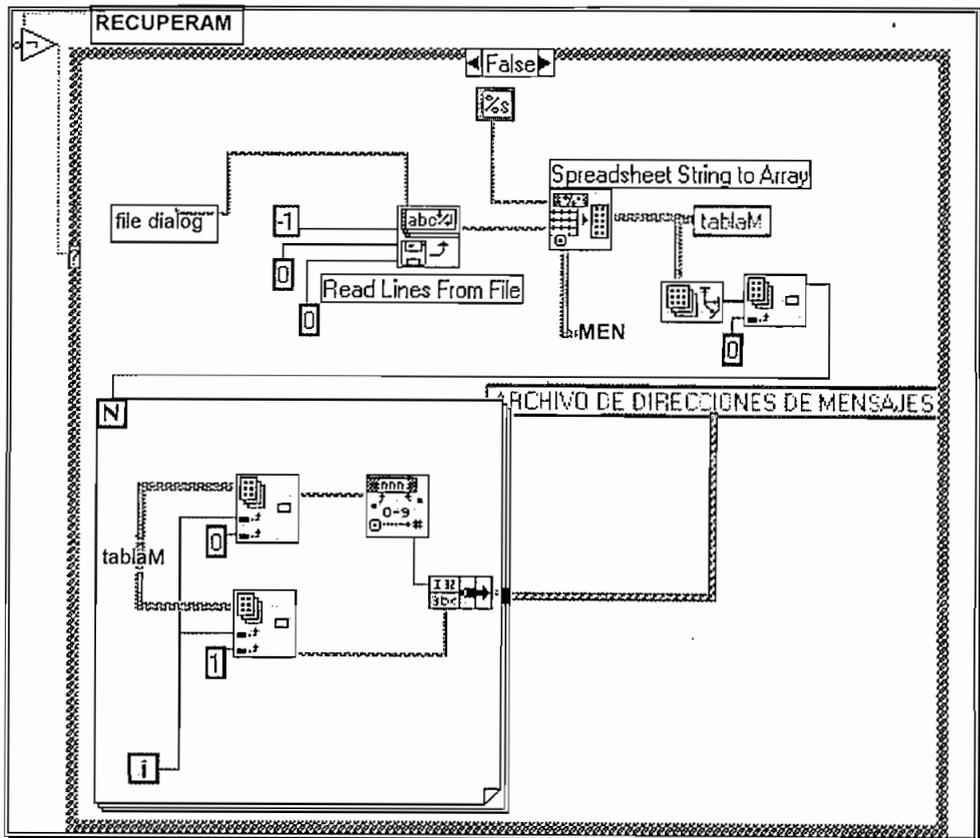


figura 19. Diagrama que permite recuperar la información de un archivo y enviarla a un "array" de "clusters".

En este caso se tiene una estructura "case" que se activará cuando se presione el control "recuperam", el cual posee la etiqueta "recuperar" en el panel. En esta estructura, se tiene el VI "read lines from file", el cual permite recuperar un archivo. Luego a este archivo recuperado se lo convierte en un "array" con el VI "spreadsheet to array", el formato del array se fija en base al formato del "array" "men" que se muestra en la figura 19. La siguiente acción es leer la dimensión del "array" resultante y realizar un lazo "for-loop" en base al número de filas del "array". En este lazo se separan en cada uno de sus componentes al "array" y se lo transfiere al "array" de "clusters" llamado "archivo de direcciones de mensajes".

#### **2.3.2.5 TRANSFERENCIA DE DIRECCIONES AL MICROCONTROLADOR**

Para la transferencia de direcciones de origen al microcontrolador, se debe utilizar como caracter de control a la letra T, lo primero que se debe hacer es recuperar el valor de la dirección que se encuentra en un "array" de "clusters". Se debe considerar que no se desea transmitir todas las direcciones que se encuentran en el "array", sino solo desde la 4 a la 22 y luego la 1 y 2, para esto en un lazo "for-loop" se descompone en primer lugar el "array" en cada uno de sus elementos, pero éstos son "clusters", así que con "unbundle by name" se descompone el "cluster" en sus elementos y se toma el elemento dirección con al cual se lo convierte en un "string". Como lo que se requiere es obtener un "string" que contenga el conjunto de todas las direcciones expresadas en alfanumérico con un caracter T inicial. Para lograr juntar los "strings", se va a utilizar un registro de desplazamiento, el cual se caracteriza por memorizar la salida anterior. Para aprovechar esta característica

inicialmente lo cargamos por la letra T y luego vamos concatenando los "strings" que se obtengan utilizando como entradas del concatenador el último resultado y el registro de desplazamiento. Al finalizar el lazo el "string" completo se lo concatenará con los "strings" parciales del otro lazo y se los transmitirá al pórtico serial. A continuación el diagrama que realiza esta función:

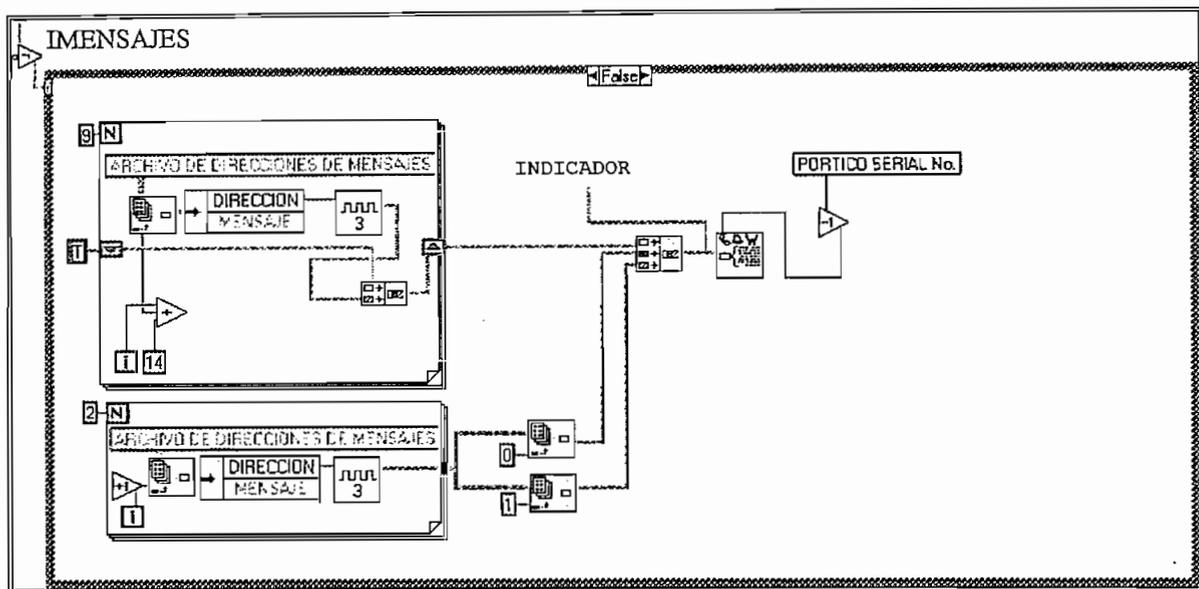


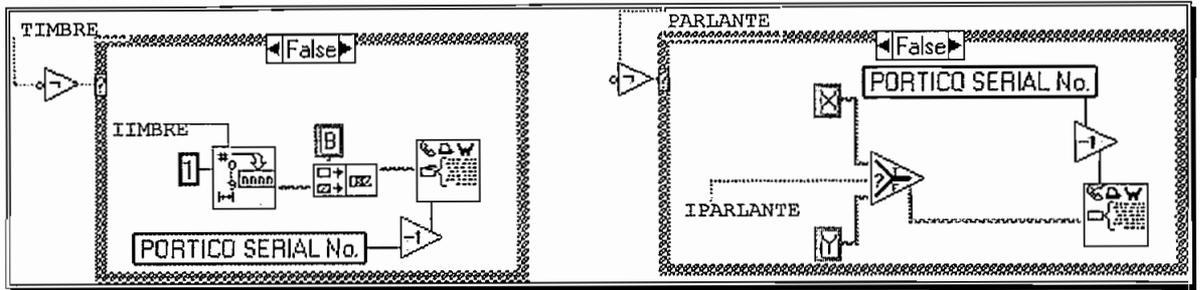
figura 20. Transferencia de direcciones al microcontrolador

### 2.3.2.6 INICIALIZACIÓN DEL NÚMERO DE TIMBRADAS Y CONTROL DEL PARLANTE:

La inicialización del Número de Timbradas y Control del parlante son dos tareas similares, ya que consisten en transmitir un caracter de control seguido en el primer caso de un número expresado en ASCII. Si se analiza el caso del número de timbradas, el caracter de control es la letra B y el número de timbradas es de un solo dígito. Lo primero que se hace es convertir el número en "string" y luego el resultado se lo concatena con la letra B y se lo envía por el

pórtico serial. En el caso del control del parlante, se enviará un caracter de control distinto para conectarlo o desconectarlo, para identificar la orden se utiliza un selector controlado por el estado lógico del control ON/OFF del parlante.

Ambos procesos están controlados por un botón de inicialización que habilita una estructura "case".



**Figura 21.** transferencia al microcontrolador del número de timbradas a esperar y transferencia de la orden de conexión/desconexión del parlante

### 2.3.2.7 GRABACIÓN DE MENSAJES EN EL CIRCUITO GRABADOR /REPRODUCTOR DE VOZ.

En el proceso de grabación de mensajes, se debe enviar por el pórtico serial al microcontrolador el caracter de control G para empezar a grabar, P para una pausa y R para reinicializar la grabación. El período de grabación en cada mensaje está definido por la tabla de grabación de mensajes, mientras que el tiempo de cada pausa está fijado en 1 segundo. Además de crear los retardos hay que visualizar el tiempo restante, esto se hace conectando el indicador de tiempo restante. El retardo se consigue repitiendo n veces lazos de 50ms de duración. El número de mensajes está fijado por el respectivo control en el "panel".

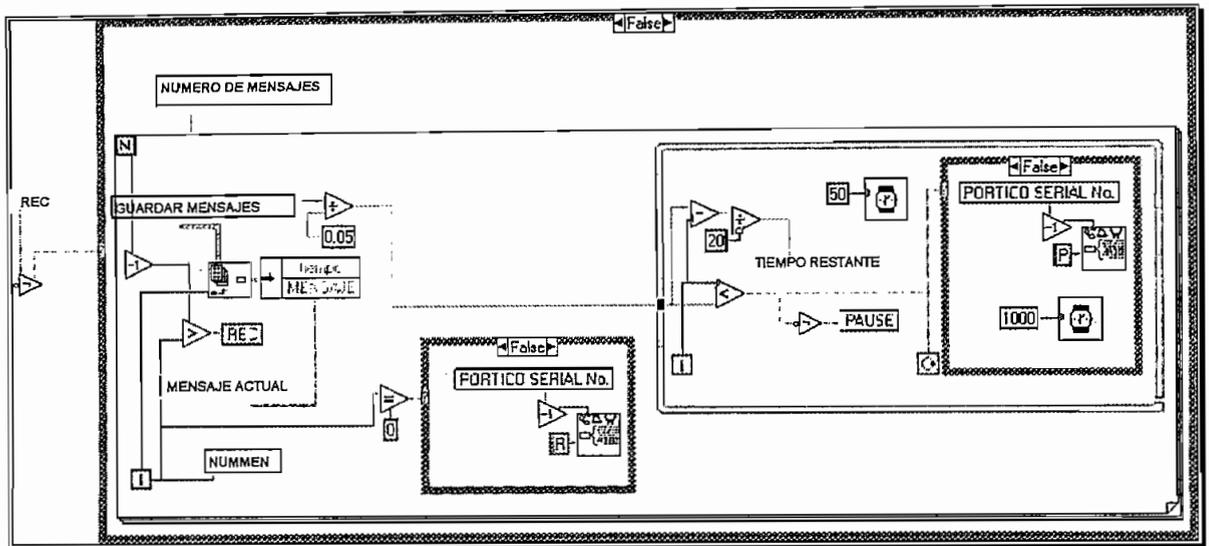


Figura 22. Diagrama de Grabación de mensajes

### 2.3.2.8 REPRODUCCIÓN AUTOMÁTICA

Este modo de funcionamiento es el más importante de este programa debido a que es el que le permite al computador obtener la información que llega desde la línea telefónica, así como responder a la misma con alguna acción hacia el microcontrolador, dentro del mismo programa o hacia un programa distinto.

En la reproducción automática se ingresa al caso verdadero de una estructura "case" cuando el número de datos en el buffer serial de entrada es once. Dentro de esta estructura se encuentra una secuencia que posee 5 escenas las cuales se describen a continuación:

En la primera escena se leen los once caracteres del pórtico serial.

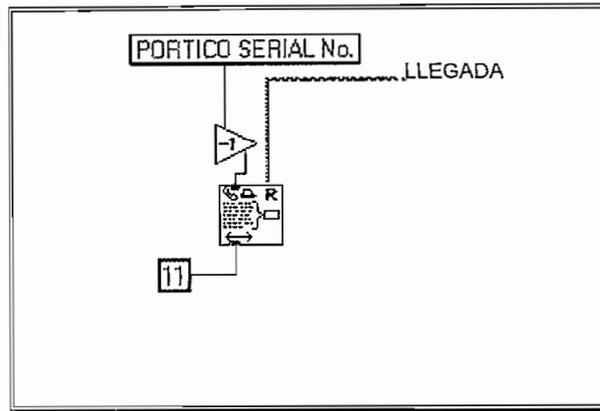


figura 23. Escena No. 0 de la secuencia para el modo de reproducción automática

En la segunda escena se obtiene un "array" que posee los valores numéricos de los 11 caracteres que han ingresado al buffer de recepción serial. También se obtiene otro "array" que contienen el valor numérico de todas las direcciones de origen de los mensajes gradados.

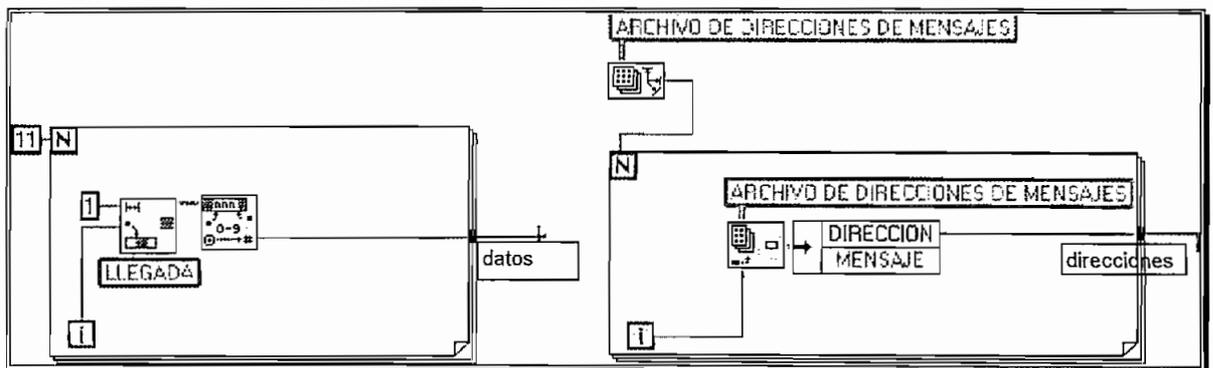


figura 24. Escena No.1 de la secuencia para el modo de reproducción automática

En la tercera escena se procede a enviar a través del pórtico serial a las direcciones de los mensajes correspondientes a los datos recibidos previamente. para esto se van extrayendo las direcciones del "array" de direcciones utilizando como

puntero a los datos recibidos. Finalmente se convierten estas direcciones en "strings" los cuales se concatenan y se envían al p rtico serial.

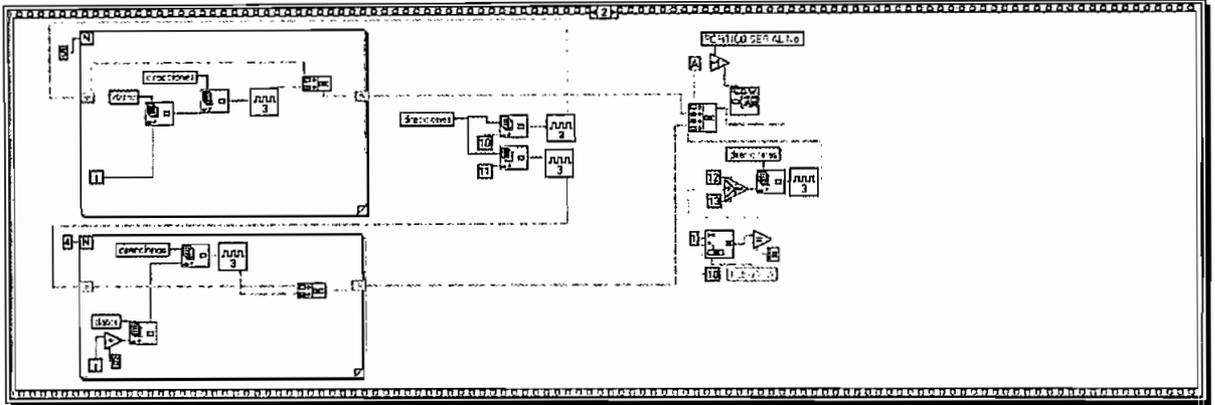


figura 25. Escena No. 2 de la secuencia para el modo de reproducci3n autom3tica

En la siguiente escena se tiene un lazo "while-loop" que espera por un dato desde el microcontrolador que corresponda a la aceptaci3n o cancelaci3n de la orden previamente dada.

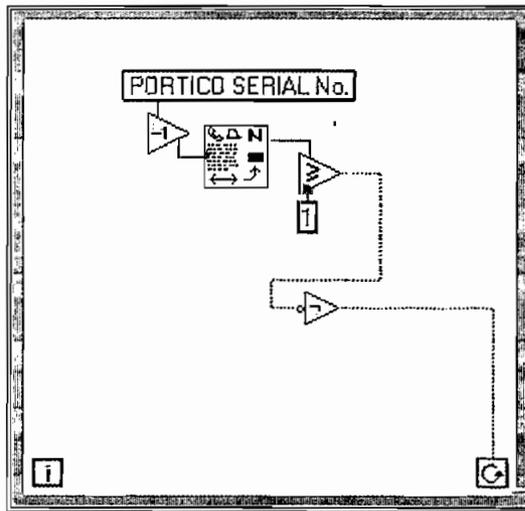


figura 26. Escena No.3 de la secuencia para el modo de reproducci3n autom3tica

En la escena No.4 se procede a identificar si el número recibido desde el pórtico serial es 1 o 2. En caso de ser 1, se estaría aceptando la orden y se procederá a comparar que la clave ingresada esté dentro del archivo de claves válidas. En caso de que sea una clave incorrecta, no se deberá efectuar ninguna acción, mientras que si la clave es correcta, se deberá proceder a guardar la orden en el archivo de órdenes recibidas. Los datos que entran a este archivo podrían ser utilizados como salidas para otros programas realizados en LabView u otro paquete computacional.

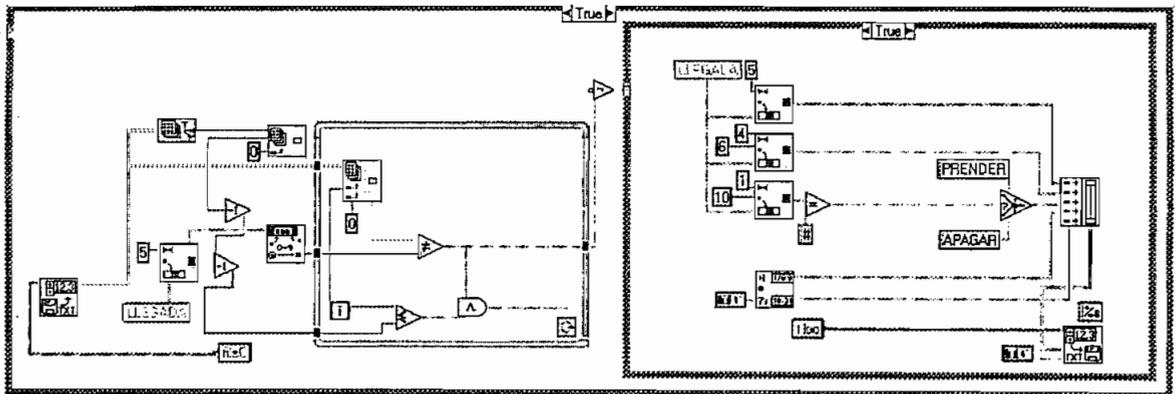


figura 27. Escena No.4

### 2.3.3 DIAGRAMA DE JERARQUÍAS.

En base a los diagramas mostrados, se puede observar que el programa completo utiliza muchas funciones llamadas VIs, la cuales se representan mediante bloques. sin embargo, cada una de éstas puede estar constituida de otras cuantas dependiendo de su complejidad. A continuación se tiene un diagrama de jerarquías donde se muestra esta situación. Los bloques superiores son los más complejos que utilizan los VIs inferiores a los cuales se encuentran conectados. De acuerdo a esto, se observa que el programa completo es un VI.



## CAPÍTULO 3. PRUEBAS y RESULTADOS

### 3.1 RESULTADOS

#### 3.1.1 CARACTERÍSTICAS CONSTRUCTIVAS

El equipo construido está alojado en una caja de plástico de las siguientes dimensiones:

20cm de largo

11cm de ancho

6cm de altura

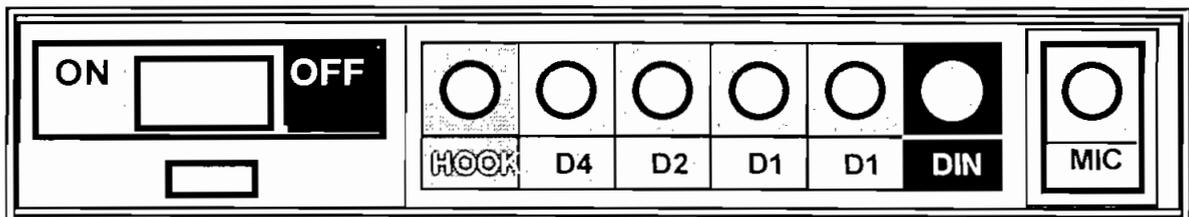
Esta caja aloja un circuito impreso que posee las siguientes dimensiones:

15 cm de largo

10 cm de ancho

El equipo posee un conector RJ-11 para la línea telefónica, un conector DB-9 macho para el pórtico serial, un conector para el micrófono externo y un cable con su respectivo enchufe para la alimentación de 115V.

El equipo posee un grupo LEDs indicadores que permiten



determinar el estado de funcionamiento del mismo. Estos se describen a continuación:

Figura 29. Tablero de control del equipo

D4,D2,D1 y D0, representan el código binario de los datos DTMF que ingresan a través de la línea telefónica. D0 es el bit menos significativo mientras que D4 es el más significativo. El LED DIN se enciende siempre que ingresa un dato DTMF, sin importar cual sea éste. El último LED de esta fila llamado HOLD es el que indica que se ha descolgado la línea telefónica. El led de forma rectangular que se encuentra en la parte inferior del interruptor de encendido es el que indica que el equipo está energizado y que la línea telefónica se ha conectado al receptor DTMF interno.

En la siguiente hoja se muestran fotografías del equipo, donde se observan sus conectores, controles externos así como también el circuito impreso con todos sus elementos.

### 3.1.2 PRESUPUESTO

En la siguiente tabla se tiene la lista de los elementos utilizados así como sus precios:

Elemento	Cantidad	Costo U.	Costo Total
74LS14	1	2500	2500
capacitores no polarizados 2.2uF 100V	4	500	2000
capacitores no polarizados 0.1uF 50V	10	350	3500
capacitores no polarizados 0.47uF 25V	1	700	700
capacitores no polarizados 1uF 25V	1	800	800
capacitores polarizados 1000uF 25V	1	3000	3000
capacitores polarizados 100uF 25V	1	2000	2000
capacitores polarizados 22uF 25V	1	1500	1500
capacitores polarizados 4.7uF 25V	1	1000	1000
resistencias varias de 1/4 W	23	150	3450
relés	2	4000	8000
receptor DTMF 273-1303	1	31000	31000
ULN2003	1	3000	3000
switch de dos posiciones y seis terminales	1	2500	2500
conector RJ45	1	2500	2500
cable gemelo #16 con enchufe incluido	1	2500	2500

LEDS	7	500	3500
resistencia tipo SIP 10k $\Omega$	1	1800	1800
crystal de cuarzo 7.3728MHz	1	4000	4000
crystal de cuarzo 3.579545MHz	1	4000	4000
diodos zener 5.1V 1W	2	1500	3000
diodos zener 15V 1W	2	1500	3000
diodos 1N4007	3	500	1500
diodos 1N4148	5	150	750
conector DB-9 macho	1	1500	1500
conector DB-9 hembra	1	1500	1500
cable de 9 hilos	1	2000	2000
transformador con tap central de 24V	1	15000	15000
LM7805	1	1500	1500
circuito impreso	1	48500	48500
microcontrolador INTEL 8751	1	82000	82000
CI grabador/reproductor de voz ISD1000A	1	85000	85000
parlante 0.25W 16 Ohm	1	6500	6500
optoacopladores 6N163	2	7000	14000
micrófono electret Radio Shack 270-092	1	16800	16800
zócalo de 8 pines	1	1500	1500
zócalo de 14 pines	1	2500	2500
zócalo de 16 pines	1	2800	2800
zócalo de 18 pines	1	3100	3100
zócalo de 28 pines	1	4200	4200
zócalo de 40 pines	1	4500	4500
caja de plástico	1	16000	16000
		<b>TOTAL S/.</b>	<b>400400</b>

tabla 6. Costos de los elementos utilizados

El circuito impreso que se utilizó es de dos lados. Utiliza aproximadamente 80 vias, las cuales fueron soldadas en ambos lados para lograr la conducción, lo mismo se hizo con los zócalos que requerían. En las siguientes hojas se indican cada uno de los lados de éste circuito impreso.

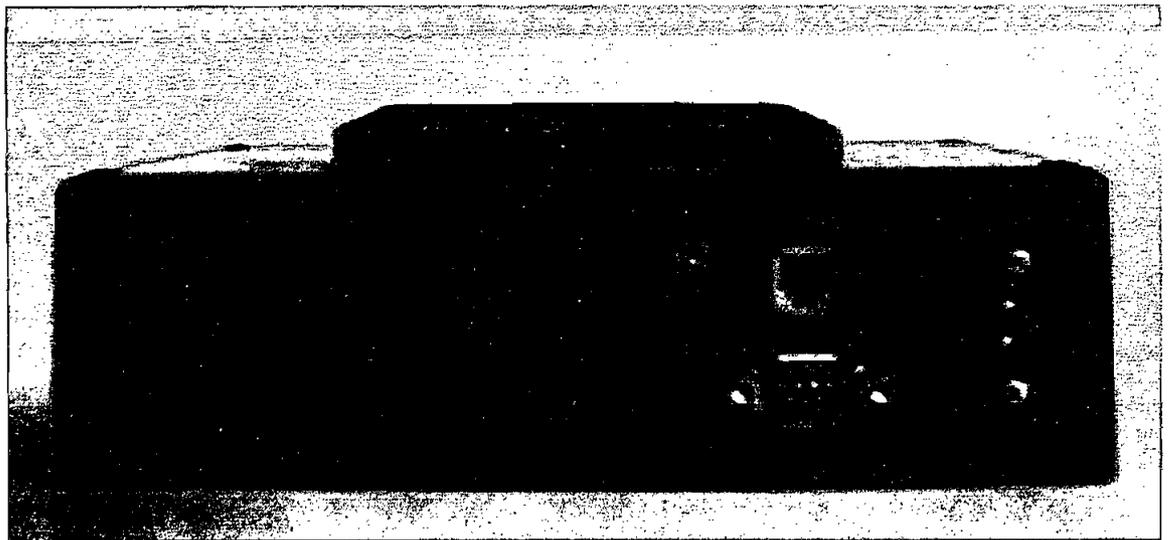
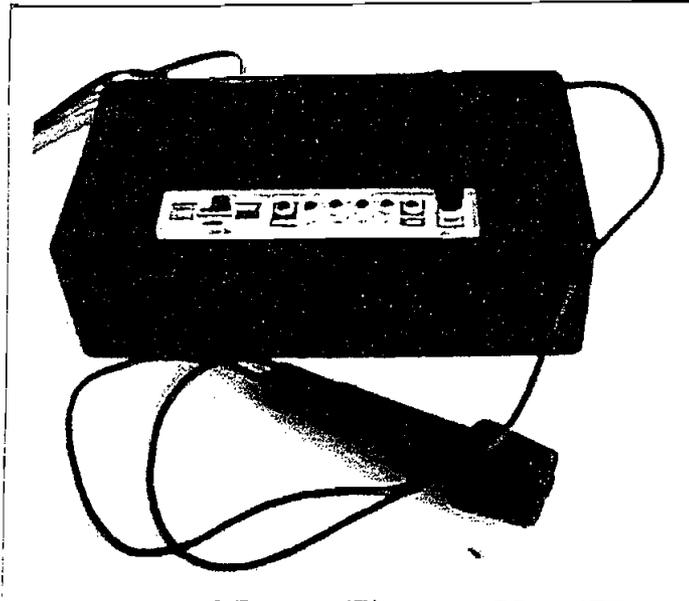
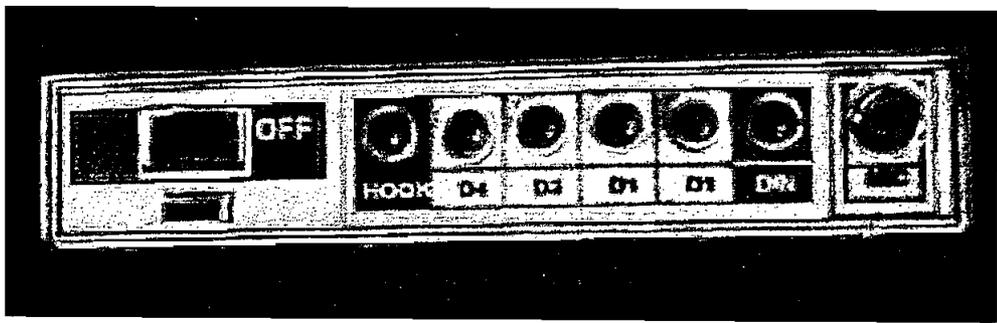


figura 30. Imágenes del equipo

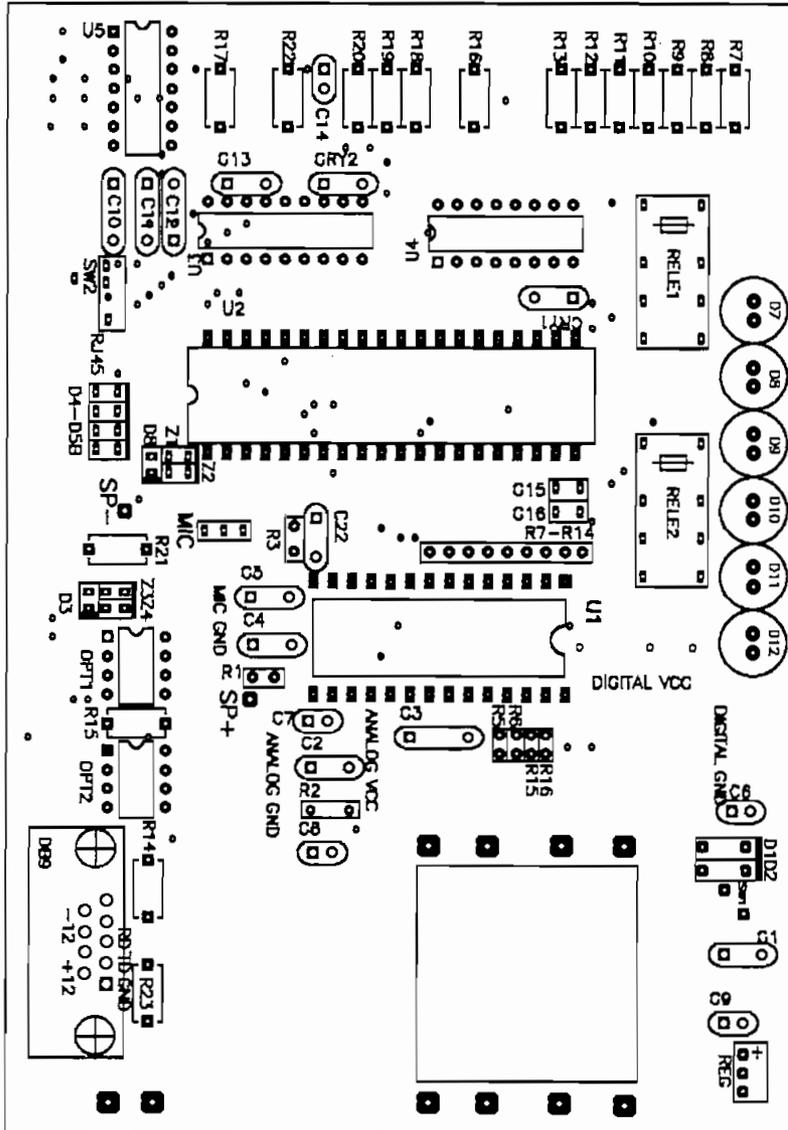


figura 31a. Componentes del circuito impreso

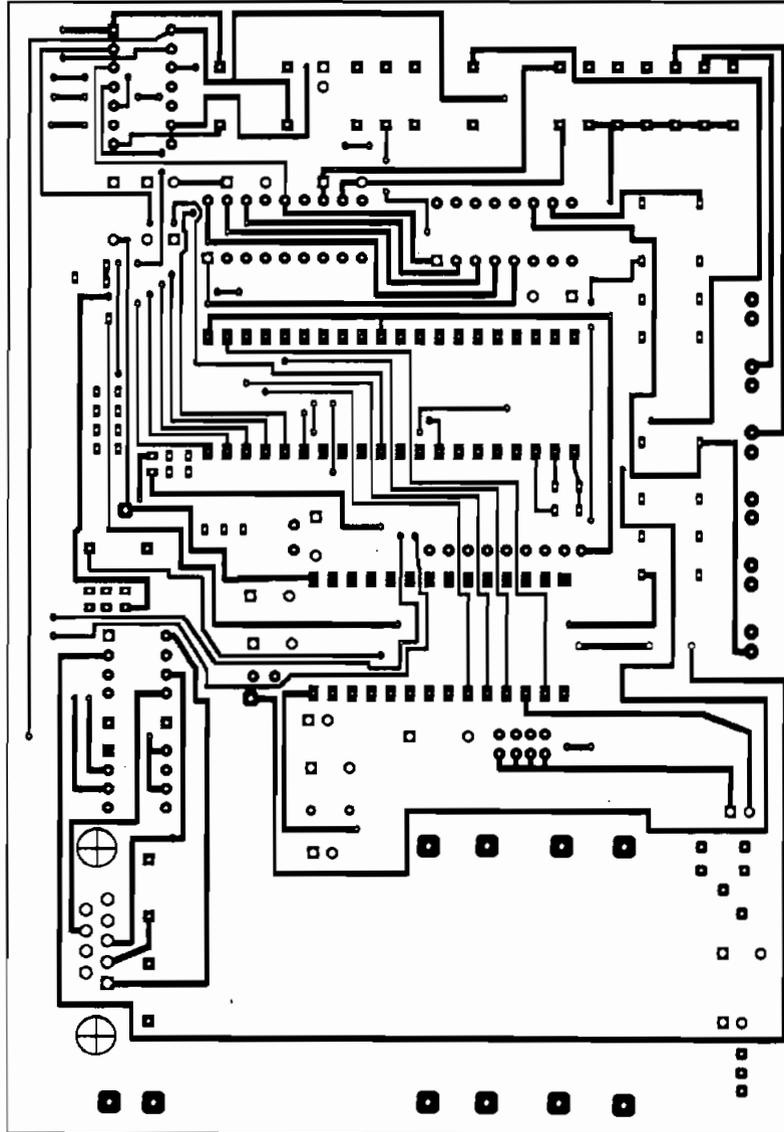


figura 31b. Cara superior del circuito impreso

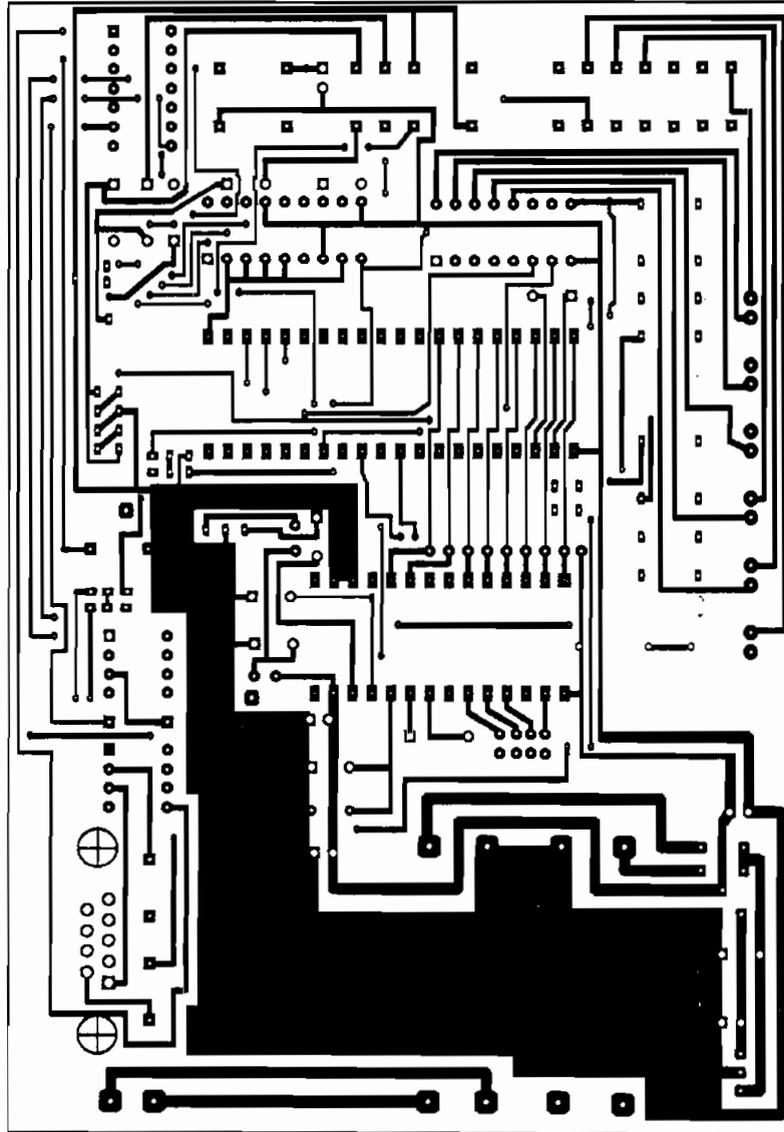


figura 31c. Cara inferior del circuito impreso

## 3.2 PRUEBAS

A continuación se describe detalladamente cada una de las funciones que realiza el equipo.

### 3.2.1 GRABACIÓN DE MENSAJES.

La grabación de mensajes se realiza con el siguiente procedimiento: En primer lugar se coloca en la tabla de grabación de mensajes, el tiempo en segundos de cada mensaje, seguido del mensaje correspondiente. Para seleccionar el número de mensaje se debe utilizar el control superior izquierdo que indica el número de mensaje. Luego de haber ingresado todos los mensajes con su respectiva duración se seleccionará en el control de número de mensajes, el número de mensajes a grabarse. Posteriormente, se presionará el botón "rec" y el computador automáticamente irá preguntando cada uno de los mensajes durante el tiempo programado para cada uno de ellos, en este tiempo paralelamente se estará grabando. Entre cada mensaje se tendrá una pausa de un segundo, la cual se indicará mediante el botón de "PAUSE" que se presionará automáticamente durante la pausa. Para poder utilizar el programa en el modo automático de reproducción, se deberá grabar 23 mensajes, los cuales se muestran a continuación:

Tiempo por mensaje	Mensaje
0.7	cero
0.7	uno
0.6	dos
0.6	tres

0.7	cuatro
0.7	cinco
0.6	seis
0.7	siete
0.7	ocho
0.7	nueve
0.7	clave
0.7	sector
0.7	prender
0.8	apagar
1	repita las
0.9	presione
0.7	para
0.9	terminar
0.9	aceptar
1.0	instrucciones
0.9	cancelar
0.9	ingrese
0.9	continuar

**tabla 5.** Archivo de grabación de mensajes

El tiempo por mensaje se debe ir escogiendo en base al número de sílabas de cada mensaje, sin embargo, debido a que el circuito grabador/reproductor de voz acepta solamente tiempos múltiplos de 0.125 segundos, se producirán tiempos en blanco, esto se debe a que la pronunciación no dura exactamente el tiempo disponible para cada mensaje y al combinar mensajes para formar frases se produce una pronunciación discontinua. Debido a que se dispone de un tiempo máximo de grabación de 20 segundos, se han seleccionado los mensajes a grabarse de tal manera que los

mensajes básicos grabados se puedan combinar para producir frases distintas. Si se tuviera disponible mayor cantidad de tiempo se podrían grabar varias frases individuales y no tener que combinar los mensajes para formarlas con lo que se eliminaría la pronunciación discontinua. Se debe notar que el tiempo que se indica en el control de grabación de mensajes no es muy exacto. Esto se puede comprobar sumando el tiempo de cada mensaje que en total no completa el número de segundos disponibles. Inicialmente se tenía programado un tiempo para cada mensaje que permitía completar exactamente los 20 segundos, pero se observó que con estos valores se conseguía grabar tres mensajes menos. En el caso de los mensajes actualmente grabados, se tiene que la suma de tiempos parciales nos dá 17.7 segundos que es menor a los 20 segundos disponibles, sin embargo el origen de la dirección del último mensaje está muy cercano a la dirección final de la memoria que es 159. Si se desea almacenar los tiempos de los mensajes grabados con sus correspondientes tiempo de duración, se puede guardarlos en el archivo de grabación de mensajes con solo presionar el botón "grabar" en el "path" escogido. Para recuperarlo solo se debe presionar el botón "recuperar".

### **3.2.2 CONTROLES PARA LA REPRODUCCIÓN MANUAL:**

Este modo de operación permite reproducir individualmente cada uno de los mensajes grabados. Para la reproducción, se ingresará una dirección entre 0 y 159 y a continuación se deberá presionar el botón "play", con lo cual el mensaje que esté en esa dirección se pronunciará hasta llegar a su final.

Este modo de operación funciona exactamente como se había previsto.

### **3.2.3 CONTROLES PARA LA INICIALIZACIÓN DE DIRECCIONES:**

Es necesario antes de utilizar el interfaz DTMF en modo automático, cargar al microcontrolador las direcciones bases de los mensajes grabados. Para determinar las direcciones correspondientes al inicio de los mensajes, es conveniente realizar una reproducción manual como ya se indicó. Debe tomarse en cuenta que mientras más cerca del inicio de la pronunciación del mensaje se seleccione como dirección de origen, las frases compuestas de varios mensajes se parecerán más a frases continuas sin cortes intermedios. Si se desea almacenar estas direcciones para evitar el tener que realizar la reproducción manual cada vez que se salga del programa o se apaguen los equipos, se los puede grabar en el ARCHIVO DE DIRECCIONES DE MENSAJES con solo presionar e botón de grabar en el "path" escogido. Luego de tener las direcciones correctas, se deberá presionar el botón de inicializar direcciones.

### **3.2.4 ARCHIVO DE CLAVES Y PERSONAS AUTORIZADAS**

Este archivo almacena las claves válidas y la persona a la cual pertenece cada una de éstas. Esta lista puede ser actualizada en cualquier momento utilizando los controles "grabar" y "recuperar" ubicados en la parte inferior al indicador de claves y personas autorizadas.

### **3.2.5 ARCHIVO DE ÓRDENES RECIBIDAS**

Este archivo permite almacenar la fecha, hora, clave, sector y orden de cada una de las instrucciones recibidas. Para visualizar un registro actualizado de las mismas, se debe presionar el botón "recuperar" que se encuentra bajo los indicadores del archivo de órdenes recibidas. Como aplicación de éste tipo de formato de instrucciones, se puede tener que la clave corresponda a una identificación personal, los dos primeros dígitos del sector correspondan al piso de un edificio, y los dos últimos dígitos permitan identificar un sector o equipo localizado en dicho piso, mientras que el último dato correspondiente a " \* " ó " # " puede indicar si se quiere prenderlo o apagarlo.

### **3.2.6 MODO AUTOMÁTICO DE REPRODUCCIÓN**

Este modo permite almacenar continuamente las órdenes almacenadas. Además en base a los datos que se obtengan, se puede transferirlos instantáneamente a otros programas de control de un edificio para que éste ejecute las instrucciones solicitadas.

### **3.2.7 INTERRUPTOR DEL PARLANTE**

Este interruptor permite que la salida del grabador/reproductor de voz esté conectada a un parlante auxiliar. La utilidad de éste parlante consiste en que se puede utilizar en el modo de reproducción manual para determinar las direcciones de origen o para revisar los mensajes grabados previamente. Cada vez que se activa o desactiva este parlante se escucha el funcionamiento de un relé que realiza esta función dentro del interfaz DTMF.

### 3.2.8 SELECTOR DEL NÚMERO DE TIMBRADAS

Este control permite escoger el número de timbradas a esperar antes de descolgar la línea telefónica; el rango está limitado entre 1 y 9 timbradas. Luego de escoger el número de timbradas con el control correspondiente, se debe presionar el botón de inicializar para que la orden sea ejecutada.

### 3.2.9 EJEMPLO DE FUNCIONAMIENTO DEL SISTEMA

A continuación se describirá un ejemplo de funcionamiento en el cual se mandan dos paquetes de instrucciones completos:

A continuación se enumeran los pasos que se debió seguir para inicializar al sistema para la ejecución del modo automático de reproducción:

- se selecciona el pórtico serial y se lo inicializa
- se selecciona el número de timbradas y se inicializa
- se recupera el archivo de direcciones de mensajes
- se inicializa las direcciones de mensajes con el botón correspondiente
- se selecciona el modo automático de reproducción

Luego de haber cumplido con estos pasos, ya se está en la posibilidad de recibir las llamadas.

Luego de marcar el número al que está conectado el equipo, el equipo espera que suenen tres timbradas antes de contestar. Se descuelga la línea y se escucha la frase "ingrese", "instrucciones" aunque en un tono que no se nota muy natural debido a la combinación de mensajes, a continuación se ingresan cinco números, el signo asterisco, cuatro números más y nuevamente el signo de número. A continuación se

escucha "clave", "uno", "dos" ,"tres", "cuatro", "cinco", "sector", "nueve", "ocho", "siete", "seis", "prender". Luego existe una pequeña pausa de 1 segundo, luego de la cual se escucha "presione", "uno", "para", "aceptar", "presione", "dos", "para", "cancelar". En este caso se presiona el número 1 y se escucha "presione", "uno", "para", "continuar", "presione", "dos", "para", "terminar". Se presiona el 1 y se escucha "ingrese", "instrucciones". Se repite las instrucciones, pero antes de terminar de ingresarlas, se escucha "repita las", "instrucciones". Esto se debe a que se posee un tiempo límite para ingresar las instrucciones, este tiempo de 10 segundos se toma a partir del primer dato presionado correspondiente al mismo grupo de instrucciones.

En el caso de no ingresar por el teclado todos los datos esperados por el microcontrolador y el computador existen tiempos de espera luego de los cuales se asumen las respuestas. Por ejemplo si no se acepta ni se cancela una orden, el microcontrolador la cancela 6 segundos luego de esperar por una respuesta. Igualmente si no se escoge ni continuar ni terminar, el microcontrolador elige terminar y cuelga 6 segundos luego de haber preguntado. En el caso de que el microcontrolador descuelgue y no reciba instrucciones en un tiempo de 23 segundos, entonces colgará.

# CAPÍTULO 4. CONCLUSIONES Y RECOMENDACIONES

## 4.1 CONCLUSIONES

-El programa LabView provee una gran versatilidad para ir formando aplicaciones complejas formadas en base a varios programas o subrutinas además se pueden correr programas que pueden interactuar entre sí al estar corriendo paralelamente en ventanas distintas.

-Los "clusters" utilizados en el LabView tienen una gran utilidad debido a que permiten empaquetar y desempaquetar objetos de cualquier tipo para utilizarlos como un solo ente o para separarlos en varios entes dependiendo la conveniencia.

-Una ventaja del sistema de interfaz DTMF implementado es que no requiere de ninguna característica especial del computador. Inclusive, en los anexos se tiene un programa realizado en Qbasic que posee características similares aunque no tan completas como las del programa LabView, pero que permitiría utilizar las características del Interfaz DTMF en un computador XT inclusive.

-El circuito integrado receptor DTMF posee una gran confiabilidad, ya que en todas las pruebas realizadas con el circuito definitivo, nunca se detectó una sola falla en la obtención de los datos DTMF. Sin embargo antes de tener armado el circuito impreso definitivo, se notó que si se ingresaban datos DTMF a través de la línea telefónica sin tener polarizado el circuito, éste cometía errores al

volverlo a polarizar. En base a este comportamiento observado se optó por colocar un interruptor en el circuito para que conecte o desconecte al mismo tiempo la polarización y la línea telefónica.

-El adecuado manejo de las tierras en el circuito impreso permitió reducir el ruido introducido en el sonido del circuito grabador/reproductor de voz, que se producía en el circuito armado en el "protoboard". Para realizar el trazado de los caminos de tierra se consideró que estos debían ser muy amplios para que su resistencia sea muy baja y no caiga voltaje en ellos, lo que les haría propensos a aceptar ruido. Otra consideración tomada fue que se debía tomar caminos individuales para los circuitos digitales y analógicos, ya que las continuas conmutaciones en los digitales podrían producir ruido que se introduciría en los circuitos analógicos.

-En el desarrollo de los circuitos de interfaz con la línea telefónica se debió considerar que las señales de la línea telefónica no son iguales en todos los sectores, por ejemplo, el valor pico que se alcanzaba en una timbrada podía variar hasta en 30 voltios dependiendo de la línea telefónica. Esto dio problemas en el funcionamiento inicial de la interfaz detectora de timbradas.

-En el manejo de los pórnicos de un microcontrolador, debe considerarse cuales pórnicos poseen resistencias de pull-up internamente, de no considerar esto, cuando se suponga que existe un 1L a la salida de uno de éstos, podría ser que se esté tan solo en un estado de alta impedancia.

-Inicialmente se estuvo utilizando una fuente "switching" en la polarización del circuito, pero se decidió cambiarla por una fuente lineal debido a que introducía mucho ruido en la línea telefónica.

-Cuando se posee un relé dentro de un circuito que posea un microcontrolador, es indispensable colocar un diodo de conmutación en paralelo con la bobina del relé. Si no se hace esto, el microcontrolador podría funcionar incorrectamente y realizar acciones impredecibles.

-Si se hubiera dispuesto de unos cuantos segundos más para pronunciar los 23 mensajes grabados en el equipo, en el modo automático, las frases formadas tendrían una pronunciación más natural y uniforme

-El procedimiento seguido para la elaboración de esta tesis permitió realizarla con el mejor aprovechamiento del tiempo. Se empezó por probar el circuito DTMF conectando a éste LEDs. Cuando ya se tenía funcionando esta etapa, se la conectó a un módulo didáctico "Microlab". Con este módulo se logró pasar la información de los datos DTMF recibidos por la línea telefónica hacia el computador. El computador inicialmente fue programado en Qbasic, debido a que se tenía un buen conocimiento de este lenguaje. El siguiente paso fue armar el circuito receptor DTMF junto con el microcontrolador en un "protoboard". Cuando ya se tenía funcionando el circuito hasta este punto, se implementó aparte el circuito grabador/reproductor de voz, el cual se operaba con "dip switches". Luego de comprobar que el circuito grabador/reproductor de voz funcionaba correctamente, se lo integró al circuito del microcontrolador y se lo empezó a manejar desde el computador. En este punto, se integró el

manejar desde el computador. En este punto, se integró el circuito a la línea telefónica, construyendo los interfaces correspondientes. Cuando ya se tenía funcionando todo el sistema en Qbasic, se creó el programa definitivo en LabView. Como último paso se construyó el circuito impreso y se lo montó en su caja correspondiente.

## 4.2 RECOMENDACIONES

-Para ampliar la memoria de grabación de mensajes, se puede utilizar varios circuitos grabadores/reproductores de voz conectados en cascada. Si se hace esto, por cada circuito ISD1000A adicionado se aumentarían 20 segundos de grabación de mensajes. El incremento de estos circuitos grabadores /reproductores de voz, vendría acompañado de un mayor número de líneas de direcciones de mensajes que controlar, por lo que se complicaría un poco más el software y el hardware del equipo.

-Al realizar conexiones de la línea telefónica a un equipo, debe considerarse que debe existir un aislamiento entre tierras, debido a que de no hacerlo, se puede introducir ruido en la línea.

-Al realizar el aislamiento con optoacopladores del pórtico serial es conveniente seleccionar los optoacopladores adecuados debido a que de no hacerlo, no se podrá tener una buena velocidad de transmisión.

-Cuando se está armando un circuito que debe utilizar la línea telefónica, se debe considerar que de un momento a otro

podría llegar un voltaje cercano a los 180 voltios debido a una timbrada. Por lo tanto, el primer circuito que se debe armar es el que permite limitar el voltaje de entrada desde la línea telefónica hacia algún circuito electrónico sensible.

- Una configuración que podría mejorar las capacidades del sistema diseñado y construido, aunque requeriría de un computador que posea una tarjeta de sonido que permita grabar y reproducir archivos de sonido, se puede describir a continuación:

El microcontrolador continuaría con todas la funciones actuales, con excepción del manejo del circuito grabador/reproductor de sonido, el cual ya no existiría y sería reemplazado por la tarjeta de sonido del computador que sería manejada directamente desde el programa del computador. Adicionalmente se requeriría un circuito de interfaz con aislamiento entre la salida de sonido del computador y la línea telefónica.

- Así como existen soluciones más complejas que utilizarían todas las características tecnológicas de un computador actual, existen otras que no requerirían ni siquiera de un computador. Por ejemplo, si se quiere controlar desde el teclado de un teléfono de tonos a algunos equipos eléctricos del hogar, se podría utilizar el circuito descolgador de la línea a más del receptor DTMF con su interfaz con la línea telefónica, el microcontrolador y unos cuantos relés. En este caso el microcontrolador cumpliría las funciones de comparar los datos DTMF recibidos con claves guardadas en su memoria de programa mediante una tabla o de lo contrario en

una memoria de datos externa no volátil. De acuerdo a los datos recibidos, el microcontrolador accionaría los relés que a su vez podrían controlar algunos equipos o contactores, dependiendo de la potencia requerida.

## **ANEXOS**

# PROGRAMA EN ASSEMBLER PARA EL MICROCONTROLADOR

```
;*****MAPA DE MEMORIA*****
;*          127  (clave)          *
;*****
;*          126  (clave)          *
;*****
;*          125  (clave)          *
;*****
;*          124  (clave)          *
;*****
;*          123  (clave)          *
;*****
;* 122(asterisco de fin de clave) *
;*****
;*          121(funcion)          *
;*****
;*          120(funcion)          *
;*****
;*          119(funcion)          *
;*****
;*          118(funcion)          *
;*****
;*          117 ( funcion * o #)  *
;*****
;r5 utilizado para el temporizador cero
;r1 del banco1 UTILIZADO PARA APUNTAR LA DIRECCION DE ADRESS
CLAVE      BIT  01H  ;INDICA QUE TIPO DE DATO INGRESA A TLFNO
DESCUELGO  BIT  02H  ;INDICA QUE DEBE COLGAR EL TELEFONO
OSCIL1     EQU      115      ;PARA RETARDO 50 ms
OSCIL2     EQU      114      ;PARA RETARDO 50 ms
```

TIMBRE	EQU	113	; PARA NUMERO DE TIMBRADAS
TIMBREAUX	EQU	112	; PARA CONTEO DE TIMBRADAS
NUM1	EQU	111	; CENTENAS DE ADRESS
NUM2	EQU	110	; DECENAS DE ADRESSS
NUM3	EQU	109	; UNIDADES DE ADRESS
CUENTAFRASES	EQU	108	; CONTADOR DE MENSAJES DEL COMPUTADOR
TTIMBRE1	EQU	107	; PARA RETARDO DE TIMBRADA
TTIMBRE2	EQU	106	; PARA RETARDO DE TIMBRADA
TTIMBRE3	EQU	105	; PARA RETARDO DE TIMBRADA
TTIMBRE4	EQU	104	; PARA TIEMPO DE LLAMADA
AUXILIAR	EQU	103	
ADRESS	EQU	101	; DIRECCION DEL LORO
; EN LA DIRECCION 100 HACIA ABAJO SE ANOTAN LAS FRASES DEL COMPUTADOR			

;100  
;99  
;98  
;97  
;96  
;95  
;94  
;93  
;92  
;91  
;90  
;89  
;88

CONTINUAR	EQU	85
INGRESE	EQU	84
CANCELAR	EQU	83
INSTRUCCIONES	EQU	82
ACEPTAR	EQU	81
TERMINAR	EQU	80
PARA	EQU	79
PRESIONE	EQU	78
REPITALAS	EQU	77
UNO	EQU	76
DOS	EQU	75
DVA	EQU	74
DVB	EQU	73
DVC	EQU	72
DVA1	EQU	71
DVB1	EQU	70
DVC1	EQU	69

ORG 0000H

LJMP INICIO

ORG 0013H

LJMP EXTERNA ;CUANDO EL DV CAMBIA DE CERO A POSITIVO

org 00bh

ljmp fintiempo ;manda a disminuir r5

org 0023h

ljmp comunicacion

ORG 0030H

INICIO:

MOV SP,#30H

;INICIALIZACION DE LOS PORTICOS

```

SETB P1.0      ;DTMF0
SETB P1.1      ;DTMF1
SETB P1.2      ;DTMF2
SETB P1.3      ;DTMF3
SETB P1.4      ;BIT DE ENTRADA DEL TIMBRE
CLR P1.5        ;CASI SIEMPRE COLGADO
SETB P1.6        ;CASI SIEMPRE EL NORMALMENTE ABIERTO
CERRADO
CLR P1.7        ;LED INDICADOR DE AYUDA

MOV P2,#00H     ;PORTICO 2 DIRECCIONANDO CERO

SETB P0.0      ;EL MODO PRESTABLECIDO ES PLAY
CLR P0.1        ;LISTO PARA RECIBIR PULSO POSITIVO
SETB P0.2      ;LISTO PARA ESPERAR UN PULSO NEGATIVO
SETB P0.3      ;LISTO PARA RECIBIR BIT DEL EOM

SETB DESCUELGO

;INICIALIZACION DE DIRECCIONES DE PRONUNCIACIONES

MOV CONTINUAR,#00
MOV INGRESE,#153
MOV CANCELAR,#143
MOV INSTRUCCIONES,#139
MOV ACEPTAR,#132
MOV TERMINAR,#125
MOV PARA,#118
MOV PRESIONE,#111
MOV REPITALAS,#104
MOV UNO,#6
MOV DOS,#13

```

```
SETB CLAVE          ;EL SIGUIENTE DATO ES DE CLAVE (CLAVE=1)
MOV R0,#127         ;PUNTERO DE MEMORIA DE CLAVES Y FUNCIONES
MOV A,#00H
```

```
MOV     SCON,#01010000B
MOV     PCON,#00H
MOV     TH1,#0FEH      ;9600 BAUDIOS
MOV     TL1,#0FEH
MOV     TMOD,#00100001B
mov     th0,#00h
mov     t10,#00h
```

```
MOV     TIMBRE,#2      ;NORMALMENTE ESPERA 7 TIMBRADAS
MOV     TIMBREAU,TIMBRE
```

```
setb    et0
SETB    IT1           ;INTERRUPCION ACTIVADA POR FLANCO
setb    tr1
SETB    EA
SETB    EX1          ;HABILITA INTERRUPCION DEL DV
setb    es
```

PRINCIPAL:

;A CONTINUACION DETERMINO SI HAY FLANCO NEGATIVO DEL TIMBRE

```
MOV     TTIMBRE1,#25
SEGUIRT1:
MOV     TTIMBRE2,#250
SEGUIRT2:
MOV     TTIMBRE3,#250
SEGUIRT3:
JNB    P1.4,SEGUIRT
```

```
DJNZ TTIMBRE3,SEGUIRT3
DJNZ TTIMBRE2,SEGUIRT2
DJNZ TTIMBRE1,SEGUIRT1
```

```
SJMP SALPRINCIPAL
```

```
SEGUIRT:
```

```
JB P1.4,ACTUART          ;SENSOR DE FLANCO DEL TIMBRE
SJMP SEGUIRT
```

```
ACTUART:
```

```
setb p1.7                ;PIN INDICADOR AUXILIAR
DJNZ TIMBREAU,PRINCIPAL
setb P1.5                 ;DESCUELGA EL TIMBRE
```

```
CLR EA
```

```
SETB P0.0 ;MODO PLAY P/R=1L
```

```
MOV P2,INGRESE          ;SE MANDA LA DIRECCION DEL MENSAJE
```

```
NOP
```

```
SETB P0.1               ;UN PULSO POSITIVO EN PD
```

```
ACALL r50ms
```

```
CLR P0.1
```

```
acall r50ms
```

```
CLR P0.2                ;UN PULSO NEGATIVO EN CE
```

```
nop
```

```
SETB P0.2
```

```
ACALL FLANCO
```

```

SETB P0.0                ;MODO PLAY P/R=1L

MOV P2, INSTRUCCIONES    ;SE MANDA LA DIRECCION DEL
MENSAJE
NOP
SETB P0.1                ;UN PULSO POSITIVO EN PD
ACALL r50ms
CLR P0.1
acall r50ms
CLR P0.2                ;UN PULSO NEGATIVO EN CE
nop
SETB P0.2

ACALL FLANCO

SETB EA

;AQUI VIENE UN RETARDO
MOV TTIMBRE1, #20        ;CUANDO CALCULE BIEN REPRESENTA N SEG.
SEGUIRD1:
MOV TTIMBRE2, #4         ;LOS SIGUIENTES LAZOS IGUALES A
1 SEG.
SEGUIRD2:
MOV TTIMBRE3, #250
SEGUIRD3:
MOV TTIMBRE4, #250
SEGUIRD4:

JNB  DESCUELGO, SALPRINCIPAL

DJNZ TTIMBRE4, SEGUIRD4
DJNZ TTIMBRE3, SEGUIRD3

```

```
DJNZ TTIMBRE2,SEGUIDR2
```

```
DJNZ TTIMBRE1,SEGUIDR1
```

```
SALPRINCIPAL:
```

```
clr P1.5 ;VUELVO A COLGAR EL TELEFONO
```

```
clr p1.7 ;LED INDICADOR AUXILIAR
```

```
SETB DESCUELGO ;INICIALIZA EL BIT PARA LA SIGUIENTE  
LLAMADA
```

```
MOV TIMBREAU, TIMBRE
```

```
SJMP PRINCIPAL
```

```
EXTERNA:
```

```
push acc
```

```
    cjne r0,#127,continuo ;distinto primer dato implica  
no reinicio t
```

```
    mov r5,#96 ;inicializo el lazo del  
temporizador
```

```
    setb tr0 ;reinicio t
```

```
continuo:
```

```
    CLR EX1 ;DESHABILITA INTERRUPCION DEL DV
```

```
    MOV A,P1 ;LEE EL PORTICO1
```

```
    ORL A,#11110000B ;SELECCIONA SOLO LOS 4 BITS
```

```
MENOS S.
```

```
    ANL A,#00001111B ;DEJA EXPRESADO EN UN NUMERO  
ENTRE 0 y F
```

```
    MOV @R0,A ;COLOCA EL PORTICO DONDE INDICA EL  
PUNTERO
```

```
    ; quito JB P3.3,$ ESPERA HASTA QUE SE SUELTE LA  
TECLA (DV)
```

```
    JNB CLAVE,FUNCION ;SALTA A FUNCION SI EL DATO ES  
FUNCION (CLAVE=0)
```

```

CJNE @R0,#00001011B,NOASTERISCO ;SALTA A NOASTERISCO SI
DATO <> *
CJNE R0,#122,REPITOCLAVE ;SI ES ASTERISCO Y NO ES ULTIMO
DATO, ENTON-
;CES SALTO Y VUELVO A INGRESAR TODA
LA CLAVE
DEC R0
CLR CLAVE ;LOS SIGUIENTES DATOS SERAN DE
FUNCIONES
SJMP FINALEXTERNA

FUNCION: ;AQUI SALTO SI EL DATO ES DE UNA
FUNCION

CJNE @R0,#00001011B,NOASTERISCOF ;SI DATO <> * SALTA A
NOASTERISCOF
CJNE R0,#117,REPITOFUNCION ;SI ES * Y NO ES EL ULTIMO
DATO,
;ENTONCES SALTO Y VUELVO A
INGRESAR
;TODA LA FUNCION
SETB CLAVE ;YA TERMINARON LOS DATOS CON EL
ASTERISCO
MOV R0,#127 ;PONGO EL PUNTERO AL INICIO
ACALL SERIAL
SJMP FINALEXTERNA

```

```

NOASTERISCOF:
CJNE @R0,#00001100B,ESDATOF ;SI DATO(F) <> * Y # SALTA A
ESDATOF
CJNE R0,#117,REPITOFUNCION ;SI ES #f Y NO ES EL ULTIMO
DATO,

```

```

;ENTONCES SALTO Y VUELVO A INGRESAR TODA
LA
;FUNCION
SETB CLAVE ;YA TERMINARON LOS DATOS CON EL
ASTERISCO
MOV R0,#127 ;PONGO EL PUNTERO AL INICIO
ACALL SERIAL
SJMP FINALEXTERNA

ESDATOF: ;ACCIONES LUEGO DE CADA DATO DE LA FUNCION <>
* Y #
DEC R0 ;BAJO UN LUGAR AL PUNTERO
CJNE R0,#116,FINALEXTERNA ;SI ES EL ULTIMO DATO DE LA CLAVE Y
<> DE *,#
;ENTONCES PERMITO INGRESAR NUEVO DATO
MOV 121,120 ;EN LAS SIGUIENTES LINEAS RECORRO UN
LUGAR LOS DATOS
;HACIA ARRIBA ELIMINANDO EL MAS ANTIGUO
MOV 120,119
MOV 119,118
MOV 118,117
MOV R0,#117 ;COLOCO EL PUNTERO EN LA LOCALIDAD
DEL * DE CLAVE

SJMP FINALEXTERNA

REPITOFUNCION:
MOV R0,#121
SJMP FINALEXTERNA

REPITOCLAVE: ;ACCIONES PARA INGRESAR NUEVAMENTE TODA
LA CLAVE

```

```

MOV R0,#127                ;POSICIONO EL PUNTERO EN PRIMERA
LOCALIDAD DECLAVE
    SJMP FINALEXTERNA     ;SALGO PARA PERMITIR QUE SE INGRESEN
NUEVOS DATOS
NOASTERISCO:              ;ACCIONES LUEGO DE CADA DATO DE LA CLAVE
<> *
    DEC R0                ;BAJO UN LUGAR AL PUNTERO
    CJNE R0,#121,FINALEXTERNA ;SI ES EL ULTIMO DATO DE LA
CLAVE Y <> *,
                                ;ENTONCES PERMITO INGRESAR NUEVO
DATO
MOV 127,126                ;EN LAS SIGUIENTES LINEAS RECORRO UN
LUGAR LOS DATOS
                                ;HACIA ARRIBA ELIMINANDO EL MAS ANTIGUO
    MOV 126,125
    MOV 125,124
    MOV 124,123
    MOV 123,122
    MOV R0,#122            ;COLOCO EL PUNTERO EN LA LOCALIDAD
DEL * DE CLAVE

    SJMP FINALEXTERNA

FINALEXTERNA:
    pop acc
    SETB EX1              ;HABILITO LA INTERRUPCION EXTERNA DEL DV
    RETI

SERIAL:                    ;ENVIA LA INFORMACION RECOGIDA AL
COMPUTADOR
    CLR ES
    PUSH ACC

```

;A CONTINUACION PARO EL TIMER

```
clr tf0
CLR TR0
mov t10,#00h
mov th0,#00h
```

```
MOV R1,#127
```

```
MOV R7,#11
```

REPITOSERIAL:

```
MOV A,@R1
CJNE A,#10,NOESCERO
ESCERO:
MOV A,#30H
SJMP MANDANUM
```

NOESCERO:

```
CJNE A,#11,NOESTERIS
ESTERIS:
MOV A,#2AH ;PONGO EN ACC ASCII DE ASTERISCO
SJMP MANDANUM
```

NOESTERIS:

```
CJNE A,#12,NOESN
ESN:
MOV A,#23H ;PONGO EN ACC ASCII DE NUMERO
SJMP MANDANUM
```

NOESN:

```
ADD A,#30H
```

MANDANUM:

```
MOV SBUF,A
JNB TI,$
CLR TI
DEC R1
```

```
DJNZ R7,REPITOSERIAL
```

```
MOV R0,#127 ;LUEGO DE TERMINAR SERIAL INICIALIZO  
PUNTERO  
POP ACC  
SETB ES  
RET
```

```
fintiempo:
```

```
PUSH ACC  
clr tf0  
mov tl0,#00h  
mov th0,#00h  
SETB tr0  
djnz r5,nuevolazo  
mov r0,#127 ;si termino 10seg empiezo desde el  
primer dato  
setb clave ;lo siguiente a ingresar es clave no  
funcion  
CLR TR0  
;a continuacion le hago decir <repita las instrucciones>  
  
SETB P0.0 ;P/R=1L  
MOV A,REPITALAS  
MOV P2,A  
nop  
SETB P0.1 ;UN PULSO POSITIVO EN PD  
ACALL r50ms  
CLR P0.1  
acall r50ms  
CLR P0.2 ;UN PULSO NEGATIVO EN CE  
NOP  
SETB P0.2
```

```
ACALL FLANCO
```

```
SETB P0.0      ;P/R=1L
```

```
MOV A, INSTRUCCIONES
```

```
MOV P2, A
```

```
nop
```

```
SETB P0.1      ;UN PULSO POSITIVO EN PD
```

```
ACALL r50ms
```

```
CLR P0.1
```

```
acall r50ms
```

```
CLR P0.2      ;UN PULSO NEGATIVO EN CE
```

```
NOP
```

```
SETB P0.2
```

```
ACALL FLANCO
```

```
nuevolazo:
```

```
POP ACC
```

```
reti
```

```
r50ms:
```

```
MOV OSCIL2, #120
```

```
CALLADO2:
```

```
MOV OSCIL1, #250
```

```
CALLADO1:
```

```
DJNZ OSCIL1, CALLADO1
```

```
DJNZ OSCIL2, CALLADO2
```

RET

COMUNICACION:

push acc

;ME FALTA PONER TIMEOUT DE ERROR EN ESTA SUBROUTINA Y EN EL  
COMPUTADOR

CLR EA

CLR RI

MOV A, SBUF

CJNE A, #71, NOGRABAR ;letra G

GRABAR:

;A CONTINUACION SE MANDAN LOS COMANDOS AL LORO PARA QUE GRABE

clr P0.0 ;P/R=0L

NOP

MOV P2, #208 ;ADRESS=11010000B (GRABAR  
SECUENCIALMENTE)

nop ;TAL VEZ INNECESARIO

SETB P0.1 ;PULSO POSITIVO EN PD

acall r50ms ;TAL VEZ UN SOLO RETARDO

CLR P0.1 ;TERMINA EL PULSO POSITIVO EN PD

acall r50ms

CLR P0.2 ;MANTENGO CE EN 0L MIENTRAS GRABO

AJMP SALIDACOM

NOGRABAR:

CJNE A,#82,NOREPITOGRAB ;letra R

REPITOGRAB:

;A CONTINUACION SE MANDAN LOS COMANDOS PARA SEGUIR GRABANDO

CLR P0.2 ;MANTENGO CE=0L MIENTRAS GRABO

AJMP SALIDACOM

NOREPITOGRAB:

CJNE A,#80,NOPAUSA ;letra P

PAUSA:

; A CONTINUACION ESTAN LOS COMANDOS PARA PARAR DE GRABAR

SETB P0.2 ;MANTENGO CE EN 1L MIENTRAS NO GRABO

AJMP SALIDACOM

NOPAUSA:

CJNE A,#72,NOESMANUAL ;letra H

MANUAL:

;MODO MANUAL DE DIRECCIONES A REPRODUCIR INSTANTANEAMENTE

JNB RI,\$ ;LEO CENTENAS DE ADDRESS

CLR RI

MOV A,SBUF

SUBB A,#30H

MOV NUM1,A

JNB RI,\$ ;LEO DECENAS DE ADDRESS

CLR RI

MOV A,SBUF

SUBB A,#30H

MOV NUM2,A

JNB RI,\$ ;LEO UNIDADES DE ADDRESS

CLR RI

MOV A,SBUF

SUBB A,#30H

```
MOV NUM3,A
```

```
;A CONTINUACION SE PRONUNCIA EL MENSAJE
```

```
SETB P0.0 ;MODO PLAY P/R=1L
```

```
MOV A,NUM1 ;EL SIGUIENTE GRUPO DE INSTRUCCIONES  
FORMA ADDRESS
```

```
MOV B,#100
```

```
MUL AB
```

```
MOV ADDRESS,A
```

```
MOV A,NUM2
```

```
MOV B,#10
```

```
MUL AB
```

```
ADD A,ADDRESS
```

```
ADD A,NUM3
```

```
MOV ADDRESS,A
```

```
MOV P2,ADDRESS ;SE MANDA LA DIRECCION DEL MENSAJE
```

```
NOP
```

```
SETB P0.1 ;UN PULSO POSITIVO EN PD
```

```
ACALL r50ms
```

```
CLR P0.1
```

```
acall r50ms
```

```
CLR P0.2 ;UN PULSO NEGATIVO EN CE
```

```
nop
```

```
SETB P0.2
```

```
;A CONTINUACION VIENE UN RETARDO PARA QUE NO SE  
INTERRUMPA
```

```
;EN LUGAR DE ESTE RETARDO SE DEBERIA UTILIZAR EL EOM
```

```
ACALL FLANCO
```

```
clr ea
```

```
AJMP SALIDACOM
```

NOESMANUAL:

CJNE A, #84, NOESCOMANDO ;letra T

ACALL LECTCOMP

ACALL TRANSNUM

MOV REPITALAS, AUXILIAR

ACALL LECTCOMP

ACALL TRANSNUM

MOV PRESIONE, AUXILIAR

ACALL LECTCOMP

ACALL TRANSNUM

MOV PARA, AUXILIAR

ACALL LECTCOMP

ACALL TRANSNUM

MOV TERMINAR, AUXILIAR

ACALL LECTCOMP

ACALL TRANSNUM

MOV ACEPTAR,AUXILIAR

ACALL LECTCOMP

ACALL TRANSNUM

MOV INSTRUCCIONES,AUXILIAR

ACALL LECTCOMP

ACALL TRANSNUM

MOV CANCELAR,AUXILIAR

ACALL LECTCOMP

ACALL TRANSNUM

MOV INGRESE,AUXILIAR

ACALL LECTCOMP

ACALL TRANSNUM

MOV CONTINUAR,AUXILIAR

ACALL LECTCOMP

ACALL TRANSNUM

MOV UNO,AUXILIAR

ACALL LECTCOMP

ACALL TRANSNUM

MOV DOS,AUXILIAR

AJMP SALIDACOM

;LO QUE VIENE LEE LOS DATOS DE LA CLAVE,SECTOR Y ORDEN (MODO  
AUTO)

NOESCOMANDO:

CJNE A,#65,NOESAUTO ;NO ES A

sjmp escapol

noesauto:

ajmp noesautol

escapol:

JNB RI,\$

CLR RI

MOV A,SBUF ;SE LEE CENTENAS DE ADDRESS

SUBB A,#30H

MOV NUM1,A

JNB RI,\$ ;LEO DECENAS DE ADDRESS

CLR RI

MOV A,SBUF

SUBB A,#30H

MOV NUM2,A

JNB RI,\$ ;LEO UNIDADES DE ADDRESS

CLR RI

```
MOV A, SBUF
SUBB A, #30H
MOV NUM3, A
```

```
ACALL DECODADDRESS
```

```
setb rs0
```

```
MOV R1, #101          ;PASO LA DIRECCION BASE A R1 del banco 1
DEC R1                ;AVANZO PUNTERO
MOV A, ADDRESS
MOV @R1, A           ;ORIGEN DE MENSAJES AQUI
```

```
;A CONTINUACION DEBO HACER LAS LECTURAS EN LAZO
```

```
MOV CUENTAFRASES, #11
```

```
DIRECCION:
```

```
JNB RI, $            ;LEO CENTENAS DE ADDRESS
CLR RI
MOV A, SBUF
SUBB A, #30H
MOV NUM1, A
```

```
JNB RI, $            ;LEO DECENAS DE ADDRESS
CLR RI
MOV A, SBUF
SUBB A, #30H
MOV NUM2, A
```

```
JNB RI, $            ;LEO UNIDADES DE ADDRESS
CLR RI
MOV A, SBUF
SUBB A, #30H
```

```
MOV NUM3,A
```

```
ACALL DECODADDRESS
```

```
DEC R1
```

```
MOV A,ADDRESS
```

```
MOV @R1,A
```

```
DJNZ CUENTAFRASES,DIRECCION
```

```
;A CONTINUACION DEBO HACER UN LAZO DE PRONUNCIACIONES
```

```
MOV CUENTAFRASES,#12
```

```
MOV R1,#101
```

```
NUEVAFRASE:
```

```
    DEC R1
```

```
    MOV A,@R1
```

```
    MOV ADDRESS,A
```

```
    SETB P0.0      ;MODO PLAY P/R=1L
```

```
    nop
```

```
    MOV P2,ADDRESS ;SE MANDA LA DIRECCION DEL MENSAJE
```

```
    nop
```

```
    SETB P0.1      ;UN PULSO POSITIVO EN PD
```

```
    ACALL r50ms
```

```
    CLR P0.1
```

```
    acall r50ms
```

```
    CLR P0.2      ;UN PULSO NEGATIVO EN CE
```

```
    nop
```

```
    SETB P0.2
```

```
ACALL FLANCO
```

```
clr ea
```

DJNZ CUENTAFRASES,NUEVAFRASE

;indico que debe presionar 1 para aceptar o 2 para cancelar

MOV R6,#10

ESPERANDO:

ACALL R50MS

DJNZ R6,ESPERANDO

SETB P0.0 ;P/R=1L

MOV A,PRESIONE

MOV P2,A

nop

SETB P0.1 ;UN PULSO POSITIVO EN PD

ACALL r50ms

CLR P0.1

acall r50ms

CLR P0.2 ;UN PULSO NEGATIVO EN CE

NOP

SETB P0.2

ACALL FLANCO

SETB P0.0 ;P/R=1L

MOV A,UNO

MOV P2,A

nop

SETB P0.1 ;UN PULSO POSITIVO EN PD

ACALL r50ms

CLR P0.1

acall r50ms

CLR P0.2 ;UN PULSO NEGATIVO EN CE

NOP

SETB P0.2

ACALL FLANCO

```
SETB P0.0      ;P/R=1L
MOV A, PARA
MOV P2,A
nop
SETB P0.1      ;UN PULSO POSITIVO EN PD
ACALL r50ms
CLR P0.1
acall r50ms
CLR P0.2      ;UN PULSO NEGATIVO EN CE
NOP
SETB P0.2
ACALL FLANCO
```

```
SETB P0.0      ;P/R=1L
MOV A,ACEPTAR
MOV P2,A
nop
SETB P0.1      ;UN PULSO POSITIVO EN PD
ACALL r50ms
CLR P0.1
acall r50ms
CLR P0.2      ;UN PULSO NEGATIVO EN CE
NOP
SETB P0.2
ACALL FLANCO
```

```
SETB P0.0      ;P/R=1L
MOV A,PRESIONE
MOV P2,A
nop
SETB P0.1      ;UN PULSO POSITIVO EN PD
```

```

ACALL r50ms
CLR P0.1
acall r50ms
CLR P0.2      ;UN PULSO NEGATIVO EN CE
NOP
SETB P0.2
ACALL FLANCO

SETB P0.0      ;P/R=1L
MOV A,DOS
MOV P2,A
nop
SETB P0.1      ;UN PULSO POSITIVO EN PD
ACALL r50ms
CLR P0.1
acall r50ms
CLR P0.2      ;UN PULSO NEGATIVO EN CE
NOP
SETB P0.2
ACALL FLANCO

SETB P0.0      ;P/R=1L
MOV A,PARA
MOV P2,A
nop
SETB P0.1      ;UN PULSO POSITIVO EN PD
ACALL r50ms
CLR P0.1
acall r50ms
CLR P0.2      ;UN PULSO NEGATIVO EN CE
NOP
SETB P0.2
ACALL FLANCO

```

```

SETB P0.0      ;P/R=1L
MOV A,CANCELAR
MOV P2,A
nop
SETB P0.1      ;UN PULSO POSITIVO EN PD
ACALL r50ms
CLR P0.1
acall r50ms
CLR P0.2      ;UN PULSO NEGATIVO EN CE
NOP
SETB P0.2
ACALL FLANCO

```

;a continuacion se debe esperar que presione el 1 o 2  
;DEBO AGREGARLE UN TIMEOUT

```

DV:
JB P3.3,SEGUIRDV      ;FLANCO NEGATIVO (AL EMPEZAR A
PRESIONAR)
SJMP DV

```

```

SEGUIRDV:
MOV DVA,#25          ;ESPERO 5 SEGUNDOS POR
CONFIRMACION

```

```

SEGUIRDVA:
MOV DVB,#250

```

```

SEGUIRDVB:
MOV DVC,#250

```

```

SEGUIRDVC:

JNB P3.3,COMPARO

```

```
DJNZ DVC,SEGUIRDVC
DJNZ DVB,SEGUIRDVB
DJNZ DVA,SEGUIRDVA
```

```
SJMP TERMINO
```

```
COMPARO:
```

```
;A CONTINUACION LEO EL VALOR DEL NUMERO PRESIONADO
```

```
MOV A,P1                ;LEE EL PORTICO1
ORL A,#11110000B        ;SELECCIONA SOLO LOS 4 BITS
MENOS S.
ANL A,#00001111B        ;DEJA EXPRESADO EN UN NUMERO
ENTRE 0 Y F
```

```
jnb p3.3,$              ;espera a que se suelte la
tecla (dv)
```

```
cjne A,#01,termino
```

```
clr ti                  ;MANDA AL COMPUTADOR SENAL DE OK
mov a,#31H
mov sbuf,a
jnb ti,$
CLR TI
```

```
SJMP TERMINO2
```

```
termino:
```

```
clr ti                  ;MANDA AL COMPUTADOR SENAL DE CANCEL
mov a,#32H
```

```

mov sbuf,a
jnb ti,$
CLR TI

TERMINO2:

SETB P0.0          ;P/R=1L
MOV A,PRESIONE
MOV P2,A
nop
SETB P0.1          ;UN PULSO POSITIVO EN PD
ACALL r50ms
CLR P0.1
acall r50ms
CLR P0.2           ;UN PULSO NEGATIVO EN CE
NOP
SETB P0.2
ACALL FLANCO

SETB P0.0          ;P/R=1L
MOV A,UNO
MOV P2,A
nop
SETB P0.1          ;UN PULSO POSITIVO EN PD
ACALL r50ms
CLR P0.1
acall r50ms
CLR P0.2           ;UN PULSO NEGATIVO EN CE
NOP
SETB P0.2
ACALL FLANCO

```

```

SETB P0.0      ;P/R=1L
MOV A, PARA
MOV P2, A
nop
SETB P0.1      ;UN PULSO POSITIVO EN PD
ACALL r50ms
CLR P0.1
acall r50ms
CLR P0.2      ;UN PULSO NEGATIVO EN CE
NOP
SETB P0.2
ACALL FLANCO

SETB P0.0      ;P/R=1L
MOV A, CONTINUAR
MOV P2, A
nop
SETB P0.1      ;UN PULSO POSITIVO EN PD
ACALL r50ms
CLR P0.1
acall r50ms
CLR P0.2      ;UN PULSO NEGATIVO EN CE
NOP
SETB P0.2
ACALL FLANCO

SETB P0.0      ;P/R=1L
MOV A, PRESIONE
MOV P2, A
nop
SETB P0.1      ;UN PULSO POSITIVO EN PD
ACALL r50ms
CLR P0.1

```

```

acall r50ms
CLR P0.2      ;UN PULSO NEGATIVO EN CE
NOP
SETB P0.2
ACALL FLANCO

SETB P0.0      ;P/R=1L
MOV A,DOS
MOV P2,A
nop
SETB P0.1      ;UN PULSO POSITIVO EN PD
ACALL r50ms
CLR P0.1
acall r50ms
CLR P0.2      ;UN PULSO NEGATIVO EN CE
NOP
SETB P0.2
ACALL FLANCO

SETB P0.0      ;P/R=1L
MOV A,PARA
MOV P2,A
nop
SETB P0.1      ;UN PULSO POSITIVO EN PD
ACALL r50ms
CLR P0.1
acall r50ms
CLR P0.2      ;UN PULSO NEGATIVO EN CE
NOP
SETB P0.2
ACALL FLANCO

SETB P0.0      ;P/R=1L

```

```

MOV A, TERMINAR
MOV P2, A
nop
SETB P0.1      ;UN PULSO POSITIVO EN PD
ACALL r50ms
CLR P0.1
acall r50ms
CLR P0.2      ;UN PULSO NEGATIVO EN CE
NOP
SETB P0.2
ACALL FLANCO

DV1:
JB P3.3, SEGUIRDV1      ;FLANCO NEGATIVO (AL EMPEZAR A
PRESIONAR)
SJMP DV1

SEGUIRDV1:
MOV DVA1, #25      ;ESPERO 5 SEGUNDOS POR CONFIRMACION
SEGUIRDVA1:
MOV DVB1, #250
SEGUIRDVB1:
MOV DVC1, #250
SEGUIRDVC1:

JNB P3.3, COMPARO1

DJNZ DVC1, SEGUIRDVC1
DJNZ DVB1, SEGUIRDVB1
DJNZ DVA1, SEGUIRDVA1

SJMP TERMINO3      ;CUELGO SI NO INGRESO NADA

```

;A CONTINUACION LEO EL VALOR DEL NUMERO PRESIONADO

COMPARO1:

```
MOV A,P1          ;LEE EL PORTICO1
ORL A,#11110000B  ;SELECCIONA SOLO LOS 4 BITS
MENOS S.
ANL A,#00001111B  ;DEJA EXPRESADO EN UN NUMERO
ENTRE 0 Y F

jnb p3.3,$        ;espera a que se suelte la
tecla (dv)

cjne A,#01,termino3

SETB P0.0 ;MODO PLAY P/R=1L

MOV P2,INGRESE    ;SE MANDA LA DIRECCION DEL
MENSAJE
NOP
SETB P0.1          ;UN PULSO POSITIVO EN PD
ACALL r50ms
CLR P0.1
acall r50ms
CLR P0.2          ;UN PULSO NEGATIVO EN CE
nop
SETB P0.2

ACALL FLANCO

SETB P0.0          ;MODO PLAY P/R=1L
```

```

MOV P2, INSTRUCCIONES           ;SE MANDA LA DIRECCION DEL
MENSAJE
NOP
SETB P0.1                       ;UN PULSO POSITIVO EN PD
ACALL r50ms
CLR P0.1
acall r50ms
CLR P0.2                       ;UN PULSO NEGATIVO EN CE
nop
SETB P0.2

ACALL FLANCO

MOV TTIMBRE1, #20
MOV TTIMBRE2, #4
MOV TTIMBRE3, #250

SJMP TERMINO4

TERMINO3:
CLR DESCUELGO                   ;CUELGO EL TELEFONO ANTES DE TIEMPO

clr p1.5                       ;CUELGO EL TELEFONO ANTES DE TIEMPO

termino4:

AJMP SALIDACOM
NOESAUTO1:
CJNE A, #66, SALIDACOM
JNB RI, $
CLR RI
MOV A, SBUF                     ;SE LEE NUMERO DE TIMBRADAS

```

```
SUBB A, #30H
MOV TIMBRE, A
MOV TIMBREAUX, TIMBRE
SJMP SALIDACOM
```

NOESTIMBRADA:

```
CJNE A, #88, NOESX ;DESCONECTA EL PARLANTE
CLR P1.6
AJMP SALIDACOM
```

NOESX:

```
CJNE A, #89, SALIDACOM ;CONECTA EL PARLANTE
SETB P1.6
```

SALIDACOM:

```
SETB EA
clr rs0
```

```
pop acc
RETI
```

DECODADDRESS:

```
MOV A, NUM1 ;EL SIGUIENTE GRUPO DE INSTRUCCIONES
FORMA ADDRESS
MOV B, #100
MUL AB
MOV ADDRESS, A
MOV A, NUM2
MOV B, #10
MUL AB
ADD A, ADDRESS
ADD A, NUM3
MOV ADDRESS, A
```

RET

FLANCO:

clr ea

ESPERO:

JB P0.3,SEGUIR ;SENSOR DE FLANCO NEGATIVO DEL EOM

SJMP ESPERO

SEGUIR:

JNB P0.3,ACTUAR

SJMP seguir

ACTUAR:

ACALL R50MS

setb ea

RET

LECTCOMP:

JNB RI,\$ ;LEO CENTENAS DE ADDRESS

CLR RI

MOV A,SBUF

SUBB A,#30H

MOV NUM1,A

JNB RI,\$ ;LEO DECENAS DE ADDRESS

CLR RI

MOV A,SBUF

SUBB A,#30H

MOV NUM2,A

JNB RI,\$ ;LEO UNIDADES DE ADDRESS

```
CLR RI
MOV A, SBUF
SUBB A, #30H
MOV NUM3, A
```

```
RET
```

```
TRANSNUM:
```

```
MOV A, NUM1           ;EL SIGUIENTE GRUPO DE INSTRUCCIONES
FORMA ADDRESS
MOV B, #100
MUL AB
MOV AUXILIAR, A
MOV A, NUM2
MOV B, #10
MUL AB
ADD A, AUXILIAR
ADD A, NUM3
MOV AUXILIAR, A
```

```
RET
```

```
END
```



# PROGRAMA PARA EL COMPUTADOR REALIZADO EN QBASIC

```
CLOSE  
CLS
```

```
baseaddr = &H3F8  
REM RTS 1  
OUT (baseaddr + 4), (INP(baseaddr + 4) OR 2)  
REM DTR 0  
OUT (baseaddr + 4), (INP(baseaddr + 4) AND &HFE)
```

```
OPEN "COM2:9600,N,8,1,RS,CS,DS,CD" FOR RANDOM AS #1  
DIM tinicial(100)  
DIM TFINAL(100)  
DIM adress(100)  
DIM DATO$(100)  
DIM q$(100)
```

```
q$(1) = "0"  
q$(2) = "1"  
q$(3) = "2"  
q$(4) = "3"  
q$(5) = "4"  
q$(6) = "5"  
q$(7) = "6"  
q$(8) = "7"  
q$(9) = "8"  
q$(10) = "9"  
q$(11) = "clave"  
q$(12) = "sector"  
q$(13) = "prender"  
q$(14) = "apagar"  
q$(15) = "repita las"  
q$(16) = "presione"  
q$(17) = "para"  
q$(18) = "terminar"  
q$(19) = "aceptar"  
q$(20) = "instrucciones"  
q$(21) = "cancelar"  
q$(22) = "ingrese"  
q$(23) = "continuar"
```

```
EMPIEZO:  
DO:
```

```

PRINT
PRINT
PRINT "INGRESE LA OPCION"
PRINT "1)GRABAR"
PRINT "2)MODO AUTOMATICO"
PRINT "3)MODO MANUAL"
PRINT "4)SELECCIONAR EL NUMERO DE TIMBRADAS"
PRINT "5)INICIALIZAR DIRECCIONES DE MENSAJE"

```

```

PRINT
INPUT OPCION
IF OPCION = 1 THEN GOTO GRABAR
IF OPCION = 2 THEN GOTO AUTO
IF OPCION = 3 THEN GOTO MANUAL
IF OPCION = 4 THEN GOTO TIMBRE
IF OPCION = 5 THEN GOTO TIEMPO
END
CLS

```

GRABAR:

```

TTOTAL = 0
INPUT "INGRESE EL NUMERO DE MENSAJES"; N
CLS
INPUT "Presione enter para empezar a grabar"; k
FOR i = 0 TO N - 1

```

GRABAR

GRABACION

```

    GOSUB retardo0
    PRINT
    tinicial(i) = TIMER
    IF i = 0 THEN PRINT #1, "G";      'EMPEZAR A
    IF i <> 0 THEN PRINT #1, "R";    'REINICIO
    PRINT "GRABANDO                "; q$(i + 1)
    IF i = 0 THEN GOSUB retardo2
    IF i = 1 THEN GOSUB retardo2
    IF i = 2 THEN GOSUB retardo1
    IF i = 3 THEN GOSUB retardo2
    IF i = 4 THEN GOSUB retardo2
    IF i = 5 THEN GOSUB retardo2
    IF i = 6 THEN GOSUB retardol
    IF i = 7 THEN GOSUB retardo2
    IF i = 8 THEN GOSUB retardo2
    IF i = 9 THEN GOSUB retardo2
    IF i = 10 THEN GOSUB retardo2
    IF i = 11 THEN GOSUB retardo2
    IF i = 12 THEN GOSUB retardo2
    IF i = 13 THEN GOSUB retardo4
    IF i = 14 THEN GOSUB retardo5
    IF i = 15 THEN GOSUB retardo4
    IF i = 16 THEN GOSUB retardol

```

```

        IF i = 17 THEN GOSUB retardo4
        IF i = 18 THEN GOSUB retardo4
        IF i = 19 THEN GOSUB retardo5
        IF i = 20 THEN GOSUB retardo4
        IF i = 21 THEN GOSUB retardo4
        IF i = 22 THEN GOSUB retardo4

        PRINT #1, "P";                                'PAUSA EN LA
GRABACION
        TFINAL(i) = TIMER
        PRINT "EL TIEMPO DEL MENSAJE No."; i + 1;
"FUE DE "; TFINAL(i) - tinicial(i); "SEGUNDOS"
        adress(i) = INT(TTOTAL / .125)
        PRINT "DIRECCION BASE DEL MENSAJE"; i + 1;
"="; adress(i)
        TTOTAL = TTOTAL + (TFINAL(i) - tinicial(i))
        TLIBRE = 20 - TTOTAL
        PRINT "TIEMPO DISPONIBLE"; TLIBRE
        PRINT

        NEXT i

LOOP

AUTO:

CUENTADATOS = 0

adress(0) = 0
adress(1) = 5
adress(2) = 11
adress(3) = 17
adress(4) = 24
adress(5) = 30
adress(6) = 36
adress(7) = 42
adress(8) = 48
adress(9) = 54
adress(10) = 61
adress(11) = 67
adress(12) = 73
adress(13) = 81

DO:
        IF LOC(1) > 0 THEN
                CUENTADATOS = CUENTADATOS + 1
                DATOS(CUENTADATOS) = (INPUT$(1, 1)) 'PUEDE
QUE ESTE MAL

```

```

END IF
IF CUENTADATOS = 11 THEN
    CUENTADATOS = 0
    NUM1 = INT(adress(10) / 100)
    NUM2 = INT((adress(10) - NUM1 * 100) / 10)
    NUM3 = adress(10) - NUM1 * 100 - NUM2 * 10

    PRINT #1, "A";          'INDICA QUE MANDARA
VALORES DE AUTO

    IRETARDO = TIMER
    DO UNTIL (FRETARDO - IRETARDO > .01)
        FRETARDO = TIMER
    LOOP

    PRINT #1, RIGHT$(STR$(NUM1), 1);    'CLAVE
    PRINT "DIRECCION CLAVE=", NUM1,
    IRETARDO = TIMER
    DO UNTIL (FRETARDO - IRETARDO > .01)
        FRETARDO = TIMER
    LOOP

    PRINT #1, RIGHT$(STR$(NUM2), 1);    'CLAVE
    PRINT NUM2,
    IRETARDO = TIMER
    DO UNTIL (FRETARDO - IRETARDO > .01)
        FRETARDO = TIMER
    LOOP

    PRINT #1, RIGHT$(STR$(NUM3), 1);    'CLAVE
    PRINT NUM3
    IRETARDO = TIMER
    DO UNTIL (FRETARDO - IRETARDO > .01)
        FRETARDO = TIMER
    LOOP

    FOR i = 1 TO 5

        NUM1 = INT(adress(VAL(DATO$(i))) /
100)
        NUM2 = INT((adress(VAL(DATO$(i))) -
NUM1 * 100) / 10)
        NUM3 = adress(VAL(DATO$(i))) - NUM1 *
100 - NUM2 * 10

        PRINT #1, RIGHT$(STR$(NUM1), 1);
        PRINT "NUMERO"; i; "="          ", NUM1,
        IRETARDO = TIMER
        DO UNTIL (FRETARDO - IRETARDO > .01)
            FRETARDO = TIMER

```

```

        LOOP

        PRINT #1, RIGHT$(STR$(NUM2), 1);
        PRINT NUM2,
        IRETARDO = TIMER
        DO UNTIL (FRETARDO - IRETARDO > .01)
            FRETARDO = TIMER
        LOOP

        PRINT #1, RIGHT$(STR$(NUM3), 1);
        PRINT NUM3
        IRETARDO = TIMER
        DO UNTIL (FRETARDO - IRETARDO > .01)
            FRETARDO = TIMER
        LOOP

    NEXT i

    REM SECTOR

    NUM1 = INT(adres(11) / 100)
    NUM2 = INT((adres(11) - NUM1 * 100) / 10)
    NUM3 = adres(11) - NUM1 * 100 - NUM2 * 10

    PRINT #1, RIGHT$(STR$(NUM1), 1);
    PRINT "DIRECCION SECTOR=", NUM1,
    IRETARDO = TIMER
    DO UNTIL (FRETARDO - IRETARDO > .01)
        FRETARDO = TIMER
    LOOP

    PRINT #1, RIGHT$(STR$(NUM2), 1);
    PRINT NUM2,
    IRETARDO = TIMER
    DO UNTIL (FRETARDO - IRETARDO > .01)
        FRETARDO = TIMER
    LOOP

    PRINT #1, RIGHT$(STR$(NUM3), 1);
    PRINT NUM3
    IRETARDO = TIMER
    DO UNTIL (FRETARDO - IRETARDO > .01)
        FRETARDO = TIMER
    LOOP

    FOR i = 7 TO 10

        NUM1 = INT(adres(VAL(DATO$(i)))) /

```

100)

```

NUM1 * 100) / 10)
100 - NUM2 * 10

NUM1,

NUM2 = INT((address(VAL(DATO$(i)))) -
NUM3 = address(VAL(DATO$(i))) - NUM1 *

PRINT #1, RIGHT$(STR$(NUM1), 1);
PRINT "DIRECCION SECTOR "; i; "=",

IRETARDO = TIMER
DO UNTIL (FRETARDO - IRETARDO > .01)
    FRETARDO = TIMER
LOOP

PRINT #1, RIGHT$(STR$(NUM2), 1);
PRINT NUM2,

IRETARDO = TIMER
DO UNTIL (FRETARDO - IRETARDO > .01)
    FRETARDO = TIMER
LOOP

PRINT #1, RIGHT$(STR$(NUM3), 1);
PRINT NUM3

IRETARDO = TIMER
DO UNTIL (FRETARDO - IRETARDO > .01)
    FRETARDO = TIMER
LOOP

NEXT i

IF DATO$(11) = "*" THEN

NUM1 = INT(address(12) / 100)
NUM2 = INT((address(12) - NUM1 * 100)
/ 10)
* 10

NUM3 = address(12) - NUM1 * 100 - NUM2

PRINT #1, RIGHT$(STR$(NUM1), 1);
PRINT "DIRECCION ENCENDIDO=", NUM1,

IRETARDO = TIMER
DO UNTIL (FRETARDO - IRETARDO > .01)
    FRETARDO = TIMER
LOOP

PRINT #1, RIGHT$(STR$(NUM2), 1);
PRINT NUM2,

```

```

IRETARDO = TIMER
DO UNTIL (FRETARDO - IRETARDO > .01)
    FRETARDO = TIMER
LOOP

PRINT #1, RIGHT$(STR$(NUM3), 1);
PRINT NUM3

IRETARDO = TIMER
DO UNTIL (FRETARDO - IRETARDO > .01)
    FRETARDO = TIMER
LOOP

END IF

IF DATOS(11) = "#" THEN

    NUM1 = INT(adress(13) / 100)
    NUM2 = INT((adress(13) - NUM1 * 100)
/ 10)
* 10
    NUM3 = adress(13) - NUM1 * 100 - NUM2

    PRINT #1, RIGHT$(STR$(NUM1), 1);
    PRINT "DIRECCION APAGADO=", NUM1,

    IRETARDO = TIMER
    DO UNTIL (FRETARDO - IRETARDO > .01)
        FRETARDO = TIMER
    LOOP

    PRINT #1, RIGHT$(STR$(NUM2), 1);
    PRINT NUM2,

    IRETARDO = TIMER
    DO UNTIL (FRETARDO - IRETARDO > .01)
        FRETARDO = TIMER
    LOOP

    PRINT #1, RIGHT$(STR$(NUM3), 1);
    PRINT NUM3

    IRETARDO = TIMER
    DO UNTIL (FRETARDO - IRETARDO > .01)
        FRETARDO = TIMER
    LOOP

END IF

```

```

DO: LOOP UNTIL LOC(1) > 0
respuesta$ = (INPUT$(1, 1))
CLS
IF respuesta$ = "2" THEN PRINT "cancelado"
IF respuesta$ = "1" THEN PRINT "aceptado"

END IF

LOOP

MANUAL:

DO UNTIL adress(MENSAJE) = 99
    INPUT "INGRESE EL NUMERO DE MENSAJE A PRONUNCIARSE";
    adress(MENSAJE)

    PRINT #1, "H"; 'H LE PREPARA AL MICRO PARA INGRESAR
DIRECCION
    NUM1 = INT(adress(MENSAJE) / 100)
    NUM2 = INT((adress(MENSAJE) - NUM1 * 100) / 10)
    NUM3 = adress(MENSAJE) - NUM1 * 100 - NUM2 * 10

    IRETARDO = TIMER
    DO UNTIL (FRETARDO - IRETARDO > .01)
        FRETARDO = TIMER
    LOOP

    PRINT #1, RIGHT$(STR$(NUM1), 1); 'TODO NUMERO QUE
ENVIE ES UNA DIRECCION
    PRINT "DIRECCION ORIGEN="
    IRETARDO = TIMER
    DO UNTIL (FRETARDO - IRETARDO > .01)
        FRETARDO = TIMER
    LOOP

    PRINT #1, RIGHT$(STR$(NUM2), 1);

    IRETARDO = TIMER
    DO UNTIL (FRETARDO - IRETARDO > .01)
        FRETARDO = TIMER
    LOOP

    PRINT #1, RIGHT$(STR$(NUM3), 1);

    IRETARDO = TIMER
    DO UNTIL (FRETARDO - IRETARDO > .01)
        FRETARDO = TIMER
    LOOP

```

```

        PRINT adres(MENSAJE), NUM1; NUM2; NUM3
LOOP
GOTO EMPIEZO

retardo0:
tiempo0 = TIMER
TIEMPO = TIMER
DO UNTIL TIEMPO - tiempo0 >= .5
TIEMPO = TIMER
LOOP
RETURN

retardo1:
TIEMPO = TIMER
DO UNTIL TIEMPO - tinicial(i) >= .6
TIEMPO = TIMER
LOOP
RETURN

retardo2:
TIEMPO = TIMER
DO UNTIL TIEMPO - tinicial(i) >= .71
TIEMPO = TIMER
LOOP
RETURN

retardo4:
TIEMPO = TIMER
DO UNTIL TIEMPO - tinicial(i) >= .96
TIEMPO = TIMER
LOOP
RETURN

retardo3:
DO UNTIL TIEMPO - tinicial(i) >= 2.98
TIEMPO = TIMER
LOOP
RETURN

retardo5:
TIEMPO = TIMER
DO UNTIL TIEMPO - tinicial(i) >= 1.2
TIEMPO = TIMER
LOOP
RETURN

TIMBRE:

```

```

INPUT "INGRESE EL NUMERO DE TIMBRADAS A ESPERAR";
NUM1

PRINT #1, "B"; 'B LE PREPARA AL MICRO PARA INGRESAR
NUMERO DE TIMBRADAS

IRETARDO = TIMER
DO UNTIL (FRETARDO - IRETARDO > .01)
    FRETARDO = TIMER
LOOP

PRINT #1, RIGHT$(STR$(NUM1), 1); 'numero de timbradas

IRETARDO = TIMER
DO UNTIL (FRETARDO - IRETARDO > .01)
    FRETARDO = TIMER
LOOP
END

```

TIEMPO:

```

adress(1) = 5
adress(2) = 11
adress(14) = 89
adress(15) = 99
adress(16) = 107
adress(17) = 113
adress(18) = 121
adress(19) = 130
adress(20) = 140
adress(21) = 147
adress(22) = 157

```

```

PRINT #1, "T"; 'PROGRAMA MENSAJES 14-22 Y 1,2

REM A CONTINUACION VIENE UN RETARDO PARA ANTES
TERMINAR DE ENVIAR EL MENSAJE
tinicial = TIMER
DO UNTIL TFINAL - tinicial > .01
    TFINAL = TIMER
LOOP

NUM1 = INT(adress(14) / 100)
NUM2 = INT((adress(14) - NUM1 * 100) / 10)
NUM3 = adress(14) - NUM1 * 100 - NUM2 * 10

```

```

PRINT #1, RIGHT$(STR$(NUM1), 1);
tinicial = TIMER
DO UNTIL TFINAL - tinicial > .01
    TFINAL = TIMER
LOOP
PRINT #1, RIGHT$(STR$(NUM2), 1);
tinicial = TIMER
DO UNTIL TFINAL - tinicial > .01
    TFINAL = TIMER
LOOP
PRINT #1, RIGHT$(STR$(NUM3), 1);
tinicial = TIMER
DO UNTIL TFINAL - tinicial > .01
    TFINAL = TIMER
LOOP

NUM1 = INT(address(15) / 100)
NUM2 = INT((address(15) - NUM1 * 100) / 10)
NUM3 = address(15) - NUM1 * 100 - NUM2 * 10

PRINT #1, RIGHT$(STR$(NUM1), 1);
tinicial = TIMER
DO UNTIL TFINAL - tinicial > .01
    TFINAL = TIMER
LOOP
PRINT #1, RIGHT$(STR$(NUM2), 1);
tinicial = TIMER
DO UNTIL TFINAL - tinicial > .01
    TFINAL = TIMER
LOOP
PRINT #1, RIGHT$(STR$(NUM3), 1);
tinicial = TIMER
DO UNTIL TFINAL - tinicial > .01
    TFINAL = TIMER
LOOP

NUM1 = INT(address(16) / 100)
NUM2 = INT((address(16) - NUM1 * 100) / 10)
NUM3 = address(16) - NUM1 * 100 - NUM2 * 10

PRINT #1, RIGHT$(STR$(NUM1), 1);
tinicial = TIMER
DO UNTIL TFINAL - tinicial > .01
    TFINAL = TIMER
LOOP
PRINT #1, RIGHT$(STR$(NUM2), 1);
tinicial = TIMER
DO UNTIL TFINAL - tinicial > .01
    TFINAL = TIMER

```

```

LOOP
PRINT #1, RIGHT$(STR$(NUM3), 1);
tinicial = TIMER
DO UNTIL TFINAL - tinicial > .01
    TFINAL = TIMER
LOOP

NUM1 = INT(address(17) / 100)
NUM2 = INT((address(17) - NUM1 * 100) / 10)
NUM3 = address(17) - NUM1 * 100 - NUM2 * 10

PRINT #1, RIGHT$(STR$(NUM1), 1);
tinicial = TIMER
DO UNTIL TFINAL - tinicial > .01
    TFINAL = TIMER
LOOP
PRINT #1, RIGHT$(STR$(NUM2), 1);
tinicial = TIMER
DO UNTIL TFINAL - tinicial > .01
    TFINAL = TIMER
LOOP
PRINT #1, RIGHT$(STR$(NUM3), 1);
tinicial = TIMER
DO UNTIL TFINAL - tinicial > .01
    TFINAL = TIMER
LOOP

NUM1 = INT(address(18) / 100)
NUM2 = INT((address(18) - NUM1 * 100) / 10)
NUM3 = address(18) - NUM1 * 100 - NUM2 * 10

PRINT #1, RIGHT$(STR$(NUM1), 1);
tinicial = TIMER
DO UNTIL TFINAL - tinicial > .01
    TFINAL = TIMER
LOOP
PRINT #1, RIGHT$(STR$(NUM2), 1);
tinicial = TIMER
DO UNTIL TFINAL - tinicial > .01
    TFINAL = TIMER
LOOP
PRINT #1, RIGHT$(STR$(NUM3), 1);
tinicial = TIMER
DO UNTIL TFINAL - tinicial > .01
    TFINAL = TIMER
LOOP

NUM1 = INT(address(19) / 100)
NUM2 = INT((address(19) - NUM1 * 100) / 10)
NUM3 = address(19) - NUM1 * 100 - NUM2 * 10

```

```

PRINT #1, RIGHT$(STR$(NUM1), 1);
tinicial = TIMER
DO UNTIL TFINAL - tinicial > .01
    TFINAL = TIMER
LOOP
PRINT #1, RIGHT$(STR$(NUM2), 1);
tinicial = TIMER
DO UNTIL TFINAL - tinicial > .01
    TFINAL = TIMER
LOOP
PRINT #1, RIGHT$(STR$(NUM3), 1);
tinicial = TIMER
DO UNTIL TFINAL - tinicial > .01
    TFINAL = TIMER
LOOP

NUM1 = INT(address(20) / 100)
NUM2 = INT((address(20) - NUM1 * 100) / 10)
NUM3 = address(20) - NUM1 * 100 - NUM2 * 10

PRINT #1, RIGHT$(STR$(NUM1), 1);
tinicial = TIMER
DO UNTIL TFINAL - tinicial > .01
    TFINAL = TIMER
LOOP
PRINT #1, RIGHT$(STR$(NUM2), 1);
tinicial = TIMER
DO UNTIL TFINAL - tinicial > .01
    TFINAL = TIMER
LOOP
PRINT #1, RIGHT$(STR$(NUM3), 1);
tinicial = TIMER
DO UNTIL TFINAL - tinicial > .01
    TFINAL = TIMER
LOOP

NUM1 = INT(address(21) / 100)
NUM2 = INT((address(21) - NUM1 * 100) / 10)
NUM3 = address(21) - NUM1 * 100 - NUM2 * 10

PRINT #1, RIGHT$(STR$(NUM1), 1);
tinicial = TIMER
DO UNTIL TFINAL - tinicial > .01
    TFINAL = TIMER
LOOP
PRINT #1, RIGHT$(STR$(NUM2), 1);
tinicial = TIMER
DO UNTIL TFINAL - tinicial > .01
    TFINAL = TIMER

```

```

LOOP
PRINT #1, RIGHT$(STR$(NUM3), 1);
tinicial = TIMER
DO UNTIL TFINAL - tinicial > .01
    TFINAL = TIMER
LOOP

NUM1 = INT(address(22) / 100)
NUM2 = INT((address(22) - NUM1 * 100) / 10)
NUM3 = adress(22) - NUM1 * 100 - NUM2 * 10

PRINT #1, RIGHT$(STR$(NUM1), 1);
tinicial = TIMER
DO UNTIL TFINAL - tinicial > .01
    TFINAL = TIMER
LOOP
PRINT #1, RIGHT$(STR$(NUM2), 1);
tinicial = TIMER
DO UNTIL TFINAL - tinicial > .01
    TFINAL = TIMER
LOOP
PRINT #1, RIGHT$(STR$(NUM3), 1);
tinicial = TIMER
DO UNTIL TFINAL - tinicial > .01
    TFINAL = TIMER
LOOP

NUM1 = INT(address(1) / 100)
NUM2 = INT((address(1) - NUM1 * 100) / 10)
NUM3 = adress(1) - NUM1 * 100 - NUM2 * 10

PRINT #1, RIGHT$(STR$(NUM1), 1);
tinicial = TIMER
DO UNTIL TFINAL - tinicial > .01
    TFINAL = TIMER
LOOP
PRINT #1, RIGHT$(STR$(NUM2), 1);
tinicial = TIMER
DO UNTIL TFINAL - tinicial > .01
    TFINAL = TIMER
LOOP
PRINT #1, RIGHT$(STR$(NUM3), 1);
tinicial = TIMER
DO UNTIL TFINAL - tinicial > .01
    TFINAL = TIMER
LOOP

NUM1 = INT(address(2) / 100)
NUM2 = INT((address(2) - NUM1 * 100) / 10)
NUM3 = adress(2) - NUM1 * 100 - NUM2 * 10

```

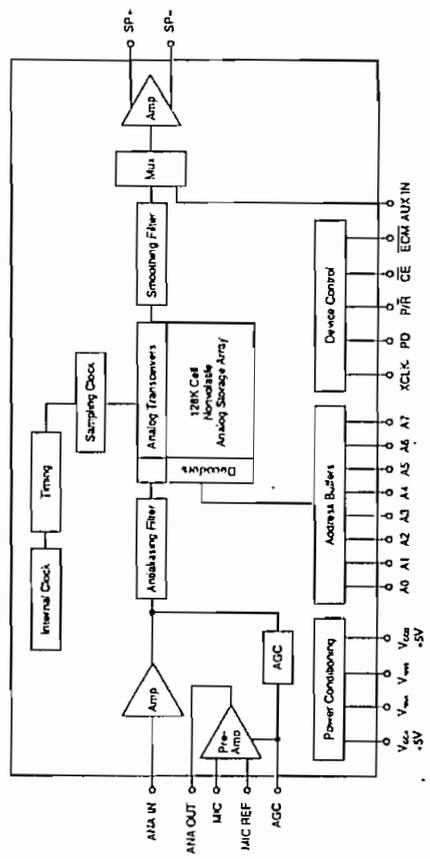
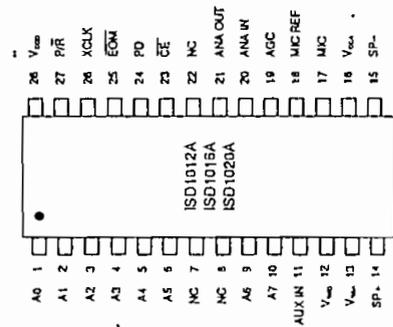
```
PRINT #1, RIGHT$(STR$(NUM1), 1);
tinicial = TIMER
DO UNTIL TFINAL - tinicial > .01
    TFINAL = TIMER
LOOP
PRINT #1, RIGHT$(STR$(NUM2), 1);
tinicial = TIMER
DO UNTIL TFINAL - tinicial > .01
    TFINAL = TIMER
LOOP
PRINT #1, RIGHT$(STR$(NUM3), 1);
tinicial = TIMER
DO UNTIL TFINAL - tinicial > .01
    TFINAL = TIMER
LOOP
```

END

INTRODUCTION

ISD's patented Direct Analog Storage Technology (DAST™) brings analog data into the semiconductor memory world. This "break through" EEPROM storage method writes analog data directly into a single cell without A/D or D/A conversion. This results in an increase in density over equivalent digital methods, and non-volatile storage of analog data.

You need only a few external components – a microphone, loudspeaker, switches, a few resistors and capacitors, and a power supply or battery – to build a complete voice record and playback system. All other devices – preamplifier, filters, AGC, power amplifier, control logic and analog storage – are built into the chip.



### To Play:

1. Set up the address you want to start recording at in the address inputs.
2. Set  $\overline{P/R}$  high.
3. Set PD low
4. Set  $\overline{CE}$  low.
5. Play begins. You can take  $\overline{CE}$  back high.

Play stops at a set End-Of-Message bit or the end of memory.

At a set End-Of-Message bit,  $\overline{EOM}$  pulses low for 16 milliseconds.

At end of memory,  $\overline{EOM}$  stays low until you take PD high.

### BASIC ADDRESSING

You can address multiple messages in an ISD1000A. Address inputs A0-A7 let you split the message area into 160 segments and let you access operation modes. The address inputs are positive logic. Each address count equals .125 seconds.

The external addresses control a register called Message Start Pointer or MSP. The MSP points to the location where the next play or record operation will begin.

To determine the value in the MSP, multiply the address by .125 seconds. For example, to start recording or playing at the 4-second boundary, load an address of 32 (32 x .125 = 4 seconds). The decimal number 32 equals 20 hex or 0010 0000 binary. Apply this binary address to the address pins.

In addressing the ISD1000A, the address is *how far "into" the message space the MSP points*. The address does not show how much memory is left.

Message addressing and operation modes are mutually exclusive. When you use an operation mode, the MSP initializes to 0 unless you select an overriding operation mode. See "Operation Modes."

ISD1000A Address	Binary								Seconds	
	A7	A6	A5	A4	A3	A2	A1	A0		
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	1	0.125
8	0	0	0	0	1	0	0	0	0	1.0
10	0	0	0	0	1	0	1	0	0	1.25
13	0	0	0	0	1	1	0	0	1	1.625
50	0	0	1	1	0	0	1	0	0	6.25
100	0	1	1	0	0	1	0	0	0	12.5
159	1	0	0	1	1	1	1	1	1	19.875
160	1	0	1	0	0	0	0	0	0	Unused address space. An ISD1000A addressed in this region defaults to an overflow. This address space is used by the operation modes.
thru 191	1	0	1	1	1	1	1	1	1	
192	1	1	0	0	0	0	0	0	0	
thru 255	1	1	1	1	1	1	1	1	1	

You can think of the ISD1000A as an analog tape recorder with the ability to position the record/play head anywhere on the tape within the address resolution.

### OPERATION MODES

The ISD1000A uses the last 64 addresses for operation modes. This part of the address map is defined by A6 and A7 both tied high. All operation mode operations start at Address 0. Later operations might start at other address locations if the proper operation mode bit is set. When the device is changed from play to record, from record to play, and when a power down cycle is executed, the internal address pointer resets to 0.

Operation mode bits do not latch. An operation mode operation is executed when  $\overline{CE}$  goes low and Address bits A6 and A7 are high. If the next  $\overline{CE}$  operation finds either of these bits low, a Message Address operation is executed. The state of any previous operation mode is lost.

**ADDRESS BIT A4 (Pin 5) – MESSAGE START POINTER RESET:** When this bit is high in operation mode, the Message Start Pointer is only reset to 0 when the record/play mode is changed or a power down cycle is executed. This option lets you *stack* recorded messages or sequentially review messages during play without having to know the start address of each message.

**ADDRESS BIT A2 (Pin 3) –  $\overline{EOM}$  CONTROL:** The  $\overline{EOM}$  output normally performs a dual role. When playing messages not ending with memory full,  $\overline{EOM}$  pulses low for 16 milliseconds to indicate an End-of-Message mark has been found.  $\overline{EOM}$  goes low and stays low when the device is in record or play and fills the message memory (Message Overflow). If  $\overline{CE}$  is high, play stops at that point. If Address bit A2 is high in operation mode, the End-of-Message indication is turned off.

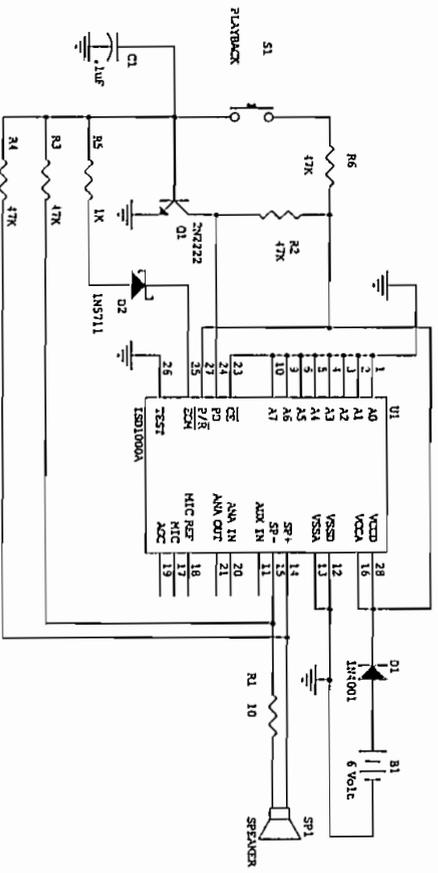
**ADDRESS BIT A3 (Pin 4) – CONTINUOUS PLAY CONTROL:** This operation mode bit lets you automatically repeat the message located at the beginning of the message space. This continues as long as  $\overline{CE}$  is held low. The only exception to this operation is that the message to be repeated cannot fill the ISD1000A to overflow.

**ADDRESS BIT A5 (Pin 6) –  $\overline{OE}$  LEVEL ACTIVATED:** The default mode is for  $\overline{CE}$  to be *edge activated* on play and *level activated* on record. With bit A5 high in operation mode,  $\overline{CE}$  is level activated during play. If Address bits A7, A6, and A5 are all high, a falling  $\overline{CE}$  starts play at the beginning of the message space and play continues while  $\overline{CE}$  is low. If  $\overline{CE}$  is taken back high, play stops and a second  $\overline{CE}$  going low restarts play at the beginning of the message space. If bit A4 is also taken high,  $\overline{CE}$  activation starts play at the message space beginning and if  $\overline{CE}$  is later taken high, message play immediately ceases. If  $\overline{CE}$  is now taken back low, play resumes at the point at which it stopped.



## Low Power Applications - Two Examples

The circuit shown below is an implementation of a low-power play-only circuit.

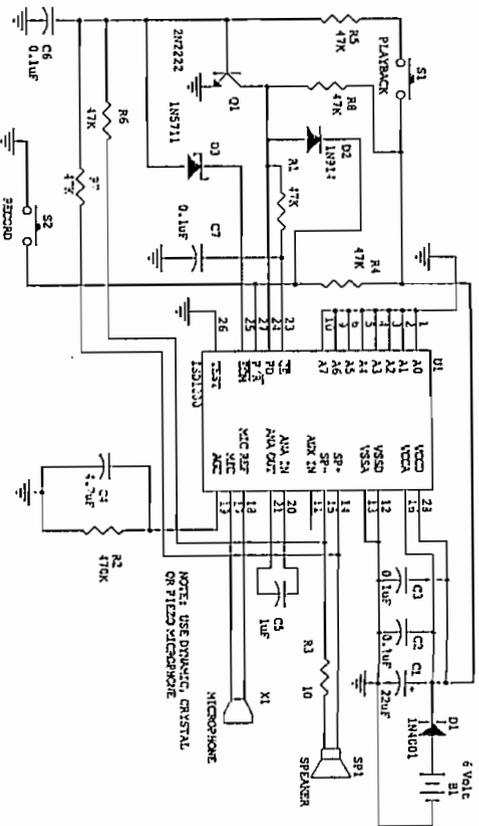


Low-Power Play-Only Schematic

To play the first message, momentarily press S1. Hold down S1 to play the entire contents of memory.

The key to this circuit is that the speaker outputs are at 1.5 volts during play and at 0 volts during power down. Initially, Q1 is turned off and R2 pulls PD up, putting the device into low-power standby. Momentarily pressing S1 makes Q1 conduct, pulling PD low. The message in the ISD1000A plays. When play starts, the speaker outputs bias to 1.5 volts, keeping Q1 on. When a set End-Of-Message bit is encountered, EOM pulses low and turns off Q1. This lets PD go high, returning the circuit to the low power state.

You can also design a circuit that achieves automatic power down in a record and play application. The following schematic is an adaptation of the previous one and shows one such implementation.

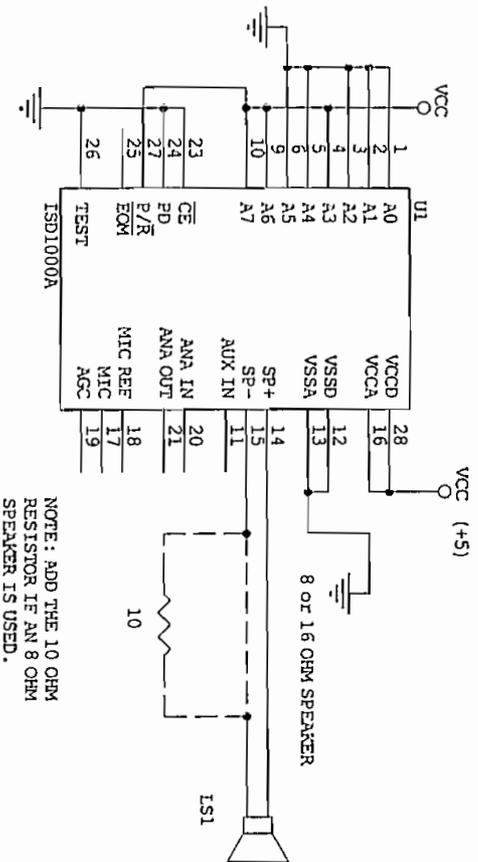


Low Power Play/Record Schematic

You begin recording by pressing S2. Recording continues as long as you press S2. Recording stops when you release S2 or when Overflow is reached. To play, momentarily press S1.

## Play Looping – Two Examples

You can use an operation mode to accomplish play looping if the message starts at 0 (the beginning of memory) and does not require the full 20 seconds of analog storage. First, record a message with all the address bits tied low. This positions this mes-

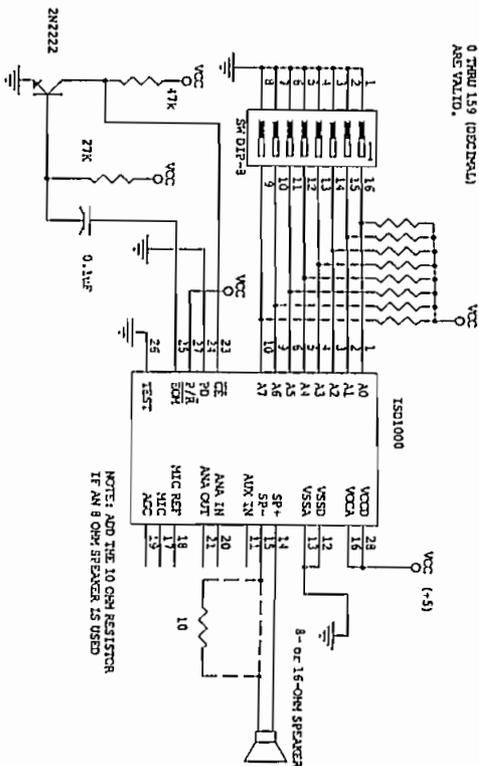


### Operational Mode Looping

sage at the beginning of the ISD1000A's address space. Next, put the device into the continuous repeat operation mode by connecting address bits A7, A6, and A3 high. With PD low, P/R high, and CE low, the first message in the ISD1000A repeats.

If CE is taken high, the ISD1000A completes the message and stops. If CE is strapped low, when you apply power, the message continuously repeats.

NOTE: ONLY ADDRESS 0 TRIM 159 (00000000) ARE VALID.



### Chip Enable (CE) Initiated Looping

You must use external circuitry to repeat a loop that starts at an address other than 0. You cannot use the CE Looping circuit if the message ends at device full. The 2N2222 transistor uses the EOM pulse at the end of a message to produce a positive pulse into CE. When this pulse falls, the dip switch address loads into the device and a new play cycle starts at that address.

### One-Minute Cascade

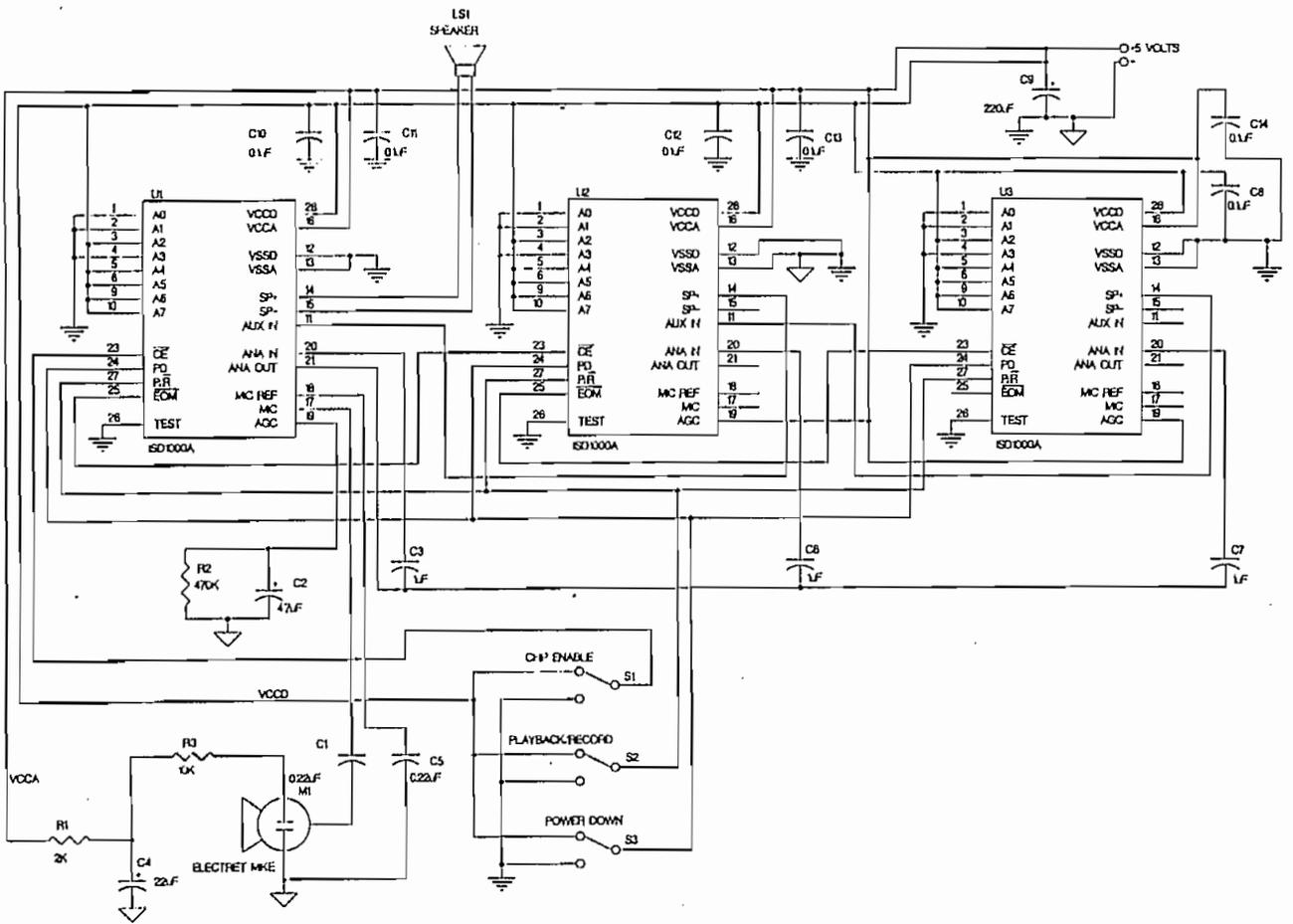
You can cascade several ISD1000As to increase a system's capacity. The method described here allows first in, first out record and play of one or more messages. In this example, three ISD1000As are cascaded for a 1 minute total record/play time.

U1 contains the mic preamp and speaker output for the system, plus 20 seconds of storage. The other ICs serve only as memory elements. U1's Ana Out (microphone preamp output) pin is connected through blocking capacitors to Ana In on each device in the cascade. Additionally, U3's SP+ is fed to U2's Aux In pin and U2's SP+ is connected to U1's Aux In pin. Thus, the Ana In pins are fed in parallel from U1's mic preamp while the speaker outputs daisy chain back to U1's speaker amplifier.

The A2 EOM Output Control operation mode sets the ISD1000A for proper cascade operation. EOM becomes chip enable for the next chip in the cascade. Bit A4 is high to make the Message Start Pointer (MSP) only reset when the system's mode changes between record and play. Bit A5 is also high, so CE is level activated during play. You can use this input to start and stop play. When an ISD1000A overflows and the EOM Output Control bit (A2) is high, EOM follows the CE input. You can think of CE as rippling through multiple devices in overflow to the current active device. Controlling CE for the first device in the cascade controls the rest of the device

With this configuration, a message is stored across chip boundaries in a manner transparent to the user. Changing from record to play resets the MSP to the beginning of the first message in the series. The next play proceeds under control of CE.

This circuit operates exactly like that for a single device

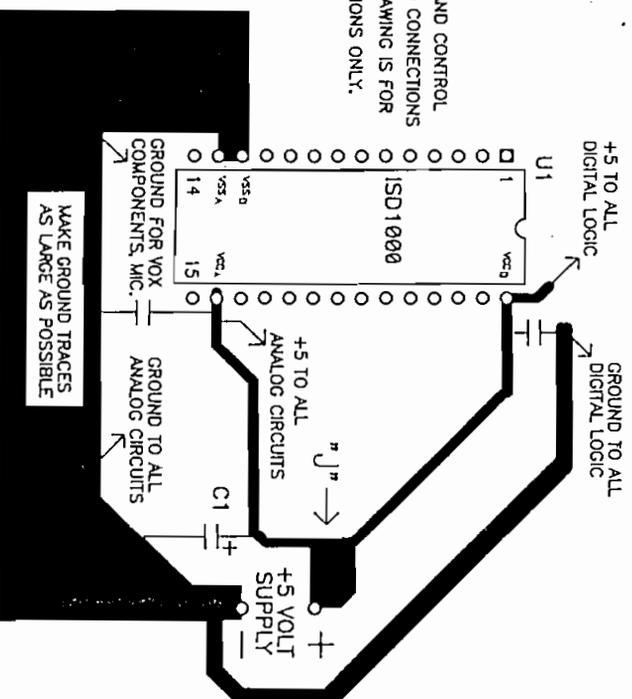


## GOOD AUDIO DESIGN PRACTICES

The ISD1000A is a high-quality, single-chip voice recording and play system. To get the full quality voice capability of the device, it is important that you follow a few guidelines as you construct your system. The following guidelines are suggested as minimum to assure satisfactory performance of your new recording system.

- Connect analog components (AGC resistors/capacitors, coupling capacitors, and so on, physically close to the ISD1000A. Use short component lead lengths.
- Use a quality microphone and connect microphone ground to a low-impedance analog ground return.
- Install a capacitor of the same value as the microphone coupling capacitor from Mic Ref to the microphone's ground.
- Provide high-frequency decoupling capacitors at the analog and digital power pins, using low ESR (Effective Series Resistance) capacitors.
- Use a separate ground return to the power supply for the VCCD (pin 28) decoupling capacitor.
- Use separate thick wire (#15 or better) or wide (30 mils) power and ground routing lines.
- Use power supplies that do not introduce their own noise source (e.g. some switching supplies introduce noise into a recording).

NOTE: ADDRESS AND CONTROL LOGIC AND AUDIO CONNECTIONS NOT SHOWN. DRAWING IS FOR POWER CONNECTIONS ONLY.



## Power Supply and Ground Distribution

There are two connections each for power (+5V) and ground. These connections minimize interference between the analog and digital portions of the ISD1000A's internal circuits. We recommend that you maintain this power separation in the manner shown on the sample printed circuit diagram. Note in particular that you should tie VSSA and VSSD together at Pins 12 and 13.

## Connecting to the Analog Path

Locate components in the ISD1000A's analog section near the pins they connect to. Referring to the printed circuit diagram, connect grounds and VCC supplies for these components as indicated. In particular, locate capacitor C1 (low frequency voltage decoupling) as close to junction "J" as is practical. (Junction "J" separates VCCA from VCCD). The components on the analog path as close as practical to the ISD1000A and place them between C1 and the ISD1000A. Tie all digital control (addresses, control switches, etc.) to the appropriate VCC or VSS feed as shown at the top of the diagram.

## Audio Drive Levels and Impedances

You can drive the ISD1000A with an audio source other than a microphone. The characteristics of the two inputs are as follows:

Mic (Pin 17):

Input Impedance  $\approx$  10k Ohms  
Max Drive Level = 20 Millivolts p-p  
AGC Control of up to 20 dB

Ana In (Pin 20):

Input Impedance  $\approx$  2.7k Ohms  
Max Drive Level = 50 Millivolts p-p

## Microphone and Speaker Selection

The above guidelines are most effective if you use good quality microphones and speakers in the system. You might want to experiment with a variety of microphones and speakers (and speaker enclosures) to get the best voice quality performance from the system.

## Battery Operation

The low operating power of the ISD1000A makes it ideal for use in portable battery-operated environments. Note that as the battery approaches its end of life, its output voltage begins to drop. In addition, the effective series resistance of the battery at end of life increases substantially. As a result of these factors, the ISD1000A's voice quality diminishes. When this happens it is time to replace the battery. In effect, the ISD1000A tells you when the battery is low! At 3.5V an internal low-voltage detect circuit in the ISD1000A causes the device to remain in play mode regardless of the status of the P/R pin. This assures reliable recordings.

## ABSOLUTE MAXIMUM RATINGS

VCC-VSS ..... -0.3 to +7.0 Volt  
Voltage Applied to Any Pin ..... (VSS-0.3V) to (VCC+0.3V)  
Storage Temperature Range ..... -85 to 302°F (-65 to 150°C)  
Temperature Under Bias ..... -85 to 257°F (-65 to 125°C)  
Lead Temperature (Soldering 10 Seconds) ..... 572°F (300°C)

## DC PARAMETERS

VCCD = VCCA = 5.0 Volts, Operating Temperature = 77°F (25°C)

Input Levels all Digital Inputs ..... Standard (TTY)  
Output Levels all Digital Outputs ..... Standard (TTY)  
Operating Supply Current (VCCA+VCCD) .....  
PD=0 ..... 25 mW  
Standby Supply Current (VCCA+VCCD) .....  
PD=1, P/R=1 ..... 10  $\mu$ A  
Speaker Output Load Impedance ..... 16 Ohm

## AC PARAMETERS

Operating Conditions: T<sub>A</sub> = 77°F (25°C), VCCA = VCCD = 5.0V, VSSA + VSSD = 0V

Symbol	Characteristics	Minimum	Typical	Maximum	Units	Conditions
FS	Internal Clock Sampling Frequency	6.24	6.4	6.56	KHz	
BW	Passband <sup>1</sup>	-	2700		Hz	
P <sub>OUT</sub>	Speaker Output Power			50	mW	REXT = 16 Ohms
V <sub>OUT</sub>	Voltage Across Speaker			2.5	V <sub>p-p</sub>	REXT = 600 Ohms
V <sub>IN1</sub>	Microphone Input Voltage			20	mV	Peak-to-Peak
V <sub>IN2</sub>	Analog In Input Voltage			50	mV	Peak-to-Peak
V <sub>IN3</sub>	Auxiliary In Input Voltage			1.25	V	Peak-to-Peak, REXT = 16 Ohms
T <sub>SET</sub>	Control/Address Set-Up Time	300			nsec	
T <sub>THOLD</sub>	Control/Address Hold Time	0			nsec	
T <sub>CE</sub>	CE Pulse Width	100			nsec	
T <sub>POH</sub>	Power-Up Delay	31.25			nsec	
T <sub>POD</sub>	Power-Down Hold	0			nsec	
T <sub>REC</sub>	Total Record Time	19.5		20	sec	
T <sub>PLAY</sub>	Total Play Time	19.5		20	sec	
T <sub>EM</sub>	EM1 Pulse Width		15.6		nsec	

Notes:

1 - Low-frequency cutoff depends on the value of external capacitors.

2 - With 12k Ohm series resistor at Ana In.



# TECHNICAL DATA

AN EXCLUSIVE RADIO SHACK SERVICE TO THE EXPERIMENTER

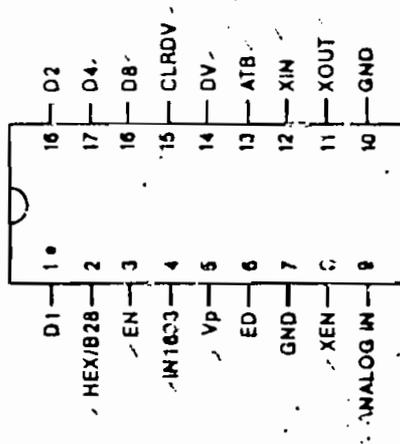
## 6V LOW POWER DTMF RECEIVER

### FEATURES

- Central Office Quality
- NO front-end band-splitting filters required
- Single, low tolerance, 6-volt supply
- Detects either 12 or 16 standard DTMF digits
- Uses inexpensive 3.879645-MHz crystal (272-1310)
- Excellent speech intelligibility
- Output in either 4-bit hexadecimal code or binary coded 2 of 5
- 18-pin DIP package for high system density
- Synchronous 2-wire interface
- Three-state outputs

### DESCRIPTION

The 276-1303 DTMF Receiver is a complete Dual Tone Multiple Frequency (DTMF) receiver detecting a selectable group of 12 or 16 standard digits. No front-end pre-filtering is needed. The only externally required components are an inexpensive 3.879645 MHz selected "colony" crystal (272-1310) and a bias trieter. Extremely high system density is made possible by wiring the clock output of a crystal connected 276-1303 DTMF Receiver to drive the time base of additional integrated circuits fabricated with low-power, complementary symmetric MOS (CMOS) processing. A maximum of only a single low



tolerance voltage supply and is packaged in a standard 18-pin plastic DIP.

The 276-1303 employs state-of-the-art circuit technology to combine digital and analog functions on the same CMOS chip using a standard digital semiconductor process. The analog input is pre-processed by 60-Hz reject and band splitting filters and then hardwired to provide AGC. Eight hardwired filters detect the individual tones. The digital post-processor times the tone durations and provides the correctly coded digital outputs. Outputs interface directly to standard CMOS facilities and are three-state enabled to facilitate bus-oriented architecture.

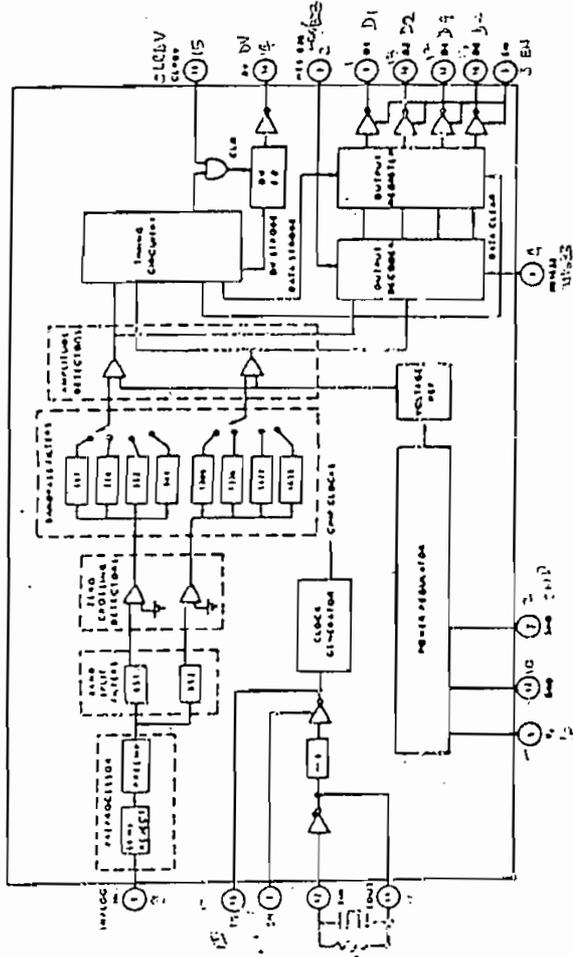


Figure 1. DTMF Receiver Block Diagram

### ABSOLUTE MAXIMUM RATINGS\*

- Supply Voltage  $V_p$  ..... +7 Volts
- Operating Temperature ..0°C to 70°C Ambient
- Storage Temperature .....-65°C to 150°C
- Power Dissipation (25°C) .....65mW (Derate above  $T_A = 25^\circ\text{C}$  @ 6.25mW/°C)
- Input Voltage .....( $V_p + .5V$ ) to  $-.5V$  (All inputs except ANALOG IN)
- ANALOG IN Voltage ..( $V_p + .5V$ ) to ( $V_p - 10V$ )

- DC Current into any input ..... +1.0mA
- Lead Temperature .....300°C (solder: 70, 10 sec.)
- Operation above absolute maximum ratings may damage the device.
- NOTE: All unused inputs must be connected to  $V_p$  or GND, as appropriate.

### ELECTRICAL CHARACTERISTICS (0°C ≤ T ≤ 70°C, $V_p = 5V \pm 10\%$ )

Parameter	Conditions	Min	Typ	Max	Units
Frequency Detect Bandwidth		±(1.5 - 2.5)	1.23	±3.5	% of $f_c$
Amplitude for Detection	each tone	-32		-2	dBm referenced to 600 Ω
Minimum Acceptable Level	1-dB 20 Hz tone 300 Hz tone	-10		+10	dB
±50-Hz Tolerance				0.4	Volts
Dual Tone Tolerance	"precise" dual tone			0.8	dB referenced to lower amplitude tone
TxD ON	MIL1500 (CUI 1700)		7		bits
Digital Output (except XOUT)	"0" level: 100 µA load "1" level: 100 µA load	0		0.5	Volts
Digital Input	"0" level "1" level	0		0.7V	Volts
Power Supply Noise	wide band			10	mV p-p
Supply Current	$T_A = 25^\circ\text{C}$		10		mA
Noise Tolerance	MIL1500 CUI 1700			-12	dB referenced to lower amplitude tone
Input Impedance	$V_p - 2V$ to $V_p - 10V$			100	kΩ at 1500

**OUTPUTS D1, D2, D4, D8 and EN**  
(Pins 1, 18, 17, 16 and 3)

Outputs D1, D2, D4, and D8 are CMOS push-pull when enabled (EN High) and open circuit (high-impedance) when disabled by pulling EN low. These digital outputs provide the code corresponding to the detected digit in the format programmed by the HEX/B28 pin. The digital outputs become valid after a tone pair has been detected and they are then cleared when a valid pause is timed.

**NIC PIN (Pin 6)**

This pin has no internal connection and may be left floating.

**1N1633 (Pin 4)**

When tied high, this pin inhibits detection of tone pairs containing the 1633-Hz component. For detection of all 16 standard digits, 1N1633 must be tied low.

**DTMF RECEIVER TIMING**

**DTMF DIALING MATRIX**

	Col 0	Col 1	Col 2	Col 3
Row 0	1	2	3	A
Row 1	4	5	6	B
Row 2	7	8	9	C
Row 3	0	*	#	D

**DTMF RECEIVER TIMING**

Note: Column 3 is for special applications and is not normally used in telephone dialing.

**DTMF RECEIVER TIMING**

**HEX/B28 (Pin 2)**

This pin selects the format of the digital output code. When HEX/B28 is tied high, the output is hexadecimal. When tied low, the output is binary coded 2 of 8. The table below describes the two output formats.

Digit	Hexadecimal		Binary Coded 2 of 8	
	D8	D4	D2	D1
1	0	0	1	0
2	0	0	1	0
3	0	0	1	0
4	0	1	0	0
5	0	1	0	0
6	0	1	0	0
7	0	1	1	0
8	1	0	0	0
9	1	0	0	1
0	1	0	1	0
*	1	0	1	1
#	1	1	0	0
A	1	1	0	1
B	1	1	1	0
C	1	1	1	1
D	0	0	0	0

**1N1633 (Pin 4)**

When tied high, this pin inhibits detection of tone pairs containing the 1633-Hz component. For detection of all 16 standard digits, 1N1633 must be tied low.

**DTMF RECEIVER TIMING**

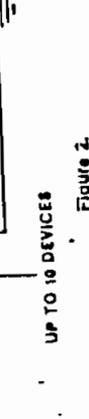
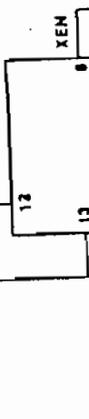
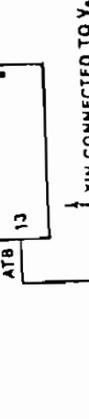
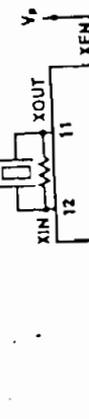
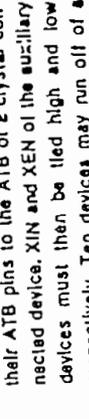
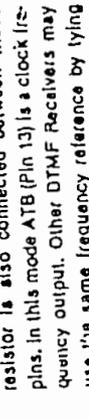
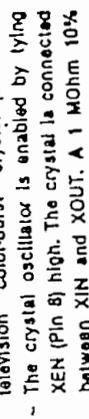
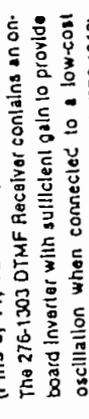
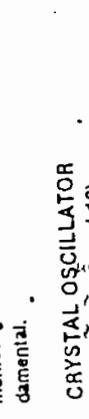
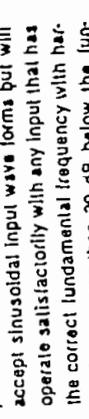
**PIN FUNCTIONS**

**ANALOG IN (Pin 9)**

This pin accepts the analog input. It is internally biased so that the input may be DC coupled as long as it does not exceed the positive supply. The 276-1303 is designed to accept sinusoidal input wave forms but will operate satisfactorily with any input that has the correct fundamental frequency with harmonics greater than 20 dB below the fundamental.

**CRYSTAL OSCILLATOR**  
(Pins 8, 11, 12 and 13)

The 276-1303 DTMF Receiver contains an on-board inverter with sufficient gain to provide oscillation when connected to a low-cost television "color-burst" crystal (272-1310). The crystal oscillator is enabled by tying XEN (Pin 8) high. The crystal is connected between XIN and XOUT. A 1 MOhm 10% resistor is also connected between these pins. In this mode ATB (Pin 13) is a clock frequency output. Other DTMF Receivers may use the same frequency reference by tying their ATB pins to the ATB of a crystal connected device. XIN and XEN of the auxiliary devices must then be tied high and low respectively. Ten devices may run off a single crystal-connected 276-1303 as shown below.



**UP TO 10 DEVICES**

## Application Guide

### For The 276-1303 Dual-Tone Multi- Frequency (DTMF) Receiver

The 276-1303 Integrated DTMF Receiver is a complete Touch-Tone detection system. It can operate in a stand-alone mode for the majority of telecommunications applications, thereby providing the most economical implementation of DTMF signaling systems possible. The IC combines precision active filters and analog circuits with digital control logic on a monolithic CMOS integrated circuit. The use of the DTMF Receiver is straightforward and the external component requirements are minimal.

This application guide describes device operation, performance, system requirements, and typical application circuits for the DTMF Receiver IC.

One Low-Group Tone	897 or 770 or 852 or 941 Hz
One High-Group Tone	1209 or 1336 or 1477 or 1633 Hz
Frequency Tolerance	$\pm 1.5\% \pm 2$ Hz
Amplitude Range	-24 dBm < A < +6dBm @ 800 Ohm (Dynamic Range 30 dB)
Relative Amplitude (Twist)	High-Group Tone -8dB < Low-Group Tone < +4 dB
Duration	40ms or longer
Inter-Tonal Pauses	40ms or longer

The 276-1303 DTMF Receiver meets or exceeds these standards.

Figure 1 shows the DTMF Receiver Block Diagram. In general terms, the detection scheme is as follows: The input signal is pre-filtered and then split into two bands, each of which contains only one DTMF tone group. The output of each band-split filter is amplified and limited by a zero-crossing

### How the DTMF Circuit Works

#### General Description of Operation

The task of a DTMF Receiver is to detect the presence of a valid tone pair on a telephone line or other transmission medium.

The presence of a valid tone pair indicates a single dialed digit; to generate a valid digit sequence, each tone pair must be separated by a valid pause.

The following table gives the established Bell system standards for a valid tone pair and a valid pause:

One Low-Group Tone	897 or 770 or 852 or 941 Hz
One High-Group Tone	1209 or 1336 or 1477 or 1633 Hz
Frequency Tolerance	$\pm 1.5\% \pm 2$ Hz
Amplitude Range	-24 dBm < A < +6dBm @ 800 Ohm (Dynamic Range 30 dB)
Relative Amplitude (Twist)	High-Group Tone -8dB < Low-Group Tone < +4 dB
Duration	40ms or longer
Inter-Tonal Pauses	40ms or longer

The 276-1303 DTMF Receiver meets or exceeds these standards.

detector. The limited signals, in the form of square waves, are passed through tone frequency band pass filters. Digital logic is then used to provide detector sampling and determine detection validity, to present the digital output data in the correct format, and to provide device timing and control.

## Description of Operation

### Noise and Speech Immunity

The two largest problems confronting a DTMF Receiver are:

1) Distinguishing between valid tone pairs (or pauses) and other stray signals (or speech) that contain valid tone pair frequencies.

2) Detecting valid tone pairs in the presence of noise, which is typically found in the telephone (or other transmission medium) environment.

The 276-1303 DTMF Receiver uses several techniques to distinguish between valid tone pairs and other stray signals. Briefly, the techniques are:

1) Pre-filtering of audio signal. Removes stray noise and dia-tone from the input audio signal and emphasizes the voice frequency domain.

2) Zero-cross detection. Limits the acceptable level of noise during detection of a tone pair. Important for speech rejection.

3) Valid tone pair/pause sampling. Samples the detection filters and checks for consistency before determining that a received tone pair or pause is valid.

### How To Use the DTMF Receiver

#### Precautions

Although static protection devices are provided on the high impedance inputs, normal handling precautions observed for CMOS devices should be used.

A destructive high current latch-up mode will occur if pin voltages are not constrained to the range between  $-0.5$  Volts and  $V_p + 0.5$  Volts (except Analog Input as described below). In applications where voltage spikes may occur, protection must be provided to ensure that the maximum voltage ratings are not exceeded. This may require the use of clamping diodes on the Analog Input to protect against ringing voltage, for example, or on the power supply to protect against supply spikes.

#### Power Supply

Excessive power supply noise should be avoided. Since the digital circuitry of the device possesses the high noise immunity characteristics of CMOS logic, limited power supply noise is required only for the analog section. The analog circuitry of the device requires lower power supply noise levels, as specified in the Electrical Characteristics. Power supply noise effects will be slightly less if the analog input is referenced to  $V_p$ . This is normally accomplished by connecting  $V_p$  to ground and utilizing a negative power supply. The effect of excessive power supply noise will cause decreased tone amplitude sensitivity and less tone detection frequency bandwidth.

#### Digital Inputs

The digital inputs are directly compatible with standard CMOS logic devices powered by  $V_p$ . The input logic levels should swing within 30% of  $V_p$  to insure detection. Any unused input must be tied to  $V_p$  or Ground.

## Analog Input

The Analog Input is the signal input pin for the device, and is specially biased to facilitate its connection to external circuitry, as shown in Figure 3. The signal level at the Analog Input pin must not exceed or fall more than a few volts below the positive supply as stated in the Electrical Specifications. If this condition cannot be guaranteed by the external circuitry, the signal must be AC coupled into the chip with a .01 uF 20% capacitor.

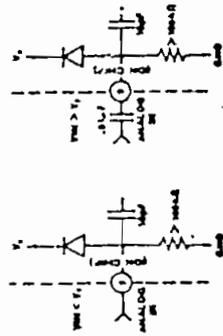


Figure 3. Direct and AC coupled configurations

## Analog Input Noise

This DTMF Receiver will tolerate wide-band input noise of up to 12dB below the lowest amplitude tone fundamental during detection of a valid tone pair. Any single interference frequency (including tone harmonics) between 1 KHz and 6 KHz should be at least 20 dB below the lowest amplitude tone fundamental. Adherence to these conditions will ensure reliable detection and full tone detection frequency bandwidth. Because of the internal band limiting, noise with frequencies above 8KHz can remain unfiltered. However, noise near the 56KHz internal switched-capacitor-filter sampling frequency will be aliased (folded back) into the audio spectrum; noise above 28(KHz therefore should be low-pass filtered with a circuit as shown in Figure 4 using a out-off

## Some of the basic guidelines are:

- 1) Maximum voltage and current ratings of the DTMF Receiver must not be exceeded; this calls for protection from ringing voltage, if applicable, which ranges from 80 to 120 Volts RMS over a 20 to 80KHz frequency range.
- 2) The interface equipment must not breakdown with high-voltage transient tests (including a 2500 Volt peak surge) as defined in the applicable document.
- 3) Phone line termination must be less than 200 Ohms DC and approximately 600 Ohms AC (200-3200 Hz).
- 4) Termination must be capable of sustaining phone line loop current (oll-hook condition) which is typically 18 to 120 mA DC.
- 5) The phone line termination must be electrically balanced in respect to ground.

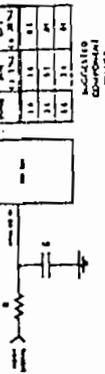


Figure 4. Filter for use in noisy environments

## Telephone Line Interface

In applications that use a DTMF Receiver to decode DTMF signals from a phone line, a DAA (Direct Access Arrangement) must be implemented. Equipment intended for connection to the public telephone network must comply with and be registered in accordance to FCC Part 68. For PBX applications refer to EIA Standard RS-464.

Figure 6 shows a more featured version of Figure 5. These added options include:

- 1) A 150 Volt surge protector to eliminate high voltage spikes.
- 2) Back-to-back Zener diodes to protect the DTMF (and optional multiplexer Op-Amp) from finger voltage.
- 3) Audio multiplexer which allows either voice or other audio to be placed on the line (a recorded message, for example) and not interfere with incoming DTMF tone detection.

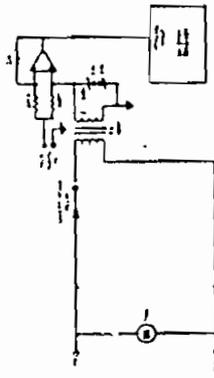


Figure 6. Full Featured Phone Line Interface

An integrated voice circuit may also be implemented for line coupling, however this approach is typically more expensive than using a transformer as shown above.

## Outputs

The digital outputs of the DTMF Receiver (Except XOUT) swing between Vp and GND and are fully compatible with standard CMOS logic devices powered from Vp and GND. The device will also interface directly to LSTTL.

Data Outputs D8, D4, D2 and D1 are three-state enabled to facilitate interface to a three-state bus. Figure 7 shows the equivalent circuit for the data outputs in the high impedance state. Care must be taken to prevent either substrate in Figure 7 from becoming forward biased or damage may result.

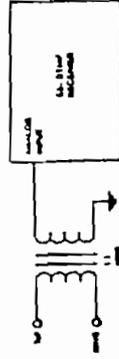


Figure 5. Simplified Phone Line Interface

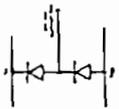
Figure 5 shows a simplified phone line interface using a 600 Ohm 1:1 line transformer. Transformers specifically designed for phone line coupling are available from many transformer manufacturers.

Public phone line termination equipment must be registered in accordance with FCC Part 68 or connected through registered-protection circuitry. Registration typically takes about six months.

### System Interface

Provision has been made on the 276-1303 for handshake interface with an outside monitoring system. In this mode, the DV strobe is polled by the monitoring system at least once every 40 ms to determine whether a new valid tone pair has been detected. If DV is high, the coded data is stored in the monitoring system and then CLR DV is pulsed high. With some systems operating in the handshake mode, it may be desirable to know when a valid pause has occurred. Ordinarily this would be indicated by the falling edge of DV. However, in the handshake mode, DV is cleared by the monitoring system each time a new valid tone pair is detected and, therefore, cannot be used to determine when a valid pause is detected. The detection of a valid pause in this case may be observed by detecting the clearing of the Data Outputs. Since, in hexadecimal format (the mode normally used with a handshake interface), the all zero state represents a commonly unused tone pair (D), the detection of a valid pause may be detected by connecting a four-input NOR gate to the device outputs and sensing the all zero state.

Figure 7. Equivalent Circuit of DTMF Receiver Data Output in High Impedance State.



### Timing

Within 40 ms of a valid tone pair appearing at the DTMF Receiver Analog Input, the Data Outputs will become valid. Seventy microseconds after the data outputs have become valid DV will be raised. DV will remain high and the outputs valid while the valid tone pair remains present. Within 40 ms after the tone pair stops, the DTMF will recognize a valid pause. DV is lowered approximately 45 ms following the end of the tone pair, and the data outputs all set to zero 4.56 ms following DV going low. DV will strobe at least for the same duration as the received tone pair.

The DTMF Receiver contains an on-chip oscillator for a 3.5795 MHz parallel resonant quartz crystal or ceramic resonator. The crystal (or resonator) is placed between XIN and XOUT in parallel with a 1 M $\Omega$  resistor, while XEN is tied high. Since the switched-capacitor-filter time base is derived from the oscillator, the tone detect band frequency tolerance is proportional to the time base tolerance. The 276-1303 frequency response and timing is guaranteed with a time base accuracy of at least .01%. To obtain this accuracy, a quartz crystal is recommended. In less critical applications a suitable ceramic resonator may be implemented.

The use of a ceramic resonator requires the addition of two 30 pF 10% capacitors; one between XIN and Ground and the other between XOUT and Ground. Extra caution should be used to avoid stray capacitance on the resonant circuit when using a ceramic resonator instead of a quartz crystal.

When the oscillator is connected as above and XEN is tied high, the ATB (Alternate Time Base) pin delivers a square wave output at one-eighth the oscillator frequency (447.443 KHz nominal). The ATB pin can be converted to a time base input by tying XEN low; ATB can then be externally driven from another device such as the ATB output of another DTMF Receiver. No crystal is required for the ATB input device; XIN must be tied high if unused. Several DTMF receivers can be driven with a single crystal.

XOUT is designed to drive a resonant circuit only and is not intended to drive additional devices. If a 3.58 MHz clock is needed for more than one device and it is desirable to use only one resonant device, and outside inverter should be used for the time base, buffered by a second inverter or buffer. The buffer output would then drive XIN of the DTMF Receiver as well as the other devices; XOUT must be left floating and XEN tied high.

The 276-1303 DTMF Receiver incorporates enough dial tone rejection circuitry to provide dial tone tolerance of up to 0 dB. Dial tone tolerance is defined as the total power of precise dial tone (350Hz and 440Hz at equal amplitudes) relative to the lowest amplitude tone in a valid tone pair. The filter of Figure 8 may be used for further dial tone rejection. This filter exhibits an elliptic highpass response that provides a minimum of 18 dB rejection at 350Hz and 24 dB rejection at 440 Hz so long as the component tolerances indicated are observed. The DTMF on-chip filter rejects 350 Hz at least 8 dB more than 440 Hz. Therefore, employing the filter of Figure 8 yields a dial tone tolerance of +24 dB.

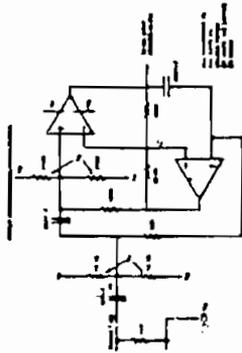


Figure 8. Dial Tone Reject Filter

### Printed Circuit Board Implementation

The 276-1303 DTMF Receiver is analog in nature and should be treated as such; circuit noise should be kept to a minimum. To be certain of this, all input and output lines should be kept away from noise sources (high frequency data or clock lines); this is especially true for the Analog Input. Noise in the ground or power supply lines can be avoided by running separate traces to support logic circuits or by running (thick wire power resistance) buses. Capacitance (power supply bypassing should be performed at the device. Refer to the Power Supply section above.

## APPLICATIONS

### 16-Channel Remote Control

DTMF Signaling provides a simple, reliable means of transmitting information over a 2-wire twisted pair. The complete schematic of a 16-channel remote control is shown in Figure 9. When one of the key pad buttons is depressed, a tone pair is sent over the transmission medium to the DTMF Receiver. The SK4514B raises one of its 16 outputs in response to the 4-bit output code from the DTMF. The output at the 4514 will remain high until the next button is depressed.

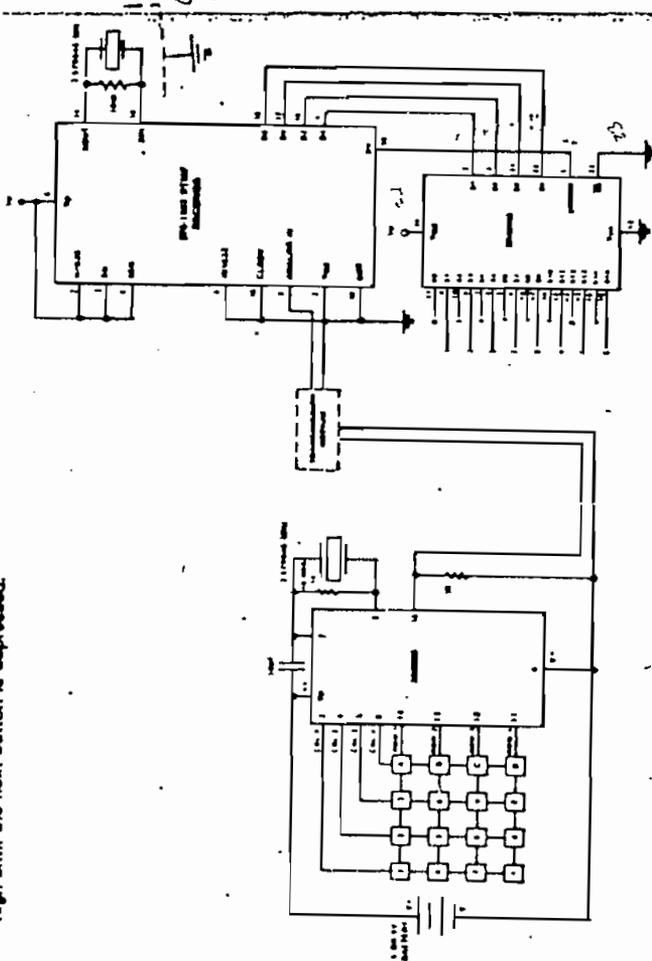


Figure 9. 16 Channel Remote Control

### 2-of-8 Output Decode

The circuit shown in Figure 10 can be used to convert the binary coded 2-of-8 to the actual 2 of 8 code for 2 of 7 if detection of 1633 Hz tone is inhibited. The output data will be valid while DV is high. If it is desired to force the eight outputs to zero when a valid tone is not present, DV should be inverted and connected to both E-NOT inputs of the SK4555B.

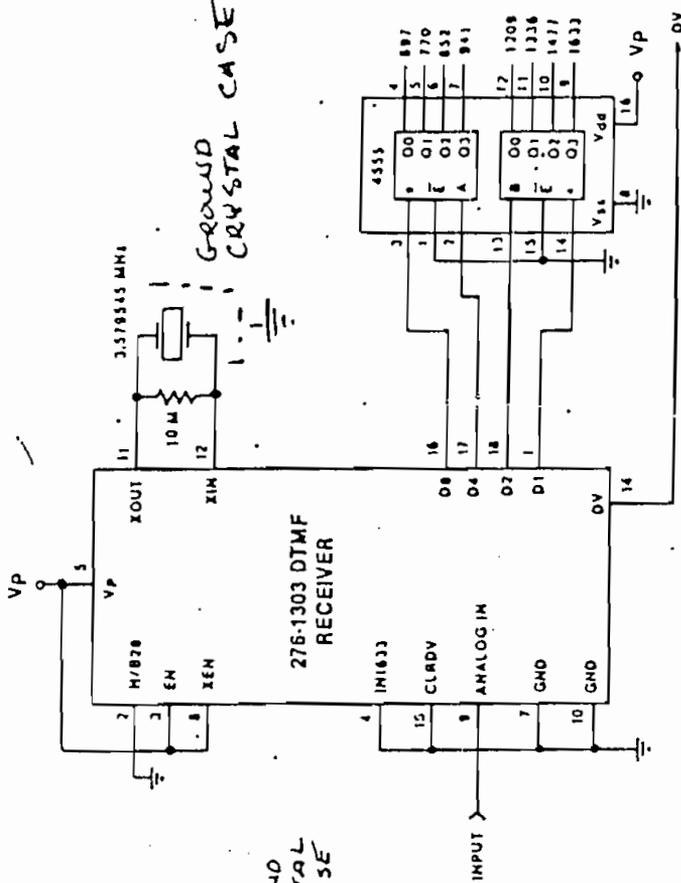


Figure 10. Touch Tone to 2-of-8 Output Converter

Note: All semiconductor components shown can be ordered through your local Radio Shack. Refer to the "SK" series number when ordering.

RADIO SHACK, A Division of Tandy Corporation

U.S.A.: FORT WORTH, TEXAS 76102  
CANADA: BARRIE, ONTARIO L4M 4W5

AUSTRALIA	BELGIUM	FRANCE	U. K.
81 Kurling Avenue Mount Druitt, N.S.W. 2170	Aut des Puits d'Herbert, 38 5140 Marcinelle (Marin)	BP 117-50072 CERGY Pontoise Cedex	Millon Road Walsleybury West Midlands B75 8 2AF