



UNIVERSIDAD SIMÓN BOLÍVAR

Ingeniería Electrónica

DISEÑO Y ELABORACION DE UN EQUIPO
EMULADOR DE IMPULSOS DE COMPUTO (EIC)
PARA FABRICACION TELEFONICA

Por

Alejandro Esteban Arells Pérez

INFORME FINAL DE CURSOS EN COOPERACION

Presentado ante la Ilustre Universidad Simón Bolívar

como Requisito Parcial para Oportar el Título de

Ingeniero Electrónico

Caracas, Febrero de 1991



UNIVERSIDAD SIMON BOLIVAR

Ingeniería Electrónica

**DISEÑO Y ELABORACION DE UN EQUIPO
EMULADOR DE IMPULSOS DE COMPUTO (EIC)
PARA TARIFICACION TELEFONICA**

Por

Alejandro Esteban Avella Pérez

INFORME FINAL DE CURSOS EN COOPERACION

Presentado ante la Ilustre Universidad Simón Bolívar

como Requisito Parcial para Optar al Título de

Ingeniero Electrónico

Sartenejas, Febrero de 1994

DISEÑO Y ELABORACION DE UN EQUIPO EMULADOR DE IMPULSOS DE COMPUTO (EIC) PARA TARIFICACION TELEFONICA

Por

Alejandro Esteban Avella Pérez

Tutor Académico: Prof. Calogero Brunscianelli

Tutor industrial: Ing. Fernando Consalvo

Empresa: SIEMENS S.A. División de Comunicaciones y Datos

RESUMEN

En el presente proyecto se ha diseñado y construido un prototipo de equipo, que emula los impulsos de cómputo utilizados para la tarificación telefónica en Venezuela. El objetivo de este desarrollo es incrementar la cantidad de facilidades, relacionadas con la tasación, que pueden ser aprovechadas en los diferentes modelos de centrales telefónicas privadas SIEMENS y que dependen directamente de la recepción de los impulsos de cómputo para su funcionamiento.

El Emulador de Impulsos de Cómputo (EIC) está controlado por el microprocesador 8031 de INTEL y tiene una capacidad de supervisión de 24 líneas urbanas como máximo, lo que cubre los requerimientos de las centrales SIEMENS de pequeña y mediana capacidad. El equipo es programable a través del puerto serial de un PC con la ayuda de un programa diseñado para correr bajo ambiente WINDOWS. Los parámetros programables son: la frecuencia de los impulsos de cómputo (50 Hz, 12 Khz o 16 Khz), el número de líneas a atender (8,16,24), las tablas de tarificación de la CANTV y la hora y fecha del día.

El EIC permitirá la comercialización de otras facilidades de las PABX SIEMENS hasta ahora no utilizadas en Venezuela (debido a que la telefonía pública generalmente no ofrece la señal de cómputo a abonados normales). Entre las facilidades de las centrales que se activan, destacan las siguientes: la presentación del costo de la llamadas en la pantalla de visualización de los teléfonos digitales a medida que estas se llevan a cabo y el registro de llamadas salientes en forma mas exacta.

Adicionalmente y como capítulo aparte, se presenta el diseño de un "Tarificador Telefónico Casero" (TTC), el cual tendría por objeto llevar el registro de llamadas realizadas en el hogar o en un establecimiento pequeño.

A mis papás, que lo han dado todo por mí y han estado siempre apoyándome;

A la memoria de mi abuela "Eca", porque siempre me quiso ver donde estoy hoy;

A mi abuelo y a mi abuela, por todo el cariño y empeño que pusieron en mi formación;

A mi hermano Armando, porque además de hermano es mi mejor amigo;

A Laura, mi novia querida, porque donde hay amor no existen las palabras...

AGRADECIMIENTOS

A **Fernando Consalvo**, mi tutor industrial, por estar en todo momento disponible a atender mis dudas y por poner todo su empeño por sacar este proyecto adelante.

A **Calogero Brunscianelli**, mi tutor académico, por oír los problemas que tuve para resolver el problema de la contestación y sugerir ideas acertadas.

Un especial agradecimiento a **Antonio Martínez**, por el apoyo incondicional durante toda la pasantía y el aporte de sus conocimientos de telefonía y programación de centrales.

A **Juan Pablo Sosa y Rafael "El Maracucho" Rodríguez**, por brindarme su experiencia en circuitos electrónicos detectores de estados de telefonía.

A los integrantes de la sección de KI: **Jesmina, Juvenal, Iván, Lula y Fortuna**, por integrarme a su grupo de trabajo y hacerme pasar bien, los primeros días dentro de la empresa.

A **Juan Carlos Labastidas, Luis Carrillo, Pedro Fonseca, Eduardo García, Tomás Nadorfy y Valmore Solórzano**, integrantes de KSE, por darme ánimo a seguir adelante y hacer agradable el sitio de trabajo.

A **Leonardo y a Pedro**, encargados del almacén, por atenderme con más que paciencia cada vez que iba a buscar componentes.

A **Omar y Félix**, integrantes del grupo del Taller, por la ayuda prestada en los detalles técnicos del trabajo.

A **Clara Concejo e Isidro "Chilo" Nordelo**, por la ayudarme a hacer el programa bajo "Windows".

A **Jose Antonio Liste**, por ayudarme con el prototipo de Wipe-Rap y hacerme ver las cosas más positivamente.

A **Carlos "Didi" Reyes, a Rafael "El Buho" Franco y a Victor Moreno**, por darme un "empujón moral" para concluir con esta pasantía.

A **Jose Gregorio Da Silva, Marycelly Ordoñez y Jose "Josechu" Espinel**, porque pude contar con ellos para resolver problemas específicos.

A **Ludovina, Raquel, Moira y Aurora**, por ayudarme en la labor de fotocopiado y encuadernación del Libro.

A **Carmen Arce**, por la ayuda en el "formateo" final del libro.

A **Luis y Gregorio**, por ayudarme a resolver problemas que no eran de la pasantía, pero que me aliviaron bastante.

A **Luis Gómez** por brindarme su experiencia en el manejo de programas y por facilitarme la información de Tarificación de la CANTV.

A **Ronald Pinto** por las diligencias que hizo por mí, especialmente la de la búsqueda de la Gaceta Oficial.

INDICE GENERAL

Resumen.....	ii
Dedicatoria.....	iii
Agradecimientos.....	iv
Indice General.....	v
Lista de Figuras.....	vii
Lista de Tablas.....	x
Lista de Abreviaturas.....	xi

INTRODUCCION 1

CAPITULO 1 MARCO TEÓRICO.....	3
A. El sistema telefónico.....	3
B. Señalización.....	5
1. Tipos de señalización.....	5
C. Señales de línea presentes en llamadas salientes y entrantes.....	11
1. Llamadas salientes.....	11
2. Llamadas entrantes.....	11
D. Servicios de telefonía que presta la CANTV.....	13
1. Servicio de telefonía urbana.....	13
2. Servicio de telefonía nacional.....	13
3. Servicio de telefonía internacional.....	14
4. Servicios especiales.....	15
E. Procedimiento de tasación.....	16
F. Tarifas.....	17
1. Conexión a la red.....	18
2. Acceso a la red.....	18
3. Uso.....	19

CAPITULO 2 ESPECIFICACIONES DEL EQUIPO Y ALTERNATIVAS DE SOLUCION

A. Especificaciones de diseño iniciales.....	23
B. Diseño del Emulador de Impulsos de Cómputo EIC.....	23
C. Alternativas del diseño del "hardware".....	25
1. Generación de la frecuencia de los impulsos de cómputo.....	25
2. Máximo número de líneas que puede atender el dispositivo.....	26
3. "Polling" vs. interrupciones.....	26
4. Detector de contestación.....	27
5. Detección de la ausencia del tono de libre por "hardware".....	28
6. Detección de tonos por procesamiento digital de señales.....	28
7. Emulador de Impulsos de Cómputo en Serie.....	28
8. Detectores compartidos vs. Detectores por línea.....	29
9. Elección del microprocesador.....	31
10. Programación del EIC.....	34
11. "Drivers" de comunicación serial.....	35
12. Filtro para el detector del tono de libre.....	35

CAPITULO 3 DESCRIPCIÓN DEL "HARDWARE"	37
A. Tarjeta Controladora.....	38
1. El microprocesador	38
2. Memorias	43
3. Mapa de memoria	43
4. Reloj-calendario.....	45
5. Oscilador de 42.5 Khz.....	46
B. Tarjetas Detectoras.....	47
1. Acople con la línea.....	47
2. Circuitos detectores	49
3. Interfaz digital hacia el CPU	62
4. Acople de impulsos a la línea	62
C. Interfase visual.....	64
D. Interfase serial.....	65
1. Conversión de niveles	65
2. Conexión de los pines del puerto serial	66
3. Conversión de DB9 a DB25.....	68
E. Fuente de alimentación	68
1. Fuente de 5V	68
2. Respaldo de Baterías:.....	69
.....	70

CAPITULO 4 DESCRIPCIÓN DEL "SOFTWARE"	71
A. Programación en Assembler del 8031.....	71
1. Estructura del programa.....	71
2. Módulo "Definición"	72
3. Módulo "Inicialización".....	78
4. Módulo "Principal"	84
5. Módulo "Tiempo"	88
6. Módulo "Display"	91
7. Módulo "Serial"	95
8. Módulo "Aritmética"	96
B. Programa en BorlandC++ para interfaz con el usuario.....	96
1. Descripción del programa	96
2. Estructura del código.....	101

CAPITULO 5 TARIFICADOR TELEFÓNICO CASERO (TTC).....	103
A. Descripción General	103
B. "Hardware"	104
C. "Software"	105
1. Definición de variables.....	106
2. Programa principal.....	106
3. Rutinas adicionales de cálculo	107
4. Rutinas adicionales para la pantalla LCD.....	111

RESULTADOS, CONCLUSIONES Y RECOMENDACIONES	114
A. Resultados	114
1. Supervisión de la línea telefónica	114
2. Conmutación de la frecuencia de los impulsos de cómputo	114
3. Modularidad en "hardware"	115
4. Modularidad en software.....	115
5. Tarificación según la CANTV	115
6. Capacidad de actualizar la hora y la fecha.....	117
7. Tarifificador para un teléfono casero	117
B. Conclusiones.....	117
C. Recomendaciones	118
 BIBLIOGRAFIA:	 120
 APENDICES	 122
1. Diagramas de conexión.....	123
2. Listado del programa en assembler para el EIC.....	131
3. Rutinas adicionales para el TTC.....	163
4. Listado del programa emulador de EPROM	171
5. Registros de programación del 8031	174
6. Tarifas telefónicas	181
7. Informaciones técnicas de algunos componentes	186
8. Glosario de telefonía general	205
9. Fundamentos de la telefonía	210
10. Tabla deCostos.....	220
11. Conectores usados.....	231
12. Analizador de base de datos para EXCEL (versión preliminar).....	233
13. Ensamblaje de un programa en assembler	235
14. Listado del programa en BorlandC++ para interfaz con el usuario	237

LISTA DE FIGURAS

Fig. 1 Partes del sistema telefónico.....	3
Fig. 2 Forma de los impulsos de cómputo.	7
Fig. 3 Discado decádico.....	8
Fig. 4 Frecuencias emitidas en el discado DTMF.	9
Fig. 5 Señalización telefónica en llamadas entrantes y salientes	12
Fig.6 Servicios de telefonía en Venezuela.....	15
Fig. 7 Aplicación de los impulsos de cómputo a la línea telefónica.....	17
Fig. 8 Ubicación del EIC.	24
Fig. 9 Estructura modular del "hardware"	37
Fig. 10 El microprocesador 8031	38
Fig. 11 El 8031 como microprocesador.....	39
Fig. 12 Conexión de los pines especiales del 8031.	40
Fig. 13 Uso de los pines del puerto 1 del 8031.	41
Fig. 14 Conexión para lograr memoria combinada	42
Fig. 15 Mapa de memoria.	43
Fig. 16 Decodificación de memoria.....	44
Fig. 17 Conexión del circuito integrado" Reloj-Calendario.....	45
Fig. 18 Oscilador de onda cuadrada de frecuencia 42,5 Khz.	46
Fig. 19 Diagrama en bloques de una tarjeta detectora	47
Fig. 20 Acople digital a línea telefónica a través de un optoacoplador	48
Fig. 21 Acople analógico con la línea telefónica a través de un transformador.	49
Fig. 22 Funcionamiento del detector de repique.	50
Fig. 23 Funcionamiento del detector de discado decádico.....	51
Fig. 24 Funcionamiento del detector de discado DTMF.....	52
Fig. 25 Diagrama en bloques del detector de tono de libre	53
Fig.26 a) Conexión del MF10 en modo 1.	55
Fig. 27 Funcionamiento del detector de tono de 425 Hz.	58
Fig. 28 Funcionamiento del integrador	60
Fig. 29 Comparador para el detector del tono de libre.....	61
Fig. 30 Error en la detección de la contestación.	62
Fig. 31. Acople de los impulsos a la línea con un relé DPDT.	63
Fig. 32 Conexión de la Pantalla LCD.	65
Fig. 33 Interfaz serial.....	66
Fig. 34 Conexión de los pines del puerto serial	67

Fig. 35 Diagrama en bloques de la fuente de alimentación	69
Fig. 36 Circuito de Respaldo de la alimentación.....	70
Fig. 37 Estructura modular del programa en assembler.....	71
Fig. 38 Estructura de la tabla de variables de análisis.....	74
Fig. 39 Estructura de la variable "Flags".....	76
Fig. 40 Estructura de la variable "Status".....	76
Fig. 41 Estructura del temporizador 0.....	80
Fig. 42 Secuencia de acciones realizadas por el programa principal.....	85
Fig. 43 Diagrama de Flujo de la rutina ANALIZO.....	87
Fig. 44 Forma de los impulsos de cómputo.....	89
Fig. 45 Diagrama de flujo del algoritmo "Ventana".....	91
Fig. 46 Variable en formato packed BCD.....	92
Fig. 47 Formato de la hora y fecha en la pantalla LCD.....	94
Fig. 48 Formato de Transmisión serial.....	96
Fig. 49 Ventana para abrir archivos.....	97
Fig. 50 Ventana de configuración del equipo.....	98
Fig. 51 Ventana para que muestra la Tabla de Tarifas a nivel Nacional.....	99
Fig. 52 Ventana del servicio de telefonía a través de operadores nacional.....	100
Fig. 54 Ventana de la Base de datos recuperada.....	101
Fig. 50 Ubicación física del "Tarificador Telefónico Casero" (TTC).....	104
Fig. 51 Diagrama en bloques del TTC.....	105
Fig. 52 Celdas de 4 bits para conversión binario-BCD.....	110
Fig. 52 Algoritmo de Conversión Binario a BCD.....	110
Fig. 53 Duración y costo de la llamada en la pantalla LCD.....	112
Fig. 54 Variables packed BCD para el costo de la llamada.....	113
Fig. 55 Circuito básico del teléfono.....	213
Fig. 56 Conexión entre Centrales Telefónicas. a) completa. b) estrella. c) combinada.....	215
Fig. 57 Segmento de la red de telefonía pública en Venezuela.....	217
Fig. 63 Formato Intel Hexadecimal.....	235
Fig. 64 Ensamblaje de un programa en assembler.....	236

LISTA DE TABLAS

Tabla 1 Resumen de la "Señalización de Línea"	7
Tabla 2 Comparación de los microcontroladores INTEL de la Familia 51.	34
Tabla 3 Funciones especiales del puerto 3.....	42
Tabla 4 Conexión de los pines del registro de status	62
Tabla 5 Variables necesarias para análisis de una línea.....	78
Tabla 6 Tabla de vectorizaciones.....	79
Tabla 7 Constantes de conteo de recarga del "Timer" 1 para las tres frecuencias.	82
Tabla 8 Constantes de conteo de 50 ms.	88
Tabla 9 Constantes de recarga del temporizador 0.....	90
Tabla 10 Modos de programación de la pantalla LCD.....	93
Tabla 11 Cadencia de los impulsos de cómputo para las llamadas de L.D.N.	181
Tabla 12 Códigos de área y Códigos de tarifa desde Caracas con las principales ciudades del país	182
Tabla 13 Tarifas por Servicio a través de Operadores, desde poblaciones incorporadas al DDN	183
Tabla 14 Tarifas para Servicio Internacional Automático y Manual.....	184
Tabla 15 Horarios para Tarifa Reducida y Tarifa Económica a nivel Internacional.....	185
Tabla 16 Días Festivos en Venezuela para el año 1994.....	185

LISTA DE ABREVIATURAS

A: Amper
ACC: Acumulador
ALE: Address Latch Enable
CANTV: Compañía Anónima de Teléfonos de Venezuela
CCITT: Comité Consultivo Internacional de Telegrafía y Telefonía
COS: Class of Service
CPA: Central Privada Automática
DTMF: Dual Tone Multi Frequency
DPTR: Data Pointer
EA: External Access
EIC: Emulador de Impulsos de Cómputo
EEPROM: Electric Erasable Programable Memory
EPROM: Erasable Programable Memory
Hz: Hertz
IE: Interrupt Enable Register
IP: Interrupt Priority Register
LCD: Liquid Cristal Display
MFC: Código Multifrecuecial de Secuencia Obligada
OE: Output Enable
PABX: Private Access Branch Xchange
PC: Program Counter
PROM: Programable Memory
PSEN: Program Strobe Enable
RAM: Random Access Memory
RD: Read
RXD: Recepción
SCON: Serial Port Control Register
SMDR: Station Message Detail Recording
SBUF: Serial Buffer
SFR: Special Function Register
SP: Stack Pointer
TCON: temporizador/Counter Control Register
TMOD: temporizador/Counter Mode Control Register
TH0-TL0, TH1-TL1: Registros del temporizador 0 y el temporizador 1
TTC: Tarificador Telefonico Casero
TXD: Transmisión
UIT: Union Internacional de Telecomunicaciones
V: Voltios
WR: Write

INTRODUCCION

La comunicación eficiente adquiere cada vez mayor importancia como factor decisivo de competitividad en todos los sectores económicos. Con el creciente uso de los servicios de comunicación de voz, textos y datos, las inversiones y los gastos por concepto de comunicaciones alcanzan un volumen significativo que debe tenerse en cuenta en la relación de gastos de cualquier empresa.

La mayoría de las centrales telefónicas privadas (CPA o PABX), son capaces de llevar un registro de las comunicaciones originadas en el sistema. Los datos registrados ofrecen estadísticas valiosas para optimizar las comunicaciones así como la posibilidad de elaborar reportes que sirven para la asignación de los gastos a los responsables correspondientes.

Sin embargo, el paquete de facilidades relacionadas con tasación que pueden ofrecer los diferentes modelos de centrales telefónicas, son parcialmente utilizados en Venezuela, debido a que los impulsos de cómputo no son aplicados a las comunicaciones. Estos impulsos son suministrados por la CANTV, sólo bajo solicitud expresa de alguna entidad que preste algún servicio a la comunidad tales como: hoteles, clínicas, farmacias, etc. Cabe destacar que este servicio no se le presta a particulares, y que además, existe una capacidad limitada para ofrecerlo por parte de las centrales públicas.

De esta manera, surge en el seno de la división de Comunicaciones y Datos de la empresa SIEMENS S.A. de Venezuela, la idea de diseñar un equipo que emule los impulsos de cómputo, con miras a activar las facilidades de tasación no utilizadas en Venezuela de las centrales de baja y mediana capacidad comercializadas actualmente por la empresa, tales como: "HICOM 102", "HICOM 125", "HICOM 130", etc.

El proyecto se desarrolló en los meses de Julio a Diciembre de 1993 (un total de 5 meses), como un curso en cooperación entre la empresa SIEMENS S.A. de Venezuela y la Universidad Simón Bolívar, bajo la modalidad conocida como "pasantía larga".

Este documento condensa todos los aspectos concernientes al diseño y elaboración del equipo Emulador de Impulsos de Cómputo (EIC). Se presenta en el primer capítulo los fundamentos de señalización y tarificación, de manera de facilitar la comprensión de los capítulos siguientes. En el capítulo 2 se puntualizan las especificaciones iniciales del proyecto y se desarrolla una discusión de las posibles alternativas de solución. Los capítulos 3 y 4 del libro son una descripción detallada del "hardware" y "software" del equipo diseñado, respectivamente. El capítulo 5 contiene la versión final del dispositivo construido, presentando además los planos del circuito impreso y un análisis de costos totales de fabricación. El último capítulo del cuerpo del trabajo consiste de la descripción del "software" y "hardware" necesarios para la fabricación de un Tarificador Telefónico Casero (TTC). Por último, se presentan los resultados, las conclusiones y las recomendaciones.

CAPITULO 1 MARCO TEÓRICO

A. El sistema telefónico

El sistema telefónico ha sido diseñado para que presente una impedancia característica A.C. en la línea de aproximadamente 600 Ohms. Cabe mencionar que, este es un valor nominal que varía notablemente con la distancia y la calidad de servicio.

La alimentación para que opere el sistema telefónico es puesta por la compañía telefónica, modalidad conocida como "alimentación por batería central" y se deriva de un conjunto de baterías conectadas en serie para producir una tensión de 48 Volt DC. Esta batería es constantemente cargada, por lo que es capaz de alimentar el sistema telefónico así haya un fallo de electricidad. Al medir la tensión V_{ab} del par telefónico se obtiene -48 V . La razón de la conexión a tierra del positivo de la batería es que al hacerlo de esta forma y no al revés, el desgaste de los cables de cobre es menor, ya que el fenómeno de electrólisis que destruye el cobre se produce en menor grado.

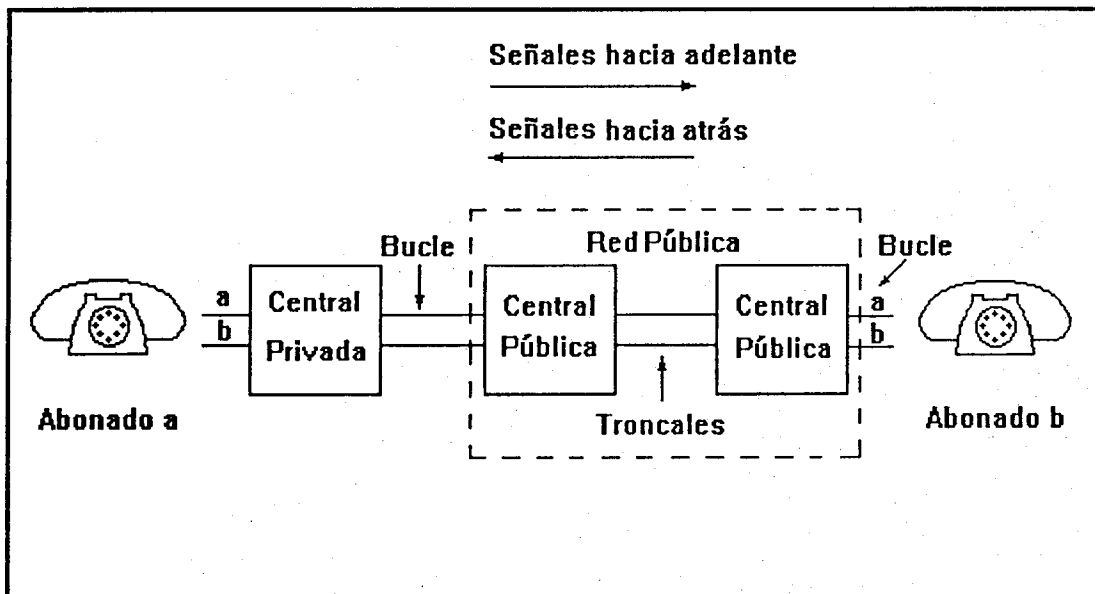


Fig. 1 Partes del sistema telefónico.

Cuando el teléfono está colgado ("on hook") el voltaje de la batería aparece completo a través de la línea telefónica, es decir hay 48 Voltios y no hay prácticamente circulación de corriente. Cuando el auricular del teléfono es levantado ("off hook"), baja la impedancia del aparato telefónico lo que causa que el voltaje a través de la línea caiga hasta 6 Voltios y comience a circular una corriente de aproximadamente 20 ma en el bucle del abonado.

En la Figura 1, se tiene un diagrama en bloques simplificado del sistema telefónico. Los integrantes fundamentales del sistema son los abonados, que son los usuarios de los servicios telefónicos. Cuando un abonado origina una llamada se le denomina abonado "a", mientras que a la persona llamada se le denomina abonado "b".

Los equipos capaces de conmutar a los abonados unos con otros son las centrales telefónicas. La red pública básica es un conjunto de centrales telefónicas públicas unidas por conductores de cobre, por los que se enrutan las llamadas. A los circuitos entre centrales se les denomina enlaces o como se les conoce más comúnmente troncales.

La voz de los abonados en un enlace común se transmite directamente a la línea sin modulación alguna. La corriente alterna de la voz se superpone a la corriente directa de la alimentación del sistema. Sin embargo, en algunos enlaces, la voz de los abonados en su tránsito por la red pública puede sufrir diversas modificaciones. Por ejemplo, la voz puede ser modulada o digitalizada, y el enlace entre centrales no necesariamente tiene que ser por un conductor bifilar. Por ejemplo, se pueden enlazar centrales vía microondas o vía satélite.

Los abonados residenciales tienen su teléfono conectado directamente al par de hilos que proporciona la compañía telefónica. La unión física del teléfono con la central telefónica que le corresponda se le denomina bucle o lazo de abonado.

Las empresas o establecimientos poseen típicamente centrales privadas con el propósito de disminuir el número de líneas necesarias y a fin de optimizar sus comunicaciones. En la Figura 1, se pone como ejemplo la conexión del abonado "a" a una central privada, que está conectada a su vez con una central terminal de la red pública.

B. Señalización

1. Tipos de señalización

El sistema de señalización presenta una división de las señales en señales de línea y señales entre registros. El primer grupo comprende principalmente señales para el control de la comunicación, antes, durante y después de la comunicación, mientras que el segundo grupo, señales entre registros, se emplean para retransmisión de la información numérica y control del establecimiento de la comunicación. A continuación se presenta una descripción general de estos dos tipos de señalización:

a) Señales de línea

La función principal de las señales de línea es la de supervisión. La central por ejemplo, supervisa el descolgado de los teléfonos, les manda el tono de invitación a marcar y se queda esperando por el número del destinatario.

Estas señales son transmitidas antes, durante y después del establecimiento de la conexión. Las señales son transmitidas mediante señalización de bucle de corriente continua (c.c.) y entre las más importantes se encuentran las siguientes:

. **El repique:** Es una señal de línea producida en las centrales públicas para avisar a los abonados que tienen una llamada en espera. Tiene una amplitud de 90 Volt. rms y una frecuencia de 25 Hz. Esta señal es producida por generadores rotativos llamados máquinas de repique o por circuitos de estado sólido que no utilizan partes móviles.

. **El tono de invitación a marcar:** Esta señal avisa al abonado, que puede empezar a marcar el número del abonado con el que se desea comunicar. Es un tono de 425 Hz continuo, que se emite desde el momento en que un abonado levanta el auricular de su teléfono, y se deja de emitir cuando el usuario disca el primer número.

• **El tono de libre:** Con esta señal se le indica al abonado llamante, que se ha seleccionado al abonado llamado y que se le está enviando el repique. Es de una frecuencia de 425 Hz, y tiene una cadencia de 1 seg. de emisión de señal y 4 seg. de silencio.

• **El tono de ocupado:** La central pública envía esta señal al abonado llamante para informarle que el abonado llamado tiene su línea ocupada. También es un tono de una frecuencia de 425 Hz, pero la cadencia es de 0,5 seg de emisión y 0,5 seg de silencio.

• **La señal de contestación:** Es una señal que se emite desde la central destino, cuando el abonado llamado contesta y se retransmite a la central de origen. Esta señal es una inversión de polaridad del par telefónico y su propósito es el de comenzar la tasación desde el momento del establecimiento de la conversación. Sin embargo, esta inversión de polaridad **no** se retransmite al lazo del abonado llamante, sino sólo hasta la central pública de origen a la cual esta conectado el abonado llamante.

• **Los impulsos de cómputo:** Son la señalización utilizada para la contabilización de llamadas, y tienen las siguientes características:

Frecuencia: 16 Khz.

Tensión: 2,2 a 4 Voltios - Distorsión del 5 %

Cadencia: 150 ms. de emisión y tiempo variable en silencio, que depende del destino de la llamada.

En la Figura 2 se aprecia más clara la forma de estos impulsos.

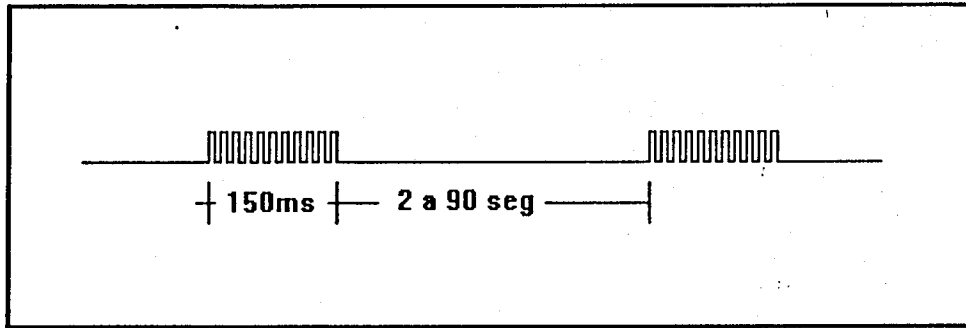


Fig. 2 Forma de los impulsos de cómputo.

En la Tabla 1 se presenta un resumen de las señales de línea que son relevantes para la detección de cambios de estado en la línea.

Tono	Frecuencia (Hz)	Cadencia (Seg)*	Voltaje (Vp)	Función
Invitación a marcar	425	Continuo	1 - 1.5	Indica al abonado "a" que puede discar.
Libre	425	<u>1</u> / 4	1 - 1.5	Indica al abonado "a" que al abonado "b" se le está enviando repique.
Ocupado	425	<u>0.5</u> / 0.5	1 - 1.5	Indica al abonado "a" que el abonado "b" tiene su línea ocupada.
Repique	25	<u>1</u> / 4	90	Corriente de llamada enviada al abonado "b" para que suene el timbre de su teléfono.

Tabla 1 Resumen de la "Señalización de Línea"

* Las cantidades subrayadas representan emisión de la señal.

El **discado** es una parte fundamental de la señalización de línea. La función del discado es informarle a las centrales públicas con que abonado nos deseamos comunicar. Existen básicamente dos formas de transmitir los dígitos del número del destinatario desde un aparato telefónico a la central pública. Ellas son: el discado decádico y el discado multifrecuencial. A continuación se describen brevemente estos dos tipos de discado:

- Discado Decádico

El **discado por pulsos o decádico** son una serie de aperturas y cierres del lazo que ocurren a una tasa de aproximadamente 10 pulsos por segundo (10 pps), durante la rotación antihoraria del dial. El número de aperturas del lazo está determinada por el dígito marcado con la excepción del número cero, en el cuál se producen 10 aperturas del lazo.

Para la detección correcta del discado decádico, debe dejarse un espacio entre dígitos (pausa interdígito) por un tiempo mayor a 500 ms, lo que se logra con un mecanismo interno del disco rotatorio. La relación de apertura del bucle es de $0.66 \pm 10\%$ (≈ 60 ms) y la relación de cierre del bucle es de $0.44 \pm 10\%$ (≈ 40 ms). En la Figura 3 se ilustra la forma de onda del discado decádico.

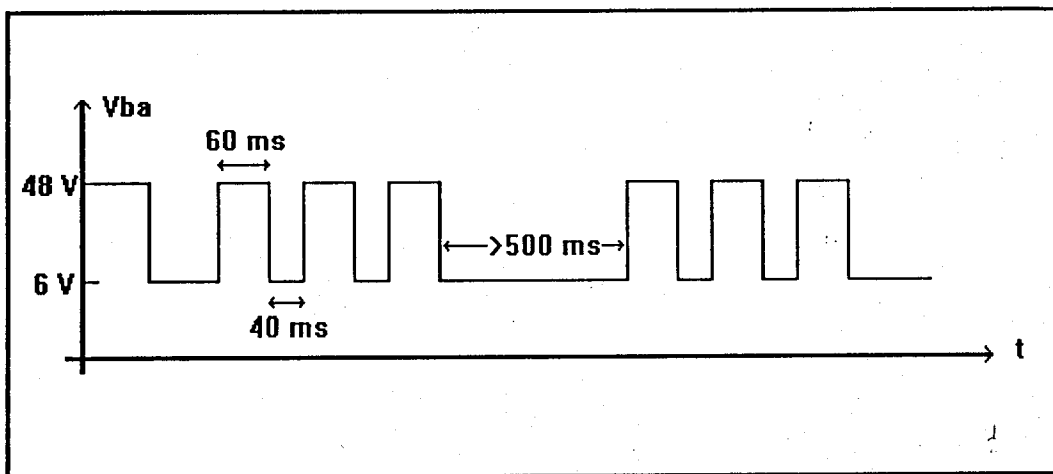


Fig. 3 Discado decádico.

- Discado Multifrecuencial

Una mejora importante en cuanto al discado por pulsos o decádico fue lograda con el **discado multifrecuencial (DTMF) "Dual Tone Multi Frequency"**. Esta mejora en la forma de discar, fue posible gracias a la tecnología de estado sólido y a la necesidad de una manera más rápida y exacta de discar.

El discado multifrecuencial se genera con un teclado de 12 botones donde están los 10 dígitos (0, 1, 2,..., 9) más dos botones adicionales usados para funciones especiales.

Cuando cualquiera de los botones es presionado, un "circuito integrado" dentro del teléfono produce un par de tonos que se encuentran en el rango de frecuencia de la voz, por lo que estos tonos pueden ser transmitidos en cualquier lugar donde las comunicaciones telefónicas estén disponibles.

Las frecuencias están divididas en 2 bandas, alta y baja, y cada botón selecciona una frecuencia de cada banda.

El sistema ha sido diseñado para expansión futura para un total de 16 botones. Los cuatro botones adicionales controlarían un octavo tono de 1633 Hz. La Figura 4 ilustra el par de tonos generados por cada uno de los 12 botones

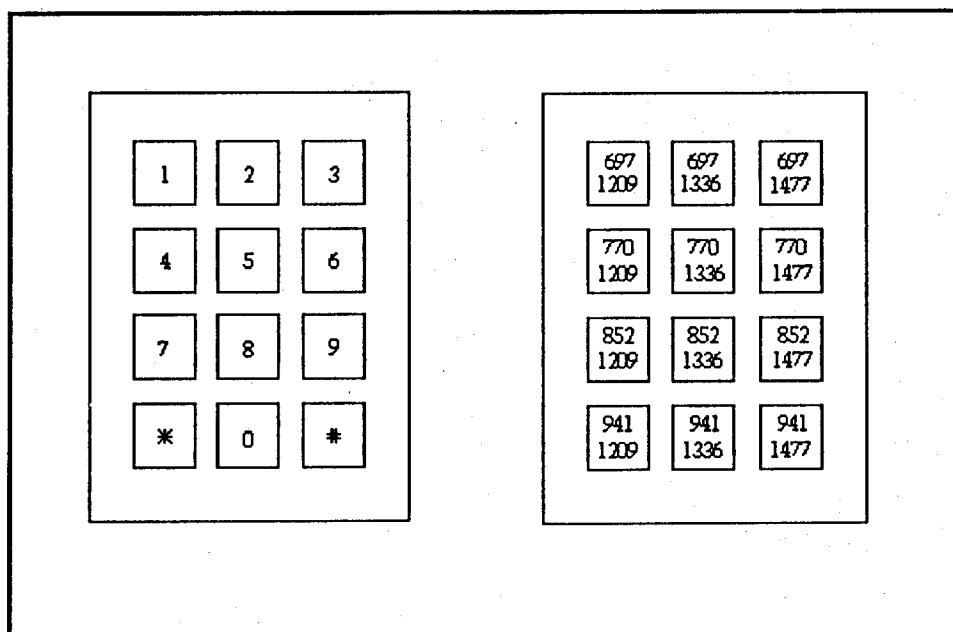


Fig. 4 Frecuencias emitidas en el discado DTMF.

Las señales multifrecuenciales deben tener una duración por dígito de más de 60 ms y tener una pausa entre dígitos mayor a 50 ms, para poder ser tomadas como válidas. La amplitud típica de estas señales es de 1.7 Vp. Los detectores DTMF tienen una tolerancia de las

frecuencias del 1.8%, lo que los hace bastante selectivos, y por lo tanto inmunes a otros tonos de señalización o ruidos que pudieran estar en la línea telefónica.

El discado multifrecuencial es entonces mucho más rápido y confiable que el discado decádico, por lo que la tendencia actual es de ir sustituyendo el discado decádico por el multifrecuencial.

b) Señalización de registro

La señalización entre registros se usa para la transmisión de información del número que el abonado llamante discó y de otro tipo de información, que hacen posible el direccionamiento de la llamada. Esta señalización no se produce en el lazo que forma el abonado y la central pública. La señalización se efectúa desde un registrador a un receptor de código (o registrador) perteneciente al paso de selección en la vía de conexión del abonado llamado. Las señales se intercambian solamente durante el breve tiempo del establecimiento de la conexión, por lo tanto el equipo de señalización puede estar concentrado en una pequeña cantidad de unidades comunes.

La señalización forzada, con código multifrecuencial (señalización MFC), es el protocolo que se usa para la comunicación entre registros. Este código usa un conjunto de frecuencias del rango vocal, distintas a las que se usa en el discado multifrecuencial de los teléfonos. Cada señal se caracteriza por la emisión simultánea de dos frecuencias. Las señales hacia adelante se denominan señales numéricas y las señales hacia atrás señales de mando. En el esquema de señalización se han previsto 6 frecuencias para señales numéricas y 6 frecuencias para señales de mando.

En Venezuela, el sistema de señalización por registro adoptado es el MFC-R2. Este sistema es utilizado por casi la totalidad de la red telefónica venezolana independientemente del sistema de central, categoría de tráfico, etc.

C. Señales de línea presentes en llamadas salientes y entrantes

1. Llamadas salientes

La corriente DC que pide el teléfono al ser descolgado es la señal que le indica a la central pública que la comunicación se ha iniciado. Seguido al descolgado los circuitos de la central pública aplican a la línea telefónica la señal de **invitación a marcar** que se presenta como un tono de 425 Hz continuo sobre la línea y a partir de este momento la central espera por dígitos marcados.

Con el **discado** del primer número la central cesa de enviar el tono de invitación a marcar y a medida que el abonado "a" marca los números las centrales van enrutando la llamada hasta que llega a su destino.

Con la certeza de cual es el destino de la llamada, la central pública comienza a mandar el repique al abonado "b" para que este sepa que lo están llamando y al mismo tiempo envía el **tono de libre** al abonado "a", de manera de informarle que su llamada ha "caído" exitosamente.

Una vez que el abonado "b" contesta, la baja impedancia que el teléfono impone sobre la línea sirve como indicador para cancelar la señal de repique y conectar al abonado "a".

Apenas se produce la contestación, la central pública comienza a aplicar impulsos al contador del abonado "a", los cuales se aplican con un ritmo que depende de la distancia a la que se encuentren los 2 abonados y la hora en que se establezca la comunicación.

En la Figura 5 (Parte 1) se hace una ilustración de la señalización presente en el lazo del abonado "a" cuando se produce una llamada saliente.

2. Llamadas entrantes

En una llamada entrante la señalización que llega de la red pública es el repique, la cual es una señal de una amplitud de 90 Vp y de una frecuencia de 25 Hz.

Cuando el abonado solicitado contesta la llamada, la central pública deja de mandar el repique, y entonces se establece la comunicación entre las dos partes.

Las centrales privadas (PABX) tienen la facilidad de transferencia, con la cual se redirige la llamada a la persona solicitada. Esta facilidad se accesa presionando un botón adicional que tienen los teléfonos (el botón de "Flash"), que al ser presionado realiza una apertura momentánea del lazo de corriente (ver Figura 5 Parte 2) indicando a la central que se debe disponer a recibir los dígitos que identifican la extensión para su posterior redireccionamiento. Otros sistemas ejecutan esta función con el gancho de colgar en lugar del botón de Flash.

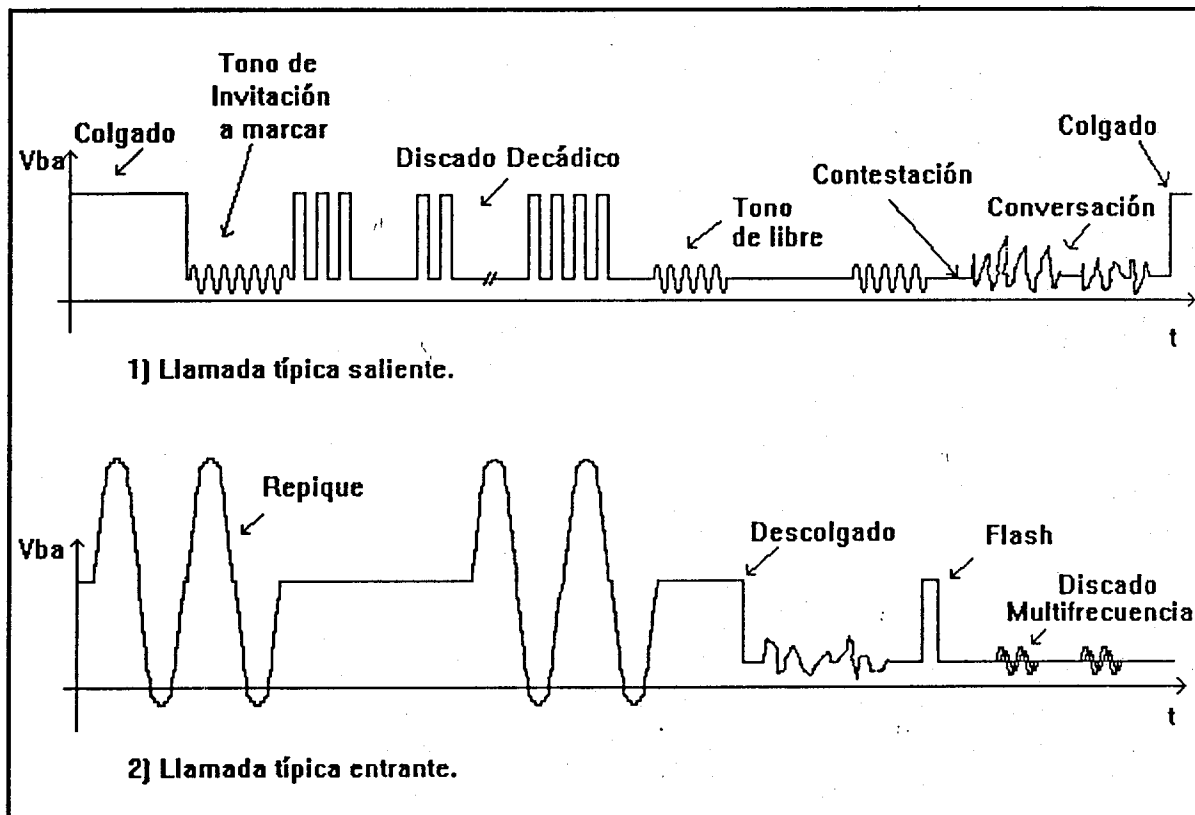


Fig. 5 Señalización telefónica en llamadas entrantes y salientes

D. Servicios de telefonía que presta la CANTV

1. Servicio de telefonía urbana

La llamada local es la que se establece entre dos (2) abonados de una misma ciudad, es decir, en una misma red local o área de numeración cerrada, mientras que toda llamada cuyo destino sea un grupo distinto de aquel en que se originó, es denominada llamada de larga distancia.

En las llamadas locales, para comunicarse con el abonado deseado solo hace falta discar su número de abonado sin necesidad de anteponer ningún tipo de código. (Ver Figura 6 Parte a)

2. Servicio de telefonía nacional

Se denominan conversaciones interurbanas o de Larga Distancia Nacional las realizadas entre abonados situados en distintas zonas de tarificación locales. La CANTV ofrece dos modalidades que son: servicio automático y servicio manual.

a) Automático

Este servicio se ofrece entre las poblaciones más importantes del país integradas al Discado Directo Nacional (D.D.N) , el cual se accede marcando el número nacional.

El número nacional esta formado por tres partes (ver Figura 6 Parte b):

- . El código de acceso
- . El código de área
- . El número del abonado

El código de acceso de larga distancia es un número de un dígito que se emplea para tener conexión con la red de larga distancia. Este código es único en todo el país y el dígito que se debe marcar es el cero "0".

El código de área Nacional está compuesto por uno o dos dígitos y tiene como función identificar las áreas de numeración cerrada. Los códigos de área nacional se encuentran en el Apéndice 6.

Tanto el código de área como el número del abonado pueden hacer uso de todos los dígitos con la excepción del "cero" y el "uno" como primer dígito, ya que estos están reservados para el acceso a larga distancia y servicios especiales respectivamente.

b) Manual

El servicio de Larga Distancia Nacional a través de operadores se presta discando el 100. Al atender el operador, el usuario le informa la población con la que se desea comunicar y el número deseado. La operadora establece la comunicación y se la transfiere luego al solicitante. (ver Figura 6 Parte c)

3. Servicio de telefonía internacional

a) Automático

El Discado Directo Internacional (D.D.I) se accede en Venezuela al marcar el 00 (cero-cero). Luego se disca sin interrupción el código del país, código de la ciudad y el número deseado (Fig. 6 Parte e). Los códigos del discado directo internacional se encuentran en el Apéndice 6.

b) Manual

Esta modalidad de comunicación se realiza a través de solicitud de llamadas internacionales a operadores, a través del 122. Al discar estos números atenderá un operador al que el usuario informa el destino deseado. El operador establece la comunicación y luego establece el enlace entre las partes interesadas. (Ver Figura 6 Parte d)

Existen dos tipos de comunicaciones básicas que se pueden realizar para hacer llamadas internacionales a través de operadores: **llamadas de estación a estación (EE)**, son las que se cobran directamente al ser exitosa la comunicación y **llamadas persona a persona (PP)** que son aquellas en las que se comienza a tasar en el momento en que la persona destino contesta personalmente, estando exentas de tarifa si el abonado llamado no se encuentra en el momento. Las llamadas persona a persona tiene un recargo en la tarifa.

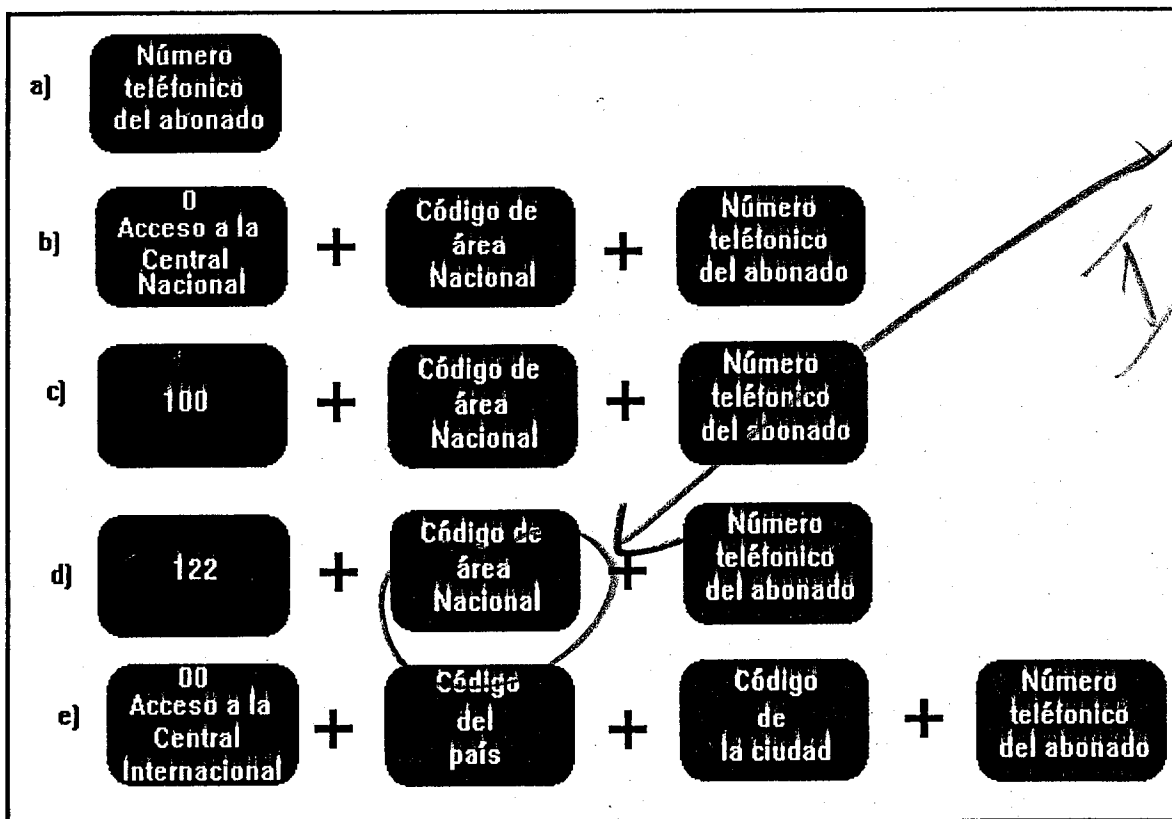


Fig. 6 Servicios de telefonía en Venezuela. a) Urbano b) Nacional automático c) Nacional Manual d) Internacional Manual e) Internacional Automático

4. Servicios especiales

Los códigos de los servicios especiales son números formados por dos o tres cifras, siendo el primero el dígito uno (llamado código de acceso) y el resto sirve para dar acceso a los citados servicios. Estos códigos son únicos para todo el Territorio Nacional.

El Servicio Horario (19) es atendido mediante equipos automáticos y se presta toda hora a nivel nacional, informando tomando como base la hora oficial del Observatorio Cajigal.

El Servicio de Información (103) es atendido por operadores y se presta con el propósito de ayudar a los interesados en obtener información sobre los números telefónicos ubicados en el Territorio Nacional, y que en un momento dado, no tengan a mano un ejemplar de la Guía Telefónica.

El Servicio de Reporte de Averías (15) es atendido por operadores y tiene por objeto centralizar los reclamos por daños sufridos en los teléfonos de los abonados.

E. Procedimiento de tasación

El procedimiento que se realiza para la contabilización de llamadas es el cómputo en base al contador de impulsos del abonado en la central. En general, este contador tiene cinco juegos móviles de cifras de tal forma que puede contar hasta 99.999 unidades de conversación. Con la recepción de cada impulso el contador avanza un paso. Lógicamente no se envían impulsos al contador, en el caso de llamadas libre de pago o llamadas no completadas.

La tasación tiene lugar al emitirse impulsos de cómputo al contador del abonado durante sus conversaciones. En las llamadas se emite un número de impulsos a intervalos que dependen de la distancia a la central destino. Así pues, la tasación esta determinada por el tiempo y la distancias entre las centrales (cómputo por tiempo y zona).

Para el cómputo por tiempo y zona se aplica el "principio de Karlsson", en donde se emplea un dispositivo común de impulsión, desde donde se emiten continuamente impulsos de tasación independientemente de cuando comienzan las diferentes llamadas.

El primer impulso de una llamada puede así emitirse en cualquier momento después de haber comenzado la llamada, emitiéndose luego los impulsos de tasación siguientes a intervalos determinados por la tarifa que rige para la llamada en cuestión (Ver Figura 7). Así la tasación de una llamada aislada no resulta exacta, pero se ha demostrado que después de cierto período de tasación estas inexactitudes se compensan mutuamente, consiguiéndose de este modo suficiente precisión.

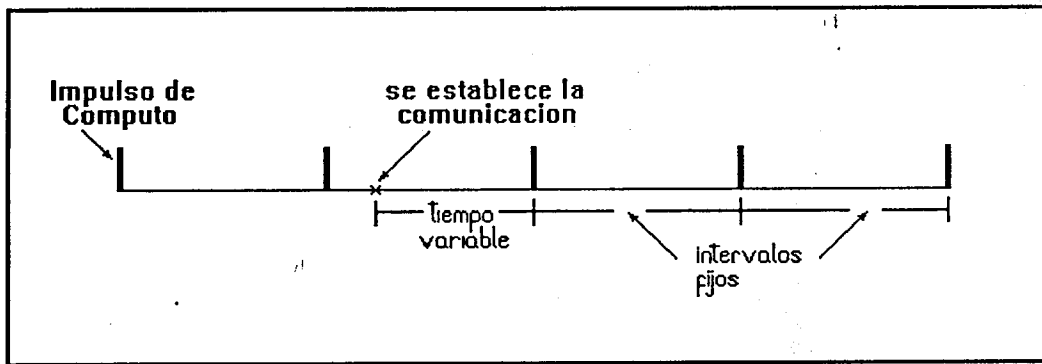


Fig. 7 Aplicación de los impulsos de cómputo a la línea telefónica

F. Tarifas

La tarifa es el pago que un individuo de la sociedad o cliente, debe realizar por la contraprestación del servicio telefónico. Las tarifas en los servicios de telecomunicaciones deben estar diseñadas de forma tal que permitan a la compañía operadora cubrir sus costos de operación, obtener una cierta ganancia y al mismo tiempo favorecer la extensión de los servicios básicos a toda la población.

Tradicionalmente, la fijación de las tarifas de los servicios básicos de telecomunicaciones ha estado sometida a diferentes tipos de regulación, con el propósito de establecer precios en un ambiente de competencia limitada. Actualmente, a la CANTV se le aplica la regulación denominada **Tope de Precios**. Este mecanismo impone al operador un límite a sus ingresos de acuerdo a las variaciones de la economía. El potencial origen de las ganancias está en la mejora de la productividad y eficiencia.

CONATEL, la Comisión Nacional de Telecomunicaciones, adscrito al MTC (Ministerio de Transporte y Comunicaciones), es el organismo encargado de hacer cumplir a la CANTV el mecanismo de regulación de Tope de Precios. Cada tres meses se realiza un ajuste de tarifas según lo establecido en el contrato de concesión y el decreto del Gobierno relativo a los ajustes tarifarios.

Las tarifas que cobra la CANTV se pueden englobar en 3 renglones que son:

- Conexión a la red
- Acceso a la red
- Uso

1. Conexión a la red

El abonado paga por la conexión de su aparato a la red pública. Este pago es único, y se efectúa en el momento de la instalación del teléfono. Cuando las condiciones del terreno son especialmente desfavorables, el abonado debe pagar también parte de los gastos de tendido de la línea hasta sus locales.

El EIC no toma en cuenta esta tarifa para el total de gastos mensuales.

2. Acceso a la red

La CANTV contempla una renta básica mensual dependiendo de la categoría del abonado. Estas categorías son:

- Teléfonos residenciales
- Teléfonos no residenciales
- PABX de bajo tráfico
- PABX de alto tráfico con o sin Discado Directo Entrante

Actualmente la CANTV también cobra una suma de dinero por concepto de digitalización y modernización de la red que depende de las 4 categorías antes mencionadas.

Dado que esta tarifa es fija, el EIC no la toma en cuenta para el cálculo de los gastos por comunicación de la PABX.

3. Uso

El consumo telefónico se mide por el número de impulsos que recibe el suscriptor. Estos impulsos son aplicados al contador particular de cada abonado. Estos contadores son leídos al principio y al final del período al que corresponda la cuenta de las tarifas, para determinar el importe, multiplicando el resultado del contador por el costo del impulso.

La factura para llamadas locales se basa en las indicaciones de este contador, por lo que no se puede dar información particular sobre cada llamada. Sin embargo, la CANTV está realizando esfuerzos por dar a los abonados el servicio de facturación detallada, en la cual se incluye el teléfono destino, la fecha, la hora, la duración, el costo, etc. Actualmente, la CANTV realiza facturación detallada por servicios a nivel nacional, internacional y por llamadas hacia celulares. A continuación se explica detalladamente los distintos tipos de tarifa que cobra la CANTV según el destino de las llamadas telefónicas.

a) Tarifas en zona urbana o local

En una zona local los gastos que tiene la Compañía de Teléfonos por conectar dos abonados, son en término medio aproximadamente constantes. Las instalaciones técnicas necesarias para cualquier comunicación urbana y la distancia entre los abonados correspondientes son aproximadamente las mismas, por consiguiente, solo puede variar la duración de la llamada y por tanto, el tiempo de ocupación de los equipos.

Se aplica un sistema de cómputo por impulsos recibidos con cupos de llamadas, lo cual significa que el abonado puede utilizar el servicio telefónico local sin que se tomen en cuenta el número de llamadas ni la duración de las mismas, hasta que alcance un cupo de impulsos determinados. Al sobrepasar el cupo, el abonado tiene que pagar un precio por cada llamada completada según el número de impulsos recibidos los cuales se aplican según la siguiente cadencia:

1 impulso cada 90 seg

Cabe destacar que para llamadas locales el costo de la llamada no varía dependiendo de la hora en que se efectúe la llamada. Para el período de Sep-Dic de 1993 el costo del impulso era:

1,18 Bs. para abonados residenciales.

1,56 Bs. para abonados no residenciales.

b) Tarifas llamadas nacionales

El volumen del equipo técnico y la parte de la red interurbana utilizada conduce necesariamente a que las tasas por llamadas nacionales sean superiores a las de las llamadas urbanas. Los gastos de una llamada de larga distancia nacional varían según la distancia y la duración de la comunicación.

Las llamadas de Larga Distancia Nacional tienen una clasificación según el horario en que se realicen. Si la llamada se efectúa entre las 7 a.m. y las 7 p.m. de un día laboral, entonces se aplicará la **Tarifa Normal**. Mientras que si las llamadas se realizan dentro del horario nocturno (desde las 7 p.m. hasta las 7 a.m.) o si se realizan en días sábados, domingos y feriados las 24 horas del día, entonces gozarán de la **Tarifa Especial**.

Hay seis grados diferentes de tarifa interurbana, cada grado se divide de acuerdo con la distancia que hay entre la central llamada y la central destino. Estas distancias son clasificadas en 6 renglones, a los que se les denomina **Código de Tarifa** del 1 al 6 a la distancia entre dos ciudades.

La CANTV ofrece dos tipos de servicio a nivel Nacional: Servicio Manual o por Operadores y Servicio Automático o Discado Directo Nacional (D.D.N). Las tarifas por estos servicios se cobran de manera diferente. A continuación se describe la manera en que se cobran cada uno de estos servicios:

• Tarifas por el Servicio Telefónico Nacional Automático o Discado Directo Nacional(D.D.N)

Para determinar cuál es la tarifa a aplicar en una comunicación de Larga Distancia Nacional, se busca primero el código de tarifa en la Tabla 12 (Códigos de área y Códigos de tarifa para las principales ciudades de Venezuela, ubicada en el Apéndice 6. Luego con este código se entra en

la Tabla 11 (Cadencia de los impulsos de cómputo para las llamadas de Larga Distancia Nacional, ubicada en el Apéndice 6, y se determina según la hora y el día cuál es la cadencia que será aplicada.

El costo del impulso es el mismo que el que se cobra para llamadas locales. Esto es:

Uso residencial: 1,18 Bs.

Uso no residencial 1,56 Bs.

•Tarifas por el Servicio Telefónico Nacional Manual o por Operadores (100)

Las tarifas para servicio manual interurbano esta compuesta por dos partes:

- a) tarifa mínima para una conexión de duración menor a 3 minutos
- b) tarifa adicional por cada minuto adicional dependiendo del grado de la comunicación.

Estas tarifas varían según la distancia, la hora y el día. Para determinar el costo por impulso se ubica primeramente el código de tarifa entre las ciudades en cuestión en la Tabla 12 (Códigos de área y Códigos de tarifa para las principales ciudades de Venezuela, ubicada en el Apéndice 6 . Luego con este código de tarifa se entra en la Tabla 13 (Tarifas por Servicio de Operadores, desde poblaciones incorporadas al Discado Directo Nacional (DDN), ubicada en el Apéndice 6, y dependiendo de la hora y la fecha se ubica el costo por impulso.

c) Tarifas llamadas internacionales

•Tarifas por el Servicio Telefónico Internacional Automático o Discado Directo Internacional (DDI):

La Tarifa se cobra en base a la duración efectiva de la comunicación, en el entendido de que las fracciones de minuto se cuentan por minutos enteros. Esto es, al pasar cada minuto se envía un impulso, el cuál tendrá un costo dependiendo del destino de la llamada.

Para determinar el costo por impulso se debe consultar la Tabla 14 (Tarifas para Servicio Internacional Automático y Manual, ubicada en el Apéndice 6, y entonces dependiendo del horario en el que se realice la llamada se ubica el costo por impulso para el destino en cuestión.

Los horarios en los que aplica la Tarifa Reducida y la Tarifa Económica dependen del continente destino de la llamada, en la Tabla 15 (Horarios para Tarifa Reducida y Tarifa

Económica a nivel Internacional, ubicada en el Apéndice 6 se puede observar la diferencia de horarios para cada uno de los tipos de tarifa.

• Tarifas por el Servicio Telefónico Internacional Manual o por Operadores (122)

Las tarifas contempladas en las Tablas 14 y 15 son aplicadas también al servicio cursado a través de operadores con la diferencia de que se cobrará una tarifa mínima de tres minutos para el caso de las llamadas EE y de cuatro minutos para el caso de las llamadas PP.

d) Tarifas por llamadas hacia teléfonos celulares

Las Tarifas por llamadas hacia la telefonía móvil celular se cobran con el mismo costo por impulso que el aplicado para las llamadas urbanas y nacionales. Esto es:

1,18 Bs. para abonados residenciales.

1,56 Bs. para abonados no residenciales.

Pero con la diferencia de que se aplicarán los impulsos con la siguiente cadencia:

- 1 impulso cada 2 segundos en la Tarifa Normal.
- 1 impulso cada 2,5 segundos en la Tarifa Especial.

e) Tarifas por llamadas especiales

En general, las llamadas por servicios especiales no se cobran; policía, bomberos, ambulancias, tránsito, información, solicitud de llamadas nacionales e internacionales, reclamos de averías, etc. Sin embargo, las llamadas de información de la hora oficial si se cobran.

CAPITULO 2 ESPECIFICACIONES DEL EQUIPO Y ALTERNATIVAS DE SOLUCIÓN

A. Especificaciones de diseño iniciales

En el momento de la propuesta del proyecto se delinearón una serie de especificaciones que a continuación se enumeran:

- El dispositivo debe supervisar las líneas telefónicas para detectar los diferentes estados en progreso de una llamada: reposo, ocupado, llamando, contestación, conversación, etc.
- Debe así mismo identificar los números discados a fin de determinar el ritmo con el cual deben ser aplicados los impulsos.
- La frecuencia de los impulsos de cómputo debe ser conmutable entre 50 Hz, 12 Khz y 16 Khz, para ajustarse a los distintos tipos de tarjetas receptoras de impulsos de cómputo fabricadas por SIEMENS.
- Debe ser modular en "software" y "hardware" para permitir una fácil ampliación y debe ser programable para poder ajustarse a posibles variaciones en la tarifa.
- La máxima desviación con respecto a la tarifa de la CANTV debe ser menor al 10%.
- Debe ser capaz de mantener la hora y fecha para poder aplicar las variaciones de la tarifa que la CANTV contempla por días festivos y tarifa nocturna.
- Opcional: Proporcionar una salida de los contadores de impulsos de cómputo con indicación de número, discado, costo de llamada, hora y fecha para aplicaciones en operación autónoma.

B. Diseño del Emulador de Impulsos de Cómputo EIC

El equipo Emulador de Impulsos de Cómputo (E.I.C) es un equipo que supervisa a todas las líneas entrantes de una central privada de mediana capacidad, y que tiene como función

aplicarle impulsos de cómputo a todas aquellas comunicaciones salientes exitosas, de la misma manera como lo haría la CANTV. Ha sido pensado como una caja adicional a la central que iría conectada en paralelo con todas las líneas urbanas de la central. La Figura 7 es un diagrama esquemático de la ubicación del EIC.

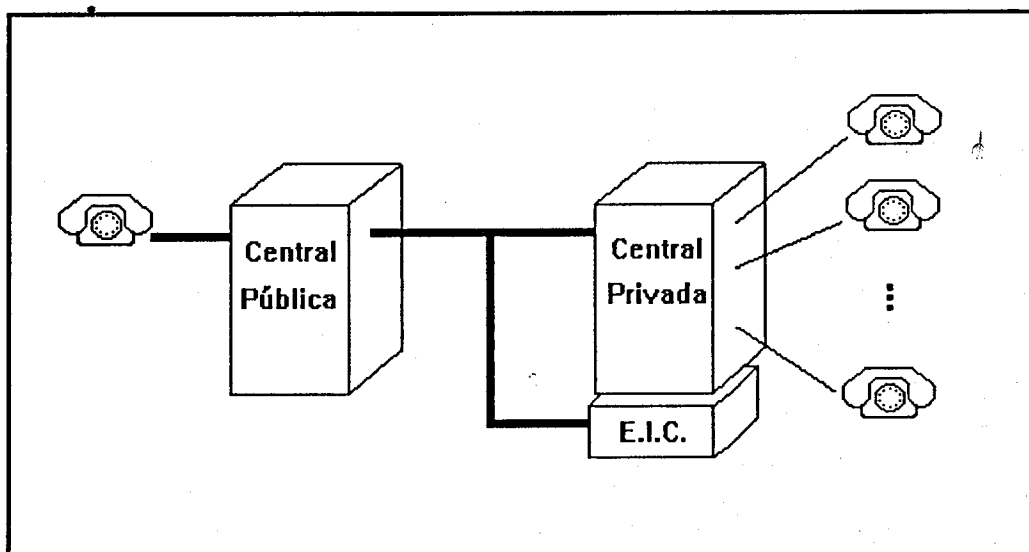


Fig. 8 Ubicación del EIC.

El equipo debe realizar una supervisión continua de las líneas urbanas que llegan a la PABX, con el propósito de detectar la ocupación saliente de alguna línea o la llegada de una llamada a la central para no tarificarla. Por otra parte el EIC debe supervisar los canales de audio ya establecidos para proceder a enviarle los impulsos de cómputo o para dejar de aplicarlos cuando termine la comunicación.

El funcionamiento del equipo se basa en un conjunto de circuitos detectores por cada línea que llevan a niveles lógicos (0-5 V) el estado de la línea. Básicamente lo que hay que detectar son cuatro estados que a continuación se enumeran:

- **Colgado o descolgado de la línea urbana**, para detectar el inicio y final de la comunicación.
- **Repique**, para detectar llamadas entrantes (ya que estas no son tarificadas).
- **Contestación del abonado llamado**, para comenzar a aplicar impulsos de cómputo.
- **Los dígitos marcados**, para planificar la cadencia de los impulsos.

Todos estos estados son agrupados en una palabra de status de 8 bits que indica en todo momento el estado de la línea. Luego una tarjeta controladora dominada por un microprocesador lee estas palabras de status secuencialmente y va tomando decisiones con respecto a cada línea.

Así, al detectar el descolgado de una línea urbana, el microprocesador se prepara a recibir los números marcados por el abonado llamante, y en la medida que los números son detectados, éste va planificando la cadencia de impulsos de cómputo según las tablas de tarificación de CANTV. Cabe mencionar que la frecuencia y la cadencia de estos impulsos es generada con el microprocesador

A posteriori y cuando el abonado llamado conteste, el microprocesador comienza a aplicar los impulsos de cómputo. Y entonces solo le queda esperar por el fin de comunicación (que se produce con el colgado de cualquiera de las dos partes), para finalizar el proceso.

Este proceso de supervisión se debe hacer simultáneamente con todas las líneas urbanas que lleguen a la central, por lo que se debe contar con un controlador suficientemente rápido para atender todos los eventos que sucedan en cada una de las líneas.

C. Alternativas del diseño del "hardware"

La estructura y funcionamiento del Emulador de Impulsos de Cómputo (EIC), descrita brevemente en la sección anterior pasó por muchas alternativas antes de llegar al diseño final propuesto. En esta sección se describen algunas de las alternativas más importantes que se estudiaron y las razones por las cuales éstas fueron desechadas.

1. Generación de la frecuencia de los impulsos de cómputo

Para generar esta onda cuadrada se podrían utilizar por ejemplo 3 osciladores controlados a cristal, cada uno de los cuales oscilará a una de las frecuencias de interés y con un conmutador manual el usuario seleccionará la frecuencia con la que desea que el equipo trabaje.

También se podría usar un oscilador basado en un circuito RC al cual se le pudiera variar 3 valores de resistencia con un interruptor para así obtener las frecuencias deseadas. Pero esta opción es compleja de instrumentar sin el uso de potenciómetros, dada la poca exactitud de valor

al tener 4 momentos en que se planifica la cadencia

→ Colgado de Abonada

de las resistencias. Por lo que la opción es mas costosa y requiere de ajuste minucioso de los osciladores.

La opción de generar la frecuencia con el microprocesador es atractiva por el ahorro de espacio y componentes y por la precisión para generar las frecuencias. Además no es necesario el ajuste de ningún circuito. La elección de la frecuencia de operación se hace por la programación inicial del dispositivo a través del puerto serial.

2. Máximo número de líneas que puede atender el dispositivo

Las centrales privadas varían bastante en cuanto a su capacidad, existen desde centrales con 2 líneas urbanas, pasando por 8, 16, 24, 32, 64, etc. En este sentido el equipo fué pensado para trabajar con centrales de pequeña y mediana capacidad. Esto es, 24 líneas urbanas o menos.

El criterio para la escogencia del número de líneas que puede manejar el EIC fué el máximo número de líneas que puede atender la central de mediana capacidad SIEMENS HICOM 130. Esta central que es el equipo que más vende SIEMENS para oficinas y establecimientos de tamaño medio., tiene una capacidad máxima de 24 líneas urbanas. De esta forma, el equipo trabajaría a plena capacidad en conjunto con la central HICOM 130.

Otras razones, que incidieron en la decisión del máximo número de líneas a manejar, fueron: el tamaño del equipo, que aumenta considerablemente mientras más líneas se atiendan; y la carga de trabajo a la que se somete al microprocesador, ya que al haber más líneas el tiempo que tiene para atender los eventos es menor.

3. "Polling" vs. interrupciones

El esquema que se planteo para resolver el problema supone la lectura cíclica de palabras de status de la líneas ("polling") en las cuales hay información acerca del progreso de la llamada. De esta manera. al leer la palabra el microprocesador hace de su conocimiento 2 informaciones básicas: Qué cambió? y En qué línea cambió?.

El esquema por interrupciones supone que cada uno de los detectores (en total 4 por cada línea: descolgado, repique, contestaciones, dígito válido) tiene la capacidad de interrumpir al microprocesador para informarle del cambio que ocurrió en su línea. De forma tal que estamos

hablando de $24 \times 4 = 96$ posibles fuentes de interrupción, lo cual se podría manejar por ejemplo con controladores de interrupciones (8259) con capacidad cada circuito integrado para atender hasta 8 interrupciones, lo que hace esta opción bastante cara.

Cabe destacar que todas estas interrupciones se deberían canalizar por los 2 pines de interrupción del 8031 (INT0 y INT1) lo cual complica más aún el "hardware" para controlar las interrupciones. Además de esta circuitería de control de interrupciones sería preciso agregar circuitos que permitan determinar en que línea fué que se produjo la interrupción. Todo esto conduce a que esta forma de resolver el problema es inevitablemente más cara.

4. Detector de contestación

Lo que se desea detectar es cuando el abonado llamado contesta. La CANTV detecta la contestación por una inversión de polaridad en la línea que se produce al momento de que la persona levanta el auricular, sin embargo, esta situación solo se produce entre el abonado llamado y la central pública a la que este se encuentra conectado, y se retransmite hasta la central pública de origen, más no en el lazo del abonado llamante. De forma tal, que del lado del abonado llamante no hay señalización que indique contestación, lo único que se puede detectar es el tono de libre de 425 Hz que manda CANTV para informar al abonado A que el abonado B tiene su circuito desocupado, y entonces asumir que la persona contestó cuando exista ausencia de este tono.

Esta forma de detectar la contestación es aproximada ya que el equipo debe esperar que no venga más el tono de libre para asumir que contestaron, lo cual introduce un error en la detección de la contestación.

El tono de libre, es un tono que se encuentra en el rango de la frecuencia vocal. Esta característica trae como consecuencia que la voz de las personas, (que tienen componentes de esta frecuencia) interfieran con la detección de este tono. Por esta razón, se tuvo que

instrumentar un detector de contestación más sofisticado, para detectar solamente el tono y no la voz de las personas (ver Capítulo 4, Descripción del detector de contestación).

5. Detección de la ausencia del tono de libre por "hardware"

La detección de la ausencia del tono de libre se podría hacer con el circuito "Missing Pulse Detector", basado en el temporizador 555. De esta manera, la detección del tono de contestación se haría totalmente por "hardware", sin necesidad de supervisar el tiempo que hay entre impulsos con el microprocesador, pero se tendría que agregar este circuito por cada línea lo que aumentaría el costo y el tamaño del prototipo.

6. Detección de tonos por procesamiento digital de señales

Una forma completamente distinta de atacar el problema de detección de tonos podría ser el uso de técnicas de procesamiento digital de señales (DSP) para el análisis del tono de 425 Hz, el repique de 25 Hz y los tonos multifrecuenciales.

En este método, se realizaría una "multiplexación" en tiempo de las 24 líneas de manera de reconstruir las señales presentes en cada una de ellas y con el uso de los algoritmos de FFT (Transformada Rápida de Fourier) reconocer si se encuentran presentes estas señales. De esta forma, se aumentaría notablemente la confiabilidad del equipo, dado que se reduce al mínimo el ajuste de circuitos de detección. Sin embargo, la disponibilidad en el mercado y el costo de los microprocesadores DSP limita grandemente la posibilidad de resolver el problema con esta tecnología.

7. Emulador de Impulsos de Cómputo en Serie

La idea de colocar el emulador de impulsos de cómputo en serie con la línea, surgió del hecho de que la voz de la persona que origina la llamada está presente en la línea en el momento de la detección del tono de libre ocasionando que el detector de tono de 425 Hz diera salidas erróneas.

En realidad, el problema no se resuelve ubicando el Emulador de Impulsos de Cómputo en serie con las líneas urbanas, ya que la voz del abonado "b" también interferirá con el detector de tono.

8. Detectores compartidos vs. Detectores por línea

Una de las ideas que se trató de implementar durante el desarrollo del proyecto fué que el conjunto de circuitos detectores fuese compartido por varias o todas las líneas que manejara el equipo, de manera de reducir el espacio y los costos.

Un esquema como este no es posible de realizar por el hecho de que los eventos en telefonía deben tener una cierta duración de tiempo para que sean válidos y además porque las señales de línea están presentes antes, durante y al final de la comunicación, lo que obliga a que los detectores estén continuamente supervisando el estado de la línea.

A continuación se discuten las razones que impiden compartir los distintos tipos de detectores:

- Para que un descolgado sea válido debe durar al menos 100 ms (para diferenciarlo del discado decádico y del Flash) lo que implicaría una ocupación del detector por un tiempo igual o mayor a 100 ms. Este hecho, frustra cualquier intento por compartir los detectores de colgado y descolgado, ya que mientras se esta atendiendo una línea se puede perder información del estado de otra línea.

Si se intentara hacerlo en un tiempo menor se podría confundir colgado y descolgado con la marcación de un dígito en discado decádico, ya que eléctricamente ambas señalizaciones son aperturas y cierre de lazo, con la diferencia de que la marcación de un dígito dura menos tiempo.

- El detector de contestación diseñado basado en la detección del tono de libre no se puede compartir. Esto se debe a que su funcionamiento depende de la presencia de la señal de 425 Hz por más de cierto tiempo, lo cual obliga a que este detector deba estar conectado permanentemente a la línea monitoreando la presencia de este tono.

- El detector de repique funciona por amplitud de voltaje, es decir cuando el voltaje A.C. presente en la línea, supera un nivel de 18 Voltios el circuito integrado detecta la señal de

repique. Dado que su funcionamiento solo depende de la amplitud de voltaje, se podría pensar en compartirlo por varias líneas, conmutándolo cíclicamente por los 24 pares.

En este caso la "traba", es precisamente este conmutador de 2 pares a 24 pares con capacidad para soportar voltajes de hasta 180 Vpp. Este es un dispositivo que no es comercial, sin embargo se podría construir en forma discreta. Por ejemplo, con una configuración de relés se podría hacer una red de conmutación como esta. Sin embargo, sería una configuración poco funcional dada la cantidad de componentes que harían falta para construirla. Por esta razón es mejor la opción de un detector de repique por cada línea.

- Quizás los únicos detectores que se podrían compartir serían los detectores de discado, ya sean los de discado decádico o los de discado multifrecuencial.

Tomando en cuenta que las comunicaciones que tiene una central no son todas salientes y que las líneas no están ocupadas el 100% del tiempo, se podría tener un número de detectores inferior al número de líneas y tener de esta manera circuitería común.

Para lograr compartir los detectores hace falta tener una red de conmutación en la cual sea posible conectar cualquiera de los detectores con cualquiera de las líneas a supervisar. De forma tal, de ir asignando los detectores a las líneas que los vayan necesitando, y liberarlos cuando ya el abonado llamante haya marcado los números por completo.

Asignar un detector a la línea que se ocupe, es un problema que se podría resolver con el uso de matrices de conmutación de estado sólido de 4 x 4 (XR2201), pero al hacer esta matriz se utilizarían más circuito integrados que los que se utilizarían usando un detector por cada línea, sin contar el esfuerzo adicional de programación y lógica que implica el proceso de asignación y liberación de los detectores.

Otra consideración importante que se debería tomar en cuenta para resolver el problema de detección con circuitería común, es el número de detectores de cada tipo que harían falta para atender a las 24 líneas.

Un criterio que se podría usar para fijar el número de detectores de discado multifrecuencial que harían falta en el banco de detectores se podría basar en el hecho de que

HICOM 130 (Central con la que trabajaría el EIC mayormente) solo tiene cuatro discadores multifrecuenciales hacia las líneas urbanas. Entonces, el banco de detectores del EIC sería funcional con 4 detectores de discado multifrecuencial.

Sin embargo las 24 líneas podrían estar, en un momento determinado, marcando todas en forma decádica, por lo que en teoría haría falta tener un detector de discado decádico por cada línea. Este caso es definitivamente poco probable, por lo que se podría pensar en tener menos detectores. Se debería estudiar entonces el tráfico de la central y proponer un número razonable de circuitos detectores.

Basados en estas aseveraciones, se podría diseñar un banco de detectores con 4 detectores multifrecuenciales y 12 detectores decádicos y satisfacer todos los casos de ocupación más comunes. Sin embargo, hay que tomar en cuenta que la probabilidad de que se ocupen más de 16 líneas en forma saliente no es nula. Por ejemplo, en el caso que se ocupasen 18 líneas, ocurriría que a dos de ellas no se le podría detectar los números marcados, causando que no se pueda tarifificar a estas líneas.

La discusión anterior justifica la alternativa de tener un detector de discado por cada línea ya que es más económico, práctico y seguro.

Sin embargo, el problema de los tipos de discado no está aún resuelto. En este sentido, la solución adoptada fue la de fabricar tarjetas que sólo sean para detectar discado multifrecuencial y tarjetas que sólo detecten discado decádico. Cada una de estas tarjetas puede atender a 8 líneas.

Por otro lado, el EIC permite que el cliente tenga líneas de distinto tipo, ya que para el equipo diseñado es transparente el tipo de las líneas una vez configurado con las tarjetas adecuadas. De esta manera, cada vez que se vaya a instalar un EIC se deberá hacer un estudio del tipo de líneas de los clientes.

9. Elección del microprocesador

La escogencia del microprocesador es un dilema que debe resolver la economía. Si su escogencia economiza tiempo, dinero, espacio y además cumple técnicamente con lo necesario entonces esa será la alternativa adecuada.

El 8031 de la casa INTEL fué el microprocesador elegido para controlar el equipo ya que sus características lo hacen uno de los más adecuados para los requerimientos técnicos de esta aplicación y es un microprocesador de bajo costo, fácil de conseguir en el mercado y con disponibilidad de la información técnica. Además es ideal porque SIEMENS fabrica este microprocesador, por lo que habrá más disponibilidad de conseguirlo en el caso de la producción en serie del equipo.

Entre las características técnicas que lo hacen ventajoso se pueden nombrar las siguientes:

- Puerto serial integrado, lo que lo hace ideal para programar las tarifas de CANTV. Esta característica lo hace superior a una gran cantidad de microprocesadores que necesitan de un circuito integrado adicional (el 8251) para la comunicación serial.

- 2 temporizadores internos, necesarios, uno para generar la frecuencia de los impulsos de cómputo y otro para controlar el ritmo con que se aplican los impulsos. Esta característica de integración lo hace superior al Z80 por ejemplo, que necesita un circuito integrado adicional (el CTC) que es el que contiene los temporizadores.

- 2 interrupciones externas, lo que hace más sencillo la interconexión de "Real Time Clock" (MM58167A), ya que éste interrumpe cada minuto y así el microprocesador puede actualizar la hora.

- La frecuencia del cristal puede ser de hasta 12 MHz, lo que conduce a que el tiempo del ciclo de instrucción sea de $1\mu s$. Este tiempo es suficientemente rápido para poder atender los eventos que debe controlar el equipo y poderles dar una respuesta efectiva.

- Tiene capacidad para direccionar hasta 64K de memoria externa de programa y 64K de memoria externa de datos, ya que tiene 16 hilos de direcciones. Este espacio de memoria

direccionable es más que suficiente para programar las tablas de tarifa y además se tiene espacio suficiente para el programa que controla el equipo.

- Tiene una memoria RAM interna de 128 bytes con la que se pueden manejar variables del programa de una manera más eficiente y rápida, ya que el conjunto de instrucciones así lo permite.

- El conjunto de instrucciones tiene 2 características que lo hacen adecuado: la primera es que tiene un grupo de instrucciones con operaciones aritméticas poderosas como lo son la multiplicación, la división, la resta y la comparación, que optimizan algunas rutinas aritméticas; la segunda característica importante es el tener instrucciones con manejo de bits (procesador "booleano"), lo que ayuda en las rutinas del análisis de status de las líneas.

- El ser un microprocesador de 8 bits, lo hace adecuado para una aplicación como la presente. Un microprocesador de 16 bits en el bus de datos o con bus de direcciones de 32 bits es quizás más de lo que hace falta. Son ejemplos de esto el 80386 o el 68000.

Entre los microcontroladores de la familia '51 hay pequeñas diferencias que podrían llevar a escoger uno u otro entre ellos. El 8031 a pesar de ser el más sencillo de la familia, se adapta perfectamente al desarrollo de este proyecto y además es el más económico de todos ellos.

Las versiones de la familia '51 con memoria de programas interna, ya sea con EPROM o PROM, salen más costosas que usar el 8031 con una memoria de programas externa y además no aportan mucho en esta aplicación ya que el programa diseñado ocupa más de 4K de memoria y además limita futuras expansiones o mejoras. Dado que el espacio de una memoria externa no es tan grande, no se justifica pagar más por su integración, a menos que sea imprescindible, que no es el caso de este proyecto.

Las versiones que integran memoria de programas son el 8051 que es idéntico al 8031, pero con la integración de un PROM de 4K. Otra versión es el 8751 que integra en el circuito integrado una EPROM de 4K. Por otro lado, están el 8052 y el 8752 que integran 8K de PROM y 8K de EPROM respectivamente.

Otros circuitos integrados de la familia son los 80x2 que tienen 3 "temporizadores" internos en lugar de 2 y 256 bytes de memoria RAM interna en vez de 128 bytes. Estas 2 características tampoco aportan mucho en la presente aplicación. En la Tabla 2 se muestra un resumen de las diferencias entre los procesadores de la familia 51.

Microcontrolador	RAM Bytes	ROM Bytes	EPROM Bytes	"temporizadores"
8031	128	--	--	2
8051	128	4K	--	2
8751	128	--	4K	2
8032	256	--	--	3
8052	256	8K	--	3
8752	256	--	8K	3

Tabla 2 Comparación de los microcontroladores INTEL de la Familia 51.

Otros microprocesadores que pudiesen ser utilizados para controlar el equipo son: alguno de la familia 6801, 6805 o 6811 fabricados por Motorola, o la familia COP800 fabricados por National o también podría ser el Z80 con sus periféricos CTC, PIO, SIO.

10. Programación del EIC

El equipo necesita varios parámetros de programación inicial para poder funcionar correctamente. A continuación se enumeran los parámetros que se deben programar:

- **Hora y fecha del día**, para poder aplicar las tarifas normales o reducidas que aplica la CANTV.
- **Tablas de tarificación de la CANTV**, para poder aplicar la cadencia de los impulsos de cómputo adecuadamente.
- **Frecuencia de los impulsos de cómputo**, con el propósito de hacer al EIC compatible con las distintas tarjetas receptoras de impulsos fabricadas por SIEMENS.
- **Número de líneas que puede manejar el EIC**, para permitir configurar el equipo a distintas capacidades.

Toda esta información podría estar en EPROM y cada vez que se deseara cambiar algunos de los parámetros, se debería sustituir la EPROM por una que tuviese las

especificaciones deseadas. Pero ésta sería una opción poco práctica, por diversas razones como lo son: el continuo cambio de tarifas, la poca versatilidad para cambiar la frecuencia de los impulsos, etc.

Otra opción sería EEPROM, con lo cual sería siempre el mismo circuito integrado con el que se realizarían los cambios de programación. Pero las EEPROM son de tres a cuatro veces más costosas que una EPROM o una RAM, y además no se consiguen con facilidad en el mercado por lo que no es una buena opción.

Otra alternativa (la realizada en este equipo), es programar el dispositivo por un puerto serial con la ayuda de un computador adicional y un programa de interfaz con el usuario que le pida los datos de configuración. La programación es guardada en la RAM, y es alimentada con una pila de NiCd de 3.6 Volts, con el propósito de no perder los datos programados en fallos del suministro eléctrico. Esta opción es a mi juicio la más práctica, por la facilidad y rapidez con que se pueden programar los datos

11. "Drivers" de comunicación serial

Se usó una configuración compuesta por: un diodo zener, una resistencia y un negador para la conversión de niveles RS-232 a 0-5 Voltios en la recepción; y un comparador alimentado entre ± 5 Voltios para la transmisión (Ver Capítulo 4)

Esta configuración para conversión de niveles es preferible que el uso de los drivers especiales para comunicación serial (MC1489, MC1488), por consideraciones de espacio y economía.

12. Filtro para el detector del tono de libre

El diseño de filtros se puede atacar por dos vías fundamentalmente: filtros activos o filtros pasivos. En ambos casos existe el problema de la criticidad de los componentes para lograr obtener la respuesta deseada.

En esta aplicación, se desea un filtro para aislar el tono de libre de 425 Hz. Dado que el acople con la línea atenúa bastante las señales, es necesario amplificar para poder analizar a las

señales adecuadamente. Por esta razón, un filtro activo es mejor opción que un filtro pasivo, ya que con los últimos no se puede obtener ganancia.

Es conveniente que el filtro sea lo más preciso y selectivo posible, ya que se desea aislar el tono de 425 Hz y rechazar las demás componentes de voz y ruido presentes en la línea. Con un filtro activo de segundo orden convencional, se podría lograr el efecto deseado, pero también se tendría el problema del ajuste del filtro debido a la poca exactitud de los componentes.

Con el filtro activo integrado MF10, la frecuencia central de la banda pasante del filtro no es función de los componentes externos, sino que es directamente proporcional a una onda cuadrada aplicada a uno de los pines del circuito integrado. El MF10 sólo requiere conexión de resistencias externas para el ajuste de la ganancia. Además elimina el uso de los condensadores, ya que los tiene integrados. Con el MF10, se pueden realizar 2 filtros de segundo orden pasabandas, con sólo 6 resistencias adicionales. Por todas estas, razones su elección resulta conveniente para esta aplicación.

CAPITULO 3 DESCRIPCIÓN DEL "HARDWARE"

La circuitería del Emulador de Impulsos de Cómputo puede ser descrita modularmente por los 5 bloques principales que a continuación se mencionan:

- Una tarjeta controladora.
- 3 tarjetas detectoras.
- Una interfase serial
- Una interfase visual
- La fuente de alimentación.

La interrelación de los bloques anteriores se puede observar en la Figura 9

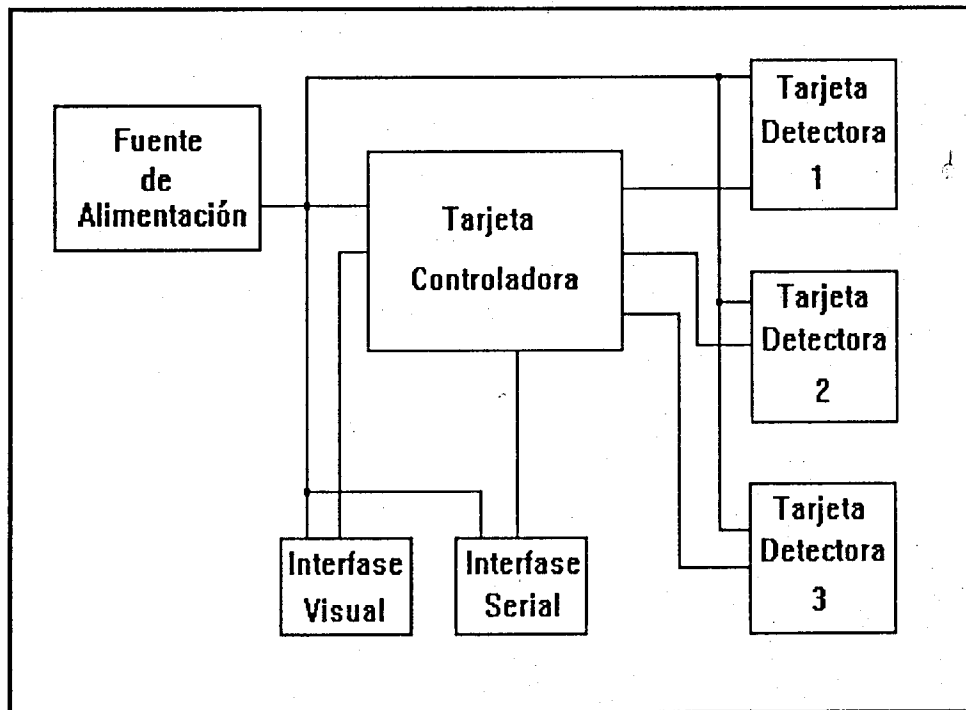


Fig. 9 Estructura modular del "hardware"

En las siguientes secciones del trabajo se explica el diagrama en bloques de la Figura 9.

Para obtener detalles de la conexión de los circuitos se debe revisar el Apéndice 1.

*Hay que decir que en el prototipo se hizo solo para
H Linea*

A. Tarjeta Controladora

1. El microprocesador

El 8031 no tiene pines dedicados exclusivamente al bus de direcciones ni tampoco dedicados solo al bus de datos. Esto es porque los circuitos integrados de la familia 51 pueden ser utilizados como microcontroladores industriales con 4 puertos de entrada o salida en donde el programa a ejecutarse está en la memoria PROM o EPROM interna (Ver Figura 10). Sin embargo, se puede utilizar como un microprocesador con un bus de direcciones y datos.

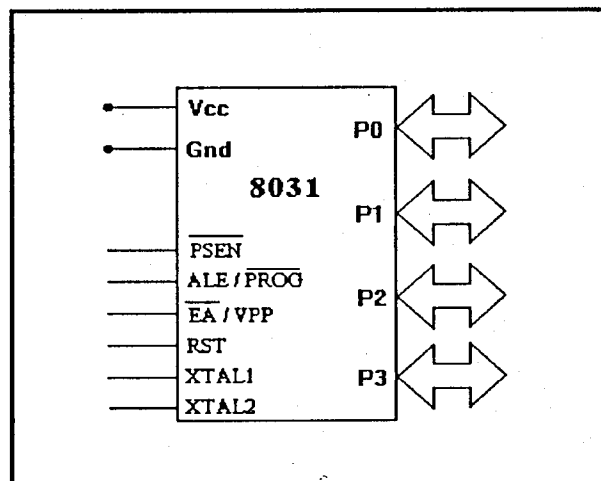


Fig. 10 El microprocesador 8031

En esta aplicación el 8031 fué utilizado como un microprocesador en donde el puerto 0 es el bus de datos (8 bits) y los puertos 0 y 2 conforman el bus de direcciones (16 bits) (véase la Figura 11).

La "multiplexación" de las 8 líneas del puerto 0, con la parte baja del bus de direcciones y el bus de datos, durante el acceso a la memoria externa de programas y datos, es una característica fundamental de la familia INTEL '51, que los diferencia de los demás microprocesadores.

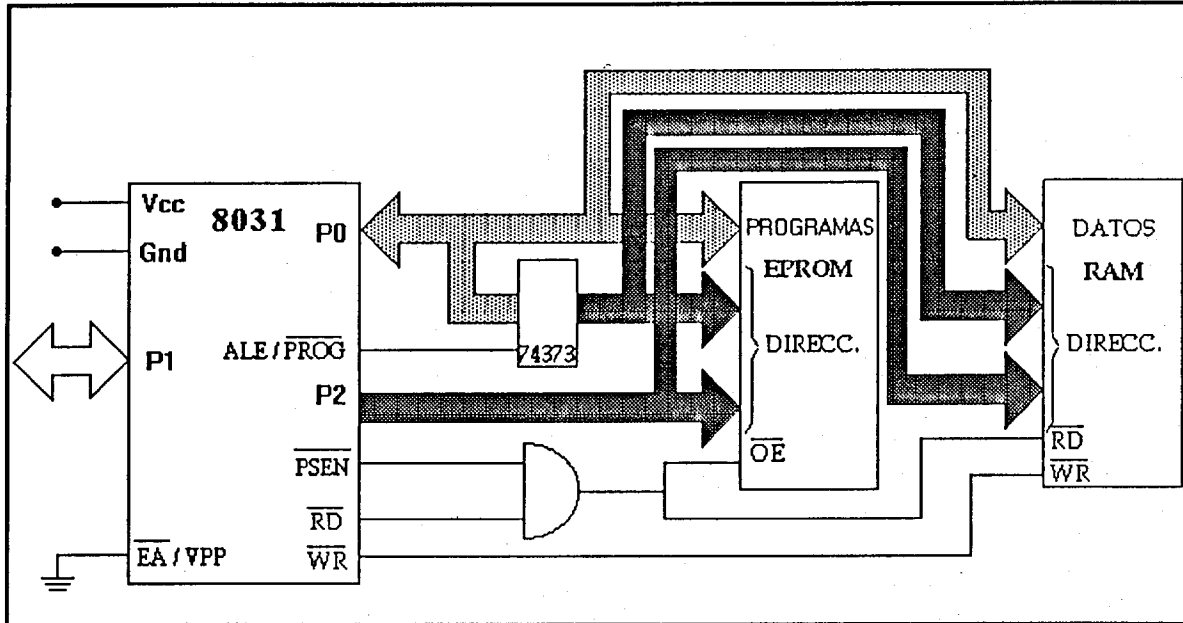


Fig. 11 El 8031 como microprocesador

- Pines especiales
- ALE

Con el multiplexaje del bus de datos y direcciones en el puerto 0, hay un ahorro de pines considerable, pero se hace necesaria la inclusión del pin ALE (Adress Latch Enable) y de un Latch (74373) para poder diferenciar los datos de las direcciones. Por el pin ALE el microprocesador emite un pulso para enclavar el byte bajo del bus de direcciones en el acceso a la memoria externa. ALE se emite con una frecuencia de 1/6 de la frecuencia del reloj, entonces dado que la frecuencia del cristal es 12 MHz, ALE tiene una frecuencia de 1 MHz. (ver Figura 12)

- "RESET"

La señal de inicialización del sistema es el "RESET" (pin 9). Un "RESET" interno del sistema se produce cuando se pone el pin RST a nivel lógico alto al menos durante 2 ciclos de máquina (24 períodos de reloj), mientras el oscilador está funcionando. Para lograr este tiempo se descarga un condensador de 10 μ f a través de una resistencia de 10 K, tal como se muestra en la Figura 12).

- EA

Al poner el pin EA (External Access) a tierra, la búsqueda de direcciones del programa se dirige a la memoria externa en todo momento. Este pin es puesto a 1, en los controladores de la familia que tienen memoria de programas interna, para que el PC busque el programa en la ROM o EPROM interna. (ver Figura 12)

- XTAL1 y XTAL2

A estos dos pines se conecta un cristal, que puede tener un valor de hasta 12 Mhz. Es necesario conectar un par de condensadores del orden de 20 pF. (ver Figura 12)

- PSEN (Program Store Enable)

Es la señal de validación para leer la memoria EPROM externa. Cada vez que está señal baja a cero voltios, el microprocesador lee el "OPCODE" de una instrucción, para luego ejecutarlo. (ver Figura 12)

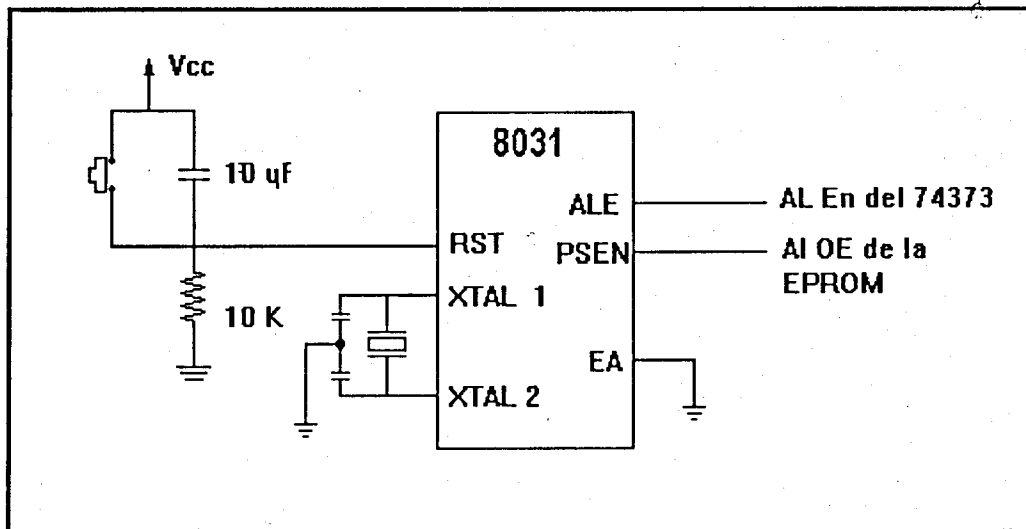


Fig. 12 Conexión de los pines especiales del 8031.

• Puerto 1

El pin P1.3 se usa para diferenciar si el equipo funciona en estado normal (1/lógico) o en estado de programación (0 lógico). Cuando se desee programar el equipo (frecuencia de los impulsos de cómputo, hora, número de líneas o tarifas) se deberá accionar este interruptor, de manera que el microprocesador cuando lea este pin entre en estado de espera de datos de programación por el puerto serie. Esta conexión se ilustra en la Figura 13.

Por el pin P1.4 se genera la frecuencia de los impulsos de cómputo (50 Hz, 12 KHz o 16 KHz) dependiendo de la programación. Esta frecuencia se genera por las interrupciones del temporizador 0, (ver su explicación en el capítulo 4 de descripción del "software"). En la Figura 13 se ilustra el funcionamiento del pin 4 del puerto 1 como salida.

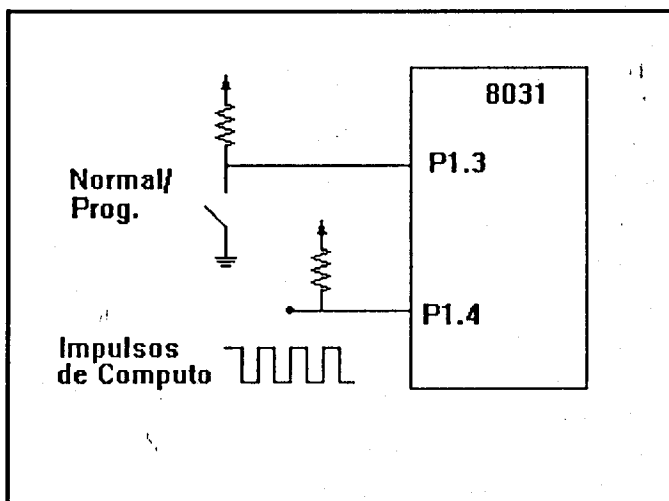


Fig. 13 Uso de los pines del puerto 1 del 8031.

• Puerto 3

Los pines del puerto 3 tienen funciones especiales tal como se especifica en la Tabla 3, sin embargo se pueden utilizar como pines de entrada/salida.

Los dos primeros pines de P3 constituyen el puerto serie RXD (P3.0) y TXD (P3.1), que le dan la capacidad al 8031 de tener comunicaciones seriales asíncronas "full duplex" (que puede recibir y transmitir simultáneamente).

Pin	Nombre	Funcion
P3.0	RXD	Entrada del puerto serie
P3.1	TXD	Salida del puerto serie
P3.2	INT0	Interrupción externa 0
P3.3	INT1	Interrupción externa 1
P3.4	T0	Entrada externa temporizador 1
P3.5	T1	Entrada externa temporizador 0
P3.6	WR	Escritura en memoria externa de datos
P3.7	RD	Lectura en memoria externa de datos

Tabla 3 Funciones especiales del puerto 3.

Por el puerto 3 también salen las señales de RD y WR necesarias para la lectura y escritura de los dispositivos externos.

El pin P3.2 (INT0) es la entrada de la interrupción externa. A este pin se conecta la interrupción del RTC ("Real Time Clock") MM58167A el cual genera un pulso positivo cada minuto. Con el flanco descendente de este pulso se cae en la rutina de atención a la interrupción externa 0, la cual actualiza la hora y la fecha y la muestra en la pantalla LCD.

• Conexión de Memoria Combinada

La memoria combinada significa que el espacio de dirección para las memorias RAM y EPROM es el mismo. La memoria combinada se logra al conectar las señales RD y PSEN a una compuerta lógica AND y conectando la salida de la compuerta al "Output Enable" de la RAM (OERAM) y al "Output Enable" de la EPROM (OEPROM). (Ver Figura 14)

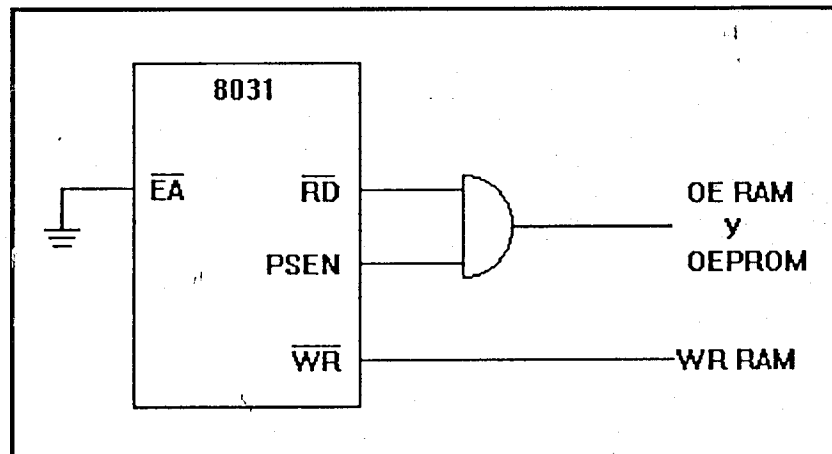


Fig. 14 Conexión para lograr memoria combinada

En la fase de experimentación se tuvo en la EPROM un programa emulador de la memoria. Este programa se queda esperando por el puerto serial a que le manden un programa de prueba. Una vez recibido el programa por completo lo transfiere a la RAM, hace un salto a la dirección de origen del programa y comienza a ejecutarlo. De esta manera, se elimina la necesidad de grabar una EPROM cada vez que se haga un cambio en el programa. El listado del programa emulador de memoria se encuentra en el Apéndice. 4

2. Memorias

El prototipo tiene 2 memorias externas, una EPROM de 32 K de capacidad (27256) en donde se encuentra grabado el programa que controla el equipo y una memoria RAM de 32 K de capacidad (62256) donde se encuentran las tablas de tarifas de la CANTV y donde se tienen también las variables del programa.

3. Mapa de memoria

El espacio direccionable por el microprocesador es de 64 K, desde la dirección 0000H hasta la FFFFH, tal como se aprecia en la Figura 15, los cuales están divididos en tres partes: Memoria de programas (32 K), memoria de datos (24 K) y periféricos (8 K).

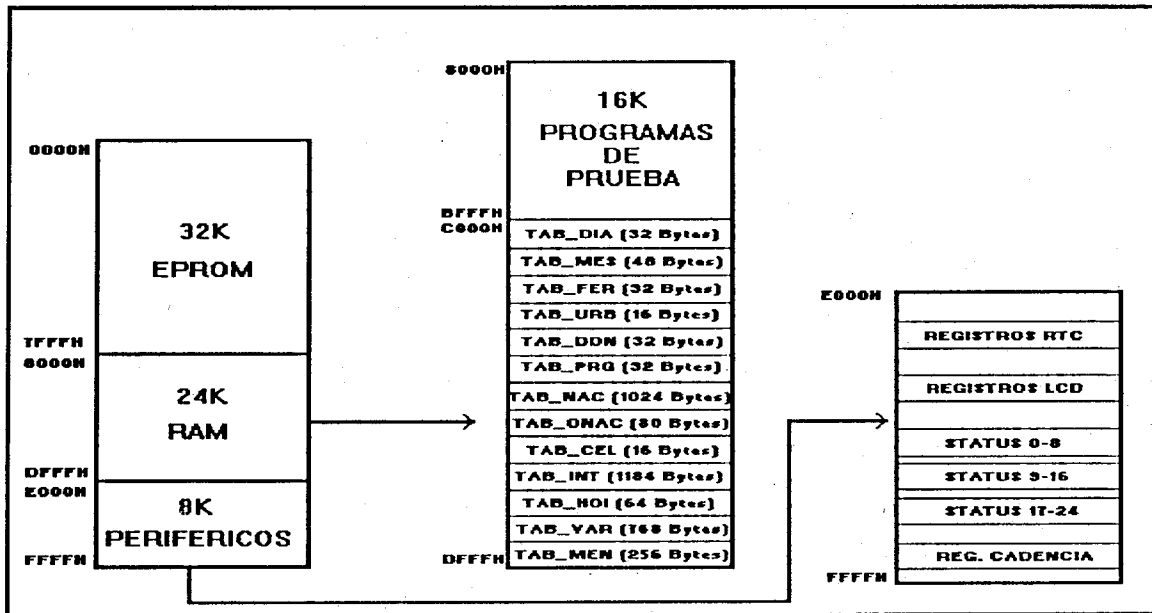


Fig. 15 Mapa de memoria.

El mapa de memoria se realiza con el circuito que ilustra la Figura 16. El circuito decodificador es el que realiza la ubicación de los componentes para que el microprocesador pueda leer y escribir sobre ellos.

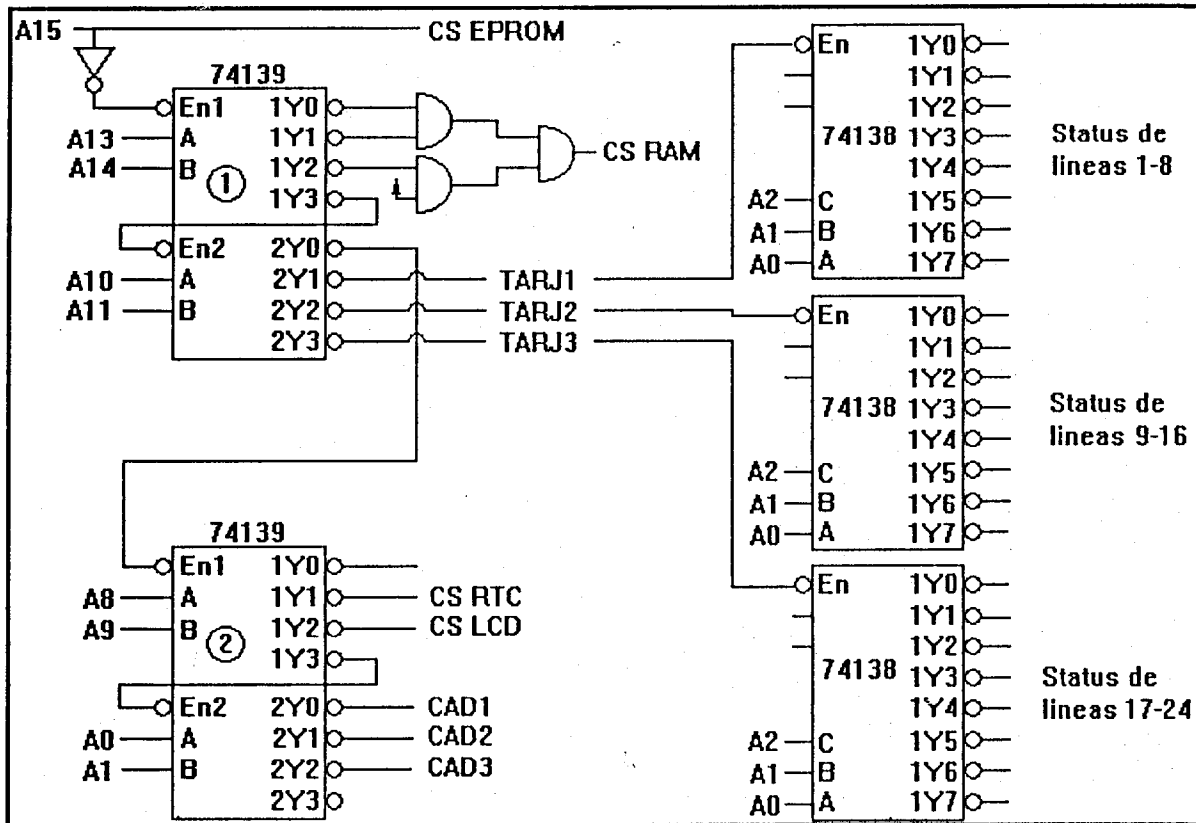


Fig. 16 Decodificación de memoria

Este circuito decodificador realiza la primera división del espacio de memoria, con la dirección 15 (A15). Cuando este pin es cero (direcciones desde 0000H hasta 7FFFH) se selecciona EPROM y cuando es 1 (desde 8000H hasta FFFFH) se ubica a la RAM o a los periféricos externos.

La memoria RAM ocupa 24 K que van desde la dirección 8000H hasta la dirección DFFFH. Estas direcciones se ubican con el AND de los 3 primeros pines del primer decodificador 74139.

Los 8 K restantes (E000H-FFFFH) se utilizan para ubicar en memoria a los periféricos externos que se enumeran a continuación:

- Status de 24 líneas urbanas
- circuito integrado reloj-calendario
- pantalla LCD
- Registros que llevan la cadencia

Se puede observar que la memoria RAM no es completamente utilizada ya que parte de su espacio de direcciones es utilizado para ubicar a los periféricos. Por otro lado se puede observar que 8K de la memoria RAM se utilizan en las tablas; la mayoría de ellas, a los fines de la tarificación de las llamadas telefónicas. Para más información acerca de las direcciones exactas de los periféricos y tablas consultar el módulo "Definición" del programa en Assembler (Apéndice 2). La función de cada tabla es explicada en el capítulo 4, en la sección relativa al programa en assembler.

4. Reloj-calendario

El circuito integrado de reloj y calendario (RTC MM58167A) se conecta al microprocesador tal como indica la Figura 17. Este circuito integrado tiene 24 direcciones de RAM interna que pueden ser accedidas tanto para escritura como para lectura y en las cuales se programa el modo de operación y se tienen los valores de la hora y fecha actual.

La función de este circuito integrado es de interrumpir cada minuto al microprocesador para que éste actualice los valores de hora y fecha. El pin de interrupción del RTC está conectado a la interrupción externa 0 (INT0) del 8031, la cual está programada para atender a la rutina de interrupción cuando ocurra un flanco descendente en el pin INTO.

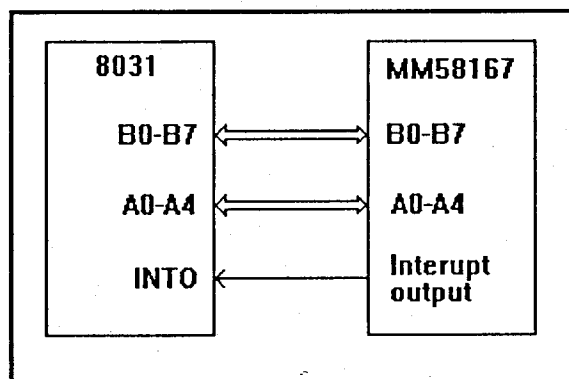


Fig. 17 Conexión del circuito integrado "Reloj-Calendario".

La base de tiempo del Reloj-Calendarario la fija un oscilador externo basado en un cristal de 32.768 Khz, con lo que logra realizar una temporización de 24 horas de reloj y 4 años de calendario (ya que no se contemplan los años bisiestos).

Puede ser operado con consumo mínimo si el pin POWER DOWN (23) se coloca en cero Voltios, deshabilitando todas las entradas y salidas excepto una de las patas de interrupción.

5. Oscilador de 42.5 Khz

Este oscilador es usado por los circuitos integrados MF10 para fijar la frecuencia central (f_0) del filtro pasabanda. Este circuito integrado necesita en uno de sus pines una onda cuadrada TTL de una frecuencia cien veces mayor. Se usa un oscilador basado en un negador con "Schimtd Trigger" (74LS14) (Ver Figura 18.) La frecuencia se fija con la siguiente fórmula:

$$f = \frac{1}{0,69.R.C};$$

Como deseamos $f = 42.5$ Khz y fijando $C = 1$ nf, se obtiene

$$R = \frac{1}{(0,69).(42,5 \times 10^3).(10^{-9})}$$

$$R = 34,1k \Rightarrow R = 33K + \text{Pot } 5K \text{ de precisión}$$

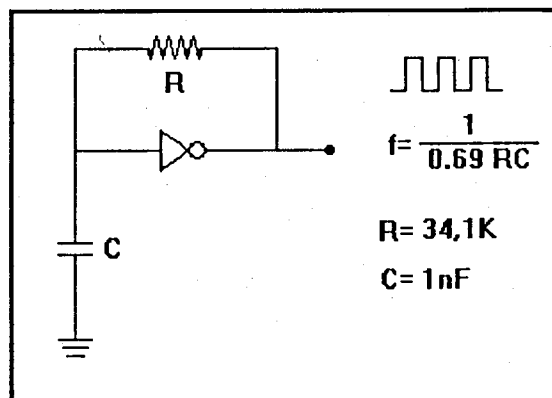


Fig. 18 Oscilador de onda cuadrada de frecuencia 42,5 Khz.

B. Tarjetas Detectoras

Cada tarjeta detectora de estado fué diseñada para atender a ocho líneas urbanas. Por cada línea urbana se necesitan circuitos de acople, un conjunto de circuitos detectores que suministren señales lógicas (0-5 Volt) dependiendo del estado de la línea y un registro que almacene la palabra de status, el cual constituye la interfaz digital con la tarjeta controladora (ver Figura 19).

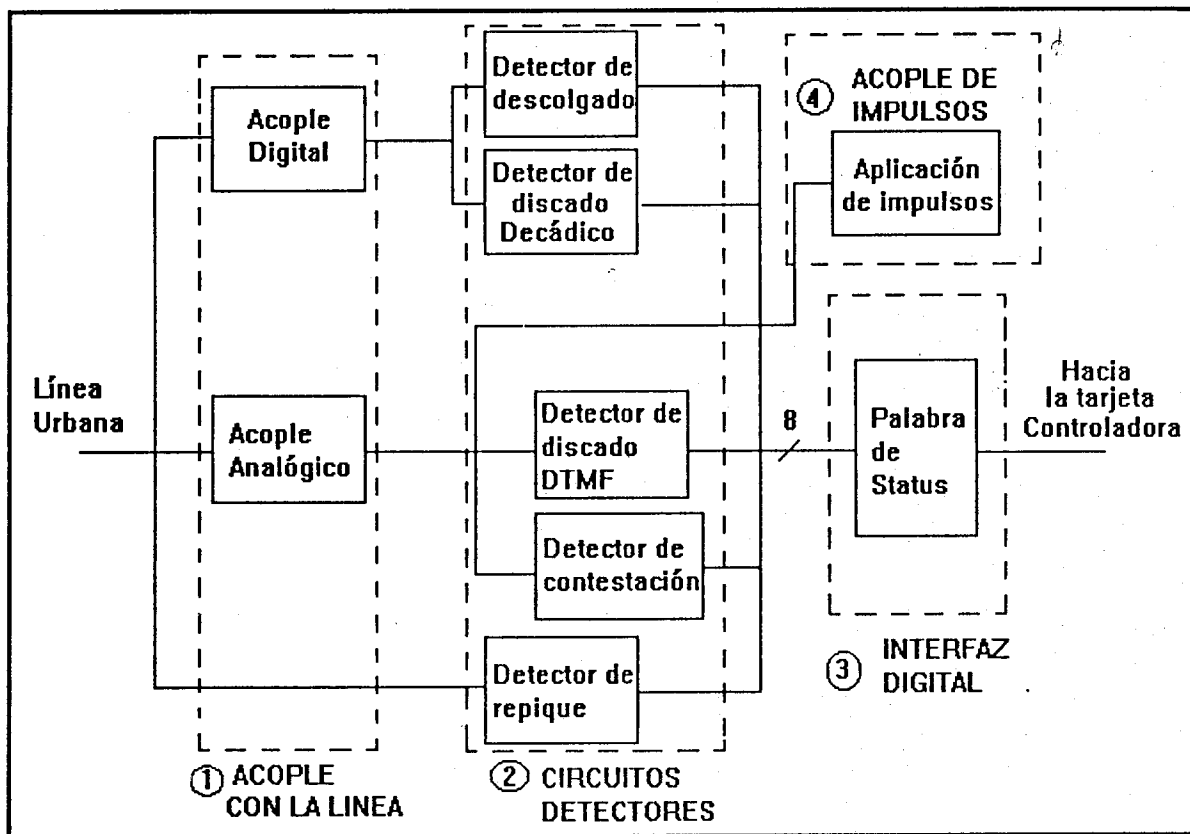


Fig. 19 Diagrama en bloques de una tarjeta detectora

Las próximas subsecciones del trabajo describen los bloques delimitados entre líneas punteadas en la Figura 19.

1. Acople con la línea

Se diseñaron dos acoples con las líneas urbanas uno para señales analógicas y otro para señales digitales.

• Acople digital

El acople digital con la línea urbana, se logra con un optoacoplador, el cual con una resistencia en serie alta (75K), logra el objetivo de tomar muestras de la línea sin cargarla (Ver Figura 20). Esta resistencia en serie debe ser alta de manera que la CANTV no detecte un descolgado. Si esta resistencia fuese menor de 2K aproximadamente, la corriente continua que circularía por el lazo del abonado sería suficiente para activar los detectores de descolgado de las centrales públicas.

Cuando la línea esta colgada la salida del optoacoplador es de 5v, mientras que cuando la línea está descolgada la salida del optoacoplador es de 0 voltios.

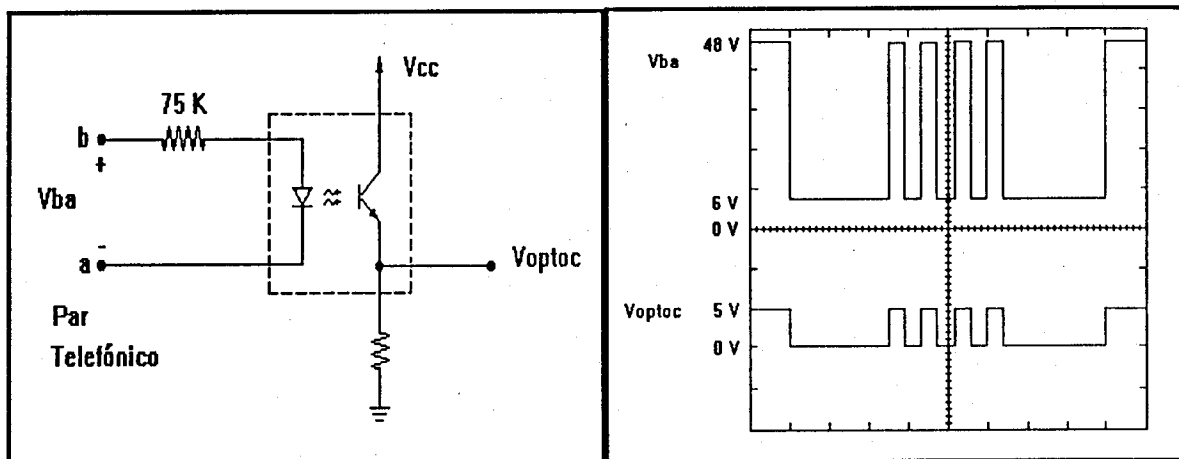


Fig. 20 Acople digital a línea telefónica a través de un optoacoplador

• Acople analógico

El acople analógico con las líneas, está formado por un condensador, una resistencia y un transformador (Ver Figura 21). El condensador impide el paso de corriente DC al transformador, con el objeto de que no se sature. La resistencia de 22K es de un valor alto para no descolgar la línea, pero no tan alto que atenúe en exceso las señales analógicas entrantes.

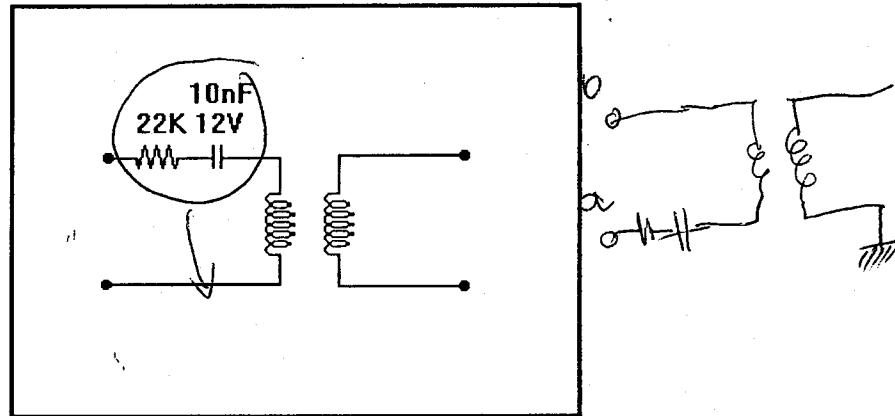


Fig. 21 Acople analógico con la línea telefónica a través de un transformador.

2. Circuitos detectores

Los circuitos detectores son los encargados de llevar los distintos estados de la línea, a señales que indiquen presencia o ausencia del estado. Un conjunto de circuitos detectores está compuesto por 4 partes:

- Detector de colgado y descolgado.
- Detector de repique.
- Detector de dígitos.
- Detector de contestación.

Las próximas secciones describen por separado cada uno de estos detectores. Para una visión global de los circuitos detectores se debe revisar el Apéndice 1.

• Detector de colgado y descolgado

a.-Para líneas decádicas:

Se utiliza el pin OH ("Off-Hook") del M-959. Este pin sube a 5 voltios cuando se descuelga la línea. Es inmune a cualquier ruido presente en la línea telefónica y no confunde al status "descolgado" con el discado decádico.

OH se conecta al bit 0 del registro de status (74LS244) correspondiente a la línea.

b.-Para líneas multifrecuenciales:

Se usa un optoacoplador el cual detecta las variaciones de corriente de la línea.

• Detector de repique

El circuito integrado PSB-6620 proporciona una salida digital de 5 voltios, cuando en la línea telefónica existe la señal de repique (25 Hz y 180 Vpp), lo cual se ilustra en la Figura 22. Este circuito integrado se acopla a la línea sin cargarla.

A la salida del detector de repique se conecta un optoacoplador para aislar la línea telefónica de la tierra del circuito. La salida del optoacoplador se conecta al bit 1 del registro de status (74LS244)

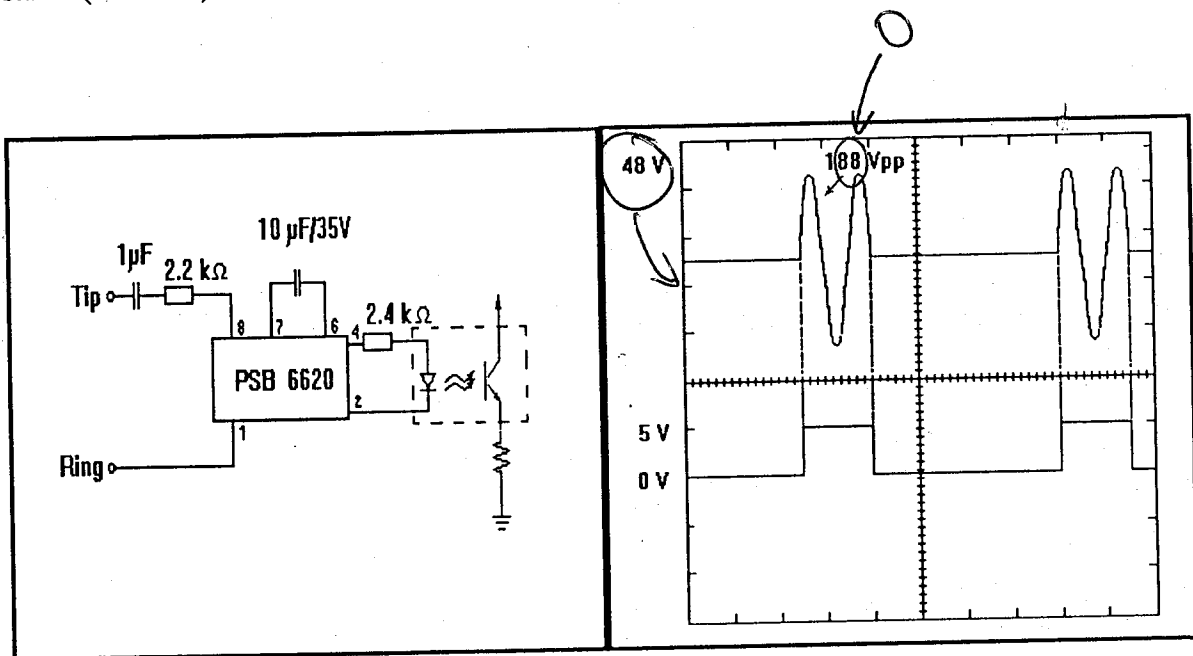


Fig. 22 Funcionamiento del detector de repique.

• Detector de dígitos

• Discado Decádico

El circuito integrado que detecta el discado decádico es el M-959. A este integrado se le conecta el pin LC ("Loop Current") a la salida del optoacoplador (Vopto). El M-959 recibe pulsos del optoacoplador y los traduce a niveles lógicos que indican si está colgada o descolgada

la línea, a través del pin OH y codifica el número de pulsos en una palabra de 4 bits, que son los pines DO-D3.

Cada vez que se detecta un dígito la salida "strobe" permanece en alto por 200 ms y por los pines D0/D3 se encuentra el dígito válido en hexadecimal.

Doscientos (200) ms de tiempo es mas que suficiente para que el microprocesador pueda leer los dígitos válidos sin perder ningún dato en la lectura secuencial de los status de las 24 líneas, ya que en ese tiempo el 8031 puede ejecutar aproximadamente 200.000 instrucciones.

El pin "strobe" se conecta al bit b3 del registro de status (74LS244). Mientras que las salidas DO-D3 se conectan a los bits b4-b7 del registro de status (74LS244) respectivamente. (Ver la información técnica en el Apéndice 7).

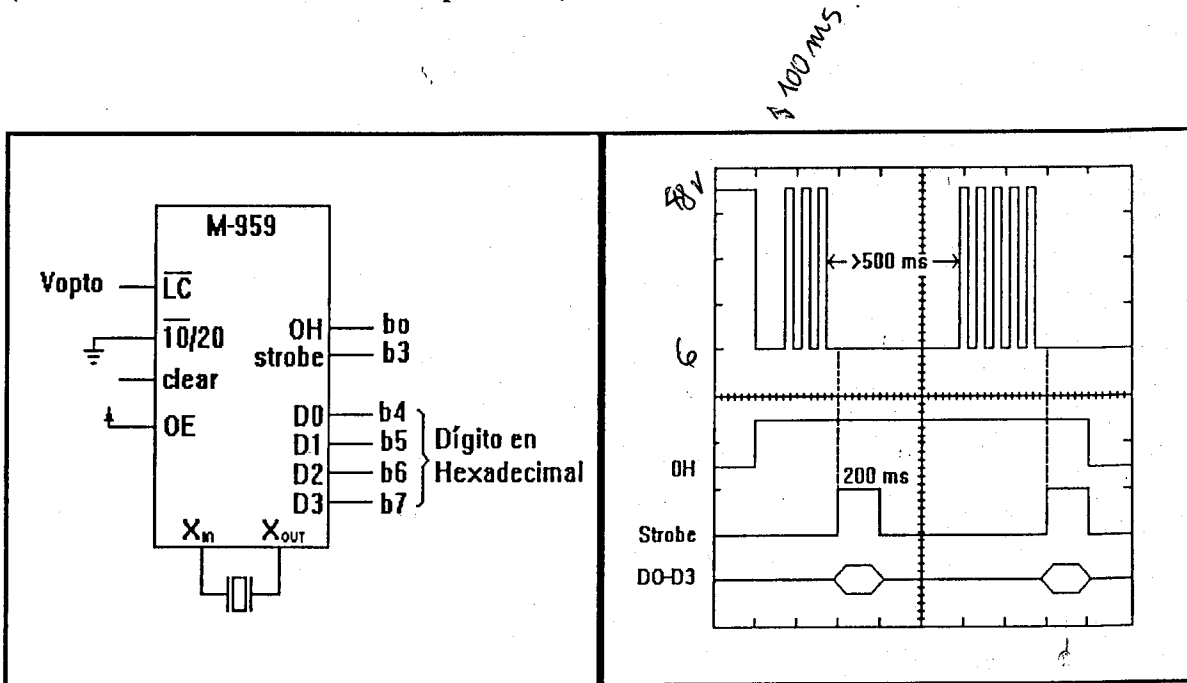


Fig. 23 Funcionamiento del detector de discado decádico

•Discado Multifrecuencial

El MC145436 es el circuito integrado que detecta el discado multifrecuencial. La entrada analógica Ain se conecta a la salida del acople analógico (transformador).

Cuando se detecta un par de tonos que simbolizan un número, el MC145436 pone en 1 la salida "Digital Valid", por el tiempo que dura la marcación del número (Ver Figura 24). Y por los pines DO-D3 saca una salida de 4bits que codifican al par de tonos en hexadecimal. (ver la información técnica en el Apéndice 7)

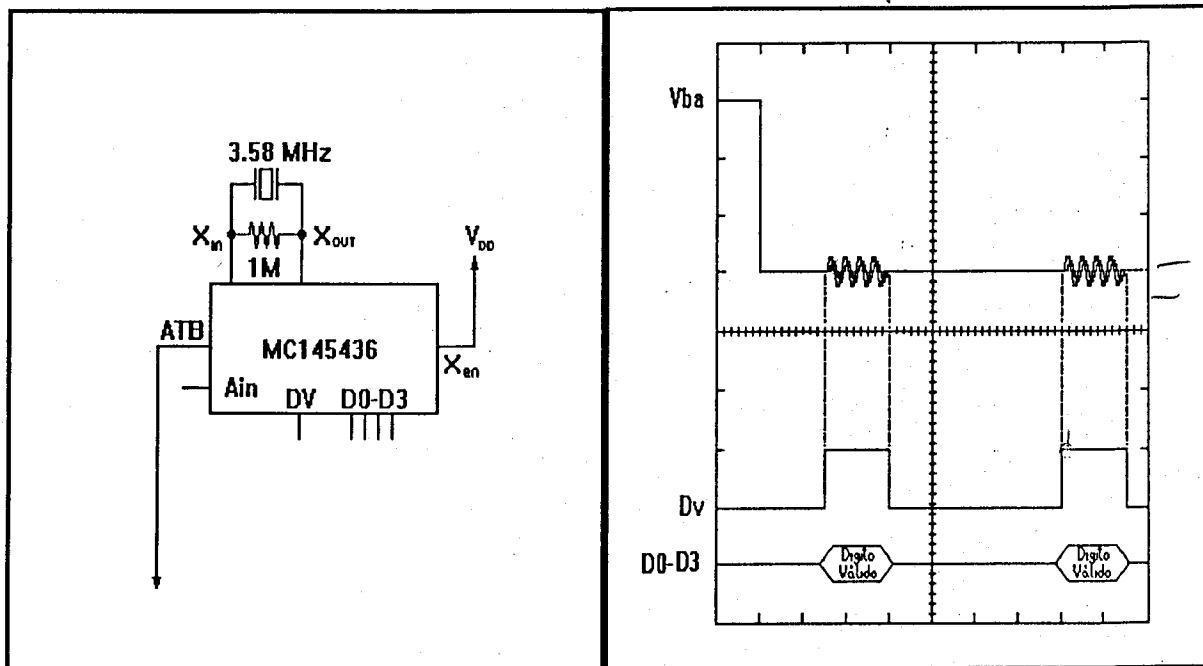


Fig. 24 Funcionamiento del detector de discado DTMF

• Detector de contestación

En el momento en que el abonado llamado contesta, ocurre una inversión de polaridad en la línea, la cual utilizan las centrales públicas como señal para empezar a tarificar. Sin embargo, este cambio en la polaridad no se retransmite al lazo del abonado llamante, el cual es justamente el punto de supervisión del EIC, por lo que se hace necesario el diseño de algún circuito que detecte la contestación de alguna otra forma.

El detectar la contestación por la ausencia del tono de libre (425 Hz, Cadencia 1 seg/4seg) constituye una forma viable de resolver el problema, ya que esta señalización la emiten las centrales públicas desde el momento en que se logra enrutar una llamada a su destino hasta el momento que se establece la conversación.

Sin embargo, la instrumentación del detector de la ausencia de tono de libre es desde el punto de vista electrónico es complicada, por las siguientes razones:

- La frecuencia del tono de libre es de 425 Hz, la cual se encuentra dentro del rango de frecuencia vocal. De esta manera se complica la diferenciación del tono de la voz humana.
- La longitud del enlace físico en las comunicaciones telefónicas es variable y dado que el tono de libre es emitido por la central destino, existe una diferencia en la amplitud del tono para distintas comunicaciones.

El circuito que se describe a continuación constituye una solución apropiada para resolver el problema de la detección del tono de libre, sin embargo, para la detección de la contestación se usa además a el microprocesador, ya que una vez detectado un tono de libre se activa un contador interno y al llegar éste a 5,5 seg se asume que hubo contestación.

El circuito de detección del tono de libre está constituido por cinco partes fundamentales: acople, filtro pasabanda, detector de tono, integrador y comparador (véase Figura 25).

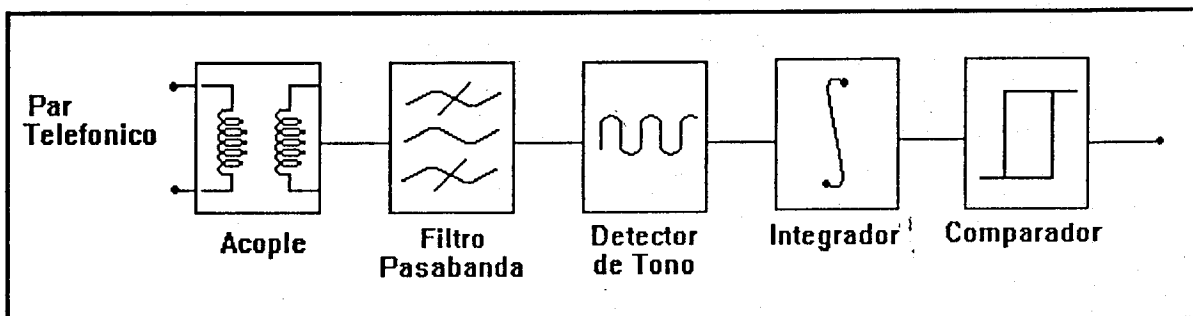


Fig. 25 Diagrama en bloques del detector de tono de libre

• Acople

El detector del tono de libre se conecta al par telefónico a través del acople por transformador, el cual deja pasar al circuito las señales de voz y los tonos de señalización. Este acople fué descrito anteriormente.

. Filtro pasabanda

Al hacer detecciones de tonos en líneas telefónicas, se presenta el inconveniente de que las amplitudes de las señales tienen variaciones dependiendo de la longitud del enlace. El problema en esta aplicación se da en comunicaciones establecidas entre abonados ubicados uno muy lejos del otro, ya que el detector de tono (bloque siguiente del detector del tono de libre) no reconoce la frecuencia de 425 Hz debido a la debilidad de la señal.

Por esta razón, el EIC debería tener circuitos que unificaran los niveles de voltaje. Por ejemplo, se podrían utilizar circuitos de control automático de ganancia, tales como los que usan las centrales telefónicas. Sin embargo, como en este caso no nos interesa amplificar la voz, se adapta mejor la utilización de un filtro pasabanda centrado a la frecuencia de 425 Hz, de manera de solo amplificar la frecuencia del tono de señalización .

Entonces, con el uso del filtro pasabanda se logra amplificar el tono de libre, lo suficiente como para que el detector de tono lo detecte en cualquier circunstancia, pero hay que tomar en cuenta que también se amplifican las componentes de la voz que se encuentran en el rango del ancho de banda del filtro. Así pues, con el uso del filtro todavía se tiene a la voz y al tono mezclados, pero a la voz se le disminuye su ancho de banda y se amplifica el tono de 425 Hz.

El filtro pasabanda utilizado es el MF10 en modo 1, en la Figura 26 se observa su conexión para 2 filtros pasabandas con $f_0 = 425$ Hz.

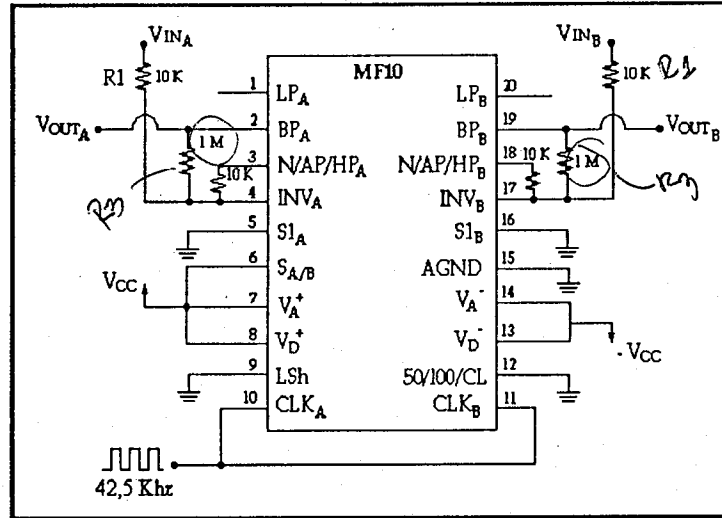


Fig.26 a) Conexión del MF10 en modo 1.

Descripción de los pines:

El MF10 tiene salidas para cada una de las funciones que los filtros pueden hacer: pasabajos, pasabanda, pasaalto, pasatodo y rechaza banda. En esta aplicación se implementaron dos filtros pasabanda por lo que la salida de los filtros serán BP_A y BP_B .

El pin $S_{A/B}$ controla un interruptor interno que sirve para dar la flexibilidad para realizar distintas configuraciones. En el modo 1 $S_{A/B}$ debe ser conectado a $+V_{CC}$, con lo que se conecta la entrada negativa del sumador a la salida pasabajos (LP).

Los pines $S1_A$ y $S1_B$ deben ser conectados a la tierra analógica (AGND). Estos pines son utilizados en los modos 4 y 5 para la entrada de señales.

El MF10 está alimentado con ± 5 V. Entonces los pines de alimentación analógica positiva (V_A^+) y la alimentación digital positiva (V_D^+) van conectados a $+5$ V, mientras que los pines de alimentación analógica negativa (V_A^-) y de alimentación digital negativa (V_D^-) van conectados a -5 V. Cabe decir, que el MF10 puede estar conectado a una con fuente de alimentación simple.

El pin "Level Shift" (LSh) acomoda los niveles del reloj dependiendo de la operación en "single supply" o "dual supply". En el caso de esta aplicación, se tiene el MF10 alimentado

± 5 V, pero el reloj está en niveles TTL (0-5 Voltios), para compensar esta diferencia se debe conectar el pin LSh a tierra.

A los pines CLKA y CLKB se conecta una onda cuadrada de 42,5 KHz, con niveles TTL generada en la tarjeta controladora. Es importante que esta onda tenga un "duty cycle" de 50%, para garantizar un funcionamiento adecuado del filtro.

El pin 50/100/CL se conecta a tierra, de esta manera se fija la frecuencia central del filtro pasabanda a una frecuencia 100 veces menor que la frecuencia de la onda cuadrada que se conecte a los pines CLKA y CLKB. En el caso particular de esta aplicación, la frecuencia central central (f_0) se fija en 425 Hz, ya que la onda cuadrada es de una frecuencia de 42,5 KHz.

El pin AGND se conecta a la mitad de la alimentación total. En este caso se conecta a tierra, pero si se usara alimentación "single supply" este pin se debería conectar a 2,5 V.

Diseño de las resistencias externas:

Para el diseño de los componentes externos del MF10 se fija la frecuencia del reloj externo F_{CLK} y luego se fija la relación de resistencias para el Q deseado. Las ecuaciones de diseño son las siguientes:

$$f_0 = \text{Frecuencia central} = \frac{F_{CLK}}{100}$$

$$Q = \text{Factor de calidad} = \frac{f_0}{B} = \frac{R_2}{R_3}$$

$$H_{oBP} = \text{Ganancia del pasabanda (en V/V) a } f_0 = -\frac{R_3}{R_1}$$

Además de estas fórmulas de diseño se deben tomar en cuenta las dos consideraciones siguientes:

- Para asegurar el funcionamiento apropiado del MF10, el producto $f_0 \times Q$ debe ser menor a 300 K. Lo cual nos dice que Q debe ser menor a 700 aproximadamente.
- Los operacionales internos pueden suministrar a su salida un máximo de 3 ma y la corriente máxima que pueden absorber es de 1,5 ma. Si tomamos en cuenta que la

alimentación es de 5 Voltios, las resistencias que se conecten deben ser mayores a 1 K aproximadamente.

Si seleccionamos los siguientes valores para las resistencias,

$$R1 = R2 = 10 \text{ K}\Omega$$

$$R3 = 1 \text{ M}\Omega$$

Se tiene que,

$$Q = \frac{R3}{R2} = \frac{1 \cdot 10^6}{1 \cdot 10^3} = 100$$

$$H_{oBP} = -\frac{R3}{R1} = -\frac{1 \cdot 10^6}{1 \cdot 10^3} = -100$$

Por otro lado, al tomar estos valores de resistencia el ancho de banda queda en,

$$B = \frac{f_o}{Q} = \frac{425}{100} = 4,25 \text{ Hz}$$

• Detector de tono

La siguiente etapa del *detector del tono de libre* es el detector de tono. Esta etapa la conforma el circuito integrado LM-567, el cual da una salida una salida lógica por el pin 8. Cuando hay presencia de tono el pin 8 baja a cero voltios, mientras que si no hay presencia de tono se mantiene la salida en 5 voltios.

La frecuencia central (f_o) del detector de tono se fija con 2 componentes externos ($R1$ y $C1$). El valor de estos componentes se obtiene con la siguiente fórmula:

$$f_o \approx \frac{1}{R1 \cdot C1}$$

Deseamos que f_o sea 425 Hz y si fijamos $C1$ en 0,1 uF, tenemos para $R1$,

$$R1 = \frac{1}{f_o \cdot C1} = \frac{1}{425 \cdot 0,1 \cdot 10^{-6}} = 23,53 \text{ K}\Omega$$

Entonces, se coloca una resistencia de 20K y un potenciómetro de 5K de precisión. De esta manera, hará falta ajustar este potenciómetro para obtener la frecuencia central deseada.

El capacitor del pin 1 (C3) se usa para obtener un retardo en tiempo de la salida del detector de tono, de esta manera se rechazan posibles ruidos que puedan activar al detector. Sin embargo, no se puede jugar libremente con su valor ya que afecta el ancho de banda. El capacitor del pin 2 (C2), tiene influencia directa sobre el ancho de banda, mientras mayor sea su valor el ancho de banda será menor.

El ancho de banda del filtro puede ser aproximado por la siguiente expresión:

$$B = 1070 \cdot \sqrt{\frac{V_i}{f_0 \cdot C_2}} \text{ en \% de } f_0$$

donde:

V_i = Voltaje de entrada, $V_i \leq 200 \text{ mV}_{\text{rms}}$

C_2 = Capacitor del pin 2 en μF

Entonces si colocamos V_i en 100 mV tenemos,

$$B = 1070 \cdot \sqrt{\frac{100 \cdot 10^{-3}}{425 \cdot 1}} = 16,41\% \text{ de } f_0$$

En la Figura 27 se tiene el diagrama de conexión del LM567 y la forma de las señales en presencia y ausencia del tono de 425 Hz.

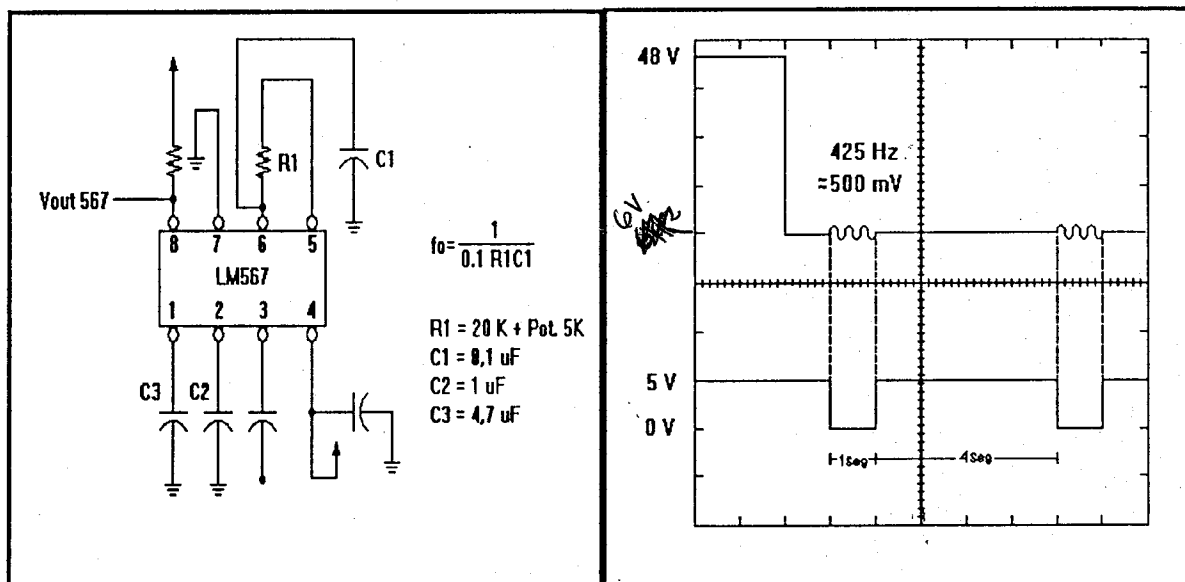


Fig. 27 Funcionamiento del detector de tono de 425 Hz.

• Integrador

Para resolver el problema de la interferencia de la voz humana con el tono de libre, se utilizó un criterio práctico, que se dedujo por medio de comprobaciones experimentales realizadas en el laboratorio en el transcurso de la pasantía.

Esta regla empírica se podría enunciar de la siguiente forma: "Las componentes de 425 Hz en la voz humana nunca superan una duración de 250 ms. Por ejemplo, la voz de los hombres en promedio tienen la componente por intervalos de 20 ms a 150 ms, mientras que la voz de las mujeres presentan esta componente por tiempos de menor duración y en algunos casos de voces muy agudas la componente de 425 Hz es casi inexistente.

Entonces, basados en el hecho de que la duración de los 425 Hz en la voz humana es bastante inferior a la duración de los 425 Hz del tono de libre, la cual es de un segundo, surgió la idea del integrador el cual va "sumando la presencia de señal de 425 Hz".

Con un integrador se puede realizar una conversión de tiempo a voltaje usando una configuración adecuada. La configuración integradora, es realizada con uno de los operacionales del circuito integrado MC1458. En la Figura 28 se puede observar la configuración utilizada. El período de tiempo que se convierte es el tiempo que la señal de salida del detector de tono ($V_{out\ 567}$) permanece en cero voltios.

Cuando $V_{out\ 567}$ se encuentra en 5 Voltios, el diodo D1 conduce y entonces la salida del amplificador se mantiene en cero voltios debido a la realimentación negativa que se produce a través del diodo D2. Cuando $V_{out\ 567}$ está en cero voltios causa que ambos diodos queden en polarizados en reverso y la salida de voltaje del amplificador crece linealmente a una tasa determinada por E_{ref} y la constante de tiempo del integrador RC. La amplitud de la rampa generada es directamente proporcional al tiempo que la señal aplicada al punto A permanezca en cero.

La fórmula de diseño del integrador es la siguiente:

$$E_o = \frac{E_{ref}}{RC} \times t$$

Donde E_O es la amplitud de la rampa de salida, t es el período de tiempo que la señal de entrada está en cero voltios, E_{ref} es el voltaje del divisor de la fuente de -5 voltios.

Con el divisor de voltaje se logra un valor de E_{ref} de -0.381 Voltios. Se desea que cuando hayan pasado más de 700 ms el voltaje de la rampa sea de 2.5 Voltios. Entonces si fijamos el valor del condensador en 0.1 μ F podemos calcular el valor de la resistencia como:

$$R = \frac{0.381}{0.1 \cdot 10^{-6} \cdot 2.5} \times 700 \cdot 10^{-3} = 1,066,800 \Omega \approx 1 \text{ M}\Omega$$

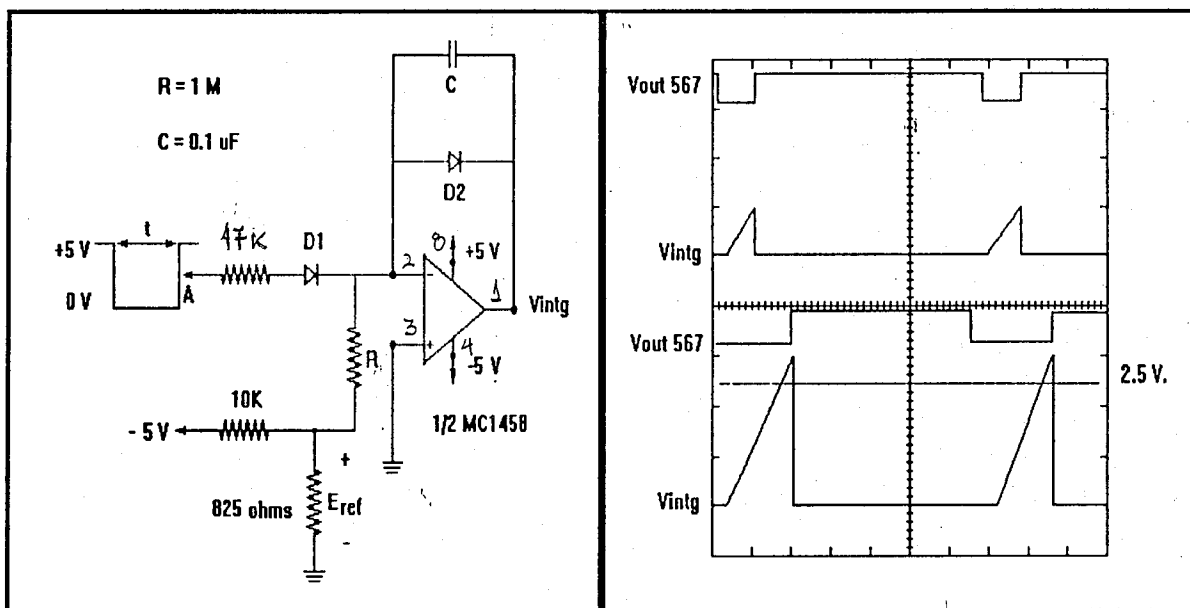


Fig. 28 Funcionamiento del integrador

• Comparador

La última etapa del detector del tono de libre la constituye un comparador realizado con un operacional sin realimentación. Se usa un operacional y no un LM311 para aprovechar el otro operacional que se encuentra en el circuito integrado MC1458. La conexión de este comparador se ilustra en la Figura 29.

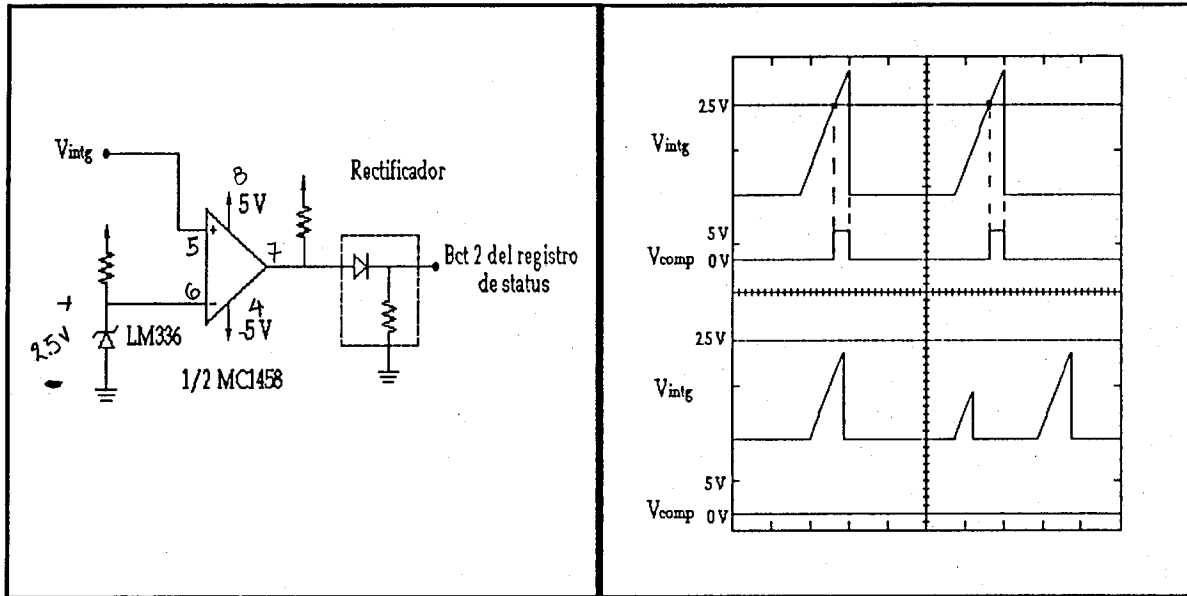


Fig. 29 Comparador para el detector del tono de libre.

El voltaje de salida del integrador (V_{intg}) se conecta al comparador al pin (+). Este voltaje se compara con un voltaje de 2.5 Voltios, el cual se obtiene del diodo zener LM336. De esta forma cuando V_{intg} sea menor de 2,5 V, la salida del comparador estará saturada en -5 V, mientras que cuando V_{intg} sea mayor de 2,5 V la salida del comparador estará saturada en +5V. (ver Figura 29). Es de hacer notar que la señal a la salida del comparador ya no tiene la cadencia del tono de libre (1seg de emisión / 4 seg de silencio), ya que la señal de emisión ahora durará aproximadamente 300 ms.

A la salida del comparador se conecta un rectificador de manera de convertir los niveles de ± 5 Voltios a niveles TTL (0 a 5 Voltios). Finalmente, se conecta la salida del rectificador al pin 2 registro de status de la línea correspondiente.

• Error en la detección de la contestación

El detector de contestación que se ha realizado, el cual está compuesto de 2 partes: detector del tono de libre y contador de tiempo en el microprocesador, realiza una detección aproximada del momento en que se establece la comunicación. El error en la detección de la contestación puede ser de hasta 5,5 seg.

En la Figura 30, se observa un ejemplo del error en la detección de la contestación. Justo después de la última emisión del tono de libre el abonado B contesta, pero no es solo sino hasta que se vence el tiempo en el contador interno del microprocesador que el EIC detecta la contestación.

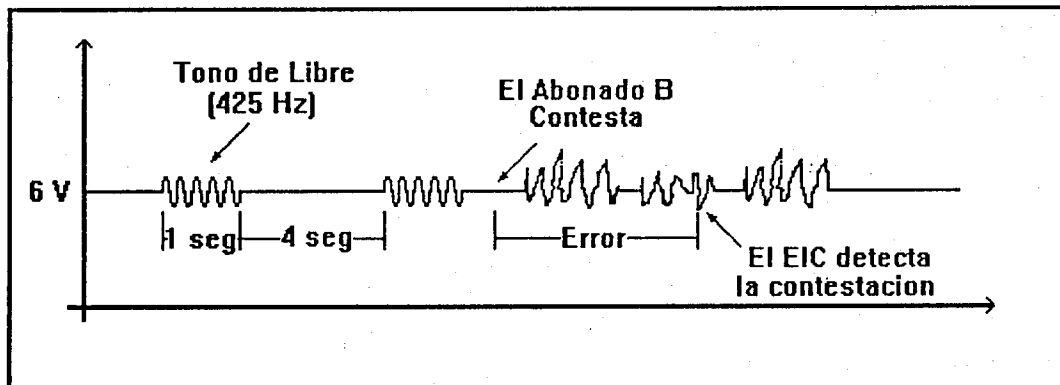


Fig. 30 Error en la detección de la contestación.

3. Interfaz digital hacia el CPU

La interfaz digital hacia el CPU la constituye, un "buffer tri state" 74LS244. A este circuito integrado se conectan los 8 bits que indican el estado de una línea de la forma que se indica en la Tabla 4. Las salidas del circuito integrado 74LS244 se conectan directamente al bus de datos

b0	Detector de colgado y descolgado (OH M-959)
b1	Detector de repique (salida optoacoplador)
b2	Detector de contestación (salida comparada)
b3	Dígito válido (DV M-959)
b4-b7	Dígito en hexadecimal

Tabla 4 Conexión de los pines del registro de status

4. Acople de impulsos a la línea

El acople de impulsos se realiza con el circuito que se ilustra en la Figura 31.

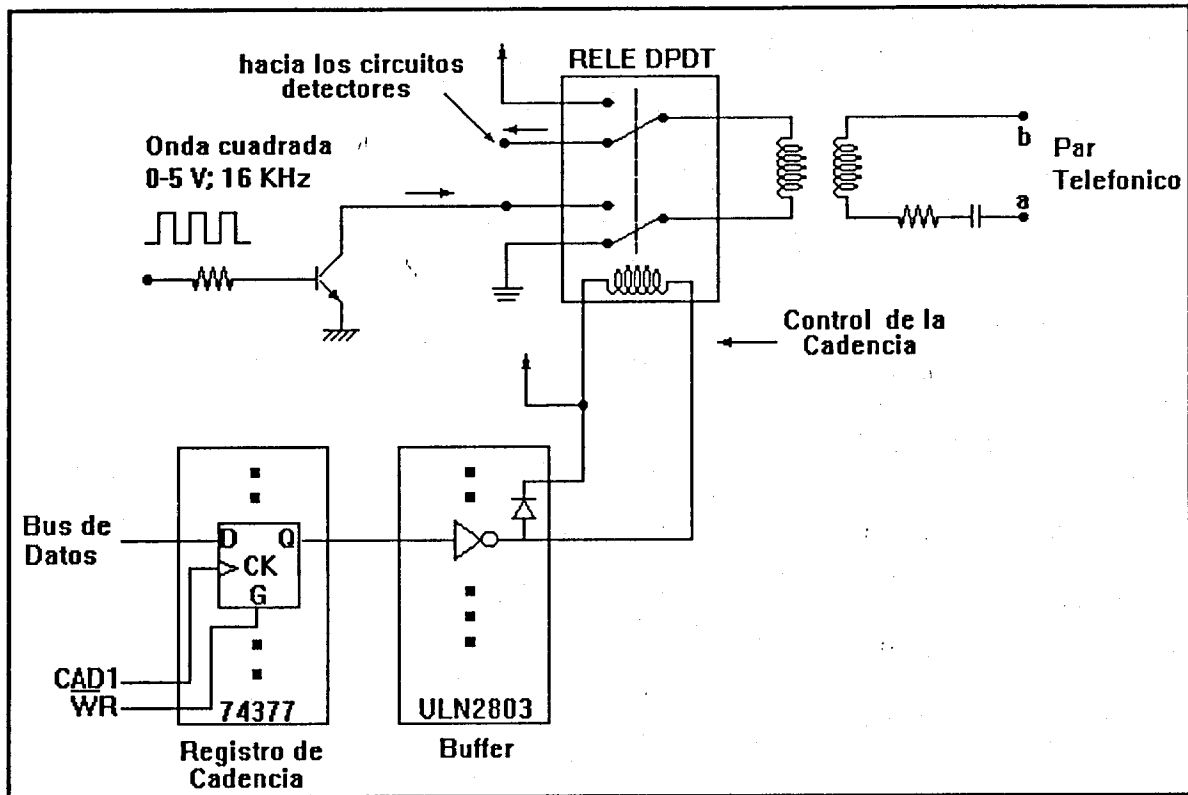


Fig. 31. Acople de los impulsos a la línea con un relé DPDT.

Se utiliza un relé doble polo doble tiro (DPDT), que en su estado de funcionamiento normal conecta la línea telefónica a los circuitos detectores analógicos.

Cuando se desea aplicar impulsos de cómputo el relé conmuta la salida del transformador, a un transistor al cual se le conecta en su base una onda cuadrada de frecuencias de 16 KHz.

La cadencia de la señal de cómputo es controlada por un bit del registro 74LS377. Este bit debe ser pasado por un "driver" de manera de suministrar la corriente necesaria para el funcionamiento del relé.

El "driver" utilizado para controlar el relé es el integrado ULN2803, el cual contiene 8 amplificadores de corriente negadores y además 8 diodos conectados a la salida del amplificador.

Los diodos se conectan en paralelo a la bobina de control del relé, con el objeto de limitar la corriente en el momento de activación del relé.

C. Interfase visual

La interfase visual la constituye una pantalla LCD (Liquid Crystal Display). Esta pantalla de cristal líquido es de 2 líneas, cada una de 16 caracteres.

El controlador HD44780 refresca en la pantalla los datos que tiene almacenados en su RAM interna. El controlador HD44780 está hecho para ser conectado al bus de datos de un sistema con microprocesador, y se controlan todas sus funciones con el manejo de 2 registros internos: el "Instruction Register" (IR) y el "Data Register" (DR). La operación interna está determinada por las señales que el microprocesador le manda. Esas señales son selección de registro (RS), read/write (R/W), el habilitador activo alto (E) y el bus de datos (b0-b7).

La señal RS es conectada al bus de direcciones AO. Cuando $R_s=0$ entonces se habilita a el registro de datos (DR) y cuando $R_s=1$ se habilita al registro de instrucciones (IR). La señal R/W selecciona si se desea escribir o leer a la pantalla LCD. Este pin se conecta al bit A1 del bus de direcciones. Cuando $A_1=0$ se escribe y cuando $A_1=1$ se lee.

El circuito de la Figura 32 ilustra la conexión realizada para habilitar la pantalla LCD. La pantalla LCD se habilita cuando se aplica un 1 lógico al pin "Enable" (E) del controlador HD44780. Este "1" lógico se obtiene cuando ocurre alguna de las dos situaciones siguientes:

- La pantalla LCD está habilitada ($CS_LCD=0$) y se desea leer ($RD=0$)
- La pantalla LCD está habilitada ($CS_LCD=0$) y se desea escribir ($WR=0$)

El efecto deseado sobre el pin "Enable" (E) se logra con la conexión de las compuertas OR y NOT de la manera como se especifica en la Figura 32.

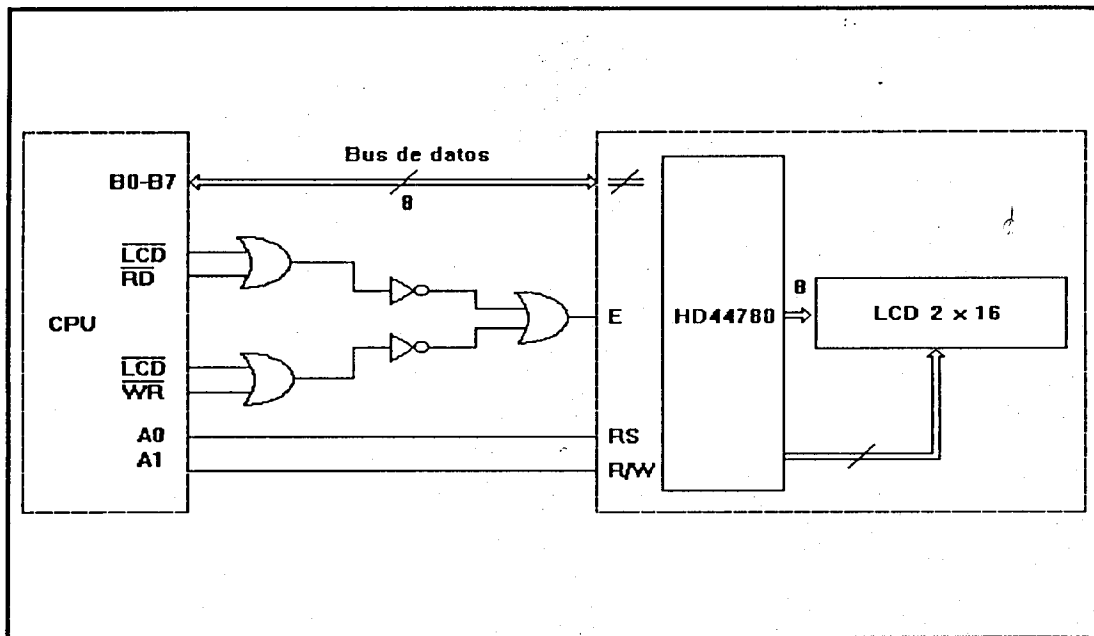


Fig. 32 Conexión del la Pantalla LCD.

D. Interfase serial

1. Conversión de niveles

El protocolo de comunicación que usan los computadores personales es el estandar internacional RS-232, sin embargo, el 8031 no utiliza este estandar por lo que es indispensable una interfaz que realice la conversión de niveles eléctricos para que la comunicación sea posible.

Las especificaciones eléctricas del RS-232 establecen que un voltaje más negativo que -3 Voltios es considerado como un 1 lógico y a un voltaje más positivo que +4 Voltios se le considera un 0 lógico.

Los datos que manda el PC por el pin 2 de su puerto serial están negados y además tienen niveles de 12 Voltios. De esta manera un 1 lógico para el PC es -12 Voltios y un cero lógico para el PC es 12 Voltios. Se hace necesario entonces una interfase que modifique estos niveles para que sean entendidos por el 8031.

Esta interfase esta formada por una resistencia, un diodo zener de 5.1 V y un negador, para la interfaz de la recepción, tal como se muestra en la Figura 33. Así cuando llegue un nivel

de +12 Voltios (0 lógico) el diodo zener quedará polarizado en directo y bajará el nivel de tensión a 5 Voltios, y luego con el negador se tendrá el nivel de 0 V entendible por el 8031.

De igual forma si viene por el pin 2 (TXD) del puerto serie un nivel de -12 Voltios el diodo zener queda polarizado en directo, por lo que se tendrá en sus terminales un voltaje de 0.7 Voltios que es interpretado por el negador TTL 7404 como un cero y al negarlo se obtiene el 1 lógico que entiende el 8031.

Para la transmisión desde el microprocesador hacia el PC igualmente hace falta una conversión de nivel y una negación. Este efecto se logra con un comparador alimentado entre +5 Voltios y -5 Voltios, al cual se le introduce la señal TXD₈₀₃₁ por el pin (-) y se le compara con un valor positivo de voltaje. Entonces +5 Voltios se convertirá en -5 Voltios (1 lógico para PC) y 0 Voltios en +5 Voltios (0 lógico para el PC).

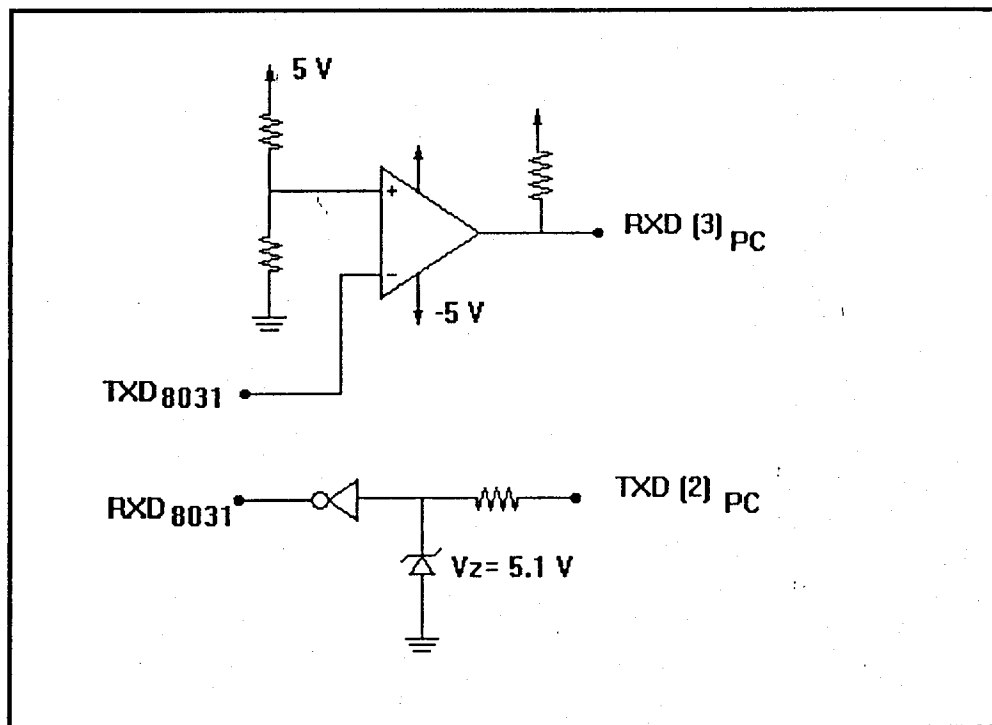


Fig. 33 Interfaz serial

2. Conexión de los pines del puerto serial

La Figura 34 muestra la conexión de los pines realizada para el puerto serial. Los pines no mostrados son omitidos en la conexión.

Al prender el computador, este aplica un "1" en el pin Data Terminal Ready (DTR, pin 20). En una comunicación serial típica el computador estaría esperando por la puesta en "1" del pin Data Set Ready (DSR pin 6) proveniente de un periférico, pero en esta aplicación se engaña al computador ya que se establece un puente DTR con DSR. Así cuando el computador se encienda y se genere un "1" por DTR inmediatamente obtendrá una respuesta por DSR, habilitando de esta forma la comunicación serial.

Cuando el computador desea transmitir datos coloca al pin "Request To Send" (RTS, pin 4) en "1" y no manda nada hasta que el periférico le conteste que se encuentra disponible a recibir datos colocando en "1" el pin "Clear To Send" (CTS, pin 5). En este caso, también se hacen puentes entre los pines RTS y CTS. Así el "1" que genera el PC cuando desea transmitir datos es el mismo que recibe por CTS.

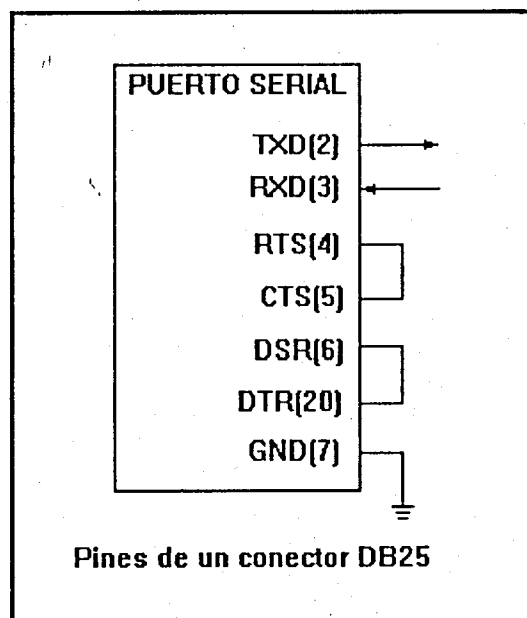


Fig. 34 Conexión de los pines del puerto serial

Para que la conexión a través del puerto serial tenga sentido se deben cruzar los cables de los pines 2 y 3. A esta conexión se le denomina "modem null". De esta manera los datos que manda el PC por el pin 2 (TXD) son recibidos por el pin 3 (RXD) del conector DB25 del EIC. El caso contrario ocurre cuando el EIC desea transmitir hacia el PC.

La transmisión de datos se realiza por el pin 2 del puerto serial (TXD), mientras que la recepción de datos se hace por el pin 3 (RXD). La tierra de la comunicación es el pin 7.

3. Conversión de DB9 a DB25

El EIC se puede programar por el puerto serial a través de un PC. La mayoría de los computadores tiene un puerto serial que es conector DB25 macho, sin embargo, muchos "LAP TOPS" tienen el puerto serial DB9 macho. Para realizar la conversión de 9 pines a 25 pines, se debe realizar la conexión que se ilustra en el Apéndice 11.

E. Fuente de alimentación

1. Fuente de $\pm 5V$

La fuente de alimentación para el prototipo montado se deriva de la corriente alterna de la red comercial (120v, 60 Hz).

El transformador utilizado reduce la tensión de la red de $120 V_{rms}$ a $24 V_{rms}$. Este transformador es "center tap", por lo que se utilizan $12V_{rms}$ para la fuente positiva y los restantes $12V_{rms}$ para la fuente negativa. Este transformador fué diseñado para soportar una corriente de 1 amperio en el secundario.

La siguiente etapa de la fuente es el puente de diodos rectificador, el cual puede suministrar hasta una corriente de un amperio como máximo.

Para la fuente de voltaje positiva se utilizan dos reguladores LM7805: uno para alimentar a la tarjeta controladora y otro para alimentar a la tarjeta detectora.

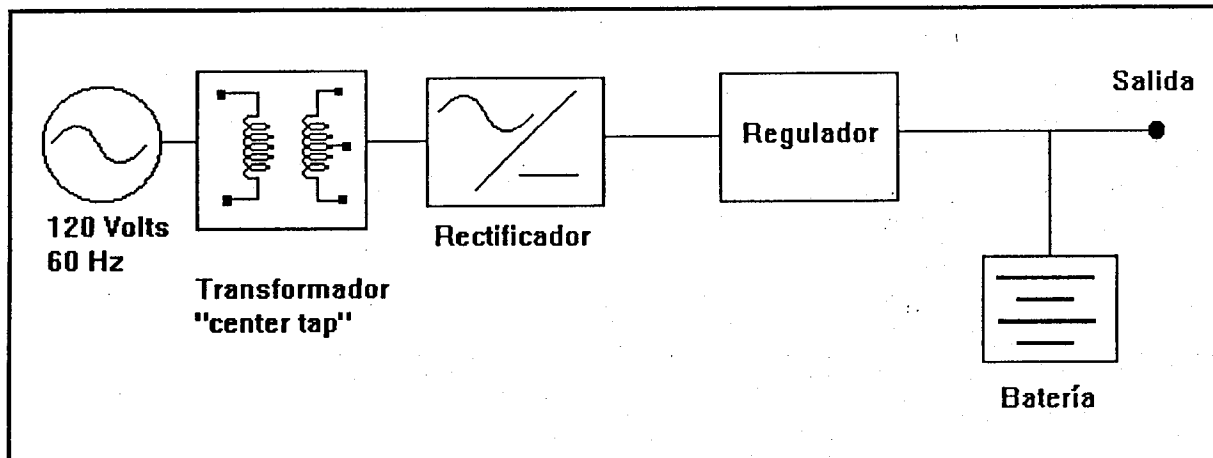


Fig. 35 Diagrama en bloques de la fuente de alimentación

Estos reguladores, con sus disipadores adecuados, pueden dar al circuito una corriente de hasta 1.5 amperios cada uno. Aunque los requerimientos de corriente son mucho menores (≈ 500 mA para todo el circuito), se prefirió sobredimensionar con el objeto de obtener una fuente estable.

Para los 5 Voltios negativos se utiliza el regulador LM7905 de plástico, el cual puede dar hasta 0,5 amperios, lo cual es mas que suficiente para los requerimientos de este diseño circuital.

La Figura 35 muestra un diagrama en bloques de los componentes de la fuente. Para obtener información detallada de su conexión ver el Apéndice 1.

2. Respaldo de Baterías:

El prototipo montado tiene una autonomía por fallos de la red de energía eléctrica o por desconexión del equipo de doce horas.

Esta autonomía es provista por una batería de NiCd de 3.6Voltios, la cual en operación normal es cargada a través de un divisor de tensión de R1 y R2 y del diodo D2, mientras que el diodo D1 conduce para darle alimentación a la RAM y al circuito integrado Reloj calendario.

Cuando no hay 5 Voltios, D1 y D2 no conducen mientras que D3 está polarizado en directo, suministrando la alimentación a la RAM y al Reloj Calendario

El integrado de Reloj Calendario comparte con la RAM el circuito de respaldo de la alimentación (Figura 36) , lográndose con esto no perder los parámetros de programación del equipo.

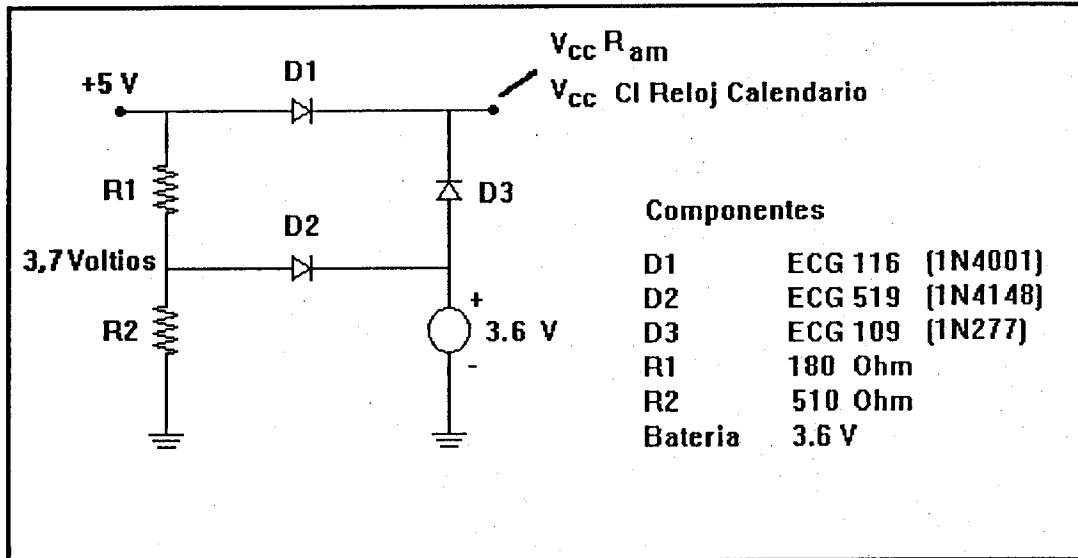


Fig. 36 Circuito de Respaldo de la alimentación.

Cuando ocurre una falla o se desconecta el equipo, el voltaje de la batería cae bruscamente hasta 2,8 Voltios aproximadamente. Mientras que en el tiempo de descarga sucesivo el voltaje de la batería baja lentamente.

Cuando el voltaje llega hasta 1,5 Voltios la RAM deja de funcionar, perdiéndose la información almacenada.

CAPITULO 4 DESCRIPCIÓN DEL "SOFTWARE"

A. Programación en Assembler del 8031

1. Estructura del programa

El programa en assembler que controla al EIC está dividido en 7 módulos, con el objeto de lograr una mayor claridad al programa. Cada uno de estos módulos agrupa las rutinas relacionadas con con la resolución de problemas específicos. La Figura 37 muestra el nombre de los módulos y su interrelación.

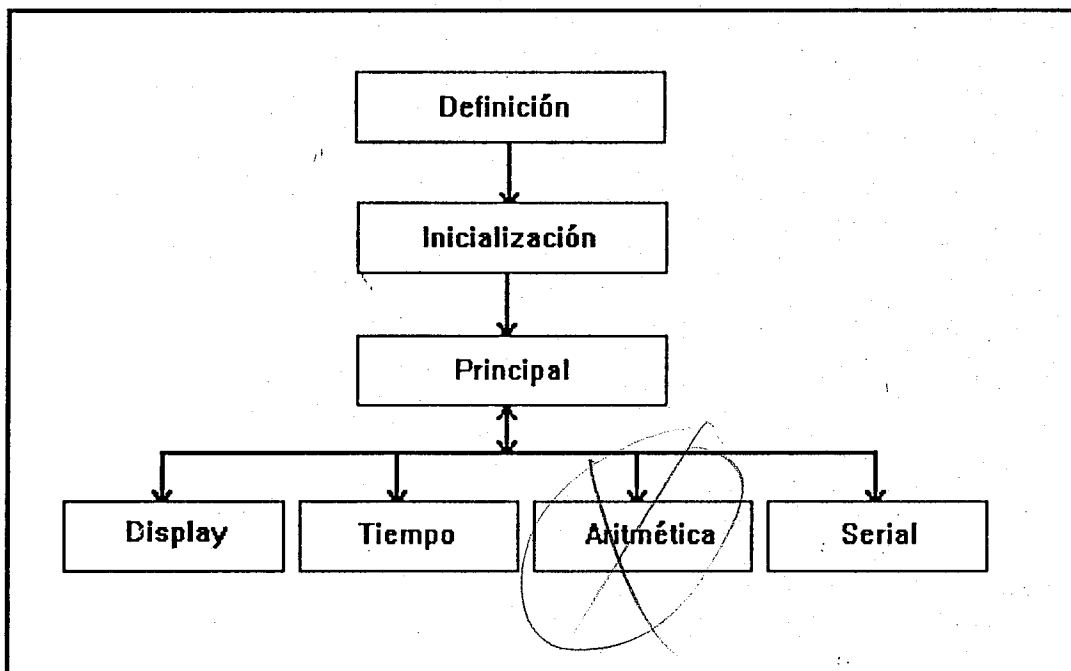


Fig. 37 Estructura modular del programa en assembler.

El código en assembler del programa se encuentra en el Apéndice 2. En las siguientes secciones se realiza una descripción funcional de las rutinas de cada uno de los módulos, es decir, se explica que hacen las rutinas, con que rutinas se relacionan y que parámetros necesitan para funcionar. Sin embargo, en las rutinas principales se hace una explicación más detallada.

2. Módulo "Definición"

Este módulo tiene por objeto definir las etiquetas de las variables y constantes utilizadas en el programa. De esta manera, el programa aumenta su legibilidad y además se facilita la labor de programación.

La definición de las etiquetas se efectúa con la pseudo-operación "EQU", la cual realiza una asignación de una constante hexadecimal a un "string" de caracteres. Cabe destacar que estas instrucciones de definición de etiquetas no se ensamblan en el código objeto que ejecuta el microprocesador. Entonces, para definir un nuevo símbolo se usa la siguiente sintaxis:

Simb EQU25H

Así, cada vez que se use el símbolo *Simb* se estará usando efectivamente el valor 25 hexadecimal. En este módulo se definen etiquetas de tres grupos principales: Periféricos, Tablas y Variables. A continuación se describen las etiquetas definidas en cada uno de estos grupos.

•Periféricos

En este módulo se definen primeramente las direcciones de los periféricos tomando en cuenta el mapa de memoria de memoria realizado en el "hardware" del equipo (Ver capítulo 3). Las direcciones de los periféricos son de 16 bits, por lo que cada constante puede ser definida como una variable de 4 dígitos en hexadecimal.

Basicamente se definen las direcciones relacionadas con 4 periféricos. Ellos son:

- Reloj-Calendario, el cual tiene asociadas 10 direcciones.
- Pantalla LCD, el cual se maneja con 3 direcciones.
- Status de las líneas urbanas, una dirección por cada línea, dando un total de 24 direcciones
- Registros para las cadencia, son 3 direcciones.

Las direcciones exactas de cada uno de estos periféricos se encuentran especificadas en el código del programa (Apéndice 2).

•Tablas

También en este módulo se realiza la definición de las direcciones de origen de las tablas usadas en el programa. A continuación se enumeran estas tablas y sus respectivas funciones. Los nombres entre paréntesis son las etiquetas de las direcciones iniciales de estas tablas:

•TABLA DE LOS DIAS DE LA SEMANA (TAB_DIA): Esta es una tabla que contiene el nombre de cada día de la semana. Los nombres se encuentran almacenados en la tabla como una abreviatura de 3 letras. La tabla se utiliza para la presentación de la fecha en la pantalla LCD.

•TABLA DE LOS MESES DEL AÑO (TAB_MES): En esta tabla se tienen todos los nombres abreviados de los meses del año en caracteres ASCII. Esta tabla se usa para la presentación de la fecha en la pantalla LCD.

•TABLA DE LOS DIAS FERIADOS EN VENEZUELA (TAB_FER): Esta tabla contiene los días feriados en Venezuela para el año 1994.

•TABLA DE TARIFAS URBANAS (TAB_URB): Esta tabla contiene en sus dos (2) primeros bytes la constante de recarga para la variable ESP para la cadencia de los impulsos de una llamada local (90 seg). Los cuatro bytes siguientes contienen el costo del impulso no residencial.

•TABLA DE LA CADENCIA (TAB_DDN): Esta tabla contiene las constantes de recarga de la variable ESP para la cadencia de los impulsos de una llamada nacional. Se tiene información de la tarifa normal y de la tarifa especial.

•TABLA DE TARIFAS NACIONALES (TAB_TNAC): En esta tabla se tienen las tarifas por llamadas Nacionales. La Tabla tiene dos campos: el primero de ellos representa el código de área; el segundo campo es el código de tarifa, ambos campos son de un byte.

•TABLA DE TARIFAS POR OPERADORES A NIVEL NACIONAL (100) (TAB_ONAC): Esta tabla contiene el costo por impulsos por conexión a 3 minutos y minuto adicional, para la Tarifa Normal y Tarifa Especial.

•TABLA DE TARIFAS CELULARES (TAB_CEL): Los dos primeros bytes de esta tabla contienen las constantes de recarga de ESPL para Tarifas Celulares. Los cuatro bytes siguientes contienen (2 bytes) el costo por impulso para llamadas residenciales y los otros dos bytes para abonados noresidenciales.

•TABLA DE TARIFAS INTERNACIONALES (TAB_INT): Esta tabla incluye la información relativa a las tarifas internacionales de la CANTV.

•TABLA DE HORARIOS INTERNACIONALES (TAB_HOI): En esta tabla hay información acerca de los horarios en los que se aplica tarifa reducida y económica según el continente destino de las llamadas.

•TABLA DE LAS VARIABLES DE ANALISIS (TAB_VAR): En esta tabla se tiene por cada línea 32 bytes de información concerniente al estado de la línea. Su estructura se puede observar en la Figura 38. Esta tabla contiene información fundamental para tomar decisiones dependiendo del estado de la línea.

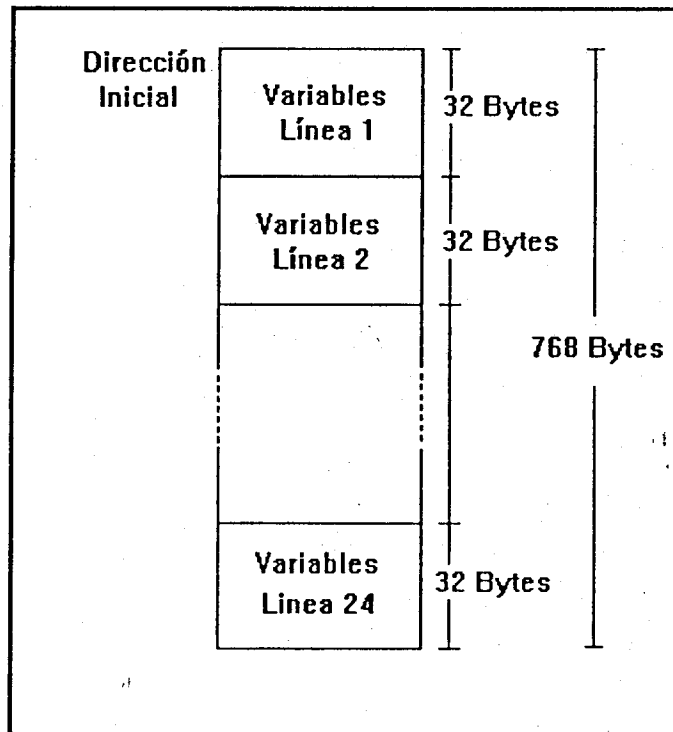


Fig. 38 Estructura de la tabla de variables de análisis.

*deber como
están ordenadas
las 32 variables*

- 0 ESPL
- 1 ESPL
- 2 C50MSCH
- 3 C50MSCL
- 4 FLAGS
- 5 STATUS
- 6 NMAR
- 7 C-REP
- 8 C-RBK

- 15 -
- 16 No1
- 17 No2

• **TABLA DE MENSAJES (TAB_MEN):** Esta tabla contiene mensajes de una longitud máxima de 16 caracteres. Cada vez que se desee escribir algún mensaje en la pantalla LCD se busca el mensaje en esta tabla. El manejo de la escritura de los mensajes en la pantalla LCD se optimiza con el uso de esta tabla.

• Variables

Las variables que se describen en esta sección son globales y son usadas para el análisis del estado de cualquier línea. Todas estas variables se encuentran en la RAM interna de 128 bytes del microprocesador. En este módulo se define la dirección en la que se encuentra cada variable con la pseudo-operación EQU. A continuación se enumeran cada una de las variables con el nombre con el que aparecen en el código en assembler y se explica brevemente su función:

• **NO_LIN:** Esta variable contiene el número de líneas que maneja el equipo. Su valor por "default" es 24, pero este valor puede ser modificado por programación a través del puerto serial. La variable puede tomar sólo cuatro valores (1, 8, 16 y 24). 4 ?

• **NO_LOOP:** El valor de esta variable es $NO_LIN / 3$, menos en el caso de que NO_LIN sea igual a 1, en cuyo caso $NO_LOOP = 1$. Esta variable se usa para hacer los lazos repetitivos del programa.] ?

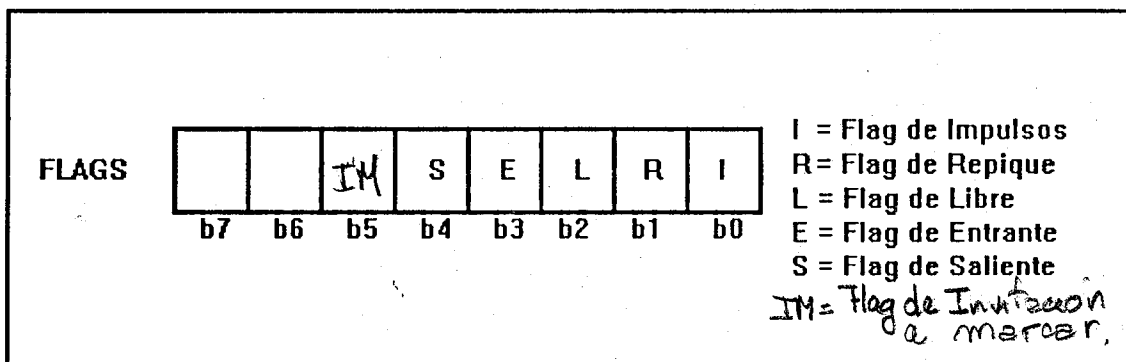
• **HORACT, MINACT, DSEACT, DMESACT, MESACT, AÑOACT, AM_PM:** En estas variables se mantiene la hora y fecha con el objeto de aplicar las tarifas por franjas horarias aplicadas por CANTV. Estas variables son actualizadas cada minuto por las interrupciones del circuito integrado Reloj-Calendarío.

• **V_IMP1, V_IMP2, V-IMP3:** En estas tres variables se construye la señal que controla la cadencia de los impulsos de cómputo. Cada variable es de un byte en donde cada bit controla la cadencia de una línea, así, cada variable controla ocho líneas. Estas variables se actualizan cada 50 ms. cuando el temporizador 0 interrumpe y luego se cargan los registros de cadencia de las tarjetas con los valores que tengan las variables en ese momento.

• ESPL, ESPH: En esta variable de 2 bytes se guarda la constante necesaria para lograr el tiempo entre impulsos de una línea en particular. El valor de ESP se determina dependiendo de los números que disquen los usuarios.

• C50MSCH, C50MSCL: Esta variable de 2 bytes es un contador que se incrementa cada vez que el temporizador 0 interrumpe. Con este contador se genera la cadencia de los impulsos de cómputo.

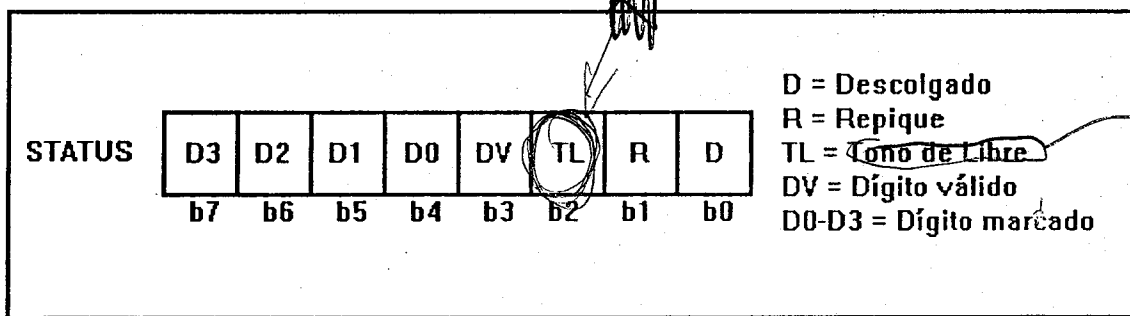
• "FLAGS": Es una variable bit a bit, en donde se puede observar el progreso del estado de una línea. Cuando un "flag" se encuentra en 1 quiere decir que la condición sucedió. La estructura del registro de "flags" se puede observar en la Figura 39.



Cada línea tiene una variable de Flags asociada que se carga en la variable global FLAGS cuando se descolga.

Fig. 39 Estructura de la variable "Flags".

• STATUS: Es la variable en donde se guarda la información leída de los registros de status de cada línea. Su contenido refleja el estado actual de una línea en particular. Su estructura bit a bit se observa en la Figura 40.



TL = Tono de Libre

Fig. 40 Estructura de la variable "Status".

- **ST_VIEJO:** Es una variable en donde se guarda el status de la lectura anterior de una línea en particular, su valor se compara con el status actual de la línea y de esa manera se detectan transiciones en las señales que conforman la palabra de status.

- **C_REP:** Esta variable es un contador que comienza su cuenta cada vez que llega un repique en una línea en particular y es incrementado cada 50 ms cuando el temporizador 0 interrumpe. Su función es determinar cuando una llamada entrante no es exitosa, es decir, si pasan más de 5 seg. después del último repique y la línea está colgada quiere decir que la llamada no fué contestada por la extensión destino de la central privada.

- **C_RBK:** Esta variable es un contador que comienza su cuenta cada vez que llega un tono de libre o "Ring Back" a una línea en particular. Este contador se incrementa cada 50 ms y cuando llega a la cuenta de 5 seg, entonces el EIC asume, dada la ausencia del tono, que la llamada fué contestada.

- **NMAR:** Cada vez que se detecta un número se incrementa el contador de manera de saber cuanto números fueron marcados en una línea en particular.

- **CAM_DES, CAM_REP, CAM_TL, CAM_DV:** En cada una de estas variables se guarda una constante cuyo valor varía dependiendo del tipo de cambio ocurrido en los bits de la variable STATUS de una línea.

- **COD:** Es una variable de 8 bits en donde se está el código de la ciudad destino con el objeto de compararlo con los códigos de las tablas de tarifas y de esta forma determinar la tarifa aplicar.

La Tabla 5 es un resumen de las variables globales de análisis. Estas variables son ocupadas momentáneamente por los valores que tengan para la línea que se está analizando, a través de la rutina RECUPERA del módulo ARITMETICA.

NOMBRE	FUNCIÓN
ESP	Tiempo de silencio de la cadencia de los impulsos de cómputo
C50MSC	Contador de 50ms para la cadencia
FLAGS	Indican el estado de la línea
STATUS	Status de una línea cualquiera leído de los registros de status
ST VIEJO	Ultimo status de la línea
C REP	Contador de "Time-out" para el repique
C RBK	Contador para decisión de contestación
NMAR	Cantidad de dígitos marcados

Tabla 5 Variables necesarias para análisis de una línea

3. Módulo "Inicialización"

Cuando botón de "RESET" es apretado, el 8031 ejecuta el algoritmo de "RESET" interno el cual escribe ceros en todos los registros del SFR(Special Function Register) excepto en los puertos, el Stack Pointer y el registro SBUF (Serial Buffer). Los puertos son inicializados a FFH (con lo que quedan programados para lectura), el Stack Pointer a 07H y el SBUF es indeterminado.

El program counter (PC) queda entonces en 0000H, y esa es la dirección en la cual el microprocesador comienza a buscar instrucciones para ejecutar. La primera instrucción que se ejecuta es un salto a la dirección de inicio del programa:

```
SJMP 100H
```

El programa se ensambla a partir de la dirección 100H, con el objeto de que no interfiera con las direcciones de vectorización de las interrupciones. Para lograr que el programa se ensamble a partir de esa dirección se usa la pseudo instrucción:

```
ORG 100H
```

De la dirección 100H en adelante se ensamblan los módulos en el siguiente orden:

- "Inicialización"
- "Principal"
- "Serial"
- "Display"

- "Aritmética"
- "Tiempo"

En el módulo "Inicialización" se realizan una serie de acciones de inicialización de periféricos y otras labores de inicialización general. A continuación se describen las acciones que se realizan en este módulo.

• Redireccionamiento de la vectorización de las interrupciones

En la parte baja de la memoria de programa se encuentran posiciones fijas de vectorización de las interrupciones. Como se puede observar en la Tabla 6, el intervalo físico entre 2 vectores de interrupciones es de 8 bytes. Este espacio es muy pequeño para tener una rutina, por lo que se hace necesario desviar el PC mediante un salto (LJMP) a una zona de la memoria capaz de contener la rutina de atención a la interrupción.

Fuente	Dirección
INT0	0003H
TIMER 0	000BH
INT1	0013H
TIMER 1	001BH
SERIAL	0023H

Tabla 6 Tabla de vectorizaciones

• Programación de los "Timers"

El 8031 tiene 2 "Timers", que pueden ser configurados como temporizadores o como contadores. Cuando un "Timer" opera como temporizador, el registro contador se incrementa cada ciclo de máquina, mientras que cuando opera como contador, el registro contador es incrementado en la transición 1 a 0 (flanco descendente) de una señal externa, la cual es aplicada

al pin T0 para el "Timer" 0 y a pin T1 para el "Timer" 1. A continuación se explica la estructura del "Timer" 0, la cual es idéntica a la del "Timer" 1.

La función de temporizador o contador es seleccionada por el bit de control C/T perteneciente al registro TMOD (Timer/Counter Mode Control Register). Cuando C/T=0, se programa al "Timer" como temporizador, mientras que cuando C/T=1 el "Timer" funciona como contador de pulsos externos.

El bit de GATE, perteneciente también al registro TMOD, se pone en cero ya que no se desea hacer un control externo del funcionamiento del temporizador (Si GATE=1 entonces se habilita la entrada exterior INTO).

Con el bit TRO, el cual se encuentra en el registro TCON (Timer\Counter Control Register), se habilita el comienzo de la cuenta de los ciclos de máquina. Cuando el registro de conteo (TH0/TLO) da una vuelta, es decir, pasa de FFFFH a 0000H, se activa la bandera de interrupción TFO el cual es un bit perteneciente al registro TCON.

Para una mejor comprensión del funcionamiento del "Timer" 0, observar la Figura 41 y revisar la estructura de los registros del 8031 que se encuentra en el Apéndice 5.

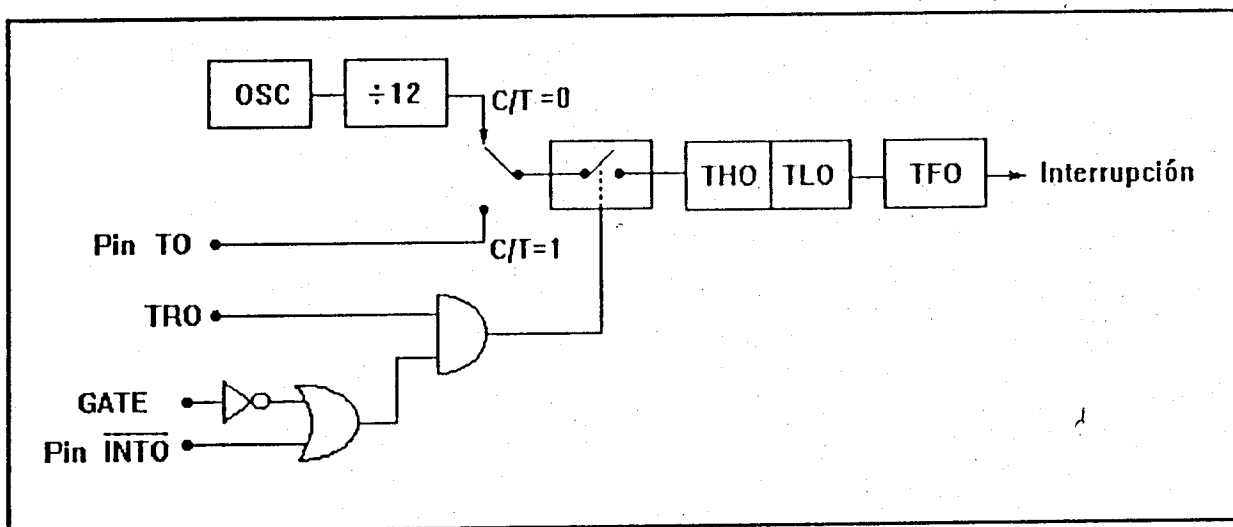


Fig. 41 Estructura del temporizador 0.

Los "Timers" se programan ambos como temporizadores de 16 bits. El "Timer" 0 se le carga una constante de manera de que interrumpa cada 50 ms y al "Timer" 1 se le carga una

constante de manera de que interrumpa cada período medio de la frecuencia de los impulsos de cómputo programada.

Para el cálculo de la constante de conteo de los temporizadores se deben tomar en cuenta los siguientes factores:

- Los temporizadores interrumpen cada vez que alcanzan la cuenta máxima, que en este caso es FFFFh.
- El ciclo de máquina del microprocesador es de 1 us.
- Al caer en una rutina de interrupción se guardan todas las variables del contexto en el "stack". Este proceso de salvamento de los registros del microprocesador tiene una duración de 5 μ s aproximadamente.

Para lograr la base de tiempo de tiempo de 50 ms, generada por las interrupciones del "Timer" 0, se deben contar 50000 ciclos de máquina, entonces se debe cargar en el registro contador (TH0/TL0) la constante 3D00H, ya que:

$$\text{FFFFH} - \text{3D00H} = \text{C200H} = 50000d.$$

Entre los requisitos del proyecto se encuentra que el equipo debe ser conmutable para 3 frecuencias de impulsos de cómputo (50 Hz, 12 KHz y 16 KHz). Para lograr estas frecuencias con el "Timer" 1 hace falta crear una variable que contenga la constante de conteo adecuada para generar cada una de las 3 frecuencias. Esta variable es de 16 bits y esta formada por dos bytes de la RAM interna (TEMPH y TEMPL).

En la Tabla 7 se resume el valor de las constantes TEMPH y TEMPL para cada una de las tres frecuencias de interés. La frecuencia de los impulsos de cómputo es por "default" 16 Khz, lo cual se logra cargando en la variable TEMP la constante FFEAh.

f	T	T/2	TEMPH\TEMPL (TEMP)
50 Hz	20 ms	10 ms	D900h
12 KHz	83.33 us	41.67 us	FFE0h
16 KHz	62.5 us	31.25 us	FFEAh

Tabla 7 Constantes de conteo de recarga del "Timer" 1 para las tres frecuencias.

•Programación del Puerto Serial

Para generar los baudios se utiliza el "Timer" 1 programado en modo 1 (Temporizador en modo de autorecarga), lo cual se logra poniendo en el registro TMOD la constante 20H. El "Timer" 1 genera los baudios sólo cuando se desea programar al EIC por el puerto serial, el resto del tiempo el "Timer" 1 genera la frecuencia de los impulsos de cómputo. Ver la estructura del registro TMOD Apéndice 5.

Para calcular la constante de recarga del "Timer" 1 (TH1) se utiliza la siguiente fórmula:

$$TH1 = 256 - \frac{\text{Frecuencia del oscilador}/384}{\text{Velocidad en baudios}}$$

Como se desea una velocidad de 2400 baudios y la frecuencia del cristal es 12 MHz se tiene que:

$$TH1 = 256 - \frac{12 \cdot 10^6 / 384}{2400} = 242.97 = 243d = F3h$$

En esta aplicación el puerto serie fué programado en el modo 1, esto es: 8 bit UART (Universal Asynchronous Receiver and Transmitter), velocidad 2400 baudios, 8 bits de datos y un bit de stop. Para lograr este modo de funcionamiento se debe cargar en el registro SCON (Serial Control) la constante 70h. Ver en el Apéndice 5 la estructura del registro SCON.

•Inicialización del número de líneas

El número de líneas que puede manejar el EIC se inicializa en 24. Por lo que el valor de la variable NO_LOOP queda en 3. Si se desea que el EIC trabaje con un número menor de

líneas se debe cambiar el valor de la variable NO_LIN al valor deseado, por programación a través del puerto serial.

• Programación del "display" LCD

El controlador del "display" (HD44780) se programa para que maneje 2 líneas y para que use el set de caracteres de 5x7 "pixels". Para programar al "display" se llama a la subrutina INST, a la cual se le pasa como parámetro la constante 38H.

Luego se limpia el "display" y se coloca el cursor en la esquina superior izquierda. Esta programación esta contenida en la subrutina PDISA.

• Programación del circuito integrado Reloj-Calendario

El RTC MM58167A (Microprocessor "Real Time Clock") tiene un conjunto de registros de control y otro conjunto de registros de datos. Los registros de control son los que permiten programar al circuito integrado para que trabaje en distintos modos de operación. Los registros que se programan son los siguientes:

- COUNTERS "RESET" (C__RST): Este registro debe ser cargado con FFH para hacer un "RESET" de todos los registros contadores del RTC.
- GO COMAND (C__GO): Al cargar FFH en este registro el circuito integrado comienza la cuenta para actualizar la hora.
- INTERRUPT CONTROL REGISTER (C_INTC): En este registro se define cada cuanto tiempo se desea que el circuito integrado interrumpa. Al cargar la constante 08H se programa al RTC para que interrumpa cada minuto.
- INTERRUPT STATUS REGISTER (C_INTS): Este registro debe ser leído cada vez que se realicen lecturas de los registros de datos o al final de una lectura sucesiva de contadores.

Luego se cargan los registros de datos con la hora 12:00:00 AM y con la fecha 1-1-94.

Si se desea cambiar la hora o fecha se debe hacer por programación a través del puerto serial.

• Programación de las interrupciones

Con los registros IE y IP se controlan las interrupciones. En el registro IE se habilitan las interrupciones y en el registro IP se manejan las prioridades (Ver Apéndice 5 para mayor información).

Se habilitan las siguientes interrupciones:

- EXTERNA O: Por donde interrumpe el RTC cada minuto.
- "Timer" O: El cual interrumpe cada 50 ms generando la base de tiempo.
- "Timer" 1: El cual genera la frecuencia de los impulsos de cómputo.

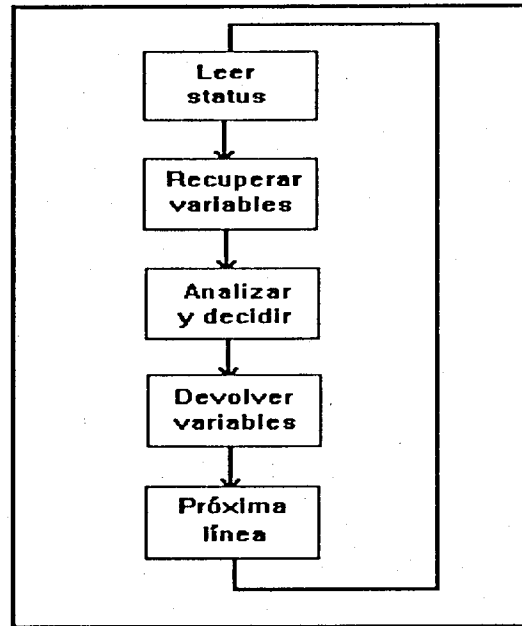
El "Timer" 1 es la interrupción de mayor prioridad ya que no se debe permitir distorsiones en la frecuencia de los impulsos de cómputo.

4. Módulo "Principal"

Dado que se tienen que manejar 24 líneas se instrumentó un sistema que se podría sintetizar en tres palabras: recuperar, analizar y guardar.

Cada línea tiene un conjunto de variables, que se encuentran en páginas contiguas de 32 bytes de la RAM, las cuales se recuperan y se guardan en las variables globales de análisis.

Con las variables recuperadas se hace un estudio del estado de la línea en base al nuevo status leído y a la información que ya se tenía de la línea. Por último se guardan de nuevo las variables en RAM, en la página correspondiente a la línea que se está analizando. En la Figura 42 se observa las acciones realizadas en el programa principal.



CAMBIOS
OCURRIDOS

Analizar

Fig. 42 Secuencia de acciones realizadas por el programa principal

- "polling" del status de las líneas urbanas

En el programa principal se realiza una lectura secuencial de los status de las líneas. La rutina que realiza este trabajo es LEO_STAT, la cual ubica la línea a leer tomando como base la dirección del status de la primera línea (STAT_01) y los registros R2 y R3.

- Detección de cambios en el status de las líneas urbanas

Primeramente se realiza un EXOR del status leído con el status anterior de la línea. Si el EXOR da cero quiere decir que no hubo cambio y entonces se prosigue con la siguiente línea. Si el EXOR da distinto de cero quiere decir que hubo algún cambio en la línea, por lo que hay que averiguar cual fué el cambio ocurrido.

La rutina "CAMBIO" que se encuentra en el módulo "Principal", realiza la función de detectar que tipo de transiciones ocurrieron en el status. La rutina tiene cuatro variables de salida:

- CAM_DES (Cambió descolgado ?)
- CAM_REP (Cambió repique ?)
- CAM_RBK (Cambió tono de libre ?)
- CAM_DV (Cambió dígito válido ?)

Al salir de la rutina, estas variables pueden tener uno de los 3 valores siguientes según el cambio ocurrido en el bit correspondiente:

- "0" si no hubo cambio
- "1" si hubo una transición de 0 a 1
- "2" si hubo una transición de 1 a 0

Cuando se detecta la contestación del abonado llamado la bandera impulso ubicada en la variable "FLAGS" de cada línea se pone en 1. De esta manera, cada vez que ocurra una interrupción del "Timer" 0 se actualizará la cadencia de los impulsos de cómputo de esa línea.

• Análisis de cambios en la línea urbana

El análisis de los cambios de la línea urbana se realiza mediante la rutina ANALIZO que se encuentra en el módulo "Principal". Su funcionamiento se basa en el análisis de la variable "FLAGS" de cada línea y las llamadas a la rutina CAMBIO?. En la Figura 43 se ilustra la secuencia de acciones que realiza la rutina ANALIZO.

Poner entre parentesis (#X)

Como lo llamo en todos lados lo llamo antes.

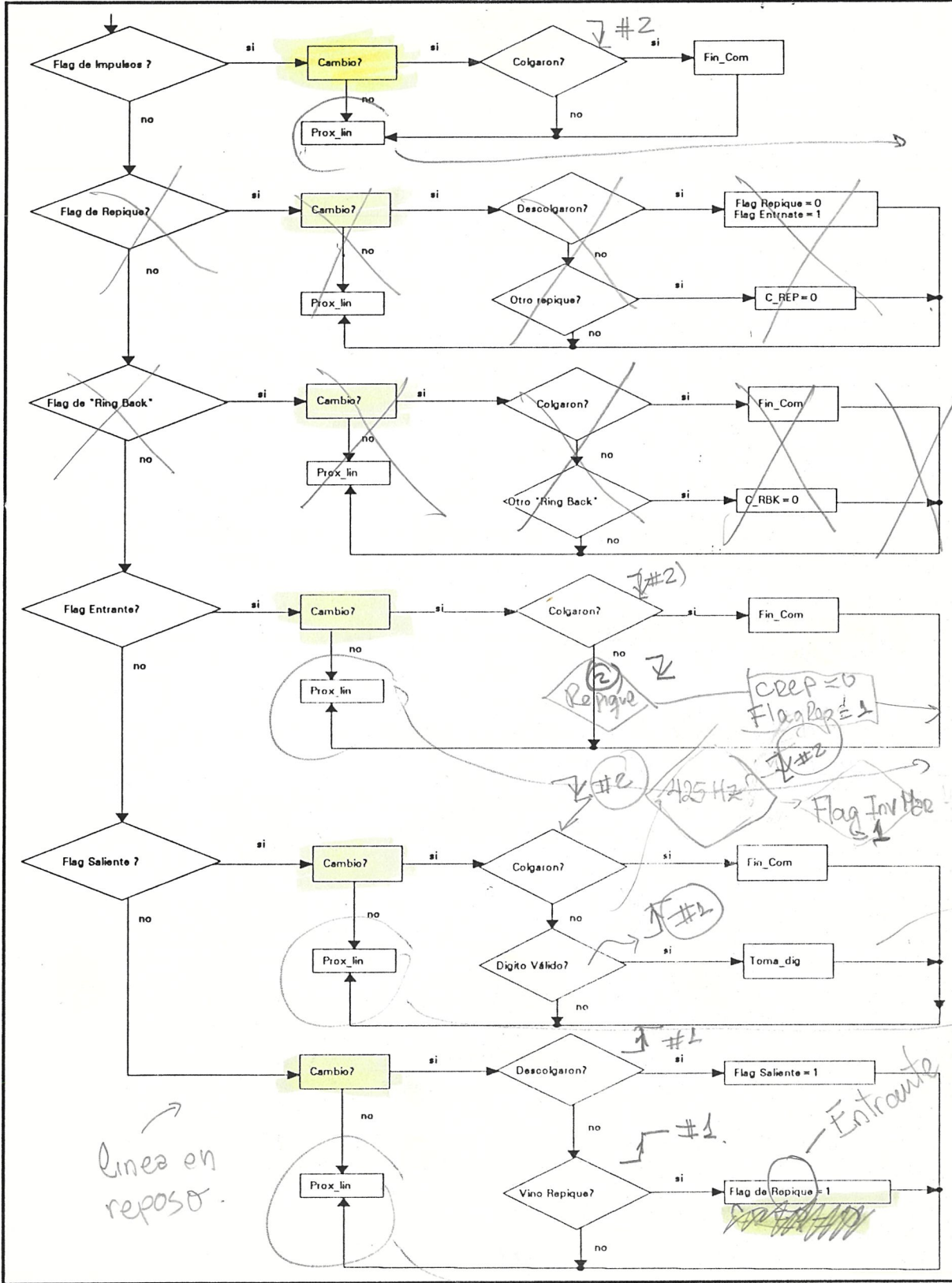


Fig.43 Diagrama de Flujo de la rutina ANALIZO.

5. Módulo "Tiempo"

•Actualización de la hora

Las interrupciones que genera el circuito integrado Reloj Calendario se manejan en la rutina de atención a la interrupción externa 0 (EXT_0). Cada minuto al producirse una interrupción se actualizan los valores de hora y fecha y se muestra en pantalla LCD a través de la llamada a la subrutina DI_HRFC los valores actualizados de hora y fecha.

Antes de salir de la rutina es preciso leer el bit de Status y el registro de Status de las interrupciones para que el circuito integrado Reloj Calendario funcione adecuadamente.

•Base de tiempo

Los eventos que involucran temporización están dominados por la base de tiempo que genera las interrupciones del "Timer" 0, el cual está programado para que interrumpa cada 50 ms. De esta manera cualquier otro tiempo que se desee lograr se obtiene como un múltiplo de 50 ms. En la Tabla 8 se observa el número de veces que se tiene que contar 50 ms para lograr alguno de los tiempos más importantes para el control del equipo.

CONSTANTE (d)	TIEMPO LOGRADO
3	150 ms
10	500 ms
20	1 seg
100	5 seg

Tabla 8 Constantes de conteo de 50 ms.

•Generación de la frecuencia de los impulsos de cómputo

La rutina de atención para las interrupciones del temporizador1 es TIM_1. En esta rutina se genera la frecuencia de los impulsos de cómputo. El temporizador 1 ha sido programado

para que interrumpa cada período medio ($T/2$) de la frecuencia a generar. Cada vez que el temporizador interrumpe se recarga la constante de conteo en los registros TL1 y TH1 y se complementa el bit 3 del puerto 1, con lo que se obtiene por el pin P1.3 una onda cuadrada de la frecuencia de interés.

• Generación de la Cadencia de los impulsos de cómputo

La cadencia de los impulsos de cómputo es: 150 ms para presencia de señal y tiempo variable de 2 a 90 segundos en silencio, tal como se observa en la Figura 44

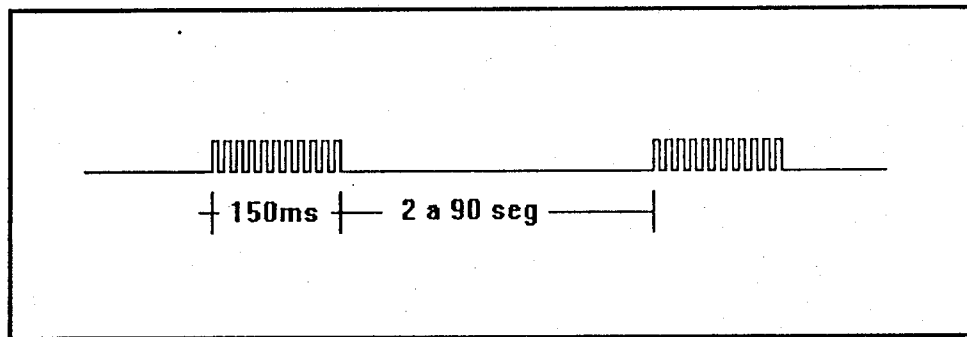


Fig. 44 Forma de los impulsos de cómputo.

Para generar el ritmo con que se aplican los impulsos se utiliza la base de tiempo generada por el temporizador 0. Dado que el tiempo entre impulsos no es constante se requiere una variable por cada línea que indique cual es el tiempo en que la señal debe permanecer en cero voltios.

Esta variable se llama ESPERA y es de 2 bytes (ESPH y ESPL). El valor que ella contiene es el número de veces que deben pasar 50 ms para llegar al tiempo deseado de silencio. La Tabla 9 muestra los valores que debe tener "ESP" para lograr los tiempos de separación entre impulsos que usa la CANTV. Es de hacer notar que todos los valores tienen un incremento de 3 unidades que son debidas a los 150 ms que dura el impulso.

TIEMPO (seg)	CONSTANTE DE RECARGA DEL "Timer" 0 (Variable Esp)
2	43d = 002Bh
2.5	53d = 0035h
3	63d = 003Fh
4	83d = 0053h
5	103d = 0067h
7.5	153d = 0099h
60	1203d = 04B0h
90	1803d = 0708h

Tabla 9 Constantes de recarga del temporizador 0

El ciclo de la cadencia se genera con 2 preguntas básicas: el contador (variable C50MSC) llegó a 3 ?, en cuyo caso se llegó al final del tiempo de presencia de impulso y se debe poner el bit en cero y una segunda pregunta:

El contador (variable C50MSC) llegó al valor prefijado en la variable "ESP"? Si es este el caso se ha llegado al fin del ciclo y se debe poner el bit de nuevo en 1, si no se continúa en la fase de silencio de la cadencia.

La Figura 45 detalla el algoritmo seguido para obtener la cadencia, en el cual se puede notar que dado que el contador para la cadencia es un variable de 2 bytes se hace necesario hacer dos comparaciones por cada pregunta de estado del contador. Esto es, primero se pregunta por la parte alta (C50MSCH) y luego por la parte baja (C50MSCL).

La rutina que realiza el proceso de generar la cadencia se llama "Ventana" (Cadencia) y se encuentra en el módulo TIEMPO del código fuente del programa en lenguaje assembler (Ver Apéndice 2).

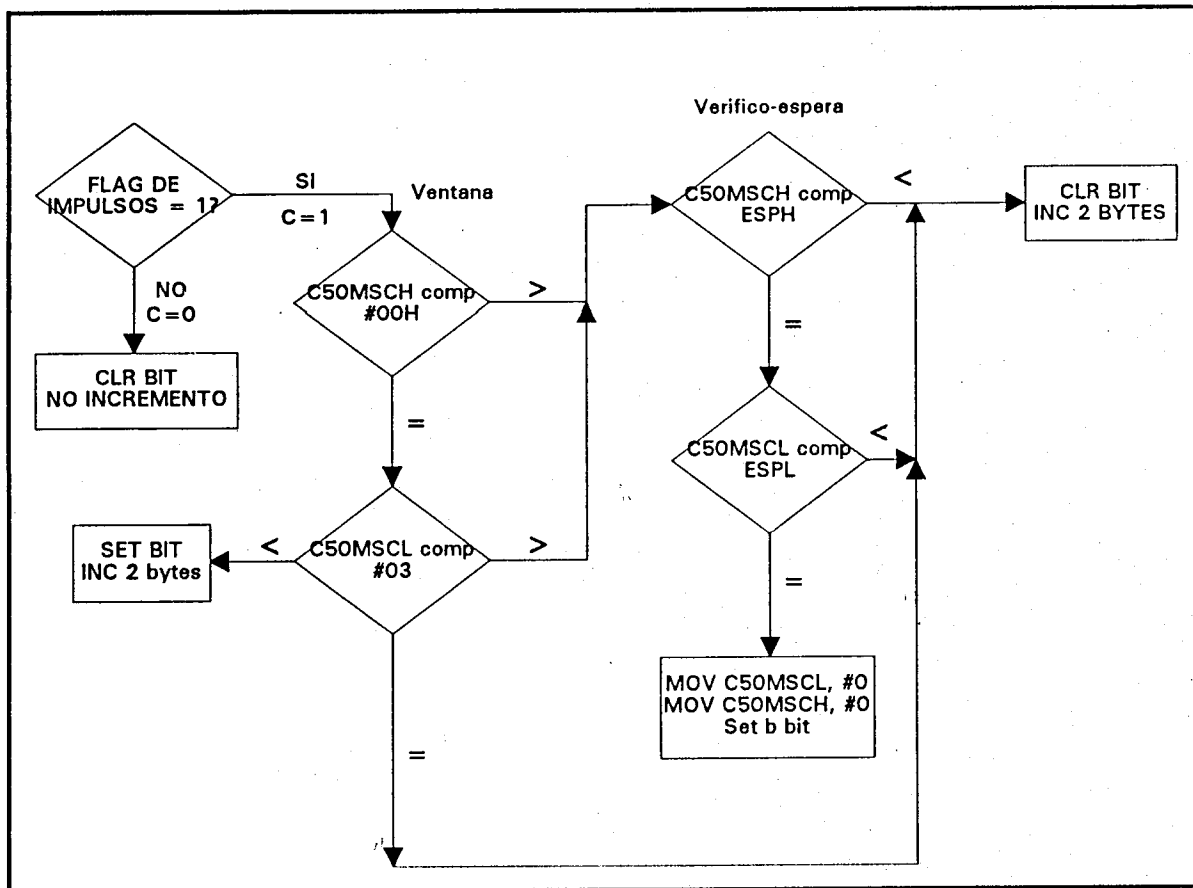


Fig. 45 Diagrama de flujo del algoritmo "Ventana"

Esta rutina contruye por cada línea, una señal que se usa para controlar el relé que aplica los impulsos a la línea telefónica. Las señales de cadencia se encuentran físicamente a la salida de 3 registros 74377, es por ello que la salida de la rutina la constituyen 3 variables de 8 bits, representando cada variable uno de los "registros de cadencia".

6. Módulo "Display"

• Rutinas de interfaz con la pantalla LCD

El control del funcionamiento de la pantalla LCD se realiza con tres rutinas. Estas son: BUSY, INST, DATOS y DISPKBCD. A continuación se describe brevemente qué hace cada una de ellas:

- BUSY: La pantalla LCD tiene un tiempo de operación que debe ser tomado en cuenta para la programación. Cuando una instrucción se está ejecutando internamente, se enciende la

bandera de ocupado o "busy flag" del HD44780 y la única operación que se puede hacer sobre la pantalla LCD es la lectura de la bandera de ocupado. Por esta razón, antes de mandar una instrucción o un dato desde el microprocesador es preciso realizar una lectura de la bandera de ocupado para asegurarse de que esté en cero. En esta rutina el microprocesador se queda en un lazo esperando que la pantalla LCD procese la información.

- **INST:** Esta rutina escribe en la dirección `IR_LCD` el código de la instrucción que se le mande en el acumulador. `IR_LCD` es la dirección de memoria en la cual se encuentra "mapeado" el registro IR (Instruction Register) del HD44780. Igualmente hace falta llamar a la rutina `BUSY` antes de mandarle otra instrucción a la pantalla LCD.

- **DATOS:** Esta rutina presenta en pantalla el carácter ASCII que se le mande en el acumulador. La rutina lo que hace es escribir el parámetro recibido en la dirección `DR_LCD`, la cual es la dirección en la que se encuentra "mapeado" el registro DR (Data register) del HD44780. Antes de salir de la rutina se llama a `BUSY` para darle tiempo a la pantalla LCD de responder adecuadamente.

- **DISPKBCD:** Esta rutina muestra en la pantalla LCD el contenido de una variable que está en formato "Packed BCD". En este formato cada "nibble del byte" representa un dígito (ver Figura 46). Algunas de las variables usadas en el programa están en este formato, ejemplo de ello son las variables de hora y fecha actualizada.

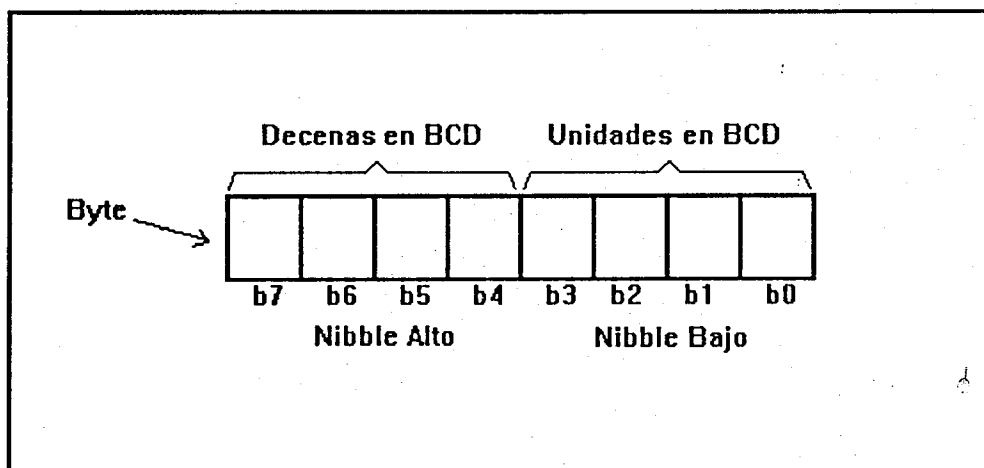


Fig. 46 Variable en formato packed BCD.

La rutina DISPKBCD extrae el nibble alto le suma 30H para convertirlo en ASCII y llama a la rutina DATOS para presentar las decenas en la pantalla LCD. El mismo procedimiento se sigue para el "nibble" bajo y la rutina termina.

El tener las variables en formato "Packed BCD" presenta dos ventajas principales: la primera es que hay un ahorro de espacio en memoria y la segunda es que simplifica la labor de presentación de los números en la pantalla LCD, ya que un número en notación binaria no puede ser presentado directamente en la pantalla LCD, debido a que para esto es preciso saber las cifras que conforman al número en decimal.

• Rutinas de programación de la pantalla LCD

Las rutinas que realizan la programación de la pantalla LCD son: PDISA, PDISB, PDISC y PDISD. Cada una de estas rutinas programa a la pantalla LCD de un modo diferente, que varía según la posición inicial del cursor y la forma en que se mueve el cursor al escribir un carácter. En la Tabla 10 se resumen los cuatro modos de programación de la pantalla LCD.

Rutina	Posición inicial del cursor	Movimiento del cursor al escribir un carácter
PDISA	Esquina superior izquierda	Avanza una posición a la derecha
PDISB	Esquina inferior izquierda	Avanza una posición a la derecha
PDISC	Esquina superior derecha	Avanza una posición a la izquierda
PDISD	Esquina inferior derecha	Avanza una posición a la izquierda

Tabla 10 Modos de programación de la pantalla LCD.

Todas las rutinas usan los comandos de programación que entiende el HD44780. Estos comandos son palabras de un byte que se cargan en el "Instruction Register" (IR) a través de la llamada a la rutina INST.

•Hora y fecha en la pantalla LCD

La rutina DIHR_FC se encarga de mostrar en la pantalla LCD la hora y la fecha en la pantalla LCD en el formato que muestra la Figura 47.

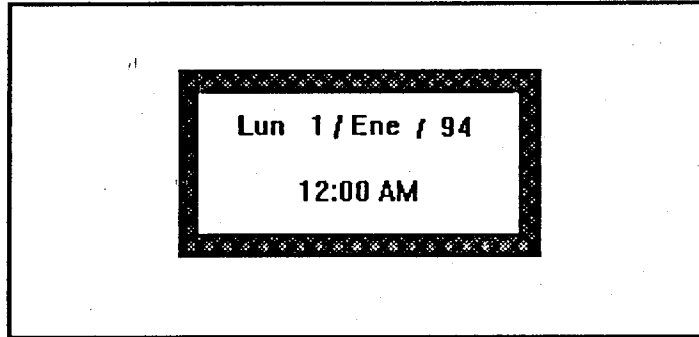


Fig. 47 Formato de la hora y fecha en la pantalla LCD

La rutina DIHR_FC llama a DISFECHA y DISHORA. La subrutina DISFECHA presenta la fecha actual en la primera línea de la pantalla LCD mientras que la subrutina DISHORA muestra la hora en la segunda línea de la pantalla LCD.

Para mostrar la fecha, se llama primeramente a la subrutina DISDIA, la cual muestra en el "display" el día de la semana en una abreviatura de 3 letras. El número que indica el día de la semana (1,2,3,...,7) sirve de índice para buscar en la tabla de nombres de día de la semana el nombre correspondiente. La rutina DISMES de igual forma toma el mes que contenga la variable MES_ACT y lo muestra en la pantalla LCD como una abreviatura de tres letras, tomando los valores que se encuentran en la tabla de nombre de los meses.

La rutina DISHORA inicialmente hace una lectura de las variables HORACT y MINACT y luego llama a la rutina DISPKBCD para mostrar a los dígitos de la hora actual en la pantalla LCD.

El circuito integrado reloj calendario coloca la hora en formato militar (0-23), y se desea presentar la hora en el formato normal (0-12). La rutina HORF12 realiza la conversión de formato realizando la siguiente operación:

$$\text{Hora}_{\text{Formato}(1-12)} = \text{Hora}_{\text{Formato}(0-23)} - 12$$

Además esta rutina determina si la hora es AM o PM, comparando la hora actual con 12. Luego en la rutina DISAM_PM, se presenta en el "display" "AM" o "PM" dependiendo del caso.

• Mensajes en la pantalla LCD

La rutina MENLCD muestra en la pantalla LCD un mensaje de 16 bits, de alguno de los que se encuentran en las tabla de mensajes. La rutina se le debe pasar como parámetro en el acumulador el índice del mensaje y en R2 la longitud del mensaje.

7. Módulo "Serial"

Los registros receptor y transmisor del puerto serie se accede por único registro que se denomina SBUF, que se encuentra entre los registros especiales de la RAM interna en la posición 99H. Escribiendo en el SBUF se carga el byte a transmitir y leyendo del SBUF se accede al byte recibido.

El puerto serie funciona por interrupciones. Al recibir un dato válido por el pin RxD, el "flag" de interrupción de la recepción (RI) se activa por "hardware" causando una interrupción. Este "flag" debe ser desactivado por "software". De igual manera cuando se finaliza la transmisión de un dato desde el 8031 al exterior, el "flag" de interrupción de la transmisión (TI) se activa por "hardware" y debe ser borrado igualmente por "software".

En el módulo serial se encuentran las rutinas que guardan la información concerniente a la configuración del equipo en las tablas apropiadas de la memoria RAM, cuando llega la programación hecha a través del programa bajo WINDOWS.

En la Figura 48 se muestra el formato de transmisión del PC al 8031. En esta figura se muestran las longitudes máximas que pueden tener cada una de las partes de la información. Cada tabla finaliza con FFh.

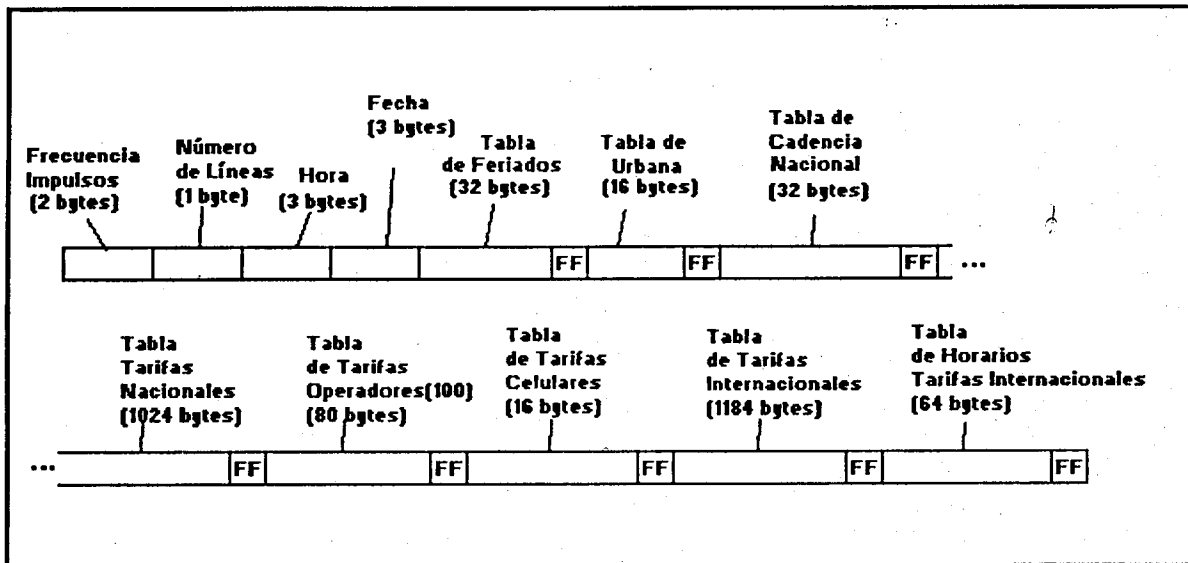


Fig. 48 Formato de Transmisión serial.

8. Módulo "Aritmética"

En el módulo "Aritmética" las rutinas que logran ubicar adecuadamente los datos en RAM., así como las rutinas de búsqueda en las tablas de tarifas de la CANTV.

B. Programa en BorlandC++ para interfaz con el usuario

1. Descripción del programa

El programa bajo Windows tiene por objeto servir de interfaz con el operador que vaya a configurar el equipo montado y además sirve como medio para recuperar los datos del equipo.

Los datos configurables por el programa son:

- Frecuencia de los impulsos de cómputo.
- Hora.
- Fecha.
- Tablas de Tarifas en Venezuela.

Para comenzar la configuración del EIC se debe abrir primeramente un archivo que contenga las tablas de tarificación de la CANTV actualizadas. Al seleccionar el menú Archivo y luego Abrir aparece la ventana que se observa en la Figura 49. En esta ventana se selecciona el archivo que contiene la Tabla de datos de configuración, y al hacer esto se habilita el menú Configurar que hasta los momentos estaba deshabilitado.

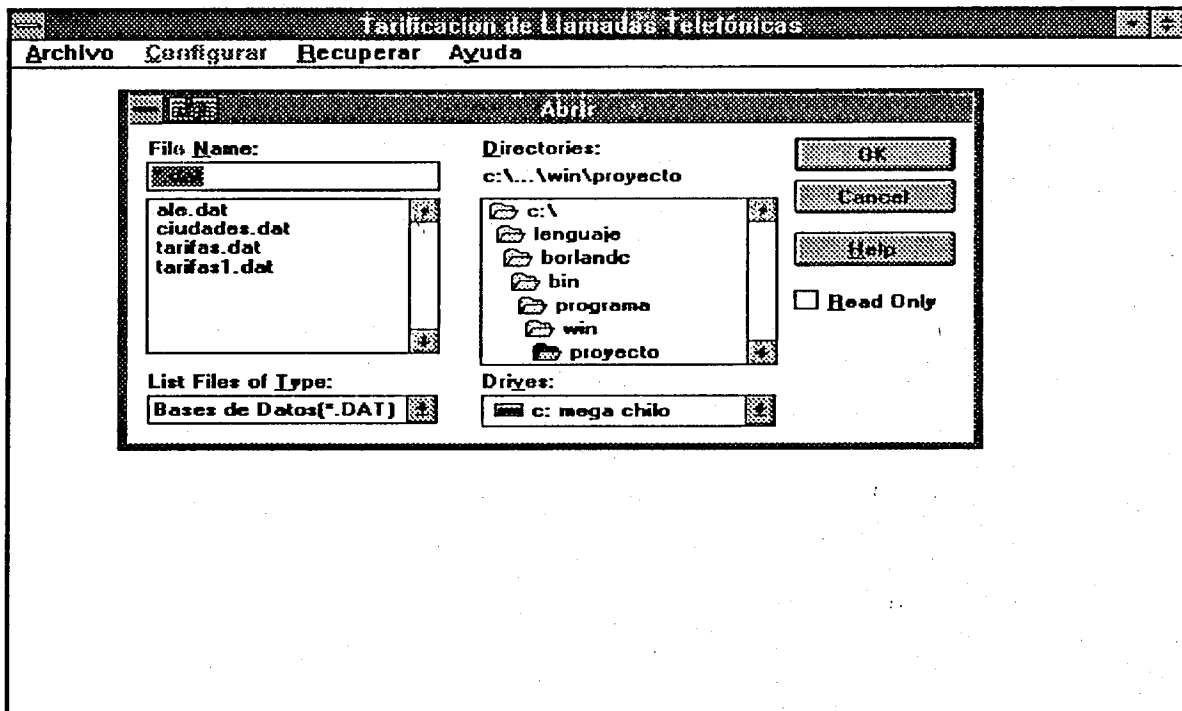


Fig. 49 Ventana para abrir archivos.

La Figura 50 muestra la ventana de configuración del equipo. En ella se observan los parámetros configurables: Frecuencia de los impulsos de cómputo, Número de líneas, Hora y Fecha y los botones para desplegar las ventanas que contienen las tablas de tarificación de la CANTV. Al seleccionar el botón de Ok el programa mandará por el puerto serial todos los parámetros configurados.

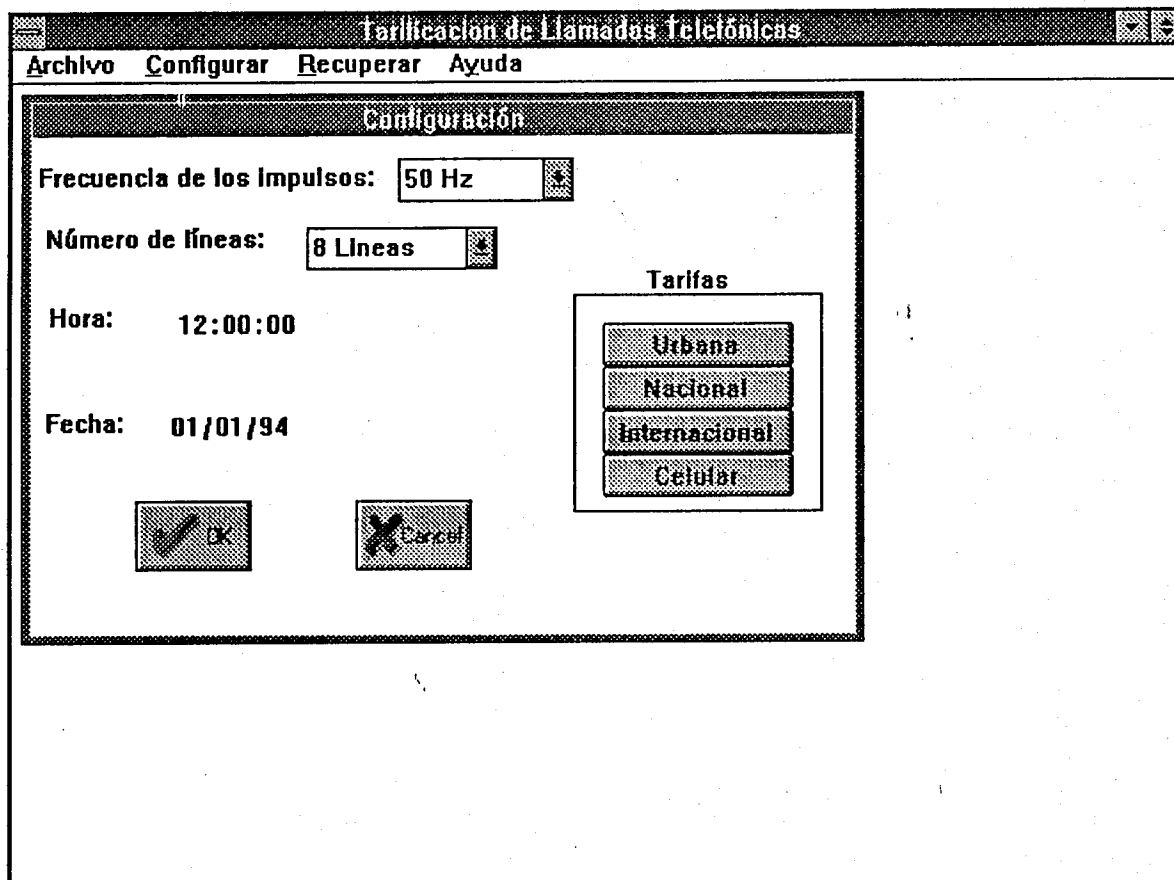


Fig. 50 Ventana de configuración del equipo.

Las tablas de tarificación de la CANTV, se muestran al apretar los botones de tarifas. Las tablas de tarifas están divididas en:

- Urbanas.
- Nacional
 - Automático.
 - Manual.
- Internacional
 - Automático.
 - Manual.

Por ejemplo, al apretar el botón de tarifas nacionales, se le pregunta al usuario si desea ver las tarifas por servicio manual o las tarifas por servicio automático a nivel nacional. En la Figura 51, se muestra la tabla de tarifas por el servicio automático o Discado Directo Nacional. En la Tabla que se muestra se tiene el código de área y el código de tarifa de las 36 ciudades más importantes del país.

The screenshot shows a software window titled "Tarificación de Llamadas Telefónicas" with a menu bar containing "Archivo", "Configurar", "Recuperar", and "Ayuda". The main area is titled "Configuración" and includes a field for "Frecuencia de los impulsos: 10 Hz". A sub-window titled "Tarifas Nacionales" is open, displaying a table of national rates. The table has three columns: "Ciudad", "Codigo de Area", and "Codigo de Tarifa". Below the table are input fields for "Costo Impulso" (Residencial and No Residencial) and a "Codigo de Tarifa..." button. The background window shows fields for "Número", "Hora", and "Fecha".

Ciudad	Codigo de Area	Codigo de Tarifa
Acarigua	55	4
Anaco	82	4
Barcelona	81	4
Barinas	73	5
Barquisimeto	51	4
Cabimas	64	5
Carora	52	5
Carúpano	94	5
Ciudad Bolívar	85	5
Coro	68	5
Cumaná	93	4

Costo Impulso:
 Residencial: [] Bs
 No Residencial: 1,56 Bs

Codigo de Tarifa...

Fig. 51 Ventana para que muestra la Tabla de Tarifas a nivel Nacional.

Si se decide ver la tabla de tarifas nacionales por el servicio a través de operadores se desplegará la ventana que se ilustra en la Figura 52. En esta Figura se puede observar además, la ventana de información acerca del tipo de Tarifa. Esta información no se puede modificar.

Tarifación de Llamadas Telefónicas

Archivo Configurar Recuperar Ayuda

Configuración

Llamadas Nacionales a través de Operadora (100)

Cadenencia : Codigo de Tarifa	Tarifa Normal	Tarifa Especial
	Información	Conexion minutos (Bs) Minuto Adicional (Bs)
1	Tarifa Normal: Se aplica los días laborales entre 7.00 am y 7.00 pm Tarifa Especial: Se aplica en los días laborales de 7.00 pm a 7.00 am, los días sábados y domingos, y los días feriados las 24 horas del día	8,5 9,5
2		4,25 14,75
3		7,0 19,0
4		8,75 26,25
5		8,25 32,75
6		10,2 34,75

Fig. 52 Ventana del servicio de telefonía a través de operadores a nivel nacional.

El programa tiene la facilidad de recuperar los datos del registro de llamadas telefónicas que realiza el prototipo montado, cuando está configurado para trabajar con una sola línea.

Para efectuar esta acción se debe seleccionar el menú **Recuperar** y el programa se comunicará serialmente con el equipo montado para leer la base de datos de las llamadas. Aparecerá una ventana como la que aparece en la Figura 54.

Al apretar el botón de **Estadísticas**, aparecerán unas ventanas con gráficos que ilustran el número y el tipo de llamadas realizadas, según la base de datos.

Hora	Fecha	Duración	Costo	Numero Destino
10:10:20	07-03-94	00:11:21	7,92	327724
10:15:35	07-03-94	00:03:13	3,12	2837627

Fig. 54 Ventana de la Base de datos recuperada.

2. Estructura del código

El programa está dividido en bloques de código para facilitar la labor de programación y para poder haer modificaciones a futuro más facilmente. El código del programa se encuentra en el Apéndice 14. A continuación se hace una descripción breve de los distintos módulos:

- **Inicialización (Proy_ini.c):** En este módulo se crea la clase de las ventanas, y se muestra la ventan principal.
- **Definición (Proyecto.def):** Se define el tipo de código, el tipo de data, tamaño del "heap", tamaño del stack, etc.
- **"Dialog Box" (Proy_dia.c):** Este módulo contiene las rutinas de manejo de mensajes de todos los "Dialog Box". Por cada una de ellas hay una subrutina que maneja los mensajes, mediante un "switch".

- **Principal (Proyecto.c):** En este módulo se encuentra "WinMain", en el cual se hace la inicialización de la aplicación, de la instancia y se entra en el lazo de manejo del ciclo de mensajes.

- **Mensajes (Proy_men.c):** En este módulo se interceptan los mensajes de la ventana principal

- **Recursos (Proyecto.rc):** Es un módulo que contiene la definición de los recursos (estilo, tamaño, tipo de letra, etc). Los recursos son: Los botones, los menu, los dialog box, etc.

- **Librería: (Proyecto.h):** Se definen los ID (identificadores) y se tienen los prototipos de las funciones.

- **Serial (Proy_ser.c):** Rutinas de manejo del puerto serial.

- **Estadísticas (Proy_etd.c):** Rutinas para análisis de la base de datos emitida por el EIC.

CAPITULO 5 TARIFICADOR TELEFÓNICO CASERO (TTC)

A.Descripción General

El Tarificador Telefónico Casero (TTC) fué desarrollado como un apartado adicional del proyecto de pasantía. La gran similitud de funciones con el EIC permitió realizar un prototipo en corto tiempo y con poco esfuerzo adicional.

El objeto del Tarificador es llevar el registro de las llamadas que se hacen por un teléfono convencional y de proveer una salida visual del costo de las llamadas en el momento en que estas se ejecutan.

Con este equipo, todas las llamadas realizadas en un período de tiempo determinado pueden ser analizadas y comparadas por las facturas emitidas por la compañía de teléfonos. Además, se pueden elaborar reportes gráficos que ilustren por ejemplo, el número de llamadas realizadas cada día del mes, o un gráfico en el que se observe el tipo de llamadas realizadas, es decir, si las llamadas fueron locales, nacionales, internacionales o de alguna otra categoría.

El TTC se conecta al par trenzado de un teléfono para poder realizar la supervisión de las llamadas. Su interfase con el exterior está constituida por dos partes: la pantalla y el puerto serial. A través de la pantalla se muestra el costo de la llamada, los dígitos marcados y la hora y fecha, mientras que por medio del puerto serial, se realiza la conexión con un computador con el objeto de analizar la base de datos almacenada por el equipo. En la Figura 50 se observa la apariencia física y la forma de conexión del TTC.

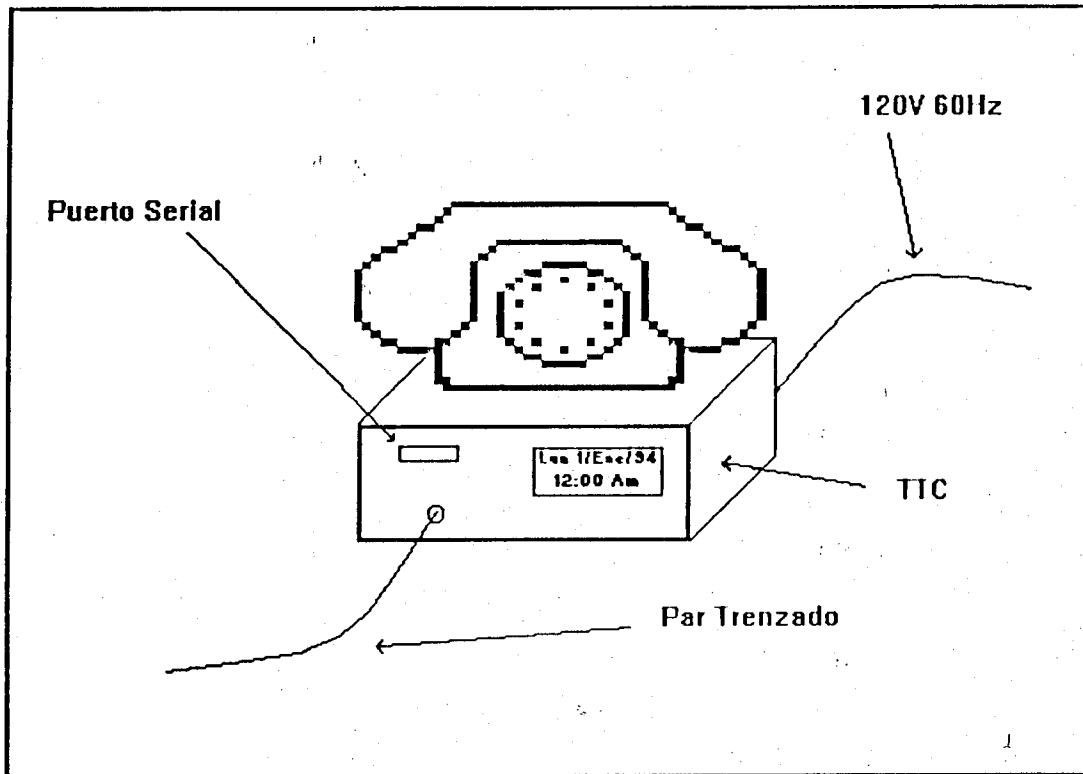


Fig. 50 Ubicación física del "Tarificador Telefónico Casero" (TTC)

El "hardware" y "software" necesarios para la implementación de este equipo, es esencia el mismo que usa el EIC. Sin embargo, se tienen pequeñas diferencias que son descritas en las próximas secciones.

B. "Hardware"

El "hardware" de control del TTC es exactamente el mismo utilizado por el Emulador de Impulsos de Cómputo (EIC). Esto es, se usa el microprocesador 8031, la EPROM 27256, la RAM 62256 y el Reloj Calendario MM58167A. Las interfases con el exterior son de igual forma las mismas utilizadas en el EIC, es decir, un puerto serial y una pantalla LCD, aunque en este caso se usan para propósitos diferentes.

Una diferencia importante con respecto al EIC, es que el Tarificador Telefónico Casero sólo debe atender una línea mientras que el EIC debe atender de 8 a 24 líneas urbanas. Entonces, sólo es necesario un conjunto de circuitos detectores de estados, lo cual reduce considerablemente el tamaño del prototipo.

Por otro lado, al ser sólo una línea la que se debe supervisar, también se eliminan los decodificadores necesarios para el mapa de memoria en memoria de los registros de estatus de las líneas urbanas.

Otra diferencia es que no hace falta acoplar los impulsos de cómputo a la línea telefónica, ya que en este caso no existe central privada que reciba estos impulsos. De esta manera, la generación de los impulsos se hace internamente y se eliminan el relé y el "driver" necesarios para aplicar los impulsos.

El "hardware" del Tarificador Telefónico Casero se puede dividir en 4 bloques fundamentales: Controlador, Interfase visual, Interfase serial y circuitos detectores (Ver Figura 51).

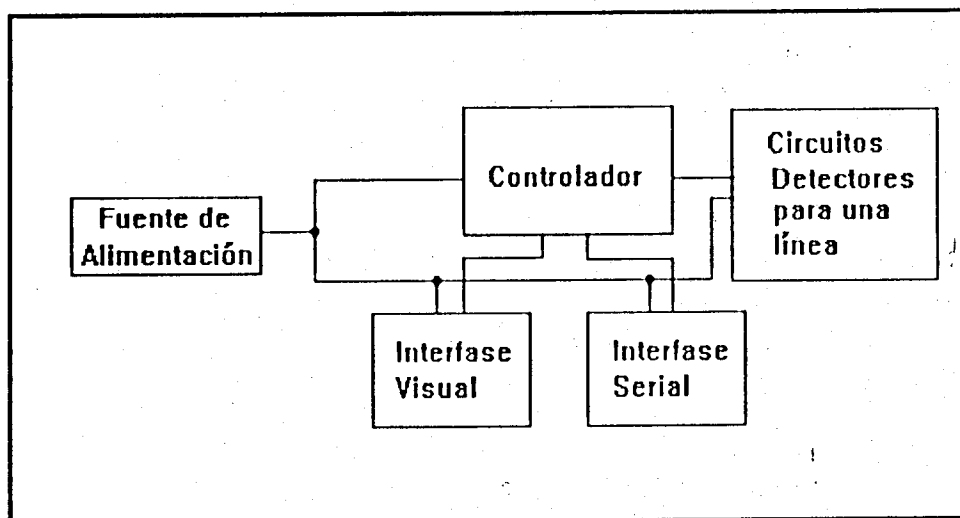


Fig. 51 Diagrama en bloques del TTC.

C. "Software"

El programa que controla al TTC, sólo debe supervisar el estado de una línea, por lo que resulta un programa más simple que el programa que controla al EIC. Así, el programa es totalmente secuencial y el microprocesador tiene tiempo para realizar otro tipo de tareas como

registro de llamadas, cálculo del costo de la llamada, actualización de la pantalla LCD, etc. En las próximas secciones se explican las rutinas adicionales desarrolladas.

1. Definición de variables

Para el Tarificador Telefónico Casero (TTC) se deben definir un serie de variables adicionales a las que ya se tenían para el EIC. A continuación se describen estas nuevas variables y la razón de su existencia.

• Variables asociadas a la duración de llamadas

- USEG, DSEG, UMIN, DMIN, UHOR y DHOR: En estas variables se guarda la duración de la llamada, la cual tiene una duración máxima de 99:59:59.

- C_50MS: Esta variable es un contador de 50 ms que se usa para actualizar la duración de un llamada. Cada vez que el contador llega a 20 ha transcurrido un segundo y se actualizan las variables de duración de la llamada.

• Variables asociadas al costo de las llamadas

- COSTO5, COSTO4, COSTO3 COSTO 2 Y COSTO1: En estas cinco variables se guarda el costo de la llamada.

- CT_IM3, CT_IM2 y CT_IM1: En estas tres variables se guarda el costo del impulso de cómputo.

- NM_IM2 y NM_IM1: En estas variables se guarda el número de impulsos de cómputo recibidos.

- DIS_MN, DIS_KL, DIS_IJ, DIS_GH, DIS_EF, DIS_CD y DIS_AB: En estas siete variables se guarda el costo de la llamada en el formato de 14 dígitos BCD. Estas variables se usan para presentar el costo de la llamada en el "display".

2. Programa principal

El programa principal del TTC es el mismo que el del EIC con la diferencia de que la variable NO_LIN es igual a 1.

La diferencia es de este programa con respecto al del EIC es que se llaman a las rutinas de cálculo de duración de una llamada y cálculo de costo de una llamada entre otras subrutinas

3. Rutinas adicionales de cálculo

Para lograr la implementación efectiva del TTC es necesario desarrollar un conjunto de rutinas adicionales. A continuación se describen las rutinas adicionales desarrolladas para el Tarificador Telefónico Casero TTC.

• Calcular el costo de una llamada

Para calcular el costo de una llamada se debe multiplicar el costo del impulso por el número de impulsos. El costo del impulso es un número con 2 decimales mientras que el número de impulsos recibidos es un número entero. Por ejemplo, para llamadas urbanas el costo, desde teléfonos residenciales, del impulso es de 1,18 Bs. y el número de impulsos se incrementa una unidad cada 90 seg. Así, para una llamada que dure 9 minutos, se mandarían 6 impulsos y el costo de la llamada que calcularía el microprocesador se haría con la siguiente cuenta:

$$118 \times 6 = 708$$

La cuenta se realiza sin tomar en consideración los decimales. Luego al presentar el resultado se corre la coma hacia la izquierda dos espacios. Por lo tanto para el ejemplo anterior el costo de la llamada sería de 7,08 Bs.

Las variables que contienen el costo de los impulsos son: CT_IM3, CT_IM2 y CT_IM1. Al ser una variable de 3 bytes, tenemos que el máximo valor que puede tomar la variable es FFFFFFFh, lo que significa que el mayor costo de impulso en bolívares podría ser 167,772.15 Bs. Esta es una cantidad más que suficiente tomando en cuenta los costos por impulsos que cobra la CANTV actualmente.

Las variables que contienen el número de impulsos recibidos son: NM_IM2, NM_IM1. Con dos bytes se fija un límite al máximo número de impulsos que se pueden recibir sin que el contador sobrepase el límite, el cual es de FFFFh = 65,535 impulsos. Si se toma el peor caso de emisión de impulsos que es un impulso cada segundo, tendríamos que la llamada más larga

que se tarificaría sin que el contador de impulsos diera la vuelta sería de aproximadamente 18 horas, el cual es un tiempo bastante exagerado para una llamada telefónica.

El costo de las llamadas se guarda en la memoria RAM interna en cinco variables, ellas son: COSTO1, COSTO2, COSTO3, COSTO4 y COSTO5. El número de variables necesarias para guardar el costo de la llamada viene dado por el máximo número que pueden tomar las variables NM_IM y CT_IM, esto es:

$$\begin{array}{rcccl}
 \underline{\text{FF FF FFh}} & \times & \underline{\text{FF FFh}} & = & \underline{\text{FF FE FF 00 01h}} \\
 \text{3 bytes} & & \text{2 bytes} & & \text{5 bytes} \\
 \text{CT_IM} & & \text{NM_IM} & & \text{COSTO}
 \end{array}$$

De esta manera el costo mas grande que puede tener una llamada sería de 10,994,947,850.25 Bs., lo cual es más que suficiente.

La rutina para calcular el costo de una llamada es MUL_AB. Esta rutina toma como parámetros a las variables CT_IM y NM_IM y da como resultado la variable COSTO. La rutina debe multiplicar un número de 3 bytes por otro de 2 bytes, pero el 8031 sólo multiplica un byte por un byte, a través del la instrucción:

```
MUL AB
```

De esta manera, para poder efectuar una multiplicación de 2 bytes por 3 bytes, hay que hacer un lazo para poder obtener el resultado. A continuación se muestra un ejemplo en donde se multiplica byte a byte.

AA	BB	CC	Costo del impulso (3 bytes)
x	EE	FF	Número de impulsos (2 bytes)
	CB	34	
	BA	45	
A9	56		
	BD	A8	
AD	DA		
9E	0C		
9F	64	A8	B8
		34	Costo de la llamada (5 bytes)

En la rutina MUL_AB, se hacen dos lazos uno dentro del otro para lograr hacer 6 repeticiones. Estos lazos repetitivos, se hacen con los registros R2 y R3. Los registros R0 y R1 se usan para direccionar indirectamente la zona de memoria en donde se encuentran las variables.

Una vez hecha la multiplicación de los dos bytes en cuestión se actualiza el resultado en la variable COSTO, tomando en cuenta el "carry" generado para la siguiente operación. La multiplicación se sigue realizando hasta lograr las seis repeticiones necesarias para lograr el resultado.

• Conversión de binario a BCD

Este algoritmo es desarrollado en el programa en lenguaje assembler en la rutina BINBCD (Ver Apéndice 3). El algoritmo de conversión de binario a BCD que se utilizó podría ser llamado desplazamiento *izquierda-suma 3*. A continuación se establecen los pasos fundamentales del algoritmo:

- 1) Colocar el número binario a convertir, *afuera y a la derecha* de un conjunto de celdas de 4 bits (Ver Figura 52), en la que se guardará el número BCD. Hacer un desplazamiento a la izquierda.

- 2) Sumar 3 (0011) al dígito formado *cuando este sea mayor que 4* (0100) en una celda. Luego de sumar 3 hacer un desplazamiento a la izquierda inmediatamente.

- 3) Repetir los pasos 2 y 3. Cuando todos los bits del número a convertir hayan sido introducidos en las celdas, el proceso de conversión concluye y los dígitos del número BCD están en las celdas.

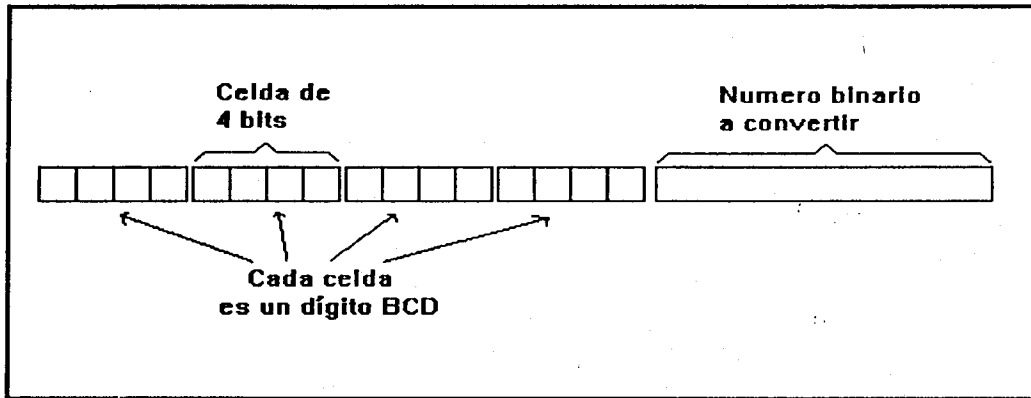


Fig. 52 Celdas de 4 bits para conversión binario-BCD

Los pasos que sigue el algoritmo de conversión de binario a BCD son ilustrados en la Figura 52, tomando como número a convertir el número $168d = 10101000d$

			10101000	= 168d
		1	0101000	Shift a la izquierda
		10	101000	Shift a la izquierda
		101	01000	Shift a la izquierda
		+11		Suma 3
		1000	01000	
		1 0000	1000	Shift a la izquierda
		10 0001	000	Shift a la izquierda
		100 0010	00	Shift a la izquierda
		1000 0100	0	Shift a la izquierda
		+11		Suma 3
		1011	0100 0	
		1 0110	1000	Shift a la izquierda
1	6	8		Resultado en BCD

Fig. 52 Algoritmo de Conversión Binario a BCD.

En el caso de este algoritmo se desea convertir los 5 bytes que conforman el costo de una llamada (COSTO1-COSTO5) a 14 dígitos en "packed BCD" (DIS-MN-DIS_AB). Esta conversión se logra repitiendo el algoritmo descrito anteriormente 40 veces.

Primeramente se realiza un desplazamiento o "shifteo" del número binario hacia las celdas BCD. Este proceso se realiza en la subrutina ROTAR. Luego se le suma 3 a todas las celdas BCD cuya cuenta sea mayor que 4, este proceso se hace en la subrutina SUM_3?. Este ciclo se repite hasta completar 40 repeticiones

Si no es un caso especial, sino se trata de un número del uno(1) al nueve(9), antes de llamar a la rutina DATOS se le suma 30h para realizar la conversión a ASCII.

- Duración y costo de una llamada en la pantalla LCD

La rutina DI_DUCOST se encarga de mostrar en la pantalla LCD la duración y el costo de la llamada según el formato de la Figura 53.

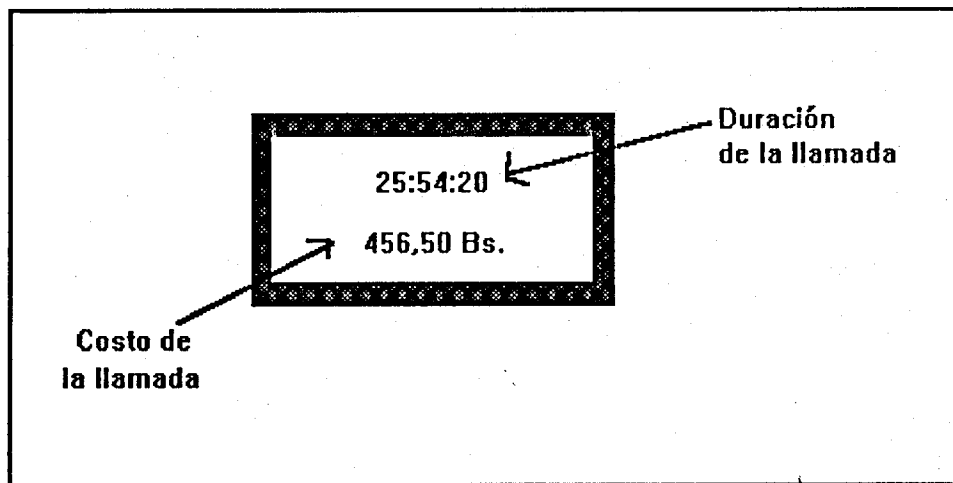


Fig. 53 Duración y costo de la llamada en la pantalla LCD.

Primeramente se programa la pantalla LCD para que el cursor comience en la esquina superior izquierda y para que se realice shift hacia la derecha con la escritura de cada carácter por medio de la rutina PDISA. Luego se muestra la duración de la llamada en la primera línea al llamar a la rutina DISDURC.

Seguidamente se programa la pantalla LCD en modo B por medio de la rutina PDISB y luego se llama a la rutina DISCOST para mostrar el costo de la llamada en la pantalla LCD. Antes de llamar a esta rutina el costo de la llamada ha sido convertido previamente a BCD y los dígitos del costo de la llamada se encuentran en las variables DIS_MN a DIS_AB. En la Figura 54 se observan los dígitos del costo de la llamada en las variables DIS_EF, DIS_CD y DIS_AB.

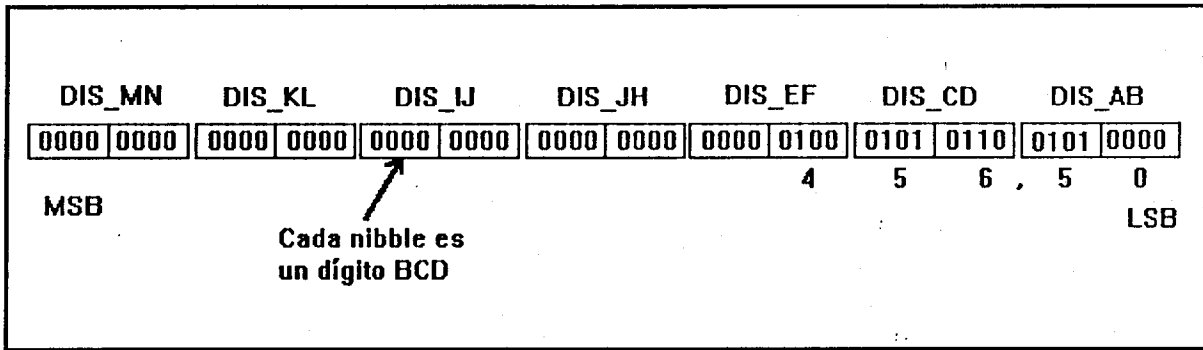


Fig. 54 Variables packed BCD para el costo de la llamada.

Antes de presentar los números en la pantalla LCD es preciso eliminar los ceros que se encuentran a la derecha del primer dígito válido del costo de la llamada. Esta acción la realiza la rutina QT_CERO.

Cuando se llega al dígito válido más significativo del costo de la llamada, se llama la subrutina BCD_DIS, la cual muestra la parte entera. Luego se escribe una coma en la pantalla LCD y se vuelve a llamar a la rutina BCD_DIS para mostrar los dos dígitos decimales.

Por último, se llama a la rutina DIS_BS para mostrar en la pantalla LCD la palabra BS.

3. Modularidad en "hardware"

El "hardware" tiene un diseño modular, ya que tiene una tarjeta controladora que es independiente de las tarjetas detectoras de estado, las cuales pueden supervisar, cada una, 8 líneas telefónicas.

El equipo en máxima capacidad puede atender a 24 líneas, las cuales pueden ser multifrecuenciales o decádicas. Sin embargo, se puede configurar al EIC, con dos tarjetas detectoras (para una capacidad de 16 líneas) o con una tarjeta detectora (para una capacidad de 8 líneas).

Si se tienen más de 24 líneas se debe instalar otro equipo emulador de impulsos de cómputo. O sea, el equipo puede ser ampliado o configurado según cada caso en particular

4. Modularidad en software

Se realizaron dos programas, uno en assembler que controla al EIC y otro en BorlandC++ para "Windows", a los fines de configurar el equipo por el puerto serial.

Ambos programas están divididos en bloques de código que resuelven tareas específicas. De esta forma, se podrán realizar modificaciones del programa, a futuro, sin afectar su estructura.

5. Tarificación según la CANTV

El EIC tiene almacenado en su memoria RAM las tablas de tarifas de la CANTV que se muestran en el Apéndice 6. Estas tablas aproximan, en buena medida, la forma en que la CANTV efectúa la tasación para los distintos tipos de llamada.

Sin embargo, para el caso de las llamadas de nivel nacional sólo se consideran las tarifas de las 36 ciudades más importantes del país, desde Caracas. En efecto, se han descartado ciudades pequeñas, como primera aproximación del prototipo, ya que entre las ciudades consideradas se cursa más del 80% del tráfico telefónico.

Ahora, dado que el EIC tiene un error de hasta 5,5 seg. en la detección de la contestación, la tarificación será imprecisa. Sin embargo, en la mayoría de los casos, no afectará el resultado global de la tasación. A continuación se citan algunos ejemplos:

- Supongamos una llamada local con una duración de 8 minutos y 45 segundos. La CANTV le enviaría 5 impulsos, lo cual es exactamente lo mismo que haría el EIC. En este caso, el error en la detección no afecta el resultado.

- En el caso de una llamada nacional con código de tarifa 6 (1 impulso cada 2 segundos), se tiene que mientras la CANTV ya habría mandado 2 impulsos probablemente, el EIC estaría aún esperando detectar la ausencia del tono de libre para comenzar a tarificar.

Llevando el error en impulsos a bolívares según la Tarifa no residencial para el período Septiembre-Diciembre de 1993, se tendría una diferencia de $2 \times 1,56Bs. = 3,12 Bs.$, en lo que cobraría la CANTV y el registro que efectuaría la central SIEMENS tarificada por el EIC.

Cabe destacar que, como la aplicación de impulsos por parte de la CANTV, se hace aplicando el principio de Karlsson, no se puede establecer con certeza el número de impulsos que lleva una llamada. Así pues, los errores que pueda tener el EIC se compensan en cierto grado con la imprecisión en la tarificación por parte de la CANTV.

- Para el caso de una llamada internacional que dure 1 minuto y 38 segundos por ejemplo, no habría ningún problema, ya que el EIC mandaría 1 impulso y la CANTV también.

Ahora para una llamada que lograra establecer la conexión pero que se "cayera" a los 3 segundos por ejemplo, sería cobrada por la CANTV como un llamada válida de duración igual a un minuto, cobrando el valor prefijado del impulso, mientras que el EIC no le daría tiempo de darse cuenta que hubo una llamada exitosa.

De los razonamientos anteriores, se puede concluir que el EIC tarifica en forma aproximada. Para conocer con mayor precisión las diferencias se sugiere montar y comparar dos líneas: una línea tarificada por la CANTV y otra por el EIC; y luego revisar al final de un período determinado el registro de llamadas hecho por cada parte, y así obtener mayores conclusiones acerca de la precisión del equipo diseñado.

Al respecto y en el transcurso de la pasantía, se hicieron gestiones para que la CANTV enviara impulsos a algunas de las líneas entrantes de SIEMENS, pero la respuesta nunca se hizo efectiva. Esta limitación de aplicar los impulsos de cómputo por parte de la CANTV es una razón más que justifica la fabricación de un equipo que emule dichos impulsos.

6. Capacidad de actualizar la hora y la fecha

El EIC actualiza la hora y la fecha a través del integrado MM58167A. De esta manera es posible aplicar las variaciones de tarifa, que aplica la CANTV dependiendo de la franja horaria y del día de la semana.

7. Tarificador para un teléfono casero

En este aspecto opcional de la pasantía, se realizó la supervisión de una línea y se proporcionó una salida visual del costo y duración de la llamada. Adicionalmente, se hizo un programa en BorlandC++ para Windows que complementa al equipo.

B. Conclusiones

- El Equipo Emulador de Impulsos de Cómputo (EIC) llegó a un grado de desarrollo intermedio, ya que se debería continuar con una etapa de pruebas de campo para solucionar posibles errores de diseño. Es decir, todavía no es factible producir en serie al equipo diseñado.

Igualmente para determinar la confiabilidad del equipo se deberán realizar pruebas de emisión de Impulsos de Cómputo, comparando los resultados que emite la facturación de la central privada con los que realiza la CANTV.

- Se deben hacer estudios para reducir los costos del equipo al máximo. En el Apéndice 10 se encuentran las tablas de costos de los equipos diseñados (EIC, TTC y prototipo montado).

- El punto crítico del diseño está en la detección de la contestación. A pesar de que el detector de la ausencia de tono de libre cumple con el objetivo, se debería reestudiar este punto y

proponer otras alternativas. En el capítulo 2 se discuten las alternativas estudiadas a este respecto.

- Con relación al Tarificador Telefónico Casero (TTC), se debería hacer un estudio de mercado para determinar la posible aceptación de un equipo como éste, antes de pensar en fabricarlo.

Por otro lado, se le deberían agregar opciones al programa bajo "Windows", de manera de hacerlo un producto más atractivo.

C. Recomendaciones

- Los módulos de tasación de las Centrales Privadas Siemens están pensados para el modo de cobrar las llamadas telefónicas en Alemania. En ese país, el costo de los impulsos es único y se varía el ritmo de aplicación de los mismos dependiendo del destino de la llamada. En cambio en Venezuela, las llamadas internacionales se cobran por un impulso cada minuto variando el costo del impulso.

Entonces se presenta el problema de que las centrales no permiten que el valor del impulso cambie tal como lo hace la CANTV, por lo que para llamadas internacionales habría que mandar una seguidilla de impulsos de forma de alcanzar el valor que se cobra en Venezuela.

Por ejemplo, para el caso de Estados Unidos habría que mandar $161Bs / 1,56Bs = 103,2 \approx 104$ impulsos. Esto implica que se debe mandar un impulso cada 550 ms. aproximadamente.

- Para las tarifas nacionales se presenta el problema de que si la ciudad origen no es Caracas, entonces los códigos de tarifa no aplican.

Se debería entonces completar el programa bajo WINDOWS y agregarle la capacidad de poder configurar la ciudad origen. Para hacer esto se debe crear una tabla de datos con los códigos de tarifa de cada una de las ciudades con cada una de las demás.

- Para el caso del Tarificador Telefónico Casero se podría realizar una aplicación para EXCEL que analice la base de datos que genera el EIC y presente reportes gráficos. En el Apéndice 12 se presenta el listado de una versión preliminar de esta aplicación.

- Para el caso del Emulador de Impulsos de Cómputo, si se desea hacer un circuito impreso, se debería diseñar una tarjeta que sea compatible con el backplane de las centrales SIEMENS. De manera de tomar las alimentaciones de ± 5 Voltios de la fuente de la central. Además de esta forma se podría lograr mayor modularidad a la hora de configurar los equipos.

- El acople con las líneas debería ser único para las señales analógicas y digitales. Una posible solución sería el uso de un optoacoplador en conjunto con un operacional, por el que se pudiesen obtener los dos tipos de señales a la vez. De esta forma, se eliminaría el uso de transformadores lo que representaría un ahorro de espacio y dinero.

- En el Tarificador Telefónico Casero se le podría agregar la facilidad de tener una alarma programada por el usuario ya que el MM58167A tiene esta facilidad. Una opción sería por ejemplo que el usuario marcará la tecla (#) y entonces el TTC mostrará en la pantalla LCD un mensaje pidiendo la hora de alarma deseada.

- El Reloj-Calendario MM58167A debería ser sustituido por el circuito integrado MC146818. Este integrado cumple con las mismas funciones y además está disponible en el almacén de SIEMENS, ya que las centrales EMS-601 y HICOM 130 lo usan.

Este "chip" no se usó en el diseño por la imposibilidad de conseguir la información técnica del componente.

En el H-95a se puede usar solo 1 cristal para todos los chips

BIBLIOGRAFIA:

Libros

1. CARISTI A., "Electronic Telephone Projects", Capitulo 1, SAMS, Indianapolis, First Edition, 1984, 11-17.
2. SINDICATO PROVINCIAL TELEFONICA, "Conmutación, Curso preparatorio a distancia para operadores técnicos", Telefónica formación, Madrid.
3. CENTRO DE ENTRENAMIENTO PARA TECNICOS EN TELECOMUNICACIONES (CETT), "Curso de Adiestramiento en Telecomunicaciones, Conmutación Telefónica", Caracas, 1973.
4. AUTOMATIC ELECTRIC INTERNATIONAL, "Principios Eléctricos de Telefonía, Curso de Instrucción", Manual 945-8005.
5. FERNANDEZ A., "Fundamentos de Telefonía básica privada", SIEMENS S.A., Dept. de Capacitación, Caracas.
6. CENTRO DE ENTRENAMIENTO PARA TÉCNICOS EN TELECOMUNICACIONES (CETT), "Telefonía General", 2da. Edición, Caracas, 1972.
7. CCITT, "Principios generales de Tarificación, Tasación y Contabilidad de los Servicios Internacionales de Telecomunicación, Recomendación de la serie D, Tomo II Fascículo II.1", Libro Azul de la IX Asamblea Plenaria, Ginebra, 1989.
8. CLAYTON G., "Experiments with Operational Amplifiers", The MacMillan Press LTD, First Edition, Surrey, Inglaterra, 1978.
9. GONZALEZ J., "Introducción a los microcontroladores 8X51 y 8X52, "hardware", "software" y Aplicaciones", McGrawHill, Madrid, España, Primera Edición, 1992.
10. MARTINEZ M., "OrCAD PCB", Editorial Paraninfo, Madrid, España, Segunda Edición, 1992.
11. ARNISON R., ROSEN D., WAITE M. y ZUCK J., "The Waite Group's Visual Basic How to", Emeyville, USA, First Edition, 1992.

12. LM ERICSSON, "Automatización de Redes Rurales. Sistema de Selectores de Coordenadas LM Ericsson, Capítulos 1 y 2", Telefonaktiebolaget, LM Ericsson, Libro No. 15504, Suecia.

Revistas

13. "Hybrids", The Lenkurt Demodulator, Vol.13, No. 1, San Carlos, California, USA, 1964.
14. "Services and Rates", Demodulator GTE Lenkurt, Vol. 31, No. 1, San Francisco California, 1982.
15. "Principios de Señalización Telefónica", El Demodulador GTE Lenkurt, No 198, San Carlos, California, 1974.
16. "El Teléfono", El Demoldulador GTE Lenkurt, No. 200, San Carlos, California, 1974.
17. "La señalización en líneas telefónicas", El Demodulador Lenkurt, No. 126, 1966.

Proyectos de grado

18. AFONSO J. y CONSALVO F., "Comunicación de datos a través de la central telefónica privada SIEMENS EMS-601", Curso en cooperación USB-SIEMENS, Caracas, 1988.
19. CARRASQUERO K. y GONZALEZ L., "Interfaz de Discado en una PABX SIEMENS para interconexión DISA", Curso en cooperación USB-SIEMENS, Caracas, 1993.
20. SUAREZ J. y GARCIA E., "Banco de Prueba para el mantenimiento rutinario en el campo de la telefonía", Curso en cooperación USB-SIEMENS, Caracas, 1993.
21. RODRIGUEZ R. y SOSA J., "Diseño e implementación de una central telefónica con capacidad para 2 líneas y 6 extensiones", Curso en cooperación USB-SIEMENS, Caracas, 1993.
22. MARTINEZ J. y CABRERA A., "Mediciones Objetivas en aparatos telefónicos", Proyecto de grado UCV, Caracas, 1965.

Manuales

23. INTEL, "Component Data Catalog", Santa Clara, California, 1982.

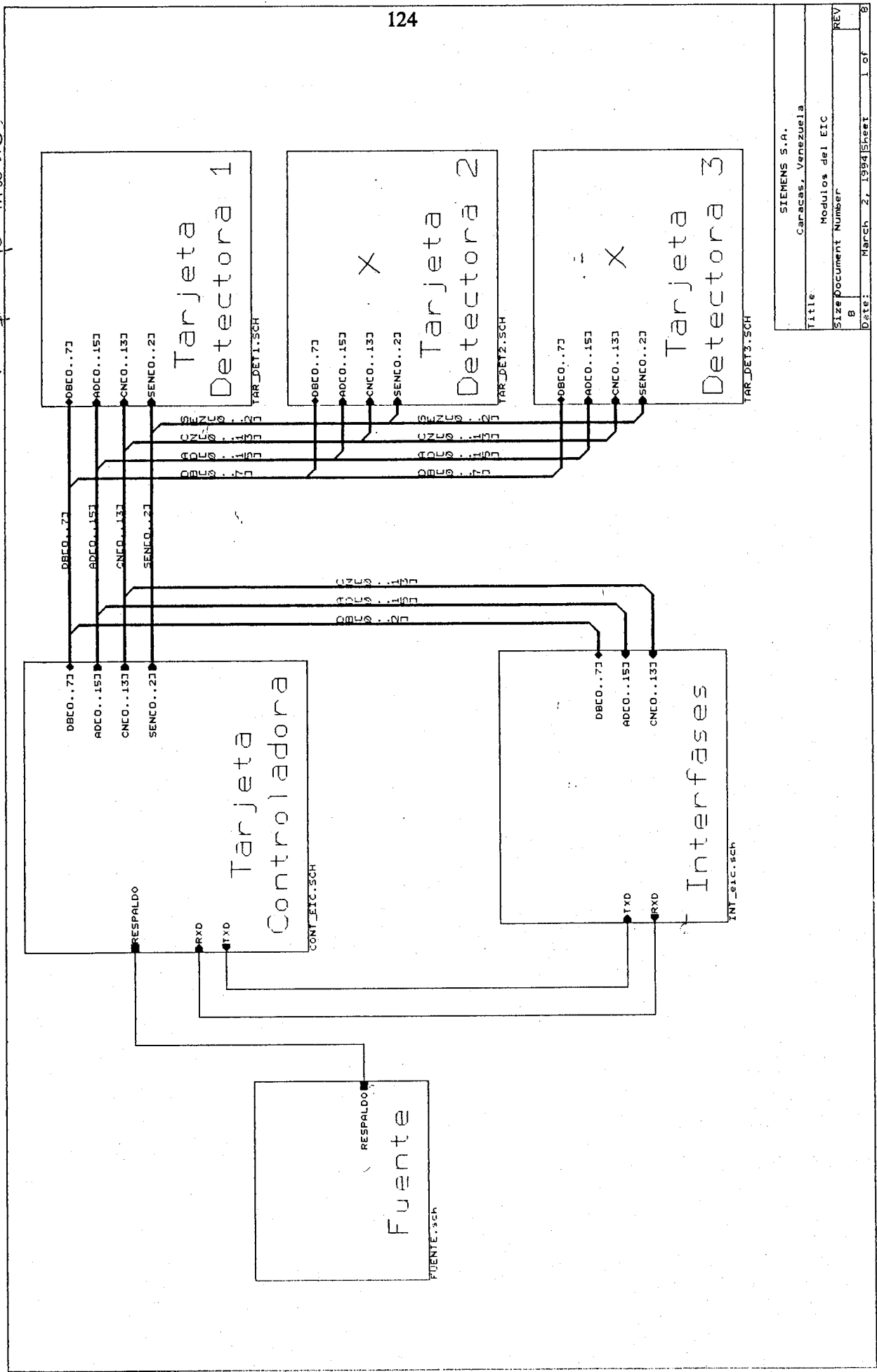
FALTA una hoja.

APENDICES

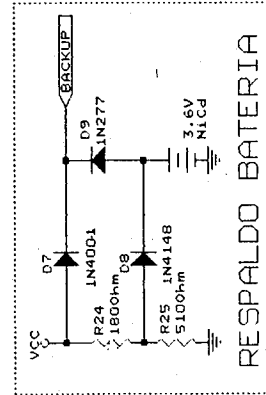
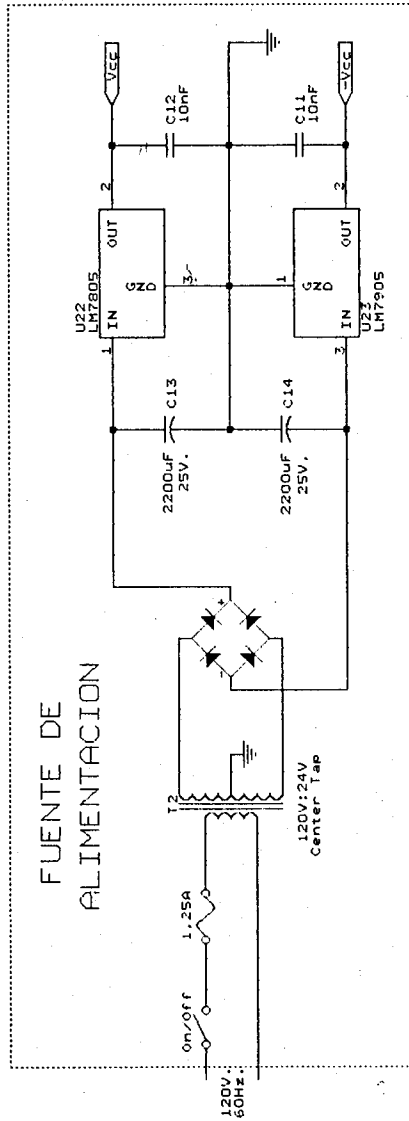
- 1. Diagramas de conexión**
- 2. Listado del programa en assembler para el EIC**
- 3. Rutinas adicionales para el TTC**
- 4. Listado del programa emulador de EPROM**
- 5. Registros de programación del 8031**
- 6. Tarifas telefónicas**
- 7. Informaciones técnicas de algunos componentes**
- 8. Glosario de telefonía general**
- 9. Fundamentos de la telefonía**
- 10. Tablas de costos**
- 11. Conectores usados**
- 12. Analizador de base de datos para EXCEL (versión preliminar)**
- 13. Ensamblaje de un programa en assembler**
- 14. Listado del programa en BorlandC++ para interfaz con el usuario**

1. DIAGRAMAS DE CONEXION

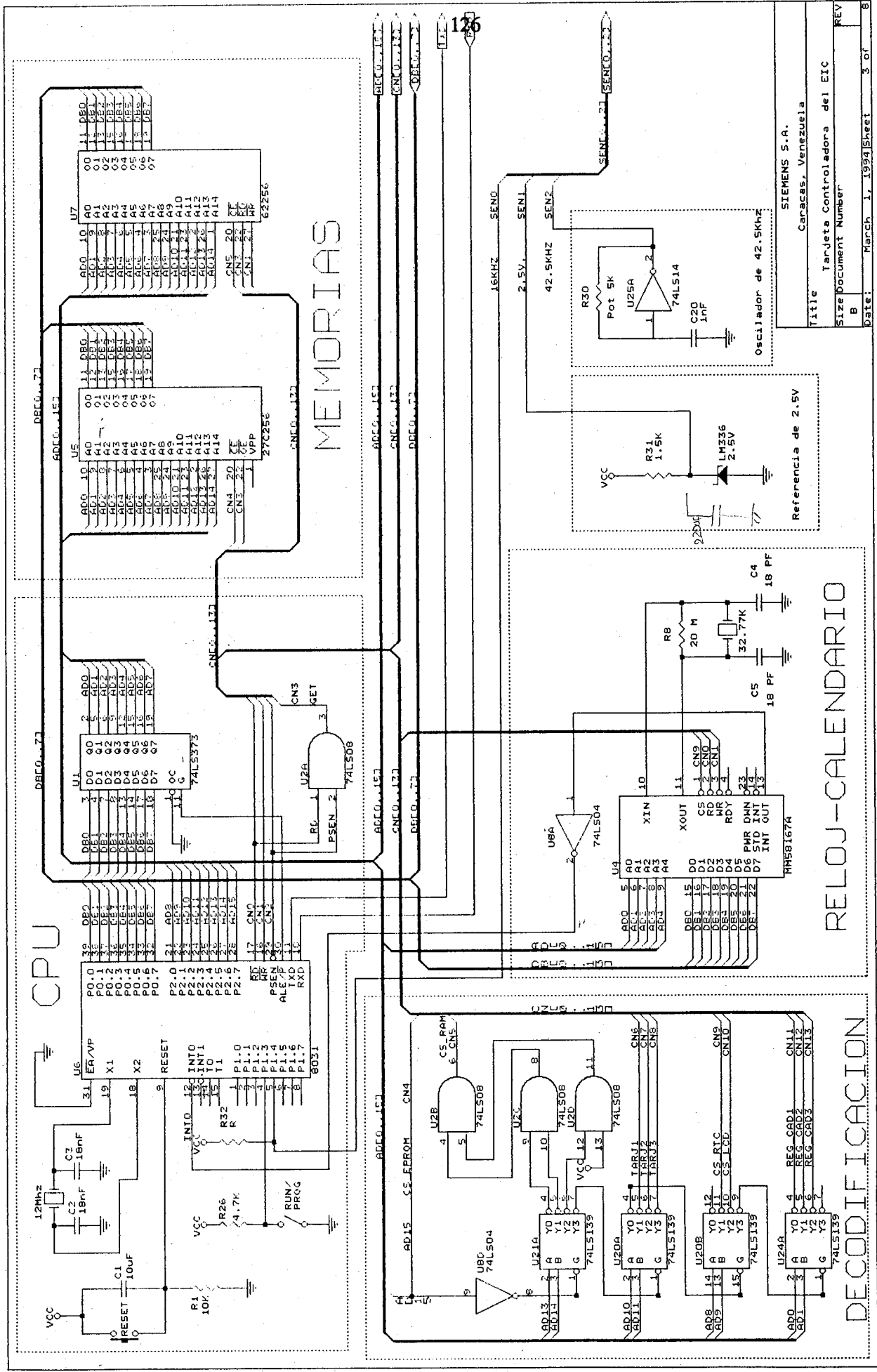
Hacer diagramas para 4 lineas.
Solo lo que yo monte.



SIEMENS S.A.	
Caracas, Venezuela	
Title	Modulos del EIC
Size	Document Number
B	REV
Date:	March 2, 1994 Sheet 1 of 8



Title		SIEMENS S.A.
Caracas, Venezuela		
Source de Alimentacion		
Size	Document Number	REV
B		
Date:	March 3, 1994	Sheet 2 of 2



SIEMENS S.A.
Caracas, Venezuela

Title Tarjeta Controladora del EIC
Size Document Number B
Date March 1, 1994 Sheet 3 of 8

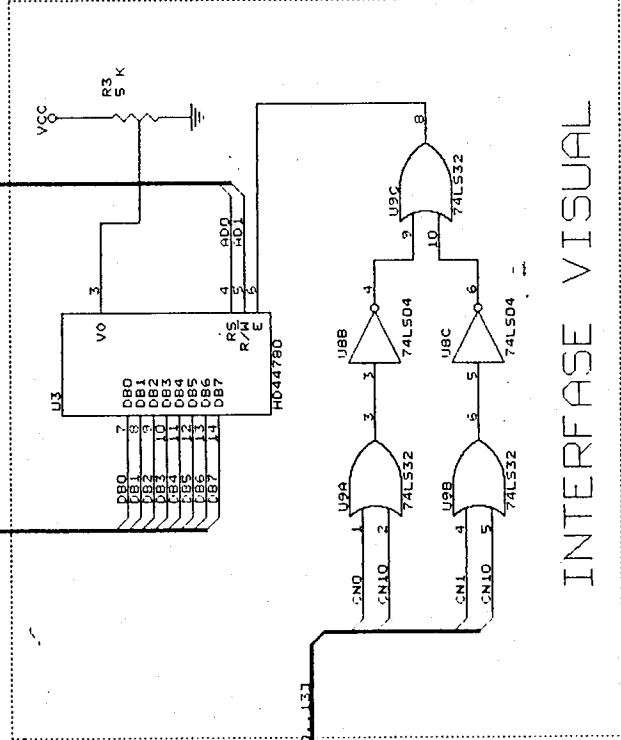
Oscilador de 42.5KHz
Referencia de 2.5V

RELOJ-CALENDARIO

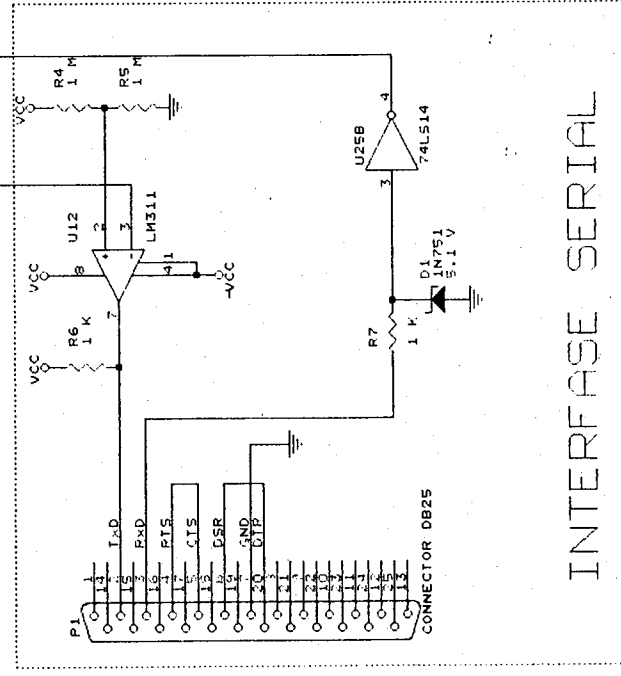
DECODIFICACION

MEMORIAS

ADCO..151
 DECO..171
 CNLO..131
 FXD
 VFD

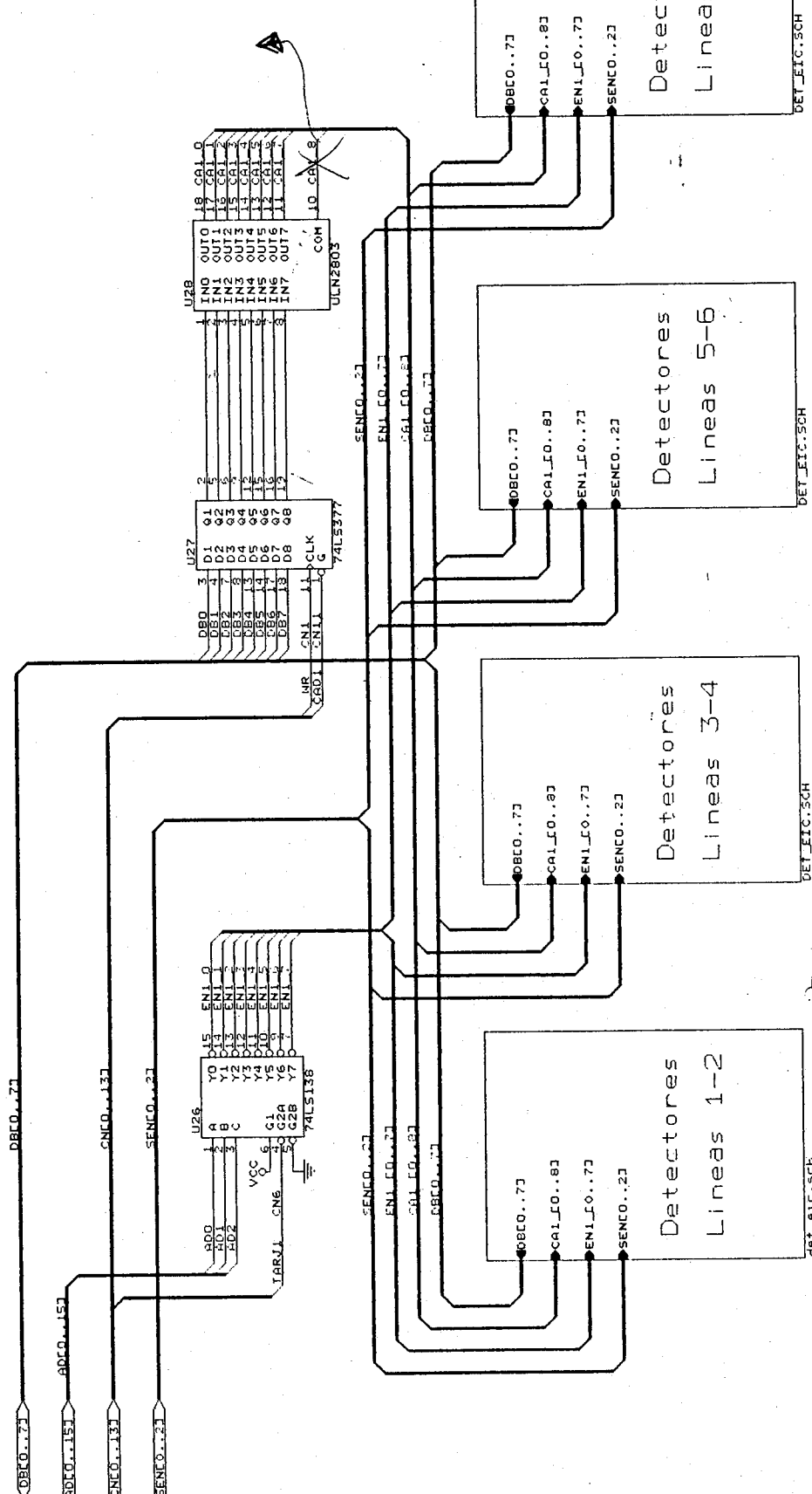


INTERFASE VISUAL

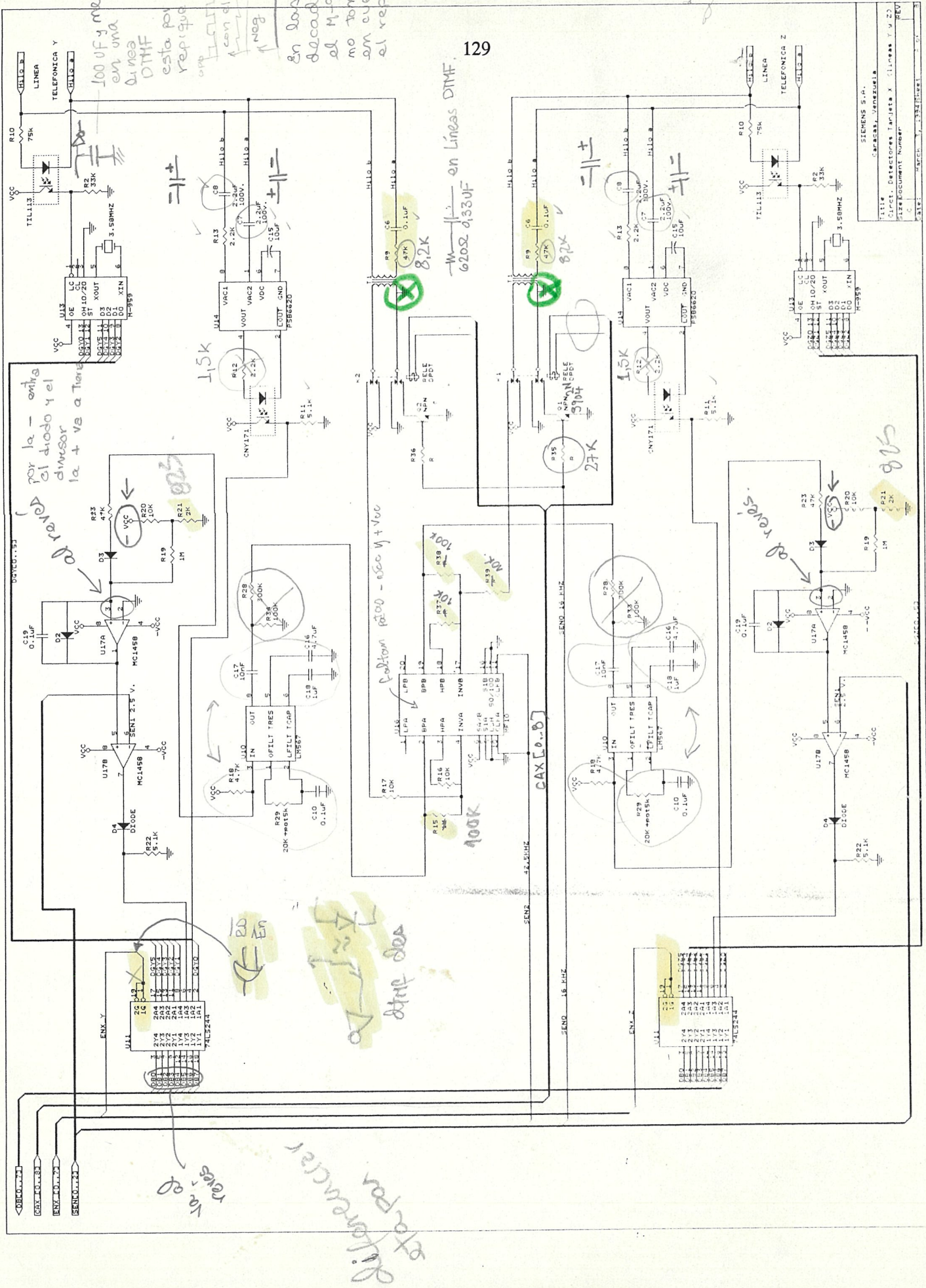


INTERFASE SERIAL

SIEMENS S.A.	
Caracas, Venezuela	
Title	Interfases del EIC
Size	Document Number
REV	B
Date:	March 3, 1994 Sheet 4 of 8



SIEMENS S.A.	
Caracas, Venezuela	
Title Tarjeta Detectora 1	
Size	Document Number
REV B	
Date:	March 3, 1994 Sheet 5 of 8



por la - entre el diodo y el divisor la + va a tierra

o tenerlo

100 uF y meg en una linea DTMF esta por el repique con el cordón

en las decodifican al M-959 no tome en cuenta el repique

W en líneas DTMF 6208 0.330UF

potron 1000 - 5500 y +VCC

100K

10K

100K

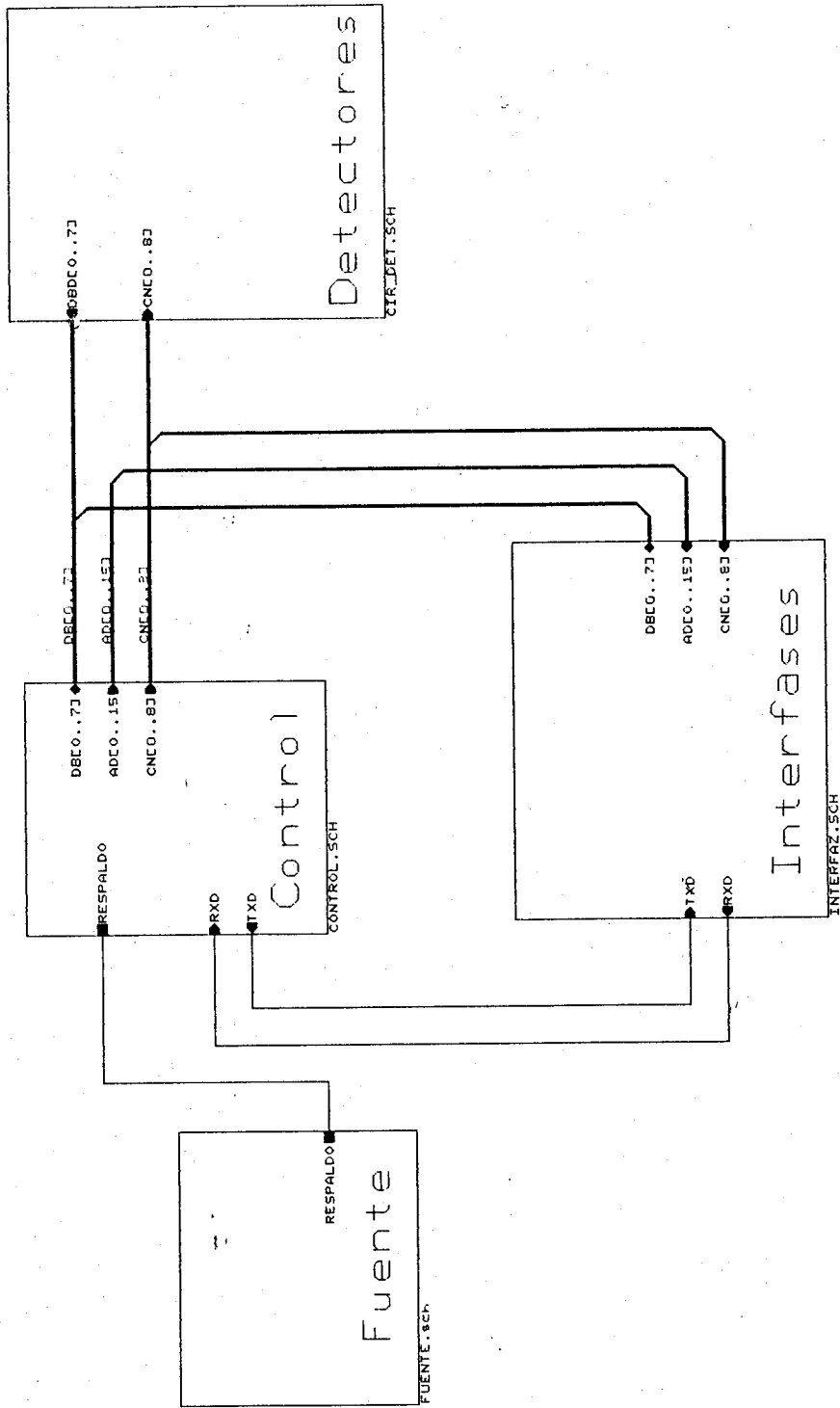
10K

10K

o tenerlo

805

TITLE	Direct. Detectores Tardista X (Lineas Y U 22)
DATE	1983
DESIGNER	SIEMENS S.P.A.
DRAWN	SIEMENS S.P.A.
CHECKED	SIEMENS S.P.A.
APPROVED	SIEMENS S.P.A.
REV	



SIEMENS S.A.	
Caracas, Venezuela	
Title	Modulos del TIC
Size	Document Number
B	REV
Date:	February 26, 1994 Sheet 1 of 5

2. LISTADO DEL PROGRAMA EN ASSEMBLER PARA EL EIC

```

;*****
;* MODULO:          "DEFINICION"
;* ARCHIVO:         DEF.ASM
;* DESCRIPCION:    DEFINICION DE VARIABLES, DIRECCIONES, CONSTANTES Y ETIQUETAS
;* FECHA:          3-OCT-93
;*****

;DIRECCIONES DE LOS PERIFERICOS

C_MINU EQU      0E103H      ;REAL TIME CLOCK - MINUTOS
C_HORA EQU      0E104H      ;REAL TIME CLOCK - HORAS
C_DSEM EQU      0E105H      ;REAL TIME CLOCK - DIA DE LA SEMANA
C_DMES EQU      0E106H      ;REAL TIME CLOCK - DIA DEL MES
C_MES EQU       0E107H      ;REAL TIME CLOCK - MES
C_INTS EQU      0E110H      ;REAL TIME CLOCK - STATUS INTERRUPCIONES
C_INTC EQU      0E111H      ;REAL TIME CLOCK - CONTROL INTERRUPCIONES
C_RST EQU       0E112H      ;REAL TIME CLOCK - RESET
C_GO EQU       0E114H      ;REAL TIME CLOCK - STATUS
C_STAT EQU      0E115H      ;REAL TIME CLOCK - INICIO DE LOS CONTADORE

IR_LCD EQU      0E200H      ;WRITE AL IR DEL DISPLAY LCD
DR_LCD EQU      0E201H      ;WRITE AL DR DEL DISPLAY LCD
BSYFLG EQU      0E202H      ;READ BUSY FLAG DEL DISPLAY LCD

STAT01 EQU      0E400H      ;STATUS DE LAS LINEAS URBANAS (TARJETA 1)
STAT02 EQU      0E401H
STAT03 EQU      0E402H
STAT04 EQU      0E403H
STAT05 EQU      0E404H
STAT06 EQU      0E405H
STAT07 EQU      0E406H
STAT08 EQU      0E407H

STAT09 EQU      0E800H      ;STATUS DE LAS LINEAS URBANAS (TARJETA 2)
STAT10 EQU      0E801H
STAT11 EQU      0E802H
STAT12 EQU      0E803H
STAT13 EQU      0E804H
STAT14 EQU      0E805H
STAT15 EQU      0E806H

```

STAT16	EQU	0E807H	
STAT17	EQU	0EC00H	;STATUS DE LAS LINEAS URBANAS (TARJETA 3)
STAT18	EQU	0EC01H	
STAT19	EQU	0EC02H	
STAT20	EQU	0EC03H	
STAT21	EQU	0EC04H	
STAT22	EQU	0EC05H	
STAT23	EQU	0EC06H	
STAT24	EQU	0EC07H	
CAD1	EQU	0E300H	;CADENCIA DE LAS LINEAS URBANAS (TARJETA 1)
CAD2	EQU	0E301H	;CADENCIA DE LAS LINEAS URBANAS (TARJETA 2)
CAD3	EQU	0E302H	;CADENCIA DE LAS LINEAS URBANAS (TARJETA 3)
;VARIABLES EN LA MEMORIA RAM INTERNA			
I:	EQU	20H	;CONTADOR MAIN (0,1,2)
J:	EQU	21H	;CONTADOR MAIN (0-23)
K:	EQU	22H	;CONTADOR MAIN (0,1,2,3,4,5,6,7)
CODH:	EQU	23H	;CODIGO DE LA CIUDAD DESTINO HIGH
CODL:	EQU	24H	;CODIGO DE LA CIUDAD DESTINO LOW
TEMPH	EQU	2BH	;CONSTANTE HIGH DE CONTEO (FREC. IMPULSOS)
TEMPL	EQU	2CH	;CONSTANTE LOW DE CONTEO (FREC. IMPULSOS)
V_IMP1	EQU	54H	;VARIABLE DE IMPULSOS TARJETA 1
V_IMP2	EQU	55H	;VARIABLE DE IMPULSOS TARJETA 2
V_IMP3	EQU	56H	;VARIABLE DE IMPULSOS TARJETA 3
NO_LIN	EQU	57H	;NUMERO DE LINEAS (1,8,16,24)
NO_LOOP	EQU	58H	;NO_LIN / 3 (1,2,3)
ESPH	EQU	59H	;ESPERA EN LA CADENCIA DE SENAL DE COMPUTO
ESPL	EQU	5AH	;ESPERA EN LA CADENCIA DE SENAL DE COMPUTO
C50MSCH:	EQU	5BH	;CONTADOR DE 50MS PARA SEÑAL DE COMPUTO
C50MSCL:	EQU	5CH	;CONTADOR DE 50MS PARA SEÑAL DE COMPUTO
FLAGS:	EQU	5DH	;FLAGS DE ESTADOS OBTENIDOS POR LA LINEA
STATUS:	EQU	60H	;STATUS DE LEIDO EN POLLING DE LINEA ACTUA
ST_VIEJO:	EQU	61H	;STATUS DEL POLLING ANTERIOR
C_REP:	EQU	62H	;CONTADOR DE 5 SEG PARA REPIQUE
C_RBK:	EQU	63H	;CONTADOR DE 5 SEG PARA RING BACK
NMAR:	EQU	64H	;CANTIDAD DE NUMEROS MARCADOS
FINPRG:	EQU	65H	;FIN DE PROGRAMACION DEL PROTOTIPO
CAM_DES:	EQU	70H	;CAMBIOS EN EL COLGADO/DESCOLGADO
CAM_REP:	EQU	71H	;CAMBIOS EN EL REPIQUE
CAM_RBK:	EQU	72H	;CAMBIOS EN EL RING BACK
CAM_DV:	EQU	73H	;CAMBIOS EN EL DIGITO VALIDO
HORACT	EQU	4DH	;HORA ACTUAL
MINACT	EQU	4EH	;MINUTOS ACTUAL

```

DSEACT EQU 4FH ;DIA DE LA SEMANA ACTUAL
DMESAC EQU 50H ;DIA DE MES ACTUAL
MESACT EQU 51H ;MES ACTUAL
ANOACT EQU 52H ;ANO ACTUAL
AM__PM EQU 53H ;AM O PM

```

```

;DIRECCIONES INICIALES DE LAS TABLAS

```

```

TAB_DIA EQU 0C000H ;TABLA DE LOS DIAS DE LA SEMANA
TAB_MES EQU 0C020H ;TABLA DE LOS MESES DEL AÑO
TAB_FER EQU 0C050H ;TABLA DE FERIADOS EN VENEZUELA
TAB_URB EQU 0C070H ;TABLA DE TARIFAS URBANAS
TAB_DDN EQU 0C090H ;TABLA DE CADENCIA DEL DDN
TAB_PRG EQU 0C0B0H ;TABLA DE PROGRAMACION
TAB_TNAC EQU 0C100H ;TABLA DE TARIFAS NACIONALES
TAB_ONAC EQU 0C500H ;TABLA DE TARIFAS POR OPERADORES (100)
TAB_CEL EQU 0C550H ;TABLA DE TARIFAS CELULARES
TAB_INT EQU 0C560H ;TABLA DE TARIFAS INTERNACIONALES
TAB_HOI EQU 0CA00H ;TABLA DE HORARIOS INTERNACIONALES
TAB_VAR EQU 0DC00H ;TABLA DE VARIABLES ASOCIADAS A CADA LINEA
TAB_MEN EQU 0DF00H ;TABLA DE MENSAJES EN EL DISPLAY

```

```

;*****
;FORMATO DE LAS TABLAS
;*****

```

```

;1) TABLA DE LOS NOMBRES DE LOS DIAS (TAB_DIA) (3 bytes x 7 = 21 bytes)
;Estructura:
;* Cada byte es una letra
;* 3 letras constituyen la abreviatura de un dia de la semana
;Esta tabla se define al final de este modulo

```

```

;2) TABLA DE LOS NOMBRES DE LOS MESES (TAB_MES) (3 bytes x 12 = 36 bytes)
;Estructura:
;* Cada byte es una letra
;* 3 letras constituyen la abreviatura del nombre de un mes
;Esta tabla se define al final de este modulo

```

```

;TABLA DE FERIADOS (TAB_FER) (2 bytes x 11 = 22 bytes)
;Final de la Tabla: FFh
;Estructura: 1
;* 1 byte dia
;* 1 byte Mes

```

```

;*****
*

```

```

;TABLA DE TARIFAS URBANAS (TAB_URB) (6 bytes en total)
;Estructura:
;* 2 bytes para la cadencia
;* 2 bytes para Costo residencial
;* 2 bytes para Costo no residencial

```

```

;90 seg -----> 1800d=708h (2 BYTES)

```

```
;11,18 Bs -----> 118d=0076h
```

```
;1,56 Bs -----> 156d=009Ch
```

```
*****
*
```

```
;TABLA DE CADENCIA DEL DDN (TAB_DDN) (3 bytes x 6 = 18 bytes)
```

```
;Estructura:
```

```
;* 1 byte codigo de tarifa
```

```
;* 1 byte Tarifa Nacional
```

```
;* 1 byte Tarifa Especial
```

```
;Constantes de recarga del Timer1 para una base de tiempo de 50 ms
```

```
;Tarifa Normal Tarifa Especial
```

```
;7,5 seg --> 153d=99h 12,0 seg --> 243d=F3h
```

```
;5,0 seg --> 103d=67h 8,0 seg --> 163d=A3h
```

```
;4,0 seg --> 83d=53h 5,5 seg --> 113d=71h
```

```
;3,0 seg --> 63d=3Fh 4,5 seg --> 93d=5dh
```

```
;2,5 seg --> 53d=35h 4,0 seg --> 83d=53h
```

```
;2,0 seg --> 43d=2Bh 3,5 seg --> 73d=49h
```

```
*****
**
```

```
;TABLA DE TARIFAS DE DISCADO DIRECTO NACIONAL (TAB_TNAC) (2 x 36 = 72 bytes)
```

```
;Estructura:
```

```
;* 2 bytes costo del impulso residencial
```

```
;* 2 bytes costo del impulso no residencial
```

```
;* luego, la tabla de codigos de area y tarifa con la siguiente estructura:
```

```
;* 1 byte codigo de area
```

```
;* 1 byte codigo de tarifa
```

```
;* fin de tabla = FFh
```

```
;codigo de area: 00-99
```

```
;codigo de tarifa: 1-6
```

```
*****
**
```

```
;TABLA DE TARIFAS POR OPERADORES NACIONAL (100) (TAB_ONAC) (13 bytes x 6 = 78 bytes)
```

```
;Estructura:
```

```
;* 1 byte Codigo de Tarifa
```

```
;* 3 bytes 3 minutos por conexion (Tarifa Normal)
```

```
;* 3 bytes 1 minuto adicional (Tarifa Normal)
```

```
;* 3 bytes 3 minutos por conexion (Tarifa Especial)
```

```
;* 3 bytes 1 minuto adicional (Tarifa Especial)
```

```
;A los costos no se le presta atencion a la coma. Es decir, 169,50=16950
```

```
*****
**
```

```
;TABLA DE TARIFAS CELULARES (TAB_CEL) (6 bytes total)
```

```
;Estructura:
```

```
;* 1 byte cadencia normal 2,0 seg = 43h
```

```
;* 1 byte cadencia especial 2,5 seg = 53h
```

```
;* 2 bytes costo residencial
```

```
;* 2 bytes costo no residencial
```

```
*****
```



```

;TABLA DE TARIFAS INTERNACIONALES (TAB_INT) (9 bytes x 35 = 315 bytes)
;Direccion Inicial: C560H
;* 1 byte Continente (1-8)
;* 2 bytesCodigo de pais
;* 2 bytes Costo Tarifa Normal
;* 2 bytes Costo Tarifa Reducida
;* 2 bytes Costo Tarifa Economica
;* Fin de tabla: FFh

```

```

;Continentes:
;1 = America del Norte
;2 = America Central
;3 = Antillas
;4 = America del Sur
;5 = Europa
;6 = Asia oriental
;7 = Asia occidental
;8 = Resto del Mundo

```

```

;*****
*
;TABLA DE FRANJAS HORARIAS (TAB_HOI) (4 bytes x 10 = 40 bytes)
;Estructura:
;* 1 byte Tipo (1-2) ---> 1= Tarifa Economica; 2= Tarifa Reducida
;* 1 byte Continente (1-8)
;* 1 byte Hora inicial
;* 1 byte Hora final
;* Fin de tabla = FFh

```

```

ORG      TAB_MEN
FCB      "Programacion"
FCB      "Alarma?"
FCB      "Tarj:"
FCB      "Linea:"
FCB      "Stat:"
FCB      "F. Impulso:"
FCB      "Khz"
FCB      "Hz"
FCB      "No. de Lineas:"
FCB      "1) Accione boton"
FCB      "de programacion"
FCB      "2) Conecte cable"
FCB      "del puerto serie"

```

```

ORG      TAB_DIA
FCB      " Lun Mar Mie Jue Vie Sab Dom"
ORG      TAB_MES
FCB      "EneFebMarAbrMayJunJulAgoSepOctNovDic"

```

```

;*****
;* MODULO:          "INICIALIZACION"
;* ARCHIVO:         INIT.ASM
;* DESCRIPCION:     INICIALIZACION DEL PROGRAMA
;* FECHA:           3-OCT-93
;*****

```

```

ORG      0000H
LJMP     0100H          ;SALTO AL INICIO DEL PROGRAMA

```

```

;REDIRECCIONAMINETO DE LA VECTORIZACION DE LAS INTERRUPCIONES

```

```

ORG      0003H          ;VECTORIZACION DE LA INTERRUPCION EXTERNA
LJMP     EXT_0

```

```

ORG      000BH          ;VECTORIZACION DE LA INTERRUPCION DEL TIME

```

```

LJMP     TIM__0

```

```

ORG      001BH          ;VECTORIZACION DE LA INTERRUPCION DEL TIME

```

```

LJMP     TIM__1

```

```

ORG      0023H          ;VECTORIZACION DE LA INTERRUPCION SERIAL
LJMP     SERIAL

```

```

ORG      0100H          ;INICIO DEL PROGRAMA EN RAM

```

```

MOV      P1,#0FFH      ;PROGRAMO PUERTO1 PARA LECTURA
;PROGRAMACION DE LOS TIMERS

```

```

MOV      TMOD,#11H     ;TIMER 0 MODO 1 TEMPORIZADOR DE 16 BITS
;TIMER 1 MODO 2 TEMPORIZADOR DE 16 BITS
MOV      TL0,#00H     ;CONSTANTES PARA LA BASE DE TIEMPO (50 MS)
MOV      TH0,#3DH

```

```

MOV      TL1,#0EAH    ;CONSTANTES PARA ONDA DE 16KHZ

```

```

MOV      TH1,#0FFH
MOV      TCON,#55H    ;RUN TIMER0 Y TIMER1
;INT0 E INT1 POR FLANCO DE BAJADA

```

```

;PROGRAMACION DEL PUERTO SERIE

```

```

MOV      SCON,#70H    ;MODO 1, 8 BIT UART, VELOCIDAD VARIABLE
;LOS 2400 BAUDIOS SON GENERADOS POR TIMER1

```

```

;INICIALIZACION DEL NUMERO DE LINEAS

```

```

MOV      NO_LIN,#24H  ;POR DEFAULT HAY 24 LINEAS
MOV      A,NO_LIN    ;NO DE LINEAS (1,8,16,24)
MOV      B,#8

```

ACTNO_LIN : MOV A, WOLIN DIV AB
 CSNE A, #1, WOLOP MOV NO_LOOP, P
 MOV NO_LOOP, #1H
 SJMP fuera
 MOV B, #8 fuera ret;

137

```
DIV        AB
MOV        NO_LOOP, A
```

;NO DE LOOP ES VARIABLE AUXILIAR

;PROGRAMACION DEL DISPLAY LCD

```
MOV        A, #38H
CALL       INST
```

;SELECCIONA A 8 BITS DE OPERACION, 2 LINEAS
 ;DE DISPLAY Y FONT DE 5x7

```
CALL       PDISA
CALL       DIHR_FC
```

;DISPLAY: LINEA1 A LA IZQUIERDA
 ;HORA Y FECHA EN EL DISPLAY

;PROGRAMACION DEL MM58167A (MICROPROCESSOR REAL TIME CLOCK)

```
MOV        DPTR, #C__RST
MOV        A, #0FFH
```

;REGISTRO PARA RESET DE LOS CONTADORES
 ;TODOS EN UNA HACE UN RESET

```
MOVX       @DPTR, A
MOV        DPTR, #C__GO
MOV        A, #0FFH
MOVX       @DPTR, A
```

```
MOV        DPTR, #C__INTC
MOV        A, #08H
MOVX       @DPTR, A
```

;REGISTRO DE CONTROL DE INTERRUPCIONES
 ;INTERRUMPE CADA MINUTO

```
MOV        DPTR, #C__INTS
MOVX       @DPTR, A
MOV        DPTR, #C__HORA
MOV        A, #12H
```

;REGISTRO DEL STATUS DE LAS INTERRUPCIONES
 ;INICIALIZO LAS HORAS

```
MOV        HORACT, A
MOVX       @DPTR, A
MOV        DPTR, #C__MINU
MOV        A, #00H
```

;INICIALIZO LOS MINUTOS

```
MOV        MINACT, A
MOVX       @DPTR, A
MOV        DPTR, #C__DMES
MOV        A, #01H
```

;INICIALIZO EL DIA

```
MOV        DMESAC, A
MOVX       @DPTR, A
MOV        DPTR, #C__DSEM
MOV        A, #01H
```

;INICIALIZO DIA DE LA SEMANA

```
MOV        DSEACT, A
MOVX       @DPTR, A
MOV        DPTR, #C__MES
MOV        A, #01H
```

;INICIALIZO EL MES

```
MOV        MESACT, A
MOVX       @DPTR, A
MOV        ANOACT, #94H
```

;INICIALIZO EL AÑO

;PROGRAMACION DE LAS INTERRUPCIONES

```
MOV        IE, #8BH
```

;HABILITO INTERRUPCIONES:
 ;EXTERNA 0 (RELOJ DEL SISTEMA)
 ;TIMER 0 (BASE DE TIEMPO DE 50MS)
 ;TIMER 1 (SEñAL DE COMPUTO 16KHZ)

```
MOV        IP, #08H
```

;PRIORIDAD A LA INTERRUPCION DEL TIMER 1

```

;*****
;* MODULO:          "PRINCIPAL"
;* ARCHIVO          MAIN.ASM
;* DESCRIPCION:     PROGRAMA PRINCIPAL
;* FECHA:           3-OCT-93
;*****

```

```

DEFSEG  INICIO,ABSOLUTE
SEG      INICIO

```

```

%INCLUDE <INIT_EPR.ASM>

```

```

;*****
;*****

```

```

;PROGRAMA PRINCIPAL

```

```

NUEVO_CI:CALL    PROG?          ;HAY QUE PROGRAMAR EL EQUIPO ?
              MOV     I,#0H      ;I CONTADOR (0,1,2)
              MOV     J,#0H      ;J CONTADOR (0-23)
PROX_TAR:MOV     K,#0H          ;K CONTADOR (0,1,2,3,4,5,6,7)

LEE_STAT:CALL    LEO_STAT        ;LEO EL STATUS ACTUAL DE LA LINEA (REGISTR
S)
              CALL    RECUPERA    CALL CAMBIO? ;RECUPERO LAS VARIABLES DE LA LINEA
              CALL    ANALIZO     ;POR DONDE VOY EN ESA LINEA?
              CALL    DEVOLVER    ;DEVUELVO LAS VARIABLES DE ESA LINEA
              INC     J
              INC     K
              MOV     A,K
              CJNE   A,#8,LEE     ;ANALICE YA LAS 8 LINEAS DE ESA TARJETA ?
              INC     I
              MOV     A,I
              CJNE   A,NO_LOOP,NEW_TAR ;YA ANALICE TODAS LAS TARJETAS PRESENTES ?
              LJMP   NUEVO_CI     ;SI ANALIZE TODAS EMPIEZO DE NUEVO
LEE:          LJMP   LEE_STAT
NEW_TAR:     LJMP   PROX_TAR

```

```

;*****
;*****

```

*leo_stat } ← Meter aqui.
Recupera }*

```

;*****
*

```

```

;RUTINA PARA SABER EL PROGRESO DEL ESTADO DE UNA LINEA

```

```

ANALIZO: MOV     A,FLAGS
          RRC     A
          JNC    FLG_REP
          CALL  CAMBIO? ;C=1 --> YA ESTOY APLICANDO IMPULSOS
          MOV     A,CAM_DES

```

;SI CAM_DES = 2 -->COLGARON. NO MAS IMPULS

S CJNE A,#2,F1

F1: CALL FIN_COM
LJMP PRX_LINFLG_REP: MOV A,FLAGS
RRC A
RRC AJNC FLG_RBK
~~CALL CAMBIO?~~

;C=1 --> YA HABIA VENIDO UN REPIQUE

MOV A,CAM_DES
CJNE A,#1,OTRO_RP
MOV FLAGS,#0

;SI CAM_DES = 1 -->ATENDIERON LLAMADA ENTR

;FLAG REPIQUE = 0

;FLAG ENTRANTE = 1

MOV A,#0000100B
ORL A,FLAGS
MOV FLAGS,A
LJMP PRX_LIN

OTRO_RP: MOV A,CAM_REP

CJNE A,#1,F2
MOV C_REP,#0

;CAM_REP = 1 --> VINO OTRO REPIQUE

;PONGO EL CONTADOR DE REPIQUE EN CERO

F2: LJMP PRX_LIN

FLG_RBK: MOV A,FLAGS
RRC A
RRC A
RRC AJNC FLG_ENT
~~CALL CAMBIO?~~

;C=1 --> YA HABIA VENIDO UN RING BACK

MOV A,CAM_DES
CJNE A,#2,OTRO_RB
CALL FIN_COM
LJMP PRX_LIN

;CAM_DES = 2 --> COLGARON

OTRO_RB: MOV A,CAM_RBK

CJNE A,#1,F3
MOV C_RBK,#0

;CAM_RBK = 1 --> VINO OTRO RING BACK

;PONGO EL CONTADOR DE RING BACK EN CERO

F3: LJMP PRX_LIN

FLG_ENT: MOV A,FLAGS
RRC A
RRC A
RRC A
RRC AJNC FLG_SAL
~~CALL CAMBIO?~~

;C=1 SE TRATA DE UNA LLAMADA ENTRANTE

MOV A,CAM_DES
CJNE A,#2,F4
CALL FIN_COM

;SOLO ESPERO A QUE QUE CUELGUEN

;CAM_DES = 2 --> COLGARON

F4: LJMP PRX_LIN

FLG_SAL: MOV A,FLAGS
RRC A
RRC A
RRC A
RRC A

Una vez que ya estoy seguro de que es saliente
ya estoy seguro de que es saliente

140

```
RRC      A
JNC      NO_FLGS      ;C=1 SE TRATA DE UNA LLAMADA SALIENTE
CALL     CAMBIO?
MOV      A,CAM_DES
CJNE     A,#2,VINO_RB ;CAM_DES = 2 --> COLGARON
CALL     FIN_COM
LJMP     PRX_LIN
VINO_RB: MOV      A,CAM_RBK
CJNE     A,#1,HAY_DIG ;CAM_RBK = 1 --> VINO UN RING BACK
MOV      A,#00000100B ;FLAG DE RING BACK=1
ORL      A,FLAGS
MOV      FLAGS,A
MOV      C_RBK,#0     ;CADA VEZ QUE INTO OCURRA INCREMENTO C_RBK
LJMP     PRX_LIN
HAY_DIG: MOV      A,CAM_DV
CJNE     A,#1,NO_FLGS
CALL     TOMA_DIG
MOV      A,#1
CJNE     A,NO_LIN,F5   ;SI SOLO TENGO UNA LINEA
CALL     PDISD
CALL     DISNUM       ;MUESTRO NUMERO MARCADO EN EL DISPLAY
F5:      LJMP     PRX_LIN

NO_FLGS: CALL     CAMBIO? ;SI SE TRATA DE UNA LINEA EN REPOSO
MOV      A,CAM_DES
CJNE     A,#1,VINO_REP ;SI DESCOLGARON...
MOV      A,#00010000B
ORL      A,FLAGS      ;FLAG SALIENTE=1
MOV      FLAGS,A
LJMP     PRX_LIN
VINO_REP:MOV      A,CAM_REP
CJNE     A,#1,PRX_LIN ;SI VINO UN REPIQUE
MOV      A,#00000010B
ORL      A,FLAGS      ;FLAG REPIQUE
MOV      FLAGS,A

PRX_LIN: RET
```

Porque sino confundido al toro de libro

```
*****
*
;RUTINA PARA DETECTAR LOS DISTINTOS TIPOS DE TRANSICIONES EN LA PALABRA DE
;STATUS DE LAS LINEAS
;DEVUELVE SUS RESULTADOS EN LAS VARIABLES: CAM_DES; CAM_REP; CAM_DV
;LOS POSIBLES RESULTADOS SON: CAM_XXX: --> 0 NO CAMBIO
;
; --> 1 TRANSICION DE 0 A 1
; --> 2 TRANSICION DE 1 A 0
```

```
CAMBIO?: MOV      A,ST_VIEJO
XRL      A,STATUS
JZ       NOCAMBIO    ;SI EXOR DA DISTINTO DE CERO HUBO CAMBIO
;SI EXOR DA CERO NO HUBO CAMBIO
MOV      R5,A        ;R5 VARIABLE AUXILIAR PARA TENER CAMBIOS
MOV      R7,#0       ;R7 CONTADOR (0,1,2,3)
```

```

PROX_BIT:MOV      A,R5
             RRC      A
             MOV      R5,A
             JNC      NOCAMBIT          ;C=0 NO HUBO CAMBIO EN ESE BIT
             MOV      A,R7              ;C=1 ESE BIT CAMBIO, PERO COMO?!!
             MOV      R4,A
             MOV      A,#0              ;PONGO ACC EN 0 PARA CREAR LA MASCARA
             INC      R4                ;R4=R7+1
             SETB     C
ROT:         RLC      A                  ;FORMACION DE UNA MASCARA PARA TESTEAR...
             CLR      C                  ;EL BIT ADECUADO
             DJNZ    R4,ROT
             ANL     A,ST_VIEJO          ;SE HACE TESTER DEL BIT
             JZ      TR_0_1             ;SI ANL = 0 ES PORQUE ST_VIEJO ERA CERO
             MOV     A,#2                ;TRANSCICION DE 1 A 0 EN ESE BIT
             SJMP    GUARDA
TR_0_1:     MOV     A,#1                ;TRANSCICION DE 0 A 1 EN ESE BIT
             SJMP    GUARDA
NOCAMBIT:   MOV     A,#0                ;ESE BIT NO CAMBIO
GUARDA:     PUSH    ACC                 ;GUARDO POR UN MOMENTO EL TIPO DE CAMBIO
             MOV     A,CAM_DES           ;BUSCO LA VARIABLE DONDE GUARDAR CAMBIO
             ADD     A,R7                ;SUMO R7 A LA DIRECCION DE CAM_DES
             MOV     R0,A                ;R0 APUNTA A LA VARIABLE ADECUADA
             POP     ACC                 ;ACUMULADOR TIENE EL CAMBIO
             MOV     @R0,A              ;GUARDO EL CAMBIO
             INC     R7
             MOV     A,R7
             CJNE   A,#04,PROX_BIT      ;ANALIZO SI YA REVISE LOS CUATRO BITS
             SJMP    SALCAM
NOCAMBIO:   MOV     CAM_DES,#0          ;SI NO HUBO CAMBIO TODAS LAS VARIABLES
             MOV     CAM_REP,#0          ;QUEDAN EN CERO
             MOV     CAM_RBK,#0
             MOV     CAM_DV,#0
SALCAM:     RET

```

```

;*****
*
;RUTINA QUE LEE EL STATUS DE UNA LINEA (REGISTROS MAPEADOS EN MEMORIA)
;PARAMETROS I (0,1,2) INDICA CUAL ES LA TARJETA
;          K (0,1,2,3,4,5,6,7) INDICA CUAL ES LA LINEA EN ESA TARJETA
;EL STATUS LEIDO SE DEVUELVE EN LA VARIABLE GLOBAL STATUS
;TARJETA1 INICIO:DC00H; TARJETA2 INICIO:DD00H; TARJETA3 INICIO:DE00H
;CADA LINEA TIENE UN REGISTRO DE 8 BITS

```

```

LEO_STAT:MOV     DPTR,#STAT01          ;DPTR AL INICIO DE LOS STATUS MAPEADOS
             MOV     A,#4
             MOV     B,I                ;I ME DICE TARJETA 1,2 O 3
             MUL     AB
             ADD     A,DPH
             MOV     DPH,A              ;DPTR AL INICIO DE UNA TARJETA
             MOV     A,K
             ADD     A,DPL
             MOV     DPL,A              ;DPTR EN LA LINEA QUE DEBO LEER

```

```

MOVX    A,@DPTR    ;LEO EL STATUS DE ESA LINEA
MOV     STATUS,A   ;VALOR LEIDO EN VARIABLE GLOBAL STATUS
RET

```

```

;*****
*

```

```

;RUTINA PARA PONER EN CERO TODAS LAS VARIABLES DE UNA LINEA

```

```

FIN_COM: CALL     INI_PAG
        MOV      A,#0
        MOV      R5,#32    ;PONGO LAS 32 VARIABLES DE LINEA EN CERO
LIMPIA:  MOVX    @DPTR,A
        DJNZ    R5,LIMPIA
        RET

```

```

;*****
*

```

```

;RUTINA PARA TOMAR UN DIGITO MARCADO Y GUARDARLO EN RAM EN DONDE CORRESPONDA
;EL DIGITO SE ENCUENTRA EN LOS CUATRO BITS MAS SIGNIFICATIVOS DE LA
;VARIABLE STATUS: DDDDXXXX

```

```

TOMA_DIG:CALL     INI_PAG
        MOV      A,DPL
        ADD     A,#16
        MOV     DPL,A    ;DPTR APUNTA AL COMIENZO DE DIGITOS MARCAD
S
        MOV     A,NMAR
        ADD     A,DPL
        MOV     DPL,A    ;DPTR AL DIGITO SIGUIENTE
        MOV     A,STATUS
        SWAP    A        ;TENGO EL DIGITO EN LA PARTE BAJA DE ACC
        ANL    A,#00001111B ;DEJO SOLO AL DIGITO
DO
        MOVX   @DPTR,A   ;GUARDO EL DIGITO EN RAM EN EL,LUGAR ADECU
        INC    NMAR
        MOV    A,NMAR
        CJNE  A,#4,SALTMD ;SI TENGO 5 DIGITOS PLANIFICO CADENCIA
        CALL  PLN_CAD
SALTMD:  RET

```

```

;*****
*

```

```

;RUTINA PARA PLANIFICAR CADENCIA DE UNA LINEA
;CUANDO EN UNA LINEA YA HAYAN MARCADO 5 DIGITOS ENTONCES ENTRA ESTA RUTINA

```

```

PLN_CAD: CALL     INI_PAG
        MOV      A,DPL
        ADD     A,#16
        MOV     DPL,A    ;ESTOY YA EN EL PRIMER DIGITO MARCADO
        MOVX   A,@DPTR
        CJNE  A,#0,LOCAL ;SI PRIMER DIGITO <> 0 --> ES LOCAL
        INC    DPTR
        MOVX   A,@DPTR

```



```

CJNE    A,#0,INTERNAC    ;SI SEGUNDO DIGITO = 0 --> ES INTERNACIONA
MOV     B,#10             ;SI SEGUNDO DIGITO <> 0 --> ES NACIONAL
MUL     AB                ;CONFORMO EL CODIGO NACIONAL
MOV     R5,A
INC     DPTR
MOVX    A,@DPTR
ADD     A,R5              ;EN ACUMULADOR TENGO CODIGO NACIONAL
MOV     CODH,#0H
MOV     CODL,A
CJNE    A,#37,OT_ESP1    ;VERIFICO CASOS ESPECIALES
CALL    ARR_COD
SJMP    BUSCA
OT_ESP1: CJNE    A,#73,BUSCA
CALL    ARR_COD
BUSCA:  CALL    BUSC_TAB    ;BUSCO EN TABLA CODIGO DEL 1 AL 6
CALL    MODESP           ;MODIFICO LAS VARIABLES DE ESPERA (HIGH Y
OW)
LJMP    SALPLN
INTERNAC:MOV    ESPH,#00H    ;CONSTANTE PARA 1 SEGUNDO
MOV     ESPL,#23
LJMP    SALPLN
LOCAL:  MOV     ESPH,#07H    ;CONSTANTE PARA 90 SEG (1803D = 070BH)
MOV     ESPL,#0BH
SALPLN: RET

```

```

;*****
*

```

```

;RUTINA QUE PREGUNTA SI SE DESEA PROGRAMAR EL EQUIPO, Y SI ESTE ES EL CASO,
;SE QUEDA ESPERANDO DATOS POR EL PUERTO SERIE
;P1.3 = 0 FUNCIONAMIENTO NORMAL
;      = 1 PROGRAMACION

```

```

PROG?:  MOV     C,P1.3
JNC     SALPROG           ;SI P1.3 = 0 SIGO CORRIENDO EL PROGRAMA
;SI P1.3 = 1 PROGRAMACION

MOV     A,#00H           ;MENSAJE #0
MOV     R6,#0FH
CALL    MENLCD           ;PONGO MENSAJE EN EL DISPLAY "PROGRAMACION

MOV     TMOD,#20H        ;TIMER1 MODO2 (TEMPORIZADOR CON AUTORECARG

)

MOV     TH1,#0F3H        ;CONSTANTE PARA 2400 BAUDIOS (F_CTAL=12MHZ
MOV     TL1,TH1          ;PARA CUBRIR LA PRIMERA PASADA
MOV     TCON,#40H        ;RUN TIMER1

MOV     IE,#90H          ;SOLO HABILITO INTERRUPCION DEL PUERTO SER

E

SEG_PROG:MOV    C,P1.3
JC      SEG_PROG

;      CALL    DISPROG
CALL    RETARDO

```

```

MOV      TMOD,#11H      ;TIMER 0 MODO 1 TEMPORIZADOR DE 16 BITS
MOV      TLO,#00H      ;TIMER 1 MODO 2 TEMPORIZADOR DE 16 BITS
MOV      TH0,#3DH      ;CONSTANTES PARA LA BASE DE TIEMPO (50 MS)
MOV      TL1,TEMPL     ;CONSTANTES DEL TIMER1 VIENE EN PROGRAMACI
N
MOV      TH1,TEMPH
MOV      TCON,#55H     ;RUN TIMERO Y TIMER1
MOV      IE,#8BH      ;INT0 E INT1 POR FLANCO DE BAJADA
SALPROG: RET          ;HABILITO DE NUEVO LAS INTERRUPCIONES

```

```

;*****
*
;RUTINA PARA MODIFICAR LAS VARIABLES DE ESPERA DE UNA LINEA DEPENDIENDO DE UN
;NUMERO QUE SE LE PASA COMO PARAMETRO QUE ES EL CODIGO DE DISTANCIA CANTV (1-6
;CADA 20D ES UN SEGUNDO, 3D ES 150MS. (PARAMETRO EN EL ACUMULADOR)

```

```

MODESP:  CJNE      A,#01,TAR2
MOV      ESPH,#00      ;CONSTANTE PARA 7.5 SEG (153D = 99H)
MOV      ESPL,#99H
SJMP     SALMOD
TAR2:    CJNE      A,#02,TAR3
MOV      ESPH,#00      ;CONSTANTE PARA 5 SEG (103D = 67H)
MOV      ESPL,#67H
SJMP     SALMOD
TAR3:    CJNE      A,#03,TAR4
MOV      ESPH,#00      ;CONSTANTE PARA 4 SEG (83D = 53H)
MOV      ESPL,#53H
SJMP     SALMOD
TAR4:    CJNE      A,#04,TAR5
MOV      ESPH,#00H     ;CONSTANTE PARA 3 SEG (63D = 3FH)
MOV      ESPL,#3FH
SJMP     SALMOD
TAR5:    CJNE      A,#05,TAR6
MOV      ESPH,#00      ;CONSTANTE PARA 2.5 SEG (53D = 35H)
MOV      ESPL,#35H
SJMP     SALMOD
TAR6:    CJNE      A,#06,SALMOD
MOV      ESPH,#00H     ;CONSTANTE PARA 2 SEG (43D = 2BH)
MOV      ESPL,#2BH
SALMOD:  RET

```

```

%INCLUDE <TIEMPO.ASM>
%INCLUDE <SERIAL.ASM>
%INCLUDE <ARITMETI.ASM>
%INCLUDE <DISPLAY.ASM>
%INCLUDE <DEF.ASM>
%INCLUDE <TTC.ASM>

```

```

;*****
;* MODULO:          "DISPLAY"
;* ARCHIVO:         DISPLAY.ASM
;* DESCRIPCION:     RUTINAS RELACIONADAS CON EL MANEJO DEL DISPLAY
;* FECHA:           3-OCT-93
;*****

```

```

;*****
*
;BUSY ESPERA QUE EL DISPLAY HAGA SUS OPERACIONES

```

```

BUSY:    MOV     A,DPH
         PUSH   ACC
         MOV    A,DPL
         PUSH   ACC
         MOV    DPTR,#BSYFLG
OCUPY:   MOVX   A,@DPTR
         RLC    A                ;ME QUEDO HASTA QUE EL DISPLAY SE DESOCUPE
         JC     OCUPY
         POP    ACC
         MOV    DPL,A
         POP    ACC
         MOV    DPH,A
         RET

```

```

;*****
*
;INST PROGRAMA AL DISPLAY LCD SEGUN EL PARAMETRO ACC

```

```

INST:    MOV     R7,A
         MOV    A,DPH
         PUSH   ACC
         MOV    A,DPL
         PUSH   ACC
         MOV    A,R7
         MOV    DPTR,#IR_LCD
         MOVX   @DPTR,A          ;PROGRAMO AL DISPOSITIVO
         CALL   BUSY             ;ESPERO QUE EL DISPLAY RECONOZCA LA PROGM.
         POP    ACC
         MOV    DPL,A
         POP    ACC
         MOV    DPH,A
         RET

```

```

;*****
*
;RUTINA QUE PRESENTA EN PANTALLA LO QUE LE MANDEN EN EL ACUMULADOR

```

```

DATOS:   MOV     R7,A                ;GUARDO EL PARAMETRO UN INSTANTE
         MOV    A,DPH
         PUSH   ACC
         MOV    A,DPL
         PUSH   ACC
         MOV    A,R7

```

```

R      MOV      DPTR,#DR_LCD
      MOVX     @DPTR,A      ;LE MANDO CARACTER EN ASCII
      CALL    BUSY         ;ESPERO QUE EL DISPLAY RECONOZCA EL CARACT

      POP     ACC
      MOV     DPL,A
      POP     ACC
      MOV     DPH,A
      RET

```

```

;*****
;RUTINA QUE MUESTRA EN EL DISPLAY UN BYTE EN "PACKED BCD" QUE SE LE PASA EN AC

```

```

DISPKBCD:MOV     R6,A      ;R6 VARIABLE AUXILIAR
          ANL     A,#11110000B ;NIBBLE ALTO
          SWAP    A
          ADD     A,#30H
          CALL    DATOS
          MOV     A,R6
          ANL     A,#00001111B ;NIBBLE BAJO
          ADD     A,#30H
          CALL    DATOS
          RET

```

```

;*****
*

```

```

;RUTINAS PARA PROGRAMAR AL DISPLAY EN DISTINTOS MODOS DE OPERACION

```

```

PDISA:  MOV     A,#01H      ;CLEAR DISPLAY
        CALL    INST
        MOV     A,#06H      ;SHIFT RIGHT
        CALL    INST
        MOV     A,#0CH      ;LINEA1 A LA IZQUIERDA
        CALL    INST
        MOV     A,#02H      ;RETURN HOME
        CALL    INST
        RET

```

```

PDISB:  MOV     A,#06H      ;SHIFT RIGHT
        CALL    INST
        MOV     A,#0C0H     ;LINEA2 A LA IZQUIERDA
        CALL    INST
        RET

```

```

PDISC:  MOV     A,#02H      ;RETURN HOME
        CALL    INST
        MOV     A,#01H      ;CLEAR DISPLAY
        CALL    INST
        MOV     A,#8FH      ;LINEA1 A LA DERECHA
        CALL    INST
        MOV     A,#07H      ;SHIFT LEFT
        CALL    INST
        RET

```

```

PDISD:  MOV     A,#02H           ;RETURN HOME
        CALL    INST
        MOV     A,#0CFH         ;LINEA2 A LA DERECHA
        CALL    INST
        MOV     A,#07H         ;SHIFT LEFT
        CALL    INST
        RET

```

```

;*****
;RUTINA QUE CONVIERTE DEL FORMATO (0-23) A FORMATO DE (1-12 PM/AM)
;LA HORA ESTA EN PACKED BCD

```

```

HORF12: CJNE    A,#12H,MENMAY
        SJMP    SAL_AM
MENMAY:  JC      SAL_AM           ;SI ES MENOR NO LE HAGO NADA
        CLR     C                ;LA RESTA TOMA EN CUENTA EL CARRY,LO BORRO
        SUBB   A,#12H           ;HORA FORMATO(0-23)-12=HORA FORMATO(1-12)
        MOV    HORACT,A
        MOV    AM__PM,#1H       ;ES PM
        SJMP   SALHRF
SAL_AM:  MOV     AM__PM,#0H       ;ES AM
SALHRF:  RET

```

```

;*****
*
;RUTINA PARA MOSTRAR AM O PM EN EL DISPLAY
;ACUMULADOR TIENE COMO PARAMETRO A LA VARIABLE AM__PM

```

```

DISAM__PM:CJNE  A,#1,AM          ;AM O PM ?
        MOV    A,#50H           ;P
        CALL   DATOS
        MOV    A,#4DH           ;M
        CALL   DATOS
        SJMP   SALDHO
AM:      MOV    A,#41H           ;A
        CALL   DATOS
        MOV    A,#4DH           ;M
        CALL   DATOS
SALDHO:  RET

```

```

;*****
*
;RUTINA PARA MOSTRAR LA HORA EN EL DISPLAY

```

```

DISHORA: MOV     R2,#3
ESPACI:  MOV     A,#20H           ;3 ESPACIOS EN BLANCO
        CALL    DATOS
        DJNZ   R2,ESPACI

        MOV    A,HORACT         ;HORA ACTUAL
        CALL   HORF12
        CALL   DISPKBCD

```

```
MOV    A,#3AH          ;DOS PUNTOS (:)
CALL   DATOS
```

```
MOV    A,MINACT        ;MINUTOS ACTUAL
CALL   DISPKBCD
```

```
MOV    A,#20H          ;ESPACIO EN BLANCO
CALL   DATOS
```

```
MOV    A,AM_PM         ;AM O PM
CALL   DISAM_PM
RET
```

```
*****
*
```

```
;RUTINA QUE MUESTRA EL DIA DE LA SEMANA EN LETRAS (LUN,MAR,...)
;PARAMETRO EN EL ACUMULADOR
```

```
DISDIA: CLR    C          ;LA RESTA TOMA EN CUENTA EL CARRY, LO BORRO
          SUBB   A,#1      ;DIA MENOS 1 PARA PODERLO USAR COMO INDICE
          MOV    B,#4
          MUL    AB        ;BUSCO EL DIA EN LA TABLA
          MOV    DPTR,#TAB_DIA
          ADD    A,DPL
          MOV    DPL,A     ;DPTR AL INICIO DEL DIA A PRESENTAR EN DISP
```

```
AY
          MOV    R2,#4H
AQUI:    MOVX   A,@DPTR    ;LAZO PARA MOSTRAR 4 LETRAS
          CALL   DATOS
          INC    DPTR
          DJNZ   R2,AQUI
          RET
```

```
*****
*
```

```
;RUTINA PARA MOSTRAR MES ACTUAL EN LETRAS
;EL MES ESTA EN PACKED BCD (___XXXX)
```

```
DISMES: SETB   C          ;VERIFICO SI ES UN MES ALTO (OCT,NOV,DIC)
          ANL   C,ACC.4    ;SI ES MES BAJO (ENE -SEP) NO HAY PROBLEMA
          JNC   NO_PROBL  ;SOLO NIBBLE BAJO
          ANL   A,#0FH     ;SUMO 10 PARA LOGRAR "MES ALTO"
          ADD   A,#0AH
NO_PROBL:CLR  C          ;LA RESTA TOMA EN CUENTA EL CARRY, LO BORR
          SUBB  A,#1      ;PARA USAR EL MES COMO INDICE
          MOV   B,#03H    ;3 ES LA CANTIDAD DE LETRAS DEL FORMATO(MM
```

```
)
          MUL   AB        ;EN EL ACUMULADOR ESTA EL OFFSET A SUMAR
          MOV   DPTR,#TAB_MES
          ADD   A,DPL
          MOV   DPL,A     ;DPTR AL INICIO DE MES A MOSTRAR
          MOV   R2,#3H
AQUI2:    MOVX  A,@DPTR   ;LAZO PARA MOSTRAR 3 LETRAS
```

```

CALL    DATOS
INC     DPTR
DJNZ   R2,AQUI2
RET

```

```

;*****
;RUTINA QUE MUESTRA LA FECHA EN EL DISPLAY
;FORMATO: Lun 1/Ene/94

```

```

DISFECHA:MOV    A,DSEACT          ;DIA DE LA SEMANA EN LETRAS
           CALL  DISDIA
           MOV   A,#20H          ;ESPACIO EN BLANCO
           CALL  DATOS
           MOV   A,DMESAC        ;DIA DEL MES ACTUAL
           CALL  DISPKBCD
           MOV   A,#2FH          ;(/)
           CALL  DATOS
           MOV   A,MESACT        ;MES ACTUAL EN LETRAS
           CALL  DISMES
           MOV   A,#2FH          ;(/)
           CALL  DATOS
           MOV   A,ANOACT        ;ANO ACTUAL
           CALL  DISPKBCD
           RET

```

```

;*****
;RUTINA QUE MUESTRA EN EL DISPLAY LA HORA Y LA FECHA EN EL DISPLAY

```

```

DIHR_FC: CALL  PDISA             ;PRIMERA LINEA EMPIEZO POR LA IZQUIERDA
           CALL  DISFECHA        ;FECHA EN PRIMERA LINEA DEL DISPLAY
           CALL  PDISB           ;SEGUNDA LINEA EMPIEZO POR LA DERECHA
           CALL  DISHORA         ;HORA EN LA SEGUNDA LINEA DEL DISPLAY
           RET

```

```

;*****
;RUTINA QUE MUESTRA EN EL DISPLAY UN MENSAJE DE 16 BYTES QUE SE ENCUENTRA EN
;LA RAM EXTERNA A PARTIR DE LA DIRECCION INICIAL DE DPTR
;ACUMULADOR TIENE QUE MENSAJE SE DESEA PRESENTAR (ACC: 0-FFH=256 MENSAJES)
;R6 ES UN PARAMETRO QUE INDICA LA LONGITUD DEL MENSAJE A MOSTRAR

```

```

MENLCD:  MOV   DPTR,#TAB_MEN
           MOV   B,#10H
           MUL   AB              ;BUSCO CUAL DE LOS MENSAJES
           ADD   A,DPL
           MOV   DPL,A

```

```
MOV      A,B
ADDC     A,DPH
MOV      DPH,A
MOST:    MOVX     A,@DPTR
          CALL    DATOS
          INC     DPTR
          DJNZ   R6,MOST
          RET
```

```
;DPTR AL INICIO DEL MENSAJE
```


El numer.

```

;*****
;* MODULO:          "ARITMETICA"
;* ARCHIVO:         ARITMETI.ASM
;* DESCRIPCION:     RUTINAS DE MANEJO DE OPERACIONES ARITMETICAS
;* FECHA:           3-OCT-93
;*****

```

```

;*****
*
;RUTINA PARA COLOCAR DPTR EN LA PAGINA ACTUAL
;TOMA COMO PARAMETRO LA VAR. GLOBAL J QUE ME INDICA QUE LINEA ESTOY ANALIZANDO

```

```

INI_PAG: MOV     DPTR,#TAB_VAR      ;DPTR AL INICIO DE LA TABLA
          MOV     A,#32             ;OFFSET PARA IR DE PAG. EN PAG.
          MOV     B,J               ;J ME DICE EN QUE LINEA ESTOY
          MUL     AB
          ADD     A,DPL
          MOV     DPL,A             ;SUMO AL BYTE BAJO DE DPTR
          MOV     A,B
          ADDC    A,DPH             ;SUMO AL BYTE ALTO DE DPTR
          MOV     DPH,A
          RET

```

Search

```

;*****
*
;RUTINA PARA BUSCAR EN LA TABLA DE LOS CODIGOS NACIONALES EL CODIGO DE DISTANC
A
;CORRESPONDIENTE A LA CIUDAD DE VENEZUELA DESTINO.

```

```

BUSC_TAB:MOV     DPTR,#TAB_NAC
LEE_CODI:MOVX    A,@DPTR
          CJNE    A,#0FFH,VERIFY    ;VERIFICO SI ES EL FIN DE LA LISTA
          SJMP    SALBSQ
VERIFY:  CJNE    A,CODH,PRX_CITY
          INC     DPTR
          MOVX    A,@DPTR
          CJNE    A,CODL,PRX_CTY
          INC     DPTR
          MOVX    A,@DPTR           ;ACUMULADOR CON CODIGO DE DISTANCIA
          SJMP    SALBSQ
PRX_CITY:INC     DPTR
PRX_CTY:  INC     DPTR
          INC     DPTR
          SJMP    LEE_CODI
SALBSQ:  RET

```

Tomo fo

```

;*****
*
;RUTINA RECUPERAR LAS VARIABLES DE UNA LINEA NECESARIAS PARA EL MAIN
;VARIABLES RECUPERADAS: ESPH, ESPL, FLAGS, ST_VIEJO, NMAR.

```

```

RECUPERA:CALL    INI_PAG           ;DPTR AL INICIO DE PAGINA ADECUADA
          MOVX    A,@DPTR
          MOV     ESPH,A           ;RECUPERO ESPH

```

```

INC     DPTR
MOVX   A,@DPTR
MOV    ESPL,A           ;RECUPERO ESPL
INC    DPTR
INC    DPTR
MOVX   A,@DPTR
MOV    FLAGS,A         ;RECUPERO FLAGS
INC    DPTR
MOVX   A,@DPTR
MOV    ST_VIEJO,A      ;RECUPERO ST_VIEJO
INC    DPTR
MOVX   A,@DPTR
MOV    NMAR,A          ;RECUPERO NMAR
RET

```

```

;*****
*

```

```

;RUTINA PARA DEVOLVER LAS VARIABLES DE UNA LINEA DESPUES DE ANALIZADAS
;VARIABLES DEVUELTAS: ESPH,ESPL,FLAGS,ST_VIEJO <-- STATUS,NMAR,C_REP,C_RBK

```

```

DEVOLVER:CALL    INI_PAG
           MOV    A,ESPH           ;GUARDO ESPH
           MOVX   @DPTR,A
           INC    DPTR
           MOV    A,ESPL
           MOVX   @DPTR,A         ;GUARDO ESPL
           INC    DPTR
           INC    DPTR
           MOV    A,FLAGS
           MOVX   @DPTR,A         ;GUARDO FLAGS
           INC    DPTR
           MOV    A,STATUS
           MOVX   @DPTR,A         ;GUARDO STATUS
           INC    DPTR
           MOV    A,NMAR
           MOVX   @DPTR,A         ;GUARDO NMAR
           INC    DPTR
           MOV    A,C_REP
           MOVX   @DPTR,A         ;GUARDO C_REP
           INC    DPTR
           MOV    A,C_RBK
           MOVX   @DPTR,A         ;GUARDO C_RBK
           RET

```

```

;*****
*

```

```

;RUTINA PARA RECUPERAR VARIABLES DE UNA LINEA ASOCIADAS A LA CADENCIA
;PARAMETRO: LINEA A RECUPERAR (0-23) EN EL ACUMULADOR

```

```

RECUPCAD:MOV    DPTR,#TAB_VAR
           MOV    B,#32
           MUL   AB

```

;OPERACION: TAB_VAR + A*32

```

      ADD      A,DPL
      MOV      DPL,A
      MOV      A,B
      ADDC    A,DPH
      MOV      DPH,A
REC_VAR: MOVX   A,@DPTR
      MOV      ESPH,A
      INC     DPTR
      MOVX   A,@DPTR
      MOV      ESPL,A
      INC     DPTR
      MOVX   A,@DPTR
      MOV      C50MSCH,A
      INC     DPTR
      MOVX   A,@DPTR
      MOV      C50MSCL,A
      INC     DPTR
      MOVX   A,@DPTR
      MOV      FLAGS,A
      RET

```

```

;*****
*

```

```

;RUTINA QUE DEVUELVE LAS VARIABLES ASOCIADAS A LA CADENCIA DE UNA LINEA
;R4 ES LE PARAMETRO QUE INDICA CUAL LINEA LE VOY A DEVOLVER LAS VARIABLES (0-2
)

```

```

DEVOLCAD: MOV    DPTR,#TAB_VAR
      MOV    B,#32
      MUL   AB
      ADD   A,DPL      ;OPERACION: TAB_VAR + A*32
      MOV   DPL,A
      MOV   A,B
      ADDC  A,DPH
      MOV   DPH,A
DEV_VAR:  MOV   A,ESPH
      MOVX  @DPTR,A
      INC  DPTR
      MOV   A,ESPL
      MOVX  @DPTR,A
      INC  DPTR
      MOV   A,C50MSCH
      MOVX  @DPTR,A
      INC  DPTR
      MOV   A,C50MSCL
      MOVX  @DPTR,A
      INC  DPTR
      MOV   A,FLAGS
      MOVX  @DPTR,A
      RET

```

```

;*****
*

```

```
;RUTINA PARA INCREMENTAR EL CONTADOR DE 50MS DE 2 BYTES
```

```
INCC50MS:MOV      R0,#C50MSCL
             MOV      R1,#C50MSCH
             CALL     INC2BYTE      ;INCREMENTO EL CONTADOR DE 50MS DE 2 BYTES
             RET
```

```
;*****
*
```

```
;RUTINA PARA INCREMENTAR DOS BYTES
;LA DIRECCION DEL BYTE BAJO SE PASA EN R0
;LA DIRECCION DEL BYTE ALTO SE PASA EN R1
```

```
INC2BYTE:MOV      A,#1
             ADD      A,@R0      ;INCREMENTO EL BYTE BAJO
             MOV      @R0,A      ;GUARDO EL RESULTADO
             MOV      A,#0
             ADDC     A,@R1      ;INCREMENTO EL BYTE ALTO,FIJANDONOS EN CAR
Y
             MOV      @R1,A      ;GUARDO EL RESULTADO
SALINC:  RET
```

```

;*****
;* MODULO:          "SERIAL"
;* ARCHIVO:         SERIAL.ASM
;* DESCRIPCION:     RUTINAS RELACIONADAS CON EL MANEJO DEL PUERTO SERIAL
;* FECHA:          3-OCT-93
;*****

```

```

;*****
;INTERRUPCION DE LA COMUNICACION SERIAL

```

```

SERIAL:  MOV     A,SBUF           ;TOMO EL DATO QUE ME LLEGO
         RETI

```

```

;*****
;RUTINA QUE ESPERA QUE EL DATO A TRANSMITIR SEA MANDADO POR COMPLETO

```

```

DELAY:  MOV     C, TI
         JNC    DELAY
         CLR    TI
         RET

```

```

;*****
;RUTINA QUE MANDA POR EL PUERTO SERIE UN NUMERO QUE ESTA EN LA DIRECCION DE
;LA MEMORIA INTERNA QUE APUNTA R0.
;PARAMETRO: R0 DIRECCION DE MEMORIA DE LO QUE SE QUIERE MANDAR

```

```

SER_DAT: MOV     DPTR, #1700H
         MOV     A, R0
         ADD    A, DPL
         MOV    DPL, A
         MOVX   A, @DPTR
         MOV    SBUF, A
         CALL   DELAY
         RET

```

```

;*****
;RUTINA QUE MANDA POR EL PUERTO SERIE MENSAJES GUARDADOS A PARTIR DE DPTR
;PARAMETRO: DPTR DIRECCION DE INICIO DEL MENSAJE

```

```

MENSER:  MOV     R2, #0FH
MANDAR:  MOVX   A, @DPTR
         MOV    SBUF, A
         CALL   DELAY
         INC    DPTR
         DJNZ   R2, MANDAR
         RET

```

```

;*****
;RUTINA QUE RECIBE LOS PARAMETROS DE PROGRAMACION POR EL PUERTO SERIAL

```

```

INICIO:  CALL    CARAC
         CALL    DATO           ;# DE DATOS
         JZ     FIN
         MOVX   @DPTR, A       ;ESCRIBO EN RAM EXTERNA

```

```
RET
```

```
*****
;FUNCION: INICIALIZAR EL PUERTO SERIE Y EL TIMER 1
```

```
INIT:  MOV     SCON,#70H ;MODO 1, 8 BIT UART, VELOCIDAD VARIABLE
        MOV     TMOD,#20H ;TIMER 1 MODO 2 (TEMPORIZADOR CON AUTORECARGA)
        MOV     TCON,#40H ;HABILITAR EL TIMER 1
        MOV     TH1,#0F3H ;CONSTANTE PARA LOGRAR LOS 2400 BAUDIOS
        MOV     TL1,TH1  ;CUBRIR LA PRIMERA PASADA
        RET
```

```
*****
;FUNCION: ASCII-HEXADECIMAL
;ENTRADA: ACUMULADOR
;SALIDA: ACUMULADOR
```

```
ASCHEX: CJNE A,#40H,LETNUM ;LETRA O NUMERO
LETNUM: JNC  LETRAS
        CLR  C
        SUBB A,#30H
        RET
LETRAS: CLR  C
        SUBB A,#37H
        RET
```

```
*****
;FUNCION: RECEPCION DE UN CARACTER POR EL PUERTO SERIAL
;ENTRADA: NINGUNA
;SALIDA: ACUMULADOR
```

```
CARAC:  JNB     RI,CARAC ;LLEGO UN CARACTER?
        CLR     RI
        MOV     A,SBUF
        RET
```

```
*****
;FUNCION: DEJAR EN EL ACUMULADOR EL DATO A GRABAR EN RAM
;ENTRADA: NINGUNA
;SALIDA: ACUMULADOR
```

```
DATO:  CALL     CARAC
        CALL     ASCHEX
        SWAP    A
        MOV     R1,A
        CALL     CARAC
        CALL     ASCHEX
        ORL     A,R1
        RET
```

```
*****
*
;FORMA EN QUE LLEGAN LOS DATOS POR EL PUERTO SERIAL
;FRECUENCIA DE LOS IMPULSOS DE COMPUTO
;LLEGARAN LAS CONSTANTES DE RECARGA DEL TIMER1:
```

```
;50 HZ ----> D900h  
;12 KHZ ----> FFE0h  
;16 KHZ ----> FFEAH
```

```
;NUMERO DE LINEAS
```

```
;1 BYTE QUE MANDE EL NUMERO DE LINEAS (8, 16 o 24)
```

```
;HORA
```

```
;1 BYTE HORAS
```

```
; *NIBBLE HIGH DECENAS DE HORA
```

```
; *NIBBLE LOW UNIDADES DE HORA
```

```
;1 BYTE MINUTOS
```

```
; *NIBBLE HIGH DECENAS DE MINUTO
```

```
; *NIBBLE LOW UNIDADES DE MINUTO
```

```
;1 BYTE SEGUNDOS
```

```
; *NIBBLE HIGH DECENAS DE SEGUNDOS
```

```
; *NIBBLE LOW UNIDADES DE SEGUNDOS
```

```
;EJEMPLO: 12d ----> 0001 0010
```

```
;FECHA
```

```
;1 BYTE DIA
```

```
; *NIBBLE HIGH DECENAS DE DIA
```

```
; *NIBBLE LOW UNIDADES DE DIA
```

```
;1 BYTE MES
```

```
; *NIBBLE HIGH DECENAS DE MES
```

```
; *NIBBLE LOW UNIDADES DE MES
```

```
;1 BYTE AÑO
```

```
; *NIBBLE HIGH DECENAS DE AÑO
```

```
; *NIBBLE LOW UNIDADES DE AÑO
```

```
❖
```

```

;*****
;* MODULO:          "TIEMPO"
;* ARCHIVO:         TIEMPO.ASM
;* DESCRIPCION:    RUTINAS RELACIONADAS CON LA TEMPORIZACION
;* FECHA:          3-OCT-93
;*****

```

```

;*****
;INTERRUPCION EXTERNA 0
;EL REAL TIME CLOCK INTERRUMPE AL MICROPROCESADOR CADA MINUTO PARA ACTUALIZAR
;LA HORA.

```

```

EXT_0:  MOV     DPTR,#C_HORA      ;ACTUALIZO LA HORA Y AL FECHA
        MOVX   A,@DPTR
        MOV    HORACT,A
        MOV    DPTR,#C_MINU
        MOVX   A,@DPTR
        MOV    MINACT,A
        MOV    DPTR,#C_DMES
        MOVX   A,@DPTR
        MOV    DMESAC,A
        MOV    DPTR,#C_MES
        MOVX   A,@DPTR
        MOV    MESACT,A
        MOV    DPTR,#C_DSEM
        MOVX   A,@DPTR
        MOV    DSEACT,A

```

```

ACT:    CALL    DIHR_FC
        MOV    DPTR,#C_STAT      ;REGISTRO DE STATUS BIT
        MOVX   A,@DPTR          ;LEO EL BIT DE STATUS
        MOV    DPTR,#C_INTS     ;REGISTRO DEL STATUS DE LAS INTERRUPCIONES
        MOVX   A,@DPTR          ;DESHABILITO LA INTERRUPCION
        RETI

```

```

;*****
;INTERRUPCION DEL TIMER1
;DEPENDIENDO DE LA PROGRAMACION INICIAL GENERA ONDAS DE 50 HZ, 12 KHZ O 16 KHZ

```

```

TIM__1:  MOV     TL1,TEMPL
        MOV     TH1,TEMPH
        CPL    P1.4
SATIM1:  RETI

```

```

;*****
;INTERRUPCION DEL TIMER0
;INTERRUMPE CADA 50MS

```

```

TIM__0:  MOV     TLO,#000H
        MOV     TH0,#03DH      ;RECARGO CONSTANTES PARA LOS 50 MS
        CALL    VENTANA        ;ACTUALIZO VENTANA DE SEÑAL DE COMPUTO

```



```

MOV      A,NO_LIN
CJNE    A,#1,SALTIO ;HAGO ESTO SI SOLO TENGO UNA LINEA
CALL    DURAC        ;ACTUALIZO DURACION DE LLAMADA
CALL    PDISC
CALL    DISDURC      ;MUESTRO EN EL DISPLAY DURACION DE LLAMADA
CALL    PDISD
CALL    DISCOST      ;MUESTRO EN EL DISPLAY COSTO DE LLAMADA
SALTIO: RETI

```

;RUTINA QUE ACTUALIZA LA "VENTANA" PARA LA SEÑAL DE COMPUTO

```

VENTANA: MOV      R2,#0H ;R2 CONTADOR PARA 3 TARJETAS DE 8 LI.(0,1,
)
MOV      R4,#0H ;R4 CONTADOR SECUENCIAL 24 LINEAS (0 - 23)
SIG_TARJ:MOV     R3,#8H ;R3 CONTADOR PARA 8 LI. (8,7,6,5,4,3,2,1)
SIG_LIN: MOV     A,R4 ;EN ACC LINEA A RECUPERAR
CALL    RECUPCAD ;RECUPERO VARIABLE ASOCIADAS A LA CADENCIA
IMP?:  MOV     A,FLAGS ;HAY QUE APLICARLE IMPULSOS A ESTA LINEA ?
RRC     A ;SI C=1 VEO POR DONDE VOY EN LA CADENCIA
JNC     CBITNO_C ;SI NO BORRO EL BIT PARA CONSERVAR POSICI

```

```

N
MOV     A,C50MSCH ;ANALIZO CONT. DE 50MS. DE CADENCIA HIGH
CJNE   A,#0,VER_ESP ;C50MSCH > 0 VERIFICO ESPERA
MOV     A,C50MSCL ;ANALIZO CONT. DE 50MS. DE CADENCIA LOW
CJNE   A,#3H,NO_IGU ;VERIFICO SI PASARON 150MS.
SJMP   CLR_BIT ;SI C50MSCL = 3 --> EMPIEZA LA ESPERA
NO_IGU: JC     SET_BIT ;SI C50MSCL < 3 --> PONGO IMPULSO
VER_ESP: MOV    A,C50MSCH ;SI C50MSCL > 3 --> VERIFICO ESPERA
CJNE   A,ESPH,CLR_BIT ;VERIFICO LA PARTE HIGH DEL C50MSCH
MOV     A,C50MSCL
CJNE   A,ESPL,CLR_BIT ;SI C50MSCL < ESPL --> ESTOY EN LA ESPERA
MOV     C50MSCL,#0 ;SI C50MSCH = ESPL --> SE ACABO LA ESPERA
MOV     C50MSCH,#0 ;EMPIEZO DE NUEVO EL CICLO BIT = 1

```

```

SET_BIT: MOV     R5,#1 ;SET BIT
MOV     R6,#1 ;INCREMENTO CONTADOR DE 50 MS
CALL    ACT_BIT
SJMP   INCONT

```

```

CLR_BIT: MOV     R5,#0 ;CLEAR BIT
MOV     R6,#1 ;INCREMENTO CONTADOR DE 50 MS
CALL    ACT_BIT
SJMP   INCONT

```

```

CBITNO_C:MOV    R6,#0 ;CLEAR BIT
MOV     R5,#0 ;NO INCREMENTO CONTADOR DE 50 MS

```

```

INCONT: CALL    V4_5S ;Verifico si debo aplicar impulsos.
MOV     A,R4
CALL    DEVOLCAD ;DEVUELVO VARIABLES DE LA LINEA DE CADENCI

```

```

INC     R4
DJNZ   R3,SIG_LIN
INC     R2
MOV     A,R2

```

```

CJNE      A,NO_LOOP,SIG_TARJ ;NO_LOOP (1,2,3), DEPENDE 8,16,24 LINEAS

MOV       A,V_IMP1           ;LAS VARIABLES DE IMPULSOS CONFORMADAS...
MOV       DPTR,#CAD1         ;LAS PONGO EN LOS TRES REGISTROS
MOVX      @DPTR,A
MOV       A,V_IMP2
MOV       DPTR,#CAD2
MOVX      @DPTR,A
MOV       A,V_IMP3
MOV       DPTR,#CAD3
MOVX      @DPTR,A

```

```
SALVEN:   RET
```

```

;*****
*
;RUTINA PARA ACTUALIZAR LAS CADENCIAS DE LAS LINEAS
;PARAMETROS: R2--> R2=0 ACTUALIZO V_IMP1
;                R2=1 ACTUALIZO V_IMP2
;                R2=2 ACTUALIZO V_IMP3
;
;                R5--> R5=0 CLEAR BIT
;                R5=1 SET BIT
;
;                R6--> R6=0 INCREMENTO CONTADOR DE 50 MS
;                R6=1 NO INCREMENTO CONTADOR DE 50 MS

```

```

ACT_BIT:  MOV     R0,#V_IMP1      ;R0 APUNTADOR A LA VARIABLE IMPULSOS 1
          MOV     A,R0
          ADD     A,R2            ;V_IMP1 + LOOP ACTUAL
          MOV     R0,A           ;R0 APUNTA A LA VARIABLE IMPULSOS ADECUADA
          MOV     A,R5
          CJNE   A,#1,CARRY0     ;ANALIZO SI ES CLR_BIT O SET_BIT
          SETB   C
          SJMP   ROTA
CARRY0:   CLR     C
ROTA:     MOV     A,@R0          ;TOMO LA VARIABLE DE IMPULSOS PARA ROTARLA
          RRC     A             ;ROTO PARA COLOCAR EL BIT DONDE CORRESPOND
          MOV     @R0,A         ;GUARDO LA MODIFICACION
          MOV     A,R6
          CJNE   A,#01H,SALACT   ;ANALIZO SI INCREMENTAR O NO CONTADOR 50MS
          CALL   INCC50MS
SALACT:   RET

```

```

;*****
*
;RUTINA PARA ACTUALIZAR C_REP Y C_RBK Y TOMAR DECISIONES SI SOBREPASO 4.5 SEG
5,5
V4_5S:   MOV     A,FLAGS
          RRC     A
          RRC     A
          JNC    RINGBK
          INC    C_REP
          MOV     A,C_REP
          ; Si FLAG REP = 0 SALTO
          ; Si FLAG REP = 1, INCREMENTO CREP

```

```

CJNE A,#90,SAL4_5
MOV  FLAGS,#0
MOV  C_REP,#0
SJMP SAL4_5
RINGBK: RRC A
JNC  SAL4_5
INC  C_RBK
MOV  A,C_RBK
CJNE A,#90,SAL4_5
MOV  A,#0
ANL  A,FLAGS
MOV  FLAGS,A
MOV  A,#0000001B
ORL  A,FLAGS
MOV  FLAGS,A
SAL4_5: RET

```

NO
NO
 ;SI PASAN MAS DE 4.5 SEG ENTRANTE FRACASO
 ; SI FLAG DE RING-BACK=0 ME SAUO
 ; SI FLAG RBK = 1, INCREMENTO CRBK
 ;SI PASAN MAS DE 4.5 SEG SALIENTE EXITOSA
 ;BORRO LOS FLAGS ANTERIORES
 ;PONGO FLAG IMPULSOS EMPIEZA TARIFICACION

```

;*****
;RUTINA QUE REALIZA RETARDO POR SOFTWARE

```

```

RETARDO: PUSH ACC
MOV  A,R0
PUSH ACC
MOV  A,R1
PUSH ACC
MOV  A,R2
PUSH ACC
MOV  R0,#44H
SAL2: MOV  R1,#20H
SAL3: MOV  R2,#0FFH
SAL4: DJNZ R2,SAL4
      DJNZ R1,SAL3
      DJNZ R0,SAL2
      POP ACC
      MOV  R2,A
      POP ACC
      MOV  R1,A
      POP ACC
      MOV  R0,A
      POP ACC
      RET

```

3. RUTINAS ADICIONALES PARA EL TTC

```

;*****
;* MODULO:          "TARIFICADOR TELEFONICO CASERO"
;* ARCHIVO          TTC.ASM
;* DESCRIPCION:     RUTINAS ADICIONALES NECESARIAS PARA EL TTC
;* FECHA:           3-OCT-93
;*****

```

```

;*****
;*                DEFINICION DE VARIABLES
;*****

```

```

;1) VARIABLES ASOCIADAS A LA DURACION DE LA LLAMADA

```

```

USEG:    EQU        25H          ;UNIDADES SEGUNDOS
DSEG:    EQU        26H          ;DECENAS SEGUNDOS
UMIN:    EQU        27H          ;UNIDADES MINUTOS
DMIN:    EQU        28H          ;DECENAS MINUTOS
UHOR:    EQU        29H          ;UNIDADES HORAS
DHOR:    EQU        2AH          ;DECENAS HORAS

C_50MS:  EQU        5EH          ;CONTADOR DE 50MS PARA DURACION LLAMADA

```

```

;2) VARIABLES PARA 14 DIGITOS EN PACKED BCD

```

```

DIS_MN:  EQU        30H          ;COSTO DE LLAMADA EN "PACKED BCD"
DIS_KL:  EQU        31H          ;MAXIMO COSTO = 10,994,947,850.25 BS.
DIS_IJ:  EQU        32H
DIS_GH:  EQU        33H
DIS_EF:  EQU        34H
DIS_CD:  EQU        35H
DIS_AB:  EQU        36H

```

```

;3) VARIABLES PARA COMPUTO DE COSTO DE UNA LLAMADA

```

```

COSTO5:  EQU        37H          ;COSTO DE LA LLAMADA EN HEXADECIMAL
COSTO4:  EQU        38H
COSTO3:  EQU        39H
COSTO2:  EQU        3AH
COSTO1:  EQU        3BH

CT_IM3:  EQU        3CH          ;COSTO DEL IMPULSO
CT_IM2:  EQU        3DH          ;MAXIMO COSTO IMPULSO = 167,772.15 BS.
CT_IM1:  EQU        3EH

NM_IM2:  EQU        3FH          ;NUMERO DE IMPULSOS
NM_IM1:  EQU        40H          ;MAXIMO NUMERO DE IMPULSOS = 65,535

```

```

INDICE: EQU      41H          ;VARIABLE AUXILIAR PARA MULTIPLICACION
TAB_REG: EQU     B000H       ;TABLA DEL REGISTRO DE LLAMADAS

```

```

;*****
*
;REGISTRO DE LLAMADAS
;El Equipo manda hacia el PC una tabla que tiene la siguiente estructura:
;* 1 byte Horas
;* 1 byte Minutos
;* 1 byte Dia
;* 1 byte Mes
;* 1 byte Ano
;* 1 byte Duracion Horas
;* 1 byte Duracion Minutos
;* 1 byte Duracion Segundos
;* 7 bytes Costo (DIS_MN-DIS_AB)
;* 1 byte vacio
;* 16 bytes Numeros marcados
;TOTAL DE CADA REGISTRO = 32 bytes
;MAXIMO NUMERO DE REGISTROS = 100

;* El formato de los bytes de Costo es "Packed BCD" (Cada Nibble contiene un
;numero, es decir se manda 14 digitos para el costo de la llamada.
;* Los numeros marcados es un byte por cada Numero

```

```

;*****
;*          DISPLAY          *
;*****

;*****
;RUTINA QUE MUESTRA EN EL DISPLAY EL NUMERO QUE MARCA EL USUARIO

```

```

DISNUM:  MOV     DPTR,#STATUS    x } MOV 4,STATUS
        MOVX   A,@DPTR
        ANL   A,#0FOH
        SWAP  A
        CJNE  A,#0AH,AST        ;CASO DE QUE SE HAYA MARCADO UN CERO
        MOV   A,#00H
        SJMP  HEXASC
AST:     CJNE  A,#0BH,NUM        ;CASO DE QUE MARQUEN ASTERISCO (*)
        MOV   A,#2AH            ;ACUMULADOR=CODIGO ASCII DEL ASTERISCO
        SJMP  MOSDIS
NUM:     CJNE  A,#0CH,HEXASC     ;CASO DE QUE MARQUEN NUMERAL (#)
        MOV   A,#23H            ;ACUMULADOR=CODIGO ASCII DEL NUMERAL
        SJMP  MOSDIS
HEXASC:  ADD   A,#30H
MOSDIS:  CALL  DATOS            ;MUESTRO NUMERO MARCADO EN LCD
SALDNM:  RET

```

```

;*****

```

```

;RUTINA AUXILIAR PARA MOSTRAR EN EL DISPLAY DIGITOS DEL COSTO DE LA LLAMADA
;EL COSTO DE LA LLAMADA ESTA EN PACKED BCD
;PARAMETROS: R1 = BYTE DE INICIO DE LOS DATOS QUE SE DESEAN MOSTRAR
;             R2 = NUMERO DE BYTES A PROCESAR

```

```

BCD_DIS: MOV     A,R2
          JZ      SALBCD           ;SI # BYTES = 0 NO MUESTRO NADA
          MOV     A,@R1
          CALL    DISPKBCD
          INC     R1               ;SIGUIENTE BYTE
          DJNZ    R2,BCD_DIS      ;SIGO MOSTRANDO HASTA QUE SE ACABEN LOS BY
ES
SALBCD:  RET

```

```

;*****
;RUTINA QUE DESCARTA LOS CEROS QUE ESTAN POR DELANTE EN EL COSTO DE UNA
;LLAMADA, PARA NO MOSTRARLOS EN EL DISPLAY.
;ESTA RUTINA ACTUA SOBRE R1 Y R2 PARA QUE SEAN LOS ADECUADOS AL LLAMAR BCD_DIS

```

```

QT_CERO: MOV     A,@R1
          ANL     A,#11110000B    ;NIBBLE ALTO
          SWAP   A                ;ROTO 4 BITS A LA DERECHA
          JNZ    SALQTC
NI_BAJ:  MOV     A,@R1
          ANL     A,#00001111B
          INC     R1               ;PROXIMO BYTE
          DEC     R2               ;2 DIGITOS MENOS
          JZ      ULM?            ;SI NIBLE BAJO=0 ME PREGUNTO SI ES EL ULTI
O
          ADD     A,#30H           ;SI EL NIBBLE BAJO NO ES CERO
          CALL    DATOS            ;LO MUESTRO
          SJMP   SALQTC           ;Y ME SALGO,CONTINUO CON LOS DEMAS DIGITOS
ULM?:    CJNE   R2,#00H,QT_CERO  ;SI NO ES EL ULTIMO SIGO ELIMINANDO CEROS
          MOV     A,#30H           ;SI ES EL ULTIMO LO MUESTRO
          CALL    DATOS
SALQTC:  RET

```

```

;*****
*
;RUTINA PARA MOSTRAR EN EL DISPLAY BS.

```

```

DIS_BS:  MOV     A,#20H           ;ESPACIO EN BLANCO
          CALL    DATOS
          MOV     A,#42H           ;PARA QUE APAREZCA LA UNIDAD MONETARIA
          CALL    DATOS
          MOV     A,#53H           ;BS. = BOLIVARES
          CALL    DATOS
          MOV     A,#2EH
          CALL    DATOS
          RET

```

```

;*****

```

;MUESTRO EN EL DISPLAY COSTO DE UNA LLAMADA
;FORMATO: XXXXXXXXXXXX,XX BS.

```
DISCOST: MOV      R1,#DIS_MN      ;COMIENZO DE VARIABLES DE COSTO DE LLAMADA
          MOV      R2,#6H        ;6 BYTES BCD PACKED (PARTE ENTERA)
          CALL     QT_CERO       ;QUITO CEROS MSB, AFECTA R1 Y R2
          CALL     BCD_DIS       ;MUESTRO EN EL DISPLAY

          MOV      A,#2CH        ;LA COMA DEL COSTO DE LA LLAMADA
          CALL     DATOS

          MOV      R1,#DIS_AB     ;DIGITOS DECIMALES
          MOV      R2,#1H        ;1 BYTE BCD PACKED (PARTE DECIMAL)
          CALL     BCD_DIS       ;MUESTRO EN EL DISPLAY
          CALL     DIS_BS
          CALL     BLANC         ;BORRO LAS VARIABLES BCD
          RET
```

;*****
;MUESTRA POR EL DISPLAY LA DURACION DE UNA LLAMADA

```
DISDURC: MOV      A,DHOR         ;DECENAS DE LAS HORAS
          ADD      A,#30H
          CALL     DATOS

          MOV      A,UHOR        ;UNIDADES DE LAS HORAS
          ADD      A,#30H
          CALL     DATOS

          MOV      A,#3AH        ;DOS PUNTOS
          CALL     DATOS

          MOV      A,DMIN        ;DECENAS DE LOS MINUTOS
          ADD      A,#30H
          CALL     DATOS
          ADD      A,#30H
          CALL     DATOS

          MOV      A,#3AH        ;DOS PUNTOS
          CALL     DATOS

          MOV      A,DSEG        ;DECENAS DE LOS SEGUNDOS
          ADD      A,#30H
          CALL     DATOS

          MOV      A,USEG        ;UNIDADES DE LOS SEGUNDOS
          ADD      A,#30H
          CALL     DATOS
          RET
```

;*****
;RUTINA QUE MUESTRA COSTO Y DURACION DE LA LLAMADA EN EL DISPLAY
;FORMATO: HH:MM:SS
;XXXXXXXXXXXX,XX BS.

```
DI_DUCST:CALL    PDISA
             CALL    DISDURC
             CALL    PDISB
             CALL    DISCOST
             RET
```

```
;*****
;*          CALCULOS
;*****
;*****
;RUTINA DE INICIALIZACION DE VARIABLES DE COSTO Y DURACION DE LLAMADA
```

```
INI_VAR: MOV     DHOR,#00H           ;DURACION DE LA LLAMADA
          MOV     UHOR,#00H
          MOV     DMIN,#00H
          MOV     UMIN,#00H
          MOV     DSEG,#00H
          MOV     USEG,#00H
          MOV     AM_PM,#00H
          MOV     CT_IM3,#00H       ;COSTO DE UN IMPULSO (3 BYTES)
          MOV     CT_IM2,#00H
          MOV     CT_IM1,#100
          MOV     NM_IM2,#00H       ;NUMERO DE IMPULSOS RECIBIDOS
          MOV     NM_IM1,#00H
          MOV     DIS_MN,#000H      ;NUMEROS EN EL DISPLAY EN DECIMAL
          MOV     DIS_KL,#000H      ;CADA BYTE CONTIENEN 2 DIGITOS (PACKED BCD)
          MOV     DIS_IJ,#000H
          MOV     DIS_GH,#000H
          MOV     DIS_EF,#000H
          MOV     DIS_CD,#000H
          MOV     DIS_AB,#000H
          MOV     COSTO1,#000H      ;COSTO DE LLAMADA=NUMERO IMP.*COSTO IMP.
          MOV     COSTO2,#000H
          MOV     COSTO3,#000H
          MOV     COSTO4,#000H
          MOV     COSTO5,#000H
          RET
```

```
;*****
;RUTINA QUE HACE UN BLANQUEO DE TODAS LAS POSICIONES PACKED BCD
```

```
BLANC:  MOV     DIS_MN,#000H
          MOV     DIS_KL,#000H
          MOV     DIS_IJ,#000H
          MOV     DIS_GH,#000H
          MOV     DIS_EF,#000H
          MOV     DIS_CD,#000H
          MOV     DIS_AB,#000H
          RET
```

```
;*****
;MUL_AB:  MULTIPLICA "COSTO DEL IMPULSO" * "NUMERO DE IMPULSOS"
;RESULTADO: "COSTO DE LLAMADA" = (5 BYTES)
```



```

MUL_AB:  MOV     R2,#0           ;CONTADOR DEL MULTIPLICANDO
         MOV     R3,#0           ;CONTADOR DEL MULTIPLICADOR
MUL:     MOV     R0,#CT_IM1      ;R0 APUNTADOR A COSTO IMPULSO
         MOV     R1,#NM_IM1      ;R1 APUNTADOR A NUMERO IMPULSOS
         MOV     A,R0
         CLR     C
         SUBB   A,R2
         MOV     R0,A           ;R0 - R2 BYTE DEL MULTIPLICANDO ACTUAL
         MOV     A,@R0          ;TOMO EL BYTE
         MOV     R4,A           ;R4 TEMPORAL BYTE DEL MULTIPLICANDO ACTUAL
         MOV     A,R1
         CLR     C
         SUBB   A,R3
         MOV     R1,A           ;R1 - R3 BYTE DEL MULTIPLICADOR ACTUAL
         MOV     A,@R1          ;TOMO EL BYTE
         MOV     R5,A           ;R5 TEMPORAL BYTE DEL MULTIPLICADOR ACTUAL
         MOV     A,R4
         MOV     B,R5
         MUL    AB             ;MULTIPLICICO COSTO * NUMERO DE IMPULSOS
         MOV     R7,A           ;R7 TEMPORAL DE RESULTADO LSB
         MOV     R6,B           ;R6 TEMPORAL DE RESULTADO MSB
         MOV     A,R2
         ADD    A,R3           ;R2+R3=INDICE ME INDICA DONDE DEBO SUMAR
         MOV     INDICE,A
         MOV     A,#COSTO1
         CLR     C
         SUBB   A,INDICE      ;R1 - INDICE = LUGAR DONDE SUMO
         MOV     R1,A
         MOV     A,@R1          ;TOMO VALOR ANTERIOR
         ADD    A,R7           ;COSTOX + RESULTADO DE LA MULT. LSB
         MOV     @R1,A         ;GUARDO RESULTADO
         DEC    R1
         MOV     A,@R1          ;TOMO VALOR ANTERIOR
         ADDC   A,R6           ;COSTOX + RESULTADO DE LA MULT.MSB + C
         MOV     @R1,A         ;GUARDO RESULTADO
         JNC    SEGUIR        ;SI NO HUBO CARRY CONTINUO EL ALGORITMO
         DEC    R1             ;SI HUBO CARRY SUMO 1 AL SIGUIENTE BYTE
         MOV     A,#0H
         ADDC   A,@R1          ;LE SUMO EL CARRY
         MOV     @R1,A
SEGUIR:  INC     R2             ;ACTUALIZO INDICE DEL MULTIPLICANDO
         MOV     A,R2
         CJNE  A,#3,MUL        ;VERIFICO SI YA SE REALIZO EL LOOP
         MOV     R2,#0
         INC   R3
         MOV     A,R3
         CJNE  A,#2,MUL        ;VERIFICO SI YA SE REALIZO EL LOOP EXTERIO
         RET

```

```

;*****
;REALIZA LA ROTACION A LA IZQUIERDA DE SIETE BYTES CONSECUTIVOS
;PARA REALIZAR LA CONVERSION DE BINARIO A BCD

```

```

ROTAR:  MOV      R1,#COSTO1      ;PRIMER BYTE A ROTAR
        MOV      R2,#12        ;PARA ROTAR 12 BYTES
        CLR      C
SIGO:   MOV      A,@R1          ;TOMO EL BYTE
        RLC      A              ;LO ROTO
        MOV      @R1,A          ;Y LO GUARDO DE NUEVO YA ROTADO
        DEC      R1             ;TOMO EL SIGUIENTE BYTE
        DJNZ     R2,SIGO        ;SIGO ROTANDO HASTA LLEGAR AL SEPTIMO BYTE
        RET

```

```

;*****
;RUTINA DE VERIFICACION PARA LA CONVERSION BINARIO A BCD
;VERIFICA SI EL DIGITO EN FORMACION ES MAYOR QUE 4, EN CUYO CASO,
;SUMO 3 UNIDADES AL DIGITO

```

```

SUM3_?: MOV      R6,#7H          ;HAY QUE VERIFICAR 7 BYTES
        MOV      R1,#DIS_MN     ;EMPIEZO POR LOS DIGITOS MAS SIGNIFICATIVO
NEXT:   MOV      A,@R1
        SWAP     A
        ANL      A,#00001111B   ;TOMO EL NIBBLE MAS SIGNIFICATIVO
        CJNE     A,#4,MAYOR     ;VERIFICACION DE LA CONVERSION
        SJMP     DEV            ;SI ES IGUAL A 4 DEJO EL DIGITO IGUAL
MAYOR:  JC       DEV            ;SI ES MENOR A 4 DEJO EL DIGITO IGUAL
        ADD      A,#3H          ;SI ES MAYOR LE SUMO 3
DEV:    SWAP     A              ;DEVUELVO EL NIBBLE A SU POSICION ORIGINAL
        MOV      R4,A
        MOV      A,@R1
        ANL      A,#00001111B   ;LIMPIO VALOR ANTERIOR
        ORL      A,R4           ;NUEVO VALOR
        MOV      @R1,A          ;GUARDO EL BYTE DE NUEVO
LSNIBB: MOV      A,@R1
        ANL      A,#00001111B   ;TOMO EL NIVEL MENOS SIGNIFICATIVO
        CJNE     A,#4,MAYOR1    ;VERIFICACION DE LA CONVERSION
        SJMP     IGUAL          ;SI ES IGUAL LO DEJO IGUAL
MAYOR1: JC       IGUAL          ;SI ES MENOR A 4 LO DEJO IGUAL
        ADD      A,#3           ;SI ES MAYOR A 4 LE SUMO 3
IGUAL:  MOV      R4,A
        MOV      A,@R1
        ANL      A,#11110000B   ;LIMPIO VALOR ANTERIOR
        ORL      A,R4
        MOV      @R1,A          ;GUARDO EL NIBBLE DE NUEVO
PROX:   INC      R1             ;PROXIMO BYTE
        DJNZ     R6,NEXT        ;HASTA QUE PROCESA 4 BYTES (8 DIGITOS)
        RET

```

```

;*****
;RUTINA DE CONVERSION DE BINARIO A BCD
;TOMA UN NUMERO DE 5 BYTES Y LO CONVIERTE EN SU EQUIVALENTE DE 8 DIGITOS BCD

```

```

BINBCD: MOV      R3,#40          ;5 BYTES = PROCESO 40 VECES
AGAIN:  CALL     ROTAR
        MOV      A,R3

```

```

      CJNE      A,#01,MENOR
      SJMP      SALBIN
MENOR: CALL     SUM3 ?
      DJNZ     R3,AGAIN
SALBIN: RET

```

```

;*****
;RUTINA QUE MODIFICA LA DURACION DE UNA LLAMADA

```

```

DURAC: MOV      A,C_50MS
      CJNE     A,#20,SALDUR ;SI NO PASO 1 SEG --> SALGO SIN MODIFICAR
      MOV     C_50MS,#00H ;SI PASO 1 SEG --> MODIFICO DURACION
SEGUN: INC      USEG
      MOV     A,#10
      CJNE     A,USEG,SALDUR ;SI NO HAN PASADO 10 SEG SALGO
      MOV     USEG,#00H ;SI PASARON INCREMENTO DECENAS DE SEGUNDO
      INC     DSEG ;Y PONGO EN CERO LAS UNIDADES
      MOV     A,#6
      CJNE     A,DSEG,SALDUR ;SI NO HAN PASADO 60 SEG SALGO
      MOV     DSEG,#00H ;SI PASARON PONGO EN CERO DECENAS DE SEGUN
O
MINU: INC      UMIN ;INCREMENTO UNIDADES DE MINUTOS
      MOV     A,#10
      CJNE     A,UMIN,SALDUR ;SI NO HAN PASADO 10 MIN SALGO
      MOV     UMIN,#00H ;SI PASARON PONGO EN CERO UNIDADES DE MINU
OS
      INC     DMIN ;INCREMENTO DECENAS DE MINUTO
      MOV     A,#6
      CJNE     A,DMIN,SALDUR ;SI NO HAN PASADO 60 MINUTOS SALGO
      MOV     DMIN,#00H ;SI PASARON PONGO EN CERO DECENAS DE MINUT
HORAS: INC      UHOR ;INCREMENTO LAS UNIDADES DE LAS HORAS
      MOV     A,#10
      CJNE     A,UHOR,SALDUR ;SI NO HAN PASADO 10 HORAS SALGO
      MOV     UHOR,#00H ;SI PASARON PONGO EN CERO LAS UNIDADES DE
ORA
      INC     DHOR ;INCREMENTO LAS DECENAS DE HORA
      MOV     A,#10
      CJNE     A,DHOR,SALDUR ;SI PASARON 99 HORAS PASO A CERO
      MOV     DHOR,#00H
SALDUR: RET

```

```

;*****
;RUTINA PARA ACTUALIZAR EL COSTO DE UNA LLAMADA CADA VEZ QUE LLEGA UN IMPULSO

```

```

ACTCOST: MOV     A,NO_LIN
      MOV     R0,#NM_IM1
      MOV     R1,#NM_IM2
      CALL    INC2BYTE ;INCREMENTO EL NUMERO DE IMPULSOS
      CALL    MUL_AB ;MULTIPLICO IMPULSOS * COSTO
      CALL    BLANC ;CELDAS BCD LES HAGO UN BLANQUEO
      CALL    BINBCD ;CONVERSION DE BINARIO A BCD
      CALL    DI_DUCST

```

4. LISTADO DEL PROGRAMA EMULADOR DE EPROM

```
;PROGRAMA EMULADOR DE MEMORIA
```

```
;El siguiente programa estará en la EPROM del prototipo, a partir de la
;línea 100H. Así pues, al prender el EQUIPO, el 8031 se quedará esperando
;a que le mandemos un programa de prueba que será cargado en RAM,
;a partir de la dirección 802Bh.
;Reconoce el formato INTEL hexadecimal (archivos compilados ".hex")
```

```
DEFSEG EPROM,ABSOLUTE
SEG EPROM
```

```
;REDIRECCION DE LAS TABLAS DE VECTORES DE INTERRUPCIONES
```

```
ORG 0000H
LJMP 0100H
```

```
ORG 0003H ;INT0
LJMP 8003H
```

```
ORG 000BH ;TIMER0
LJMP 800BH
```

```
ORG 0013H ;INT1
LJMP 8013H
```

```
ORG 001BH ;TIMER1
LJMP 801BH
```

```
ORG 0023H ;RI + TI
LJMP 8023H
```

```
ORG 0100H
```

```
;*****
```

```
MAIN: CALL INIT
INICIO: CALL CARAC
        CJNE A,#3AH,INICIO;RECONOCE LOS DOS PUNTOS
        CALL DATO ;# DE DATOS
        JZ FIN ;SI EL # DE DATOS=0 FINALIZO LA TRANSMISION?
        MOV R2,A ;R2 CONTADOR DE # DE DATOS
        CALL DATO ;DIRECCION DE RAM HIGH
        MOV DPH,A
        CALL DATO ;DIRECCION DE RAM LOW
```

```

MOV     DPL,A
CALL   DATO      ;00 DEL FORMATO
LAZO:  CALL   DATO      ;RECONOCE LOS DATOS
MOVX   @DPTR,A  ;ESCRIBO EN RAM EXTERNA
INC    DPTR
DJNZ   R2,LAZO
CALL   DATO      ;RECONOCE EL CHECKSUM
SJMP   INICIO
FIN:   LJMP   802BH   ;ORIGEN DE LOS PROGRAMAS EN LA RAM

```

```

;*****
;FUNCION: INICIALIZAR EL PUERTO SERIE Y EL TIMER 1
;ENTRADA: NINGUNA
;SALIDA: NINGUNA

```

```

INIT:  MOV     SCON,#70H ;MODO 1, 8 BIT UART, VELOCIDAD VARIABLE
MOV     TMOD,#20H ;TIMER 1 MODO 2 (TEMPORIZADOR CON AUTORECARGA)
MOV     TCON,#40H ;HABILITAR EL TIMER 1
MOV     TH1,#0F3H ;CONSTANTE PARA LOGRAR LOS 2400 BAUDIOS
MOV     TL1,TH1  ;CUBRIR LA PRIMERA PASADA
RET

```

```

;*****
;FUNCION: ASCII-HEXADECIMAL
;ENTRADA: ACUMULADOR
;SALIDA: ACUMULADOR

```

```

ASCHEX: CJNE A,#40H,LETNUM ;LETRA O NUMERO
LETNUM: JNC  LETRAS
CLR    C
SUBB  A,#30H
RET
LETRAS: CLR C
SUBB  A,#37H
RET

```

```

;*****
;FUNCION: RECEPCION DE UN CARACTER POR EL PUERTO SERIAL
;ENTRADA: NINGUNA
;SALIDA: ACUMULADOR

```

```

CARAC: JNB    RI,CARAC ;LLEGO UN CARACTER?
CLR    RI
MOV    A,SBUF
RET

```

```

;*****
;FUNCION: DEJAR EN EL ACUMULADOR EL DATO A GRABAR EN RAM
;ENTRADA: NINGUNA
;SALIDA: ACUMULADOR

```

```
DATO:  CALL    CARAC
        CALL    ASCHEX
        SWAP    A
        MOV     R1,A
        CALL    CARAC
        CALL    ASCHEX
        ORL    A,R1
        RET
        END
```

PSW **Program Status Word:** Contiene información de estado de la CPU en cada ciclo de instrucción. El Cuadro 2.1 muestra el formato de la palabra de estado.

Cuadro 2.1.

PSW																							
<table border="1" style="width: 100%; text-align: center;"> <tr> <td>b₇</td> <td>b₆</td> <td>b₅</td> <td>b₄</td> <td>b₃</td> <td>b₂</td> <td>b₁</td> <td>b₀</td> </tr> <tr> <td>C</td> <td>AC</td> <td>F0</td> <td>RS₁</td> <td>RS₀</td> <td>OV</td> <td>—</td> <td>P</td> </tr> </table>								b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀	C	AC	F0	RS ₁	RS ₀	OV	—	P
b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀																
C	AC	F0	RS ₁	RS ₀	OV	—	P																
BIT	NOMBRE Y COMENTARIO																						
b ₀	P : Flag de paridad del Acumulador (ACC). — Si P = 1 entonces el número de <i>unos</i> del ACC es <i>impar</i> . — Si P = 0 entonces el número de <i>unos</i> del ACC es <i>par</i> .																						
b ₁	Flag disponible y definible por el usuario.																						
b ₂	OV : Flag de Overflow (sobrepasamiento) (ver Apéndice B).																						
b ₃ - b ₄	RS₀ - RS₁ : Selección del banco de registros. <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>RS₁</th> <th>RS₀</th> <th>BANCOS</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Banco 0 (00-07H)</td> </tr> <tr> <td>0</td> <td>1</td> <td>Banco 1 (08-0FH)</td> </tr> <tr> <td>1</td> <td>0</td> <td>Banco 2 (10-17H)</td> </tr> <tr> <td>1</td> <td>1</td> <td>Banco 3 (18-1FH)</td> </tr> </tbody> </table>							RS ₁	RS ₀	BANCOS	0	0	Banco 0 (00-07H)	0	1	Banco 1 (08-0FH)	1	0	Banco 2 (10-17H)	1	1	Banco 3 (18-1FH)	
RS ₁	RS ₀	BANCOS																					
0	0	Banco 0 (00-07H)																					
0	1	Banco 1 (08-0FH)																					
1	0	Banco 2 (10-17H)																					
1	1	Banco 3 (18-1FH)																					
b ₅	F0 : Flag 0. De propósito general. Definible por el usuario.																						
b ₆	AC : Flag Acarreo Auxiliar. Operaciones en BCD.																						
b ₇	C : Flag de Acarreo. (Ver Apéndice B).																						

IP (INTERRUPT PRIORITY REGISTER)								
	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀
	X	X	PT2	PS	PT1	PX1	PT0	PX0
BIT	NOMBRE Y COMENTARIO							
b ₀	PX0 : - Si PX0 = 1 define alta prioridad a interrupción INTO							
b ₁	PT0 : - Si PT0 = 1 define alta prioridad a interrupción <i>Timer 0</i> .							
b ₂	PX1 : - Si PX1 = 1 define alta prioridad a interrupción INT1.							
b ₃	PT1 : - Si PT1 = 1 define alta prioridad a interrupción <i>Timer 1</i> .							
b ₄	PS : - Si PS = 1 define alta prioridad a interrupción puerto serie.							
b ₅	PT2 : - Si PT2 = 1 define alta prioridad a interrupción <i>Timer 2</i> .							
b ₆	Reservado.							
b ₇	Reservado.							

IE (INTERRUPT ENABLE REGISTER)								
	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀
	EA	X	EI2	ES	EI1	EI0	EI0	EX0
BIT	NOMBRE Y COMENTARIO							
b ₀	EX0 : - Si EX0 = 1 habilita interrupción externa INTO - Si EX0 = 0 inhabilita.							
b ₁	EI0 : - Si EI0 = 1 habilita interrupción del <i>Timer 0</i> . - Si EI0 = 0 inhabilita.							
b ₂	EI1 : - Si EI1 = 1 habilita interrupción externa INT1. - Si EI1 = 0 inhabilita.							
b ₃	EI1 : - Si EI1 = 1 habilita interrupción del <i>Timer 1</i> . - Si EI1 = 0 inhabilita.							
b ₄	ES : - Si ES = 1 habilita interrupción del puerto serie. - Si ES = 0 inhabilita.							
b ₅	EI2 : - Si EI2 = 1 habilita interrupción por sobrepasamiento o captura del <i>Timer 2</i> . - Si EI2 = 0 inhabilita.							
b ₆	Reservada.							
b ₇	EA : - Si EA = 1 habilita individualmente a todas las interrupciones que en este registro están a uno. - Si EA = 0 no reconoce ninguna interrupción.							

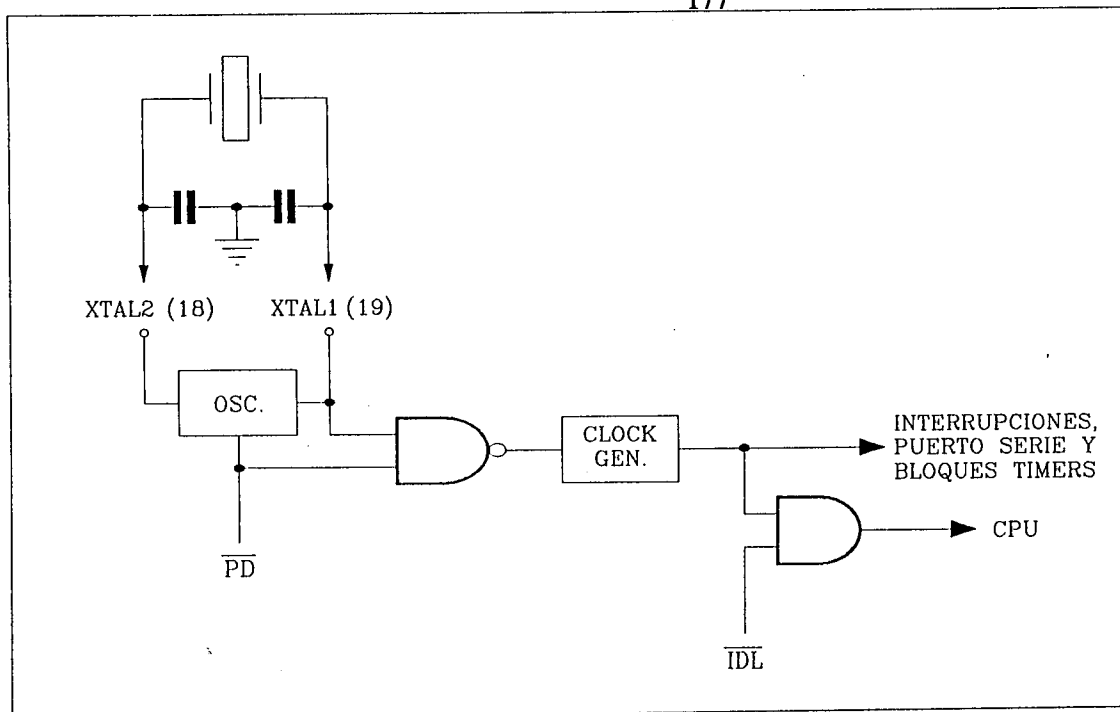


Figura C.1.

Cuadro C.1.

PCON								
BIT	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀
	SMOD	—	—	—	GF1	GF0	PD	IDL
BIT	NOMBRE Y COMENTARIO							
b ₀	IDL : Bit <i>Modo Idle</i> . Si IDL = 1 entonces activa este modo de operación.							
b ₁	PD : Bit <i>Power Down</i> . Si PD = 1 entonces activa el <i>Modo Power Down</i> .							
b ₂	GF0 : <i>Flag</i> bit de propósito general.							
b ₃	GF1 : <i>Flag</i> bit de propósito general.							
b ₄ -b ₅ -b ₆	Reservados para futuras ampliaciones.							
b ₇	SMOD : Bit duplicador de baudios. Si SMOD = 1 entonces duplica la frecuencia de reloj del <i>Timer 1</i> cuando éste se utiliza como generador reloj de baudios en la comunicación serie en los <i>Modos 1, 2 y 3</i> .							
- Si se activa simultáneamente PD e IDL el microcontrolador toma el <i>Modo Power Down</i> , que es preferencial sobre el <i>Modo Idle</i> . - Después de la operación <i>Reset</i> , PCON toma el siguiente valor: 0XXX0000. - En los dispositivos HMOS, PCON contiene solamente el bit SMOD.								

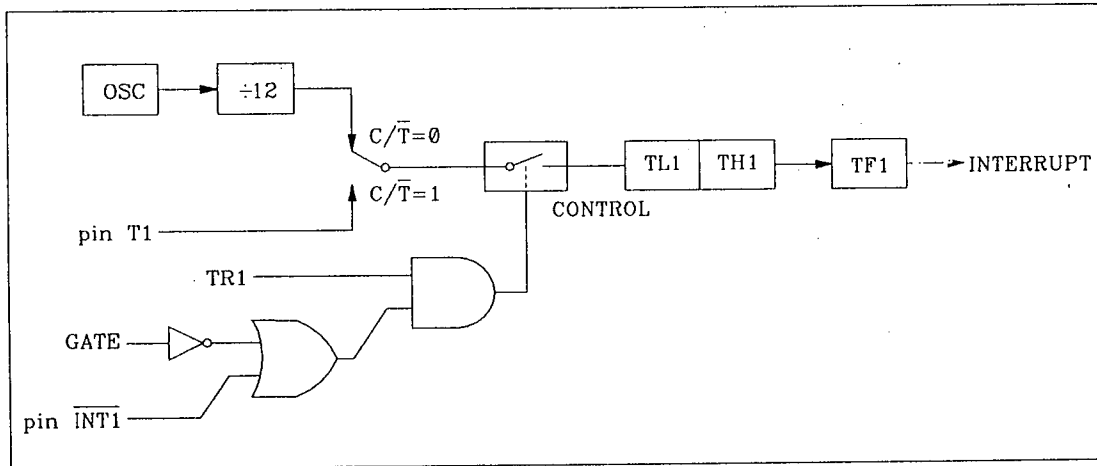
EL REGISTRO DE CONTROL DEL PUERTO SERIE (SCON)

SCON								
	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀
	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
BIT	NOMBRE Y COMENTARIO							
b ₀	RI : <i>Flag</i> de interrupción de la recepción. Se activa por <i>hardware</i> al finalizar la recepción del 8.º bit en el <i>Modo 0</i> o hacia la mitad del intervalo de tiempo del bit de stop en los otros modos (excepto ver SM2). Debe ser desactivado por <i>software</i> .							
b ₁	TI : <i>Flag</i> de interrupción en la transmisión. Se activa por <i>hardware</i> al final de la transmisión del 8.º bit en el <i>Modo 0</i> o al comienzo del bit de stop en los otros modos. Debe ser desactivado por <i>software</i> .							
b ₂	RB8 : En los <i>Modos 2 y 3</i> es el 9.º bit que se recibe. En <i>Modo 1</i> , si SM2 = 0, RB8 es el bit de stop. En <i>Modo 0</i> no se utiliza.							
b ₃	TB8 : Corresponde al 9.º bit de datos en los <i>Modos 2 y 3</i> . Es programable por el usuario. Habitualmente es el bit de paridad.							
b ₄	REN : - Si REN = 1 (por <i>software</i>) permite la recepción - Si REN = 0 no la permite.							
b ₅	SM2 : - En <i>Modo 2 y 3</i> , Si SM2 = 1 entonces RI no se activará si el 9.º bit de datos (RB8) es igual a cero. - En <i>Modo 1</i> , si SM2 = 1 entonces RI no se activará si el bit de stop no se ha recibido. - En <i>Modo 0</i> , SM2 debe estar a cero.							

SCON																																	
	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀																									
	SM0	SM1	SM2	REN	TB8	RB8	TI	RI																									
BIT	NOMBRE Y COMENTARIO																																
b ₆ - b ₇	SM0-SM1 : Especifica el modo según la tabla: <table border="1" data-bbox="406 1533 1291 1701"> <thead> <tr> <th>SM0</th> <th>SM1</th> <th>MODO</th> <th>DESCRIPCION</th> <th>VELOCIDAD</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>Desplaza 8 bit</td> <td>Reloj/12</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>8 bit-UART*</td> <td>Variable</td> </tr> <tr> <td>1</td> <td>0</td> <td>2</td> <td>9 bit-UART</td> <td>Reloj/64 o reloj/32</td> </tr> <tr> <td>1</td> <td>1</td> <td>3</td> <td>9 bit-UART</td> <td>Variable</td> </tr> </tbody> </table>								SM0	SM1	MODO	DESCRIPCION	VELOCIDAD	0	0	0	Desplaza 8 bit	Reloj/12	0	1	1	8 bit-UART*	Variable	1	0	2	9 bit-UART	Reloj/64 o reloj/32	1	1	3	9 bit-UART	Variable
SM0	SM1	MODO	DESCRIPCION	VELOCIDAD																													
0	0	0	Desplaza 8 bit	Reloj/12																													
0	1	1	8 bit-UART*	Variable																													
1	0	2	9 bit-UART	Reloj/64 o reloj/32																													
1	1	3	9 bit-UART	Variable																													

Temporizadores y contadores

TIMER 0 (T0) Y TIMER 1 (T1)



TMOD																											
b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀																				
GATE	C/T	M1	M0	GATE	C/T	M1	M0																				
← Timer 1 →				← Timer 0 →																							
BIT	NOMBRE Y COMENTARIO																										
b ₀ - b ₁	<table border="1"> <thead> <tr> <th>MODO</th> <th>M1</th> <th>M0</th> <th>MODO DE OPERACION</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>Temporizador de 13 bits</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>Temporizador/Contador de 16 bits</td> </tr> <tr> <td>2</td> <td>1</td> <td>0</td> <td>Temporizador/Contador de 8 bits con Auto-recarga</td> </tr> <tr> <td>3</td> <td>1</td> <td>1</td> <td>Contadores multiples especificos</td> </tr> </tbody> </table>							MODO	M1	M0	MODO DE OPERACION	0	0	0	Temporizador de 13 bits	1	0	1	Temporizador/Contador de 16 bits	2	1	0	Temporizador/Contador de 8 bits con Auto-recarga	3	1	1	Contadores multiples especificos
MODO	M1	M0	MODO DE OPERACION																								
0	0	0	Temporizador de 13 bits																								
1	0	1	Temporizador/Contador de 16 bits																								
2	1	0	Temporizador/Contador de 8 bits con Auto-recarga																								
3	1	1	Contadores multiples especificos																								
b ₂	C/T : Selecciona temporizador o contador - Si C/T = 0 entonces temporiza con los pulsos del reloj interno. - Si C/T = 1 entonces cuenta los pulsos que llegan por T0 (pin 14).																										
b ₃	GATE : Habilita la entrada exterior INT0 (pin 12). - Si GATE = 1 entonces habilita INT0 si TR0 = 1 (control por hardware). - Si GATE = 0 entonces deshabilita INT0 y depende exclusivamente de TR0 (control por software). (TR0 es un bit del Reg. TCON y se activa o desactiva por software).																										
b ₄ - b ₅ - b ₆ y b ₇	Configuración del <i>Timer 1</i> . Igual que para el <i>Timer 0</i> , sustituyendo: T0 por T1 (pin 15) INT0 por INT1 (pin 13) TR0 por TR1																										

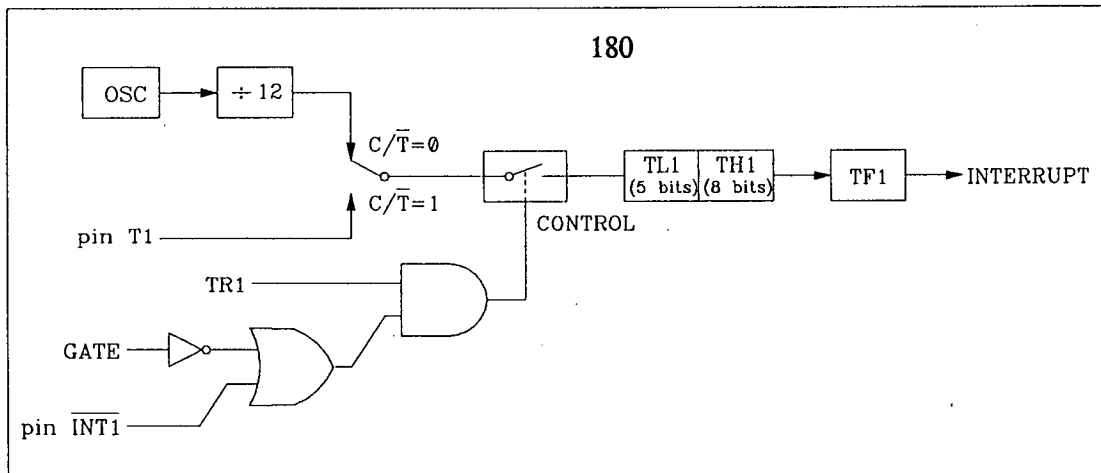


Figura 9.2.

Cuadro 9.2.

TCON								
	b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀
	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
BIT	NOMBRE Y COMENTARIO							
b ₀	IT0 : Control interrupción externa 0 (INT0). - Si IT0 = 0 entonces es activa por nivel bajo. - Si IT0 = 1 entonces es activa por flanco de bajada.							
b ₁	IE0 : Flag de interrupción para la interrupción externa 0 (INT0). - Se pone a uno cuando se detecta interrupción externa. - Se repone automáticamente al atender la interrupción por flanco.							
b ₂	IT1 : Control interrupción externa 1 (INT1).							
b ₃	IE1 : Flag de interrupción para la interrupción externa 1 (INT1).							
b ₄	TR0 : Habilita temporizador/contador 0. - Si TR0 = 1 entonces habilita temporizador/contador 0. - Si TR0 = 0 entonces deshabilita temporizador/contador 0.							
b ₅	TF0 : Flag de Overflow (sobrepasamiento) del Timer 0. Se repone automáticamente al atender la interrupción.							
b ₆	TR1 : Habilita temporizador/contador 1.							
b ₇	TF1 : Flag de Overflow (sobrepasamiento) del Timer 1.							

Códigos de Area y Tarifa de las ciudades más importantes del país:

Caracas con:

Ciudad	Código de Area	Código de Tarifa
Acarigua	✓ 55	4
Anaco	✓ 82	4
Barcelona	✓ 81	4
Barinas	✓ 73	5
Barquisimeto	✓ 51	4
Cabimas	92 64	5
Carora	52	5
Carúpano	94	5
Ciudad Bolívar	85	5
Coro	68	5
Cumaná	93	4
El Tigre	83	5
Guarenas	✓ 36	1
La Guaira	✓ 31	1
La Victoria	✓ 44	2
Los Teques	32	1
Maracaibo	✓ 61	6
Maracay	43	2
Maturín	91	5
Mérida	74	6
Porlamar	95	5
Puerto Ayacucho	48	6
Puerto Cabello	42	3
Puerto La Cruz	81	4
Puerto Ordaz	✓ 86	6
Punto Fijo	69	5
San Antonio (Táchira)	76	6
San Carlos	✓ 58	4
San Cristóbal	76	6
San Felipe	54	4
San Fernando de Apure	47	4
San Juan de los Morros	46	2
Trujillo	72	5
Tucupita	87	6
Valencia	41	3
Valera	71	5
Valle de La Pascua	35	4

Tabla 12 Códigos de área y Códigos de tarifa desde Caracas con las principales ciudades del país

3) LLAMADAS NACIONALES A TRAVES DE OPERADORES (100)

Cadencia:

3 minutos conexión básica, luego 1 impulso cada minuto, pero el costo del impulso depende de la distancia, de la hora, de la fecha y del día:

Código de Tarifa	Distancia (Kms)	Tarifa Normal		Tarifa Especial	
		Conexión 3 minutos (Bolívares)	Minuto Adicional (Bolívares)	Conexión 3 minutos (Bolívares)	Minuto Adicional (Bolívares)
1	0 - 40	46,50	15,50	28,50	9,50
2	41 - 90	70,50	23,50	44,25	14,75
3	91 - 170	88,50	29,50	57,00	19,00
4	171 - 300	121,50	40,50	78,75	26,25
5	301 - 500	148,50	49,50	98,25	32,75
6	501 en adelante	169,50	56,50	104,25	34,75

Tabla 13 Tarifas por Servicio a través de Operadores, desde poblaciones incorporadas al DDN

Tarifa Normal: se aplica los días laborales entre las 7am. y las 7 pm.

Tarifa Especial: se aplica en los días laborales de 7 pm. a 7 am., los días sábados y domingos y los días feriados las 24 horas del día.

4) LLAMADAS HACIA TELÉFONOS CELULARES

Cadencia:

Depende de la hora:

Tarifa Normal: 1 impulso cada 2 seg.

Tarifa Especial: 1 impulso cada 2,5 seg.

Tarifa Normal: se aplica los días laborales entre las 7am. y las 7 pm.

Tarifa Especial: se aplica en los días laborales de 7 pm. a 7 am., los días sábados y domingos y los días feriados las 24 horas del día.

Costo del Impulso:

Uso residencial: 1,18 Bs.

Uso no residencial : 1,56 Bs.

5) LLAMADAS A TRAVÉS DEL DISCADO DIRECTO INTERNACIONAL (D.D.I.)**Cadencia:**

1 impulso cada minuto, pero el costo del impulso varía según el destino y la hora.

CONTINENTE	PAIS	CODIGO	TARIFA NORMAL	TARIFA REDUCIDA	TARIFA ECONÓMICA
América del Norte	Estados Unidos	1	161	130	111
	Canada	11	161	130	111
	México	502	347	278	226
América Central	Panamá	507	230	190	170
	Resto de América Central	50X	204	166	0
Antillas	Cuba	541	227	182	0
	República Dominicana	531	230	184	150
	Puerto Rico	508	219	180	143
	Resto de Antillas	1809	215	172	140
América del Sur	Chile	56	281	230	184
	Perú	51	201	169	131
	Ecuador	593	200	160	130
	Colombia	57	200	160	130
	Bolivia	591	278	223	181
	Brasil	55	318	255	207
	Argentina	54	326	261	212
	Uruguay	598	326	261	212
	Resto de América del Sur	59X	280	224	182
Europa	Italia y Vaticano	39	320	256	208
	Portugal	351	324	260	211
	España	34	211	169	138
	Francia	33	215	172	140
	Suiza	41	219	176	163
	Grecia	30	221	179	158
	Alemania	49	358	287	253
	Reino Unido	44	362	290	236
	Bélgica	32	360	288	234
	Resto de Europa	3X o 4X	363	291	236
Asia Oriental	Japón	81	364	292	0
	Taiwan	850	364	292	0
	Hong Kong	852	364	292	0
	Resto de Asia Oriental	8X	364	293	0
Asia Occidental	Israel	972	362	290	0
	Resto de Asia Occidental	9X	364	292	0
Resto del Mundo		XX	369	296	0

Tabla 14 Tarifas para Servicio Internacional Automático y Manual

FRANJAS HORARIAS PARA TARIFAS INTERNACIONALES

Tarifa Reducida	América del Norte América del Sur Antillas	18 h - 24 h
	Asia Occidental Resto del Mundo	18 h - 7 h
	Asia Oriental	22 h - 7 h
	América Central	16 h - 24 h
	Europa	16 h - 20 h
Tarifa Económica	Todos los países a excepción de Europa.	00 h - 7 h
	Europa	20 h - 7 h

Tabla 15 Horarios para Tarifa Reducida y Tarifa Económica a nivel Internacional.

6) LLAMADAS INTERNACIONALES A TRAVES DE OPERADORES (122)

Estación a Estación: Se aplican las mismas tarifas por minuto de uso del servicio de Discado Directo Internacional (DDI) pero con una tarifa mínima de 3 minutos, aunque la comunicación dure menos tiempo.

Persona a Persona: Se aplican las mismas tarifas por minuto de uso del Discado Directo Internacional (DDI), pero con una tarifa mínima de 4 minutos aunque la duración dure menos tiempo.

7) DIAS FERIADOS

Año Nuevo	1 Enero
Carnaval	14-15 Febrero
Jueves Santo	31 Marzo
Viernes Santo	1Abril
Declaración de la Independencia	19 Abril
Día del Trabajo	1 Mayo
Batalla de Carabobo	24 Junio
Independencia	5 Julio
Natalicio de Simón Bolívar	24 Julio
Día de la Raza	12 Octubre
Navidad	25 Diciembre

Tabla 16 Días Festivos en Venezuela para el año 1994

7) INFORMACIONES TÉCNICAS DE LOS COMPONENTES

a) Chip Reloj-Calendario MM5817A



MM58167A Microprocessor Real Time Clock

General Description

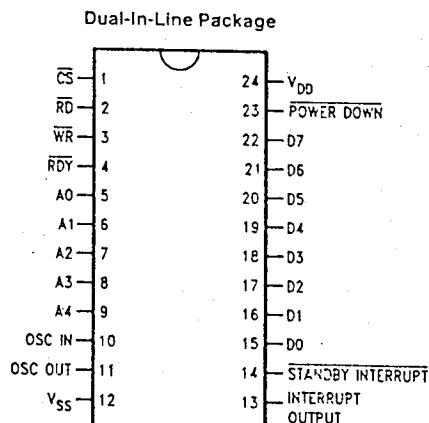
The MM58167A is a low threshold metal gate CMOS circuit that functions as a real time clock in bus oriented microprocessor systems. The device includes an addressable real time counter, 56 bits of RAM, and two interrupt outputs. A **POWER DOWN** input allows the chip to be disabled from the rest of the system for standby low power operation. The time base is a 32,768 Hz crystal oscillator.

Features

- Microprocessor compatible (8-bit data bus)
- Milliseconds through month counters

- 56 bits of RAM with comparator to compare the real time counter to the RAM data
- 2 INTERRUPT OUTPUTS with 8 possible interrupt signals
- **POWER DOWN** input that disables all inputs and outputs except for one of the interrupts
- Status bit to indicate rollover during a read
- 32,768 Hz crystal oscillator
- Four-year calendar (no leap year)
- 24-hour clock

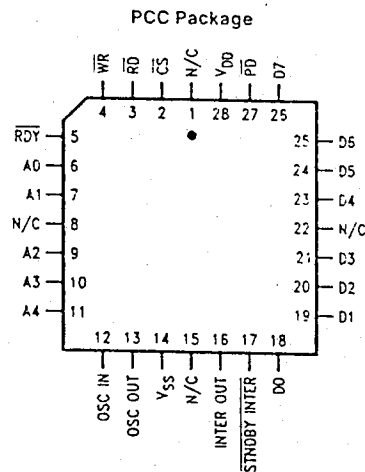
Connection Diagrams



Top View

TL/F/6148-1

Order Number MM58167AN
See NS Package Number N24A



Top View

TL/F/6148-2

Order Number MM58167AV
See NS Package Number V28A

MM58167A

Functional Description

Real Time Counter

The real time counter is divided into 4-bit digits with 2 digits being accessed during any read or write cycle. Each digit represents a BCD number and is defined in Table I. Any unused bits are held at a logical zero during a read and ignored during a write. An unused bit is any bit not necessary to provide a full BCD number. For example tens of hours cannot legally exceed the number 2, thus only 2 bits are necessary to define the tens of hours. The other 2 bits in the tens of hours digit are unused. The unused bits are designated in Table I as dashes.

The addressable portion of the counter is from milliseconds to months. The counter itself is a ripple counter. The ripple delay is less than 60 μ s above 4.5V and 300 μ s at 2.2V.

RAM

56 bits of RAM are contained on-chip. These can be used for any necessary power down storage or as an alarm latch for comparison to the real time counter. The data in the RAM can be compared to the real time counter on a digit basis. The only digits that are not compared are the unit ten thousandths of seconds and tens of days of the week (these are unused in the real time counter). If the two most significant bits of any RAM digit are ones, then this RAM location will always compare. The rule of thumb for an "alarm" interrupt is: All nibbles of higher order than specified are set to C hex (always compare). All nibbles lower than specified are set to "zero". As an example, if an alarm is to occur everyday at 10:15 a.m., configure the bits in RAM as shown in Table II.

The RAM is formatted the same as the real time counter, 4 bits per digit, 14 digits, however there are no unused bits.

The unused bits in the real time counter will compare only to zeros in the RAM.

An address map is shown in Table III.

Interrupts and Comparator

There are two interrupt outputs. The first is the INTERRUPT OUTPUT (a true high signal). This output can be programmed to provide 8 different output signals. They are: 10 Hz, once per second, once per minute, once per hour, once a day, once a week, once a month, and when a RAM/real time counter comparison occurs. To enable the output a one is written into the interrupt control register at the bit location corresponding to the desired output frequency (Figure 1). Once one or more bits have been set in the interrupt control register, the corresponding counter's rollover to its reset state will clock the interrupt status register and cause the interrupt output to go high. To reset the interrupt and to identify which frequency caused the interrupt, the interrupt status register is read. Reading this register places the contents of the status register on the data bus. The interrupting frequency will be identified by a one in the respective bit position. Removing the read will reset the interrupt.

The second interrupt is the STANDBY INTERRUPT (open drain output, active low). This interrupt occurs when enabled and when a RAM/real time counter comparison occurs. The STANDBY INTERRUPT is enabled by writing a one on the D0 line at address 16H or disabled by writing a zero on the D0 line. This interrupt is not triggered by the edge of the compare signal, but rather by the level. Thus if the compare is enabled when the STANDBY INTERRUPT is enabled, the interrupt will turn on immediately.

TABLE I. Real Time Counter Format

Counter Addressed		Units				Max BCD Code	Tens				Max BCD Code
		D0	D1	D2	D3		D4	D5	D6	D7	
Milliseconds	(00H)	—	—	—	—	0	D4	D5	D6	D7	9
Hundredths and Tenths Sec	(01H)	D0	D1	D2	D3	9	D4	D5	D6	D7	9
Seconds	(02H)	D0	D1	D2	D3	9	D4	D5	D6	—	5
Minutes	(03H)	D0	D1	D2	D3	9	D4	D5	D6	—	5
Hours	(04H)	D0	D1	D2	D3	9	D4	D5	—	—	2
Day of the Week	(05H)	D0	D1	D2	—	7	—	—	—	—	0
Day of the Month	(06H)	D0	D1	D2	D3	9	D4	D5	—	—	3
Month	(07H)	D0	D1	D2	D3	9	D4	—	—	—	1

(—) indicates unused bits

MM58167A

Absolute Maximum Ratings

188

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

Voltage at All Pins $V_{SS} - 0.3V$ to $V_{DD} + 0.3V$
 Operating Temperature $0^{\circ}C$ to $70^{\circ}C$

Storage Temperature $-65^{\circ}C$ to $+150^{\circ}C$
 $V_{DD} - V_{SS}$ 6.0V
 Lead Temperature (Soldering, 10 sec.) $300^{\circ}C$
 ESD rating is to be determined.

Electrical Characteristics $V_{SS} = 0V, 0^{\circ}C \leq T_A \leq 70^{\circ}C$

Parameter	Conditions	Min	Max	Units
Supply Voltage				
V_{DD}	Outputs Enabled	4.5	5.5	V
V_{DD}	POWER DOWN Mode	2.2	5.5	V
Supply Current				
I_{DD} , Dynamic	Outputs TRI-STATE* $f_{IN} = 32.768$ kHz, $V_{DD} = 5.5V$ $V_{IH} \geq V_{DD} - 0.3V$ $V_{IL} \leq V_{SS} + 0.3V$		20	μA
I_{DD} , Dynamic	Outputs TRI-STATE $f_{IN} = 32.768$ kHz, $V_{DD} = 5.5V$ $V_{IH} = 2.0V, V_{IL} = 0.8V$		5	mA
Input Voltage				
Logical Low		0.0	0.8	V
Logical high		2.0	V_{DD}	V
Input Leakage Current	$V_{SS} \leq V_{IN} \leq V_{DD}$	-1	1	μA
Output Impedance	I/O and INTERRUPT OUT			
Logical Low	$V_{DD} = 4.5V, I_{OL} = 1.6$ mA		0.4	V
Logical High	$V_{DD} = 4.5V, I_{OH} = -400$ μA $I_{OH} = -10$ μA	2.4		V
TRI-STATE	$V_{SS} \leq V_{OUT} \leq V_{DD}$	$0.8 V_{DD}$ -1	1	μA
Output Impedance	\overline{RDY} and $\overline{STANDBY INTERRUPT}$ (Open Drain Devices)			
Logical Low, Sink	$V_{DD} = 4.5V, I_{OL} = 1.6$ mA		0.4	V
Logical High, Leakage	$V_{OUT} \leq V_{DD}$		10	μA

Functional Description (Continued)

TABLE II. Clock RAM Bit Map for Alarm Interrupt Everyday at 10:15 a.m.

Function	Address					Data							
						Hi Nibble				Lo Nibble			
	4	3	2	1	0	7	6	5	4	3	2	1	0
Milliseconds	0	1	0	0	0	0	0	0	0	No RAM Exists			
Hundredths and Tenths of Seconds	0	1	0	0	1	0	0	0	0	0	0	0	0
Seconds	0	1	0	1	0	0	0	0	0	0	0	0	0
Minutes	0	1	0	1	1	0	0	0	1	0	1	0	1
Hours	0	1	1	0	0	0	0	0	1	0	0	0	0
Day of Week	0	1	1	0	1	No RAM Exists				1	1	X	X
Day of Month	0	1	1	1	0	1	1	X	X	1	1	X	X
Months	0	1	1	1	1	1	1	X	X	1	1	X	X

TABLE III. Address Codes and Function

A4	A3	A2	A1	A0	Function
0	0	0	0	0	0 Counter—Milliseconds
0	0	0	0	1	7 Counter—Hundredths and Tenths of Seconds
0	0	0	1	0	2 Counter—Seconds
0	0	0	1	1	4 Counter—Minutes
0	0	1	0	0	5 Counter—Hours
0	0	1	0	1	6 Counter—Day of Week
0	0	1	1	0	3 Counter—Day of Month
0	0	1	1	1	8 Counter—Month
0	1	0	0	0	17 RAM—Milliseconds
0	1	0	0	1	18 RAM—Hundredths and Tenths of Seconds
0	1	0	1	0	19 RAM—Seconds
0	1	0	1	1	20 RAM—Minutes
0	1	1	0	0	21 RAM—Hours
0	1	1	0	1	22 RAM—Day of Week
0	1	1	1	0	23 RAM—Day of Month
0	1	1	1	1	24 RAM—Months
1	0	0	0	0	25 Interrupt Status Register
1	0	0	0	1	26 Interrupt Control Register
1	0	0	1	0	27 Counters Reset
1	0	0	1	1	28 RAM Reset
1	0	1	0	0	29 Status Bit
1	0	1	0	1	30 GO Command
1	0	1	1	0	31 STANDBY INTERRUPT
1	1	1	1	1	Test Mode

All others unused

The comparator is a cascaded exclusive NOR. Its output is latched 61 μs after the rising edge of the 1 kHz clock signal (input to the milliseconds counter). This allows the counter to ripple through before looking at the comparator. For operation at less than 4.5V, the thousandths of seconds counter should not be included in a compare because of the possibility of having a ripple delay greater than 61 μs. (For output timing see Interrupt Timing.)

Power Down Mode

The POWER DOWN input is essentially a second chip select. It disables all inputs and outputs except for the STANDBY INTERRUPT. When this input is at a logical zero, the device will not respond to any external signals. It will, however, maintain timekeeping and turn on the STANDBY INTERRUPT if programmed to do so. (The programming must be done before the POWER DOWN input goes to a

Functional Description (Continued)

logical zero.) When switching V_{DD} to the standby or power down mode, the POWER DOWN input should go to a logical zero at least 1 μ s before V_{DD} is switched. When switching V_{DD} all other inputs must remain between $V_{SS} - 0.3V$ and $V_{DD} + 0.3V$. When restoring V_{DD} to the normal operating mode, it is necessary to insure that all other inputs are at valid levels before switching the POWER DOWN input back to a logical one. These precautions are necessary to insure that no data is lost or altered when changing to or from the power down mode.

Counter and RAM Resets; GO Command

The counters and RAM can be reset by writing all 1's (FF) at address 12_H or 13_H respectively.

A write pulse at address 15_H will reset the thousandths, hundredths, tenths, units, and tens of seconds counters. This GO command is used for precise starting of the clock. The data on the data bus is ignored during the write. If the seconds counter is at a value greater than 39 when the GO is issued, the minute counter will increment; otherwise the minute counter is unaffected. This command is not necessary to start the clock, but merely a convenient way to start precisely at a given minute.

Status Bit

The status bit is provided to inform the user that the clock is in the process of rolling over when a counter is read. The status bit is set if this 1 kHz clock occurs during or after any counter read. This tells the user that the clock is rippling through the real time counter. Because the clock is rippling, invalid data may be read from the counter. If the status bit is set following a counter read, the counter should be reread.

The status bit appears on D0 when address 14_H is read. All the other data lines will zero. The bit is set when a logical one appears. This bit should be read every time a counter read or after a series of counter reads are done. The trailing edge of the read at address 14_H will reset the status bit.

Using the Rollover Status Bit

If a single read of any clock counter is made, it should be followed by reading the rollover status bit.

Example: Read months, then read rollover status.

If a sequential read of the clock counters is made, then the rollover status bit should be read after the last counter is read.

Example: Read hours, minutes, seconds, then read the rollover status.

Oscillator

The oscillator used in the standard Pierce parallel resonant oscillator. Externally, 2 capacitors, a 20 M Ω resistor and the crystal are required. The 20 M Ω resistor is connected between OSC IN and OSC OUT to bias the internal inverter in the linear region. For micropower crystals a resistor in series with the oscillator output may be necessary to insure the crystal is not overdriven. This resistor should be approximately 200 k Ω . The capacitor values should be typically 20 pF-25 pF. The crystal frequency is 32,768 Hz.

The oscillator input can be externally driven, if desired. In this case the oscillator output should be left floating and the oscillator input levels should be within 0.3V of the supplies.

A ground line or ground plane between pins 9 and 10 may be necessary to reduce interference of the oscillator by the A4 address.

Control Lines

The READ, WRITE, AND CHIP SELECT signals are active low inputs. The READY signal is an open drain output. At the start of each read or write cycle the READY line (open drain) will pull low and will remain low until valid data from a chip read appears on the bus or data on the bus is latched in during a write. READ and WRITE must be accompanied by a CHIP SELECT (see Figures 3 and 4 for read and write cycle timing).

During a read or write, address bits must not change while chip select and control strobes are low.

Test Mode

The test mode is for production testing. It allows the counters to count at a higher than normal rate. In this mode the 32,768 kHz oscillator input is connected directly to the ten thousandths of seconds counter. The chip select and write lines must be low and the address must be held at 1F_H.

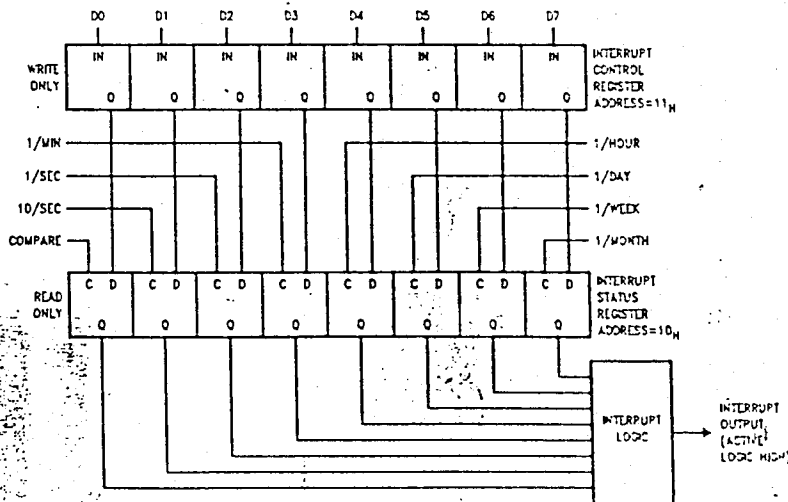
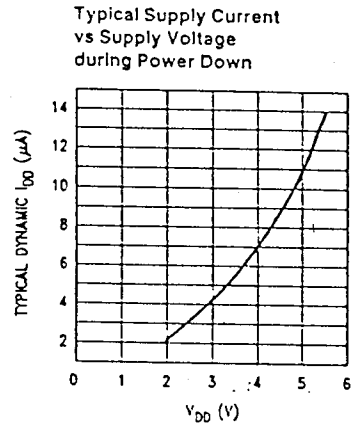
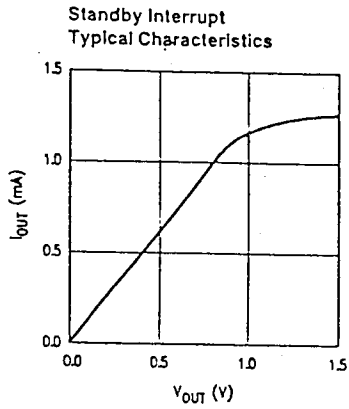


FIGURE 1. Interrupt Register Format

TL/F/6148-3



TL/F/614E-4

FIGURE 2

TL/F/614B-5

Interrupt Timing 0°C ≤ T_A ≤ 70°C, 4.5V ≤ V_{DD} ≤ 5.5V, V_{SS} = 0V

Symbol	Parameter	Min	Max	Units
t _{INTON}	Status Register Clock to INTERRUPT OUTPUT (Pin 13) High (Note 1)		5	µs
t _{SBYON}	Compare Valid to STANDBY INTERRUPT (Pin 14) Low (Note 1)		5	µs
t _{INTOFF}	Trailing Edge of Status Register Read to INTERRUPT OUTPUT Low		5	µs
t _{SBYOFF}	Trailing Edge of Write Cycle (D0 = 0; Address = 16 _H) to STANDBY INTERRUPT Off (High Impedance State)		5	µs

Note 1: The status register clocks are: the corresponding counter's rollover to its reset state or the compare becoming valid. The compare becomes valid 61 µs after the 1/10,000 of a second counter is clocked, if the real time counter data matches the RAM data.

Read Cycle Timing 0°C ≤ T_A ≤ 70°C, 4.5V ≤ V_{DD} ≤ 5.5V, V_{SS} = 0V

Symbol	Parameter	Min	Max	Units
t _{AR}	Address Bus Valid to Read Strobe (Note 3)	100		ns
t _{CSR}	Chip Select to Read Strobe (Note 2)	0		ns
t _{RRY}	Read Strobe to Ready Strobe		150	ns
t _{RYD}	Ready Strobe to Data Valid		800	ns
t _{AD}	Address Bus Valid to Data Valid		1050	ns
t _{RH}	Data Hold Time from Trailing Edge of Read Strobe	0		ns
t _{HZ}	Trailing Edge of Read Strobe to TRI-STATE Mode		250	ns
t _{RYH}	Read Hold Time after Ready Strobe	0		ns
t _{RA}	Address Bus Hold Time from Trailing Edge of Read Strobe	50		ns
t _{RYDV}	Rising Edge of Ready to Data Valid		100	ns

Note 2: When reading, a deselect time of 500 ns minimum must occur between counter reads. Deselect is: CS = 1 or (WR) • (RD) = 1.

Note 3: If t_{AR} = 0 and Chip Select, Address Valid or Read are coincident then they must exist for 1050 ns.

Write Cycle Timing $0^{\circ}\text{C} \leq T_A \leq 70^{\circ}\text{C}$, $4.5\text{V} \leq V_{DD} \leq 5.5\text{V}$, $V_{SS} = 0\text{V}$

Symbol	Parameter	Min	Max	Units
t_{AW}	Address Valid to Write Strobe	100		ns
t_{CSW}	Chip Select to Write Strobe	0		ns
t_{DW}	Data Valid before Write Strobe	100		ns
t_{WRY}	Write Strobe to Ready Strobe		150	ns
t_{RY}	Ready Strobe Width		600	ns
t_{RYH}	Write Hold Time after Ready Strobe	0		ns
t_{WD}	Data Hold Time after Write Strobe	110		ns
t_{WA}	Address Hold Time after Write Strobe	50		ns

Note 4: If data changes while \overline{CS} and \overline{WR} are low, then they must remain coincident for 1050 ns after the data change to ensure a valid write. Data bus loading is 100 pF. Ready output loading is 50 pF and 3 k Ω pull-up.
 Input and output AC timing levels:
 Logical one = 2.0V
 Logical zero = 0.8V

Read and Write Cycle Timing Diagrams

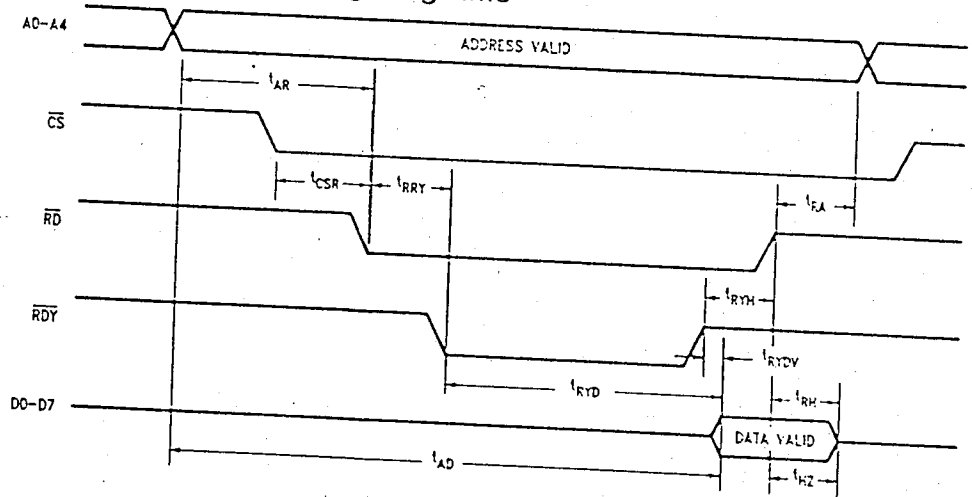


FIGURE 3. Read Cycle Timing

TL/F/6148-6

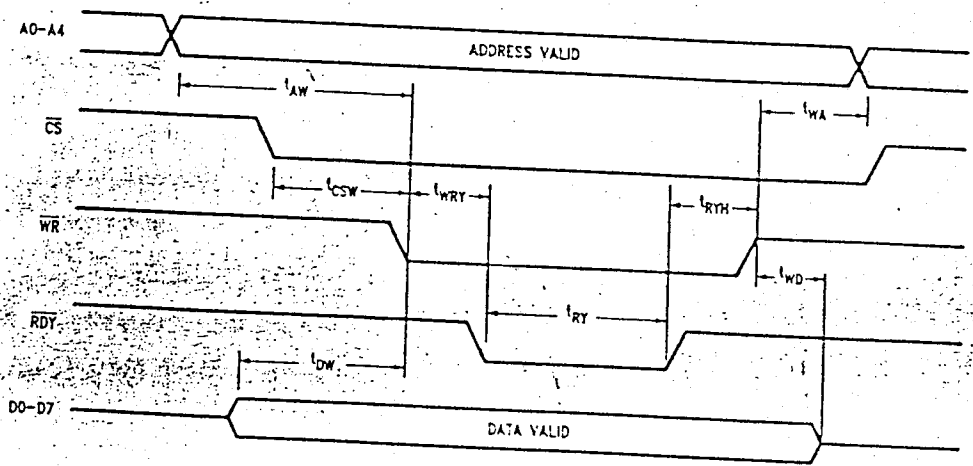
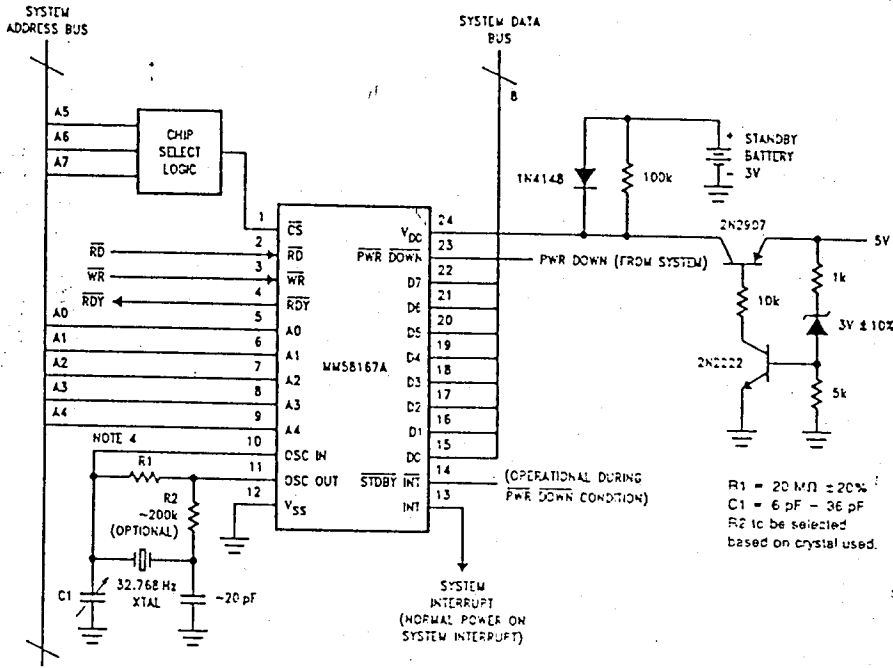


FIGURE 4. Write Cycle Timing

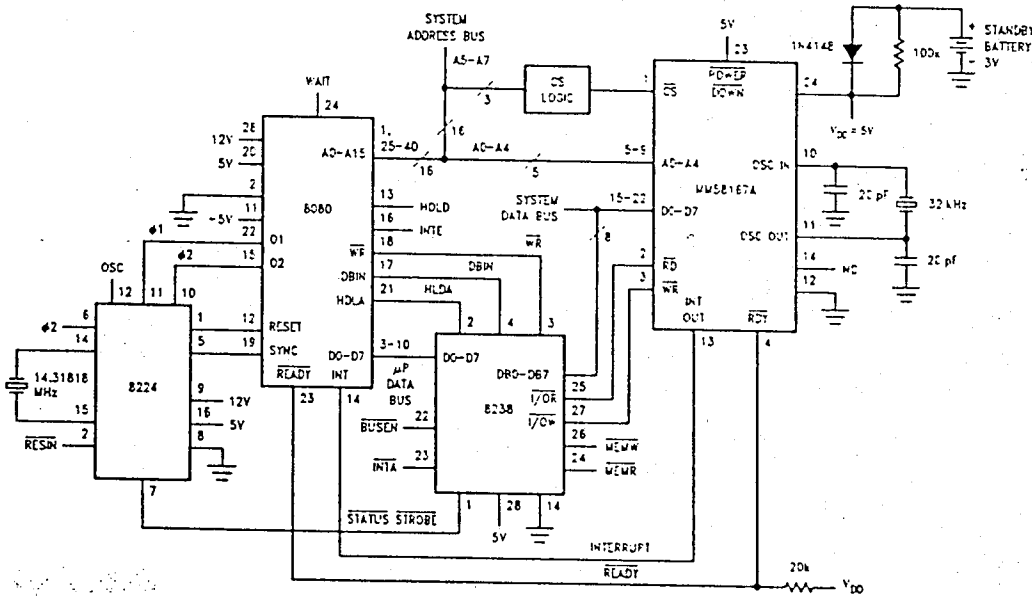
TL/F/6148-7

Typical Applications



Note 5: A ground line or ground plane guard trace should be included between pins 9 and 10 to insure the oscillator is not disturbed by the address line.
 FIGURE 5. Typical Connection Diagram

TL/F/6148-8



Note 6: Must use 8238 or equivalent logic to insure advanced I/O pulse; so that the ready output of the MM58167A is valid by the end of 42 during the T2 microcycle.
 Note 7: $I_{A2} \geq I_{AS8080} + I_{D8238} + I_{WR58167A}$
 FIGURE 6. 8080 System Interface with Battery Backup

TL/F/6148-9

MMS8167A

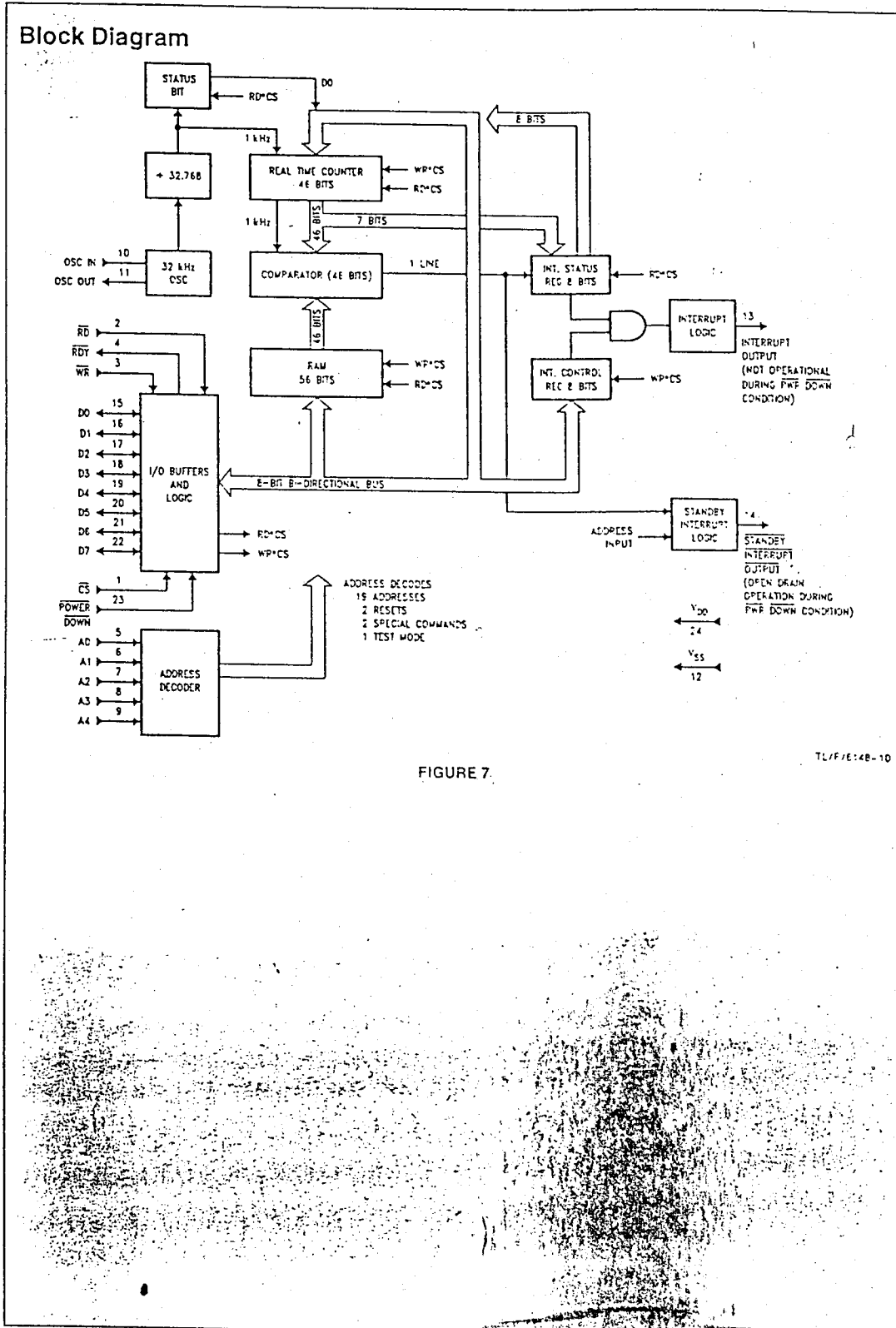


FIGURE 7.

TL/F/E:48-10

b) Detector de Discado Decádico M-959

M-959 DIAL PULSE COUNTER AND HOOK STATUS MONITOR

General Description

The Telone[®] M-959 is a low power dial pulse counter and hook status monitor. The M-959 is contained in a 14 pin package and requires no external components except a single 3.579 MHz television color bar crystal.

The M-959 is typically connected to a loop current sensing circuit which is connected in series with the voice pair (Tip and Ring) of a telephone line. The M-959 receives pulses from the loop current sense circuit and translates them into logic level outputs indicating hook status and decoded dialed digits. Logic inputs to the M-959 select dialed digit speeds and control Data and Strobe outputs supporting bus interrupt driven implementations.

Features

- Meets AT&T and RS-464 recommendations for Break Make ratios and other signal timing.
- Time guarded dial pulse counting.
- 10 or 20 PPS dialing speeds, pin selectable.
- Data output tri-stateable.
- Valid data output strobe.
- Data strobe control for use in interrupt driven environments.
- Independent hook status monitoring.
- Low power CMOS construction.

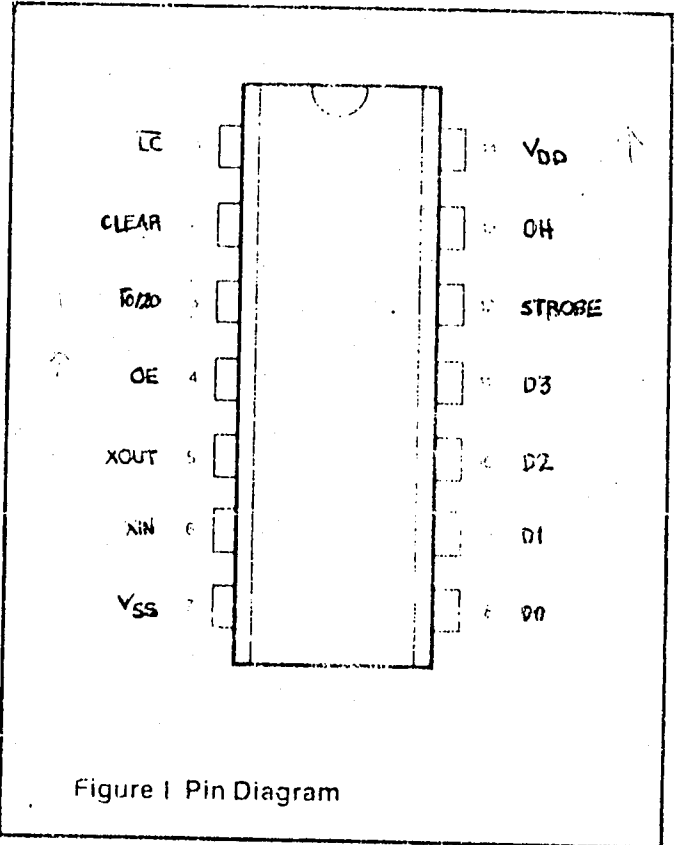


Figure 1 Pin Diagram

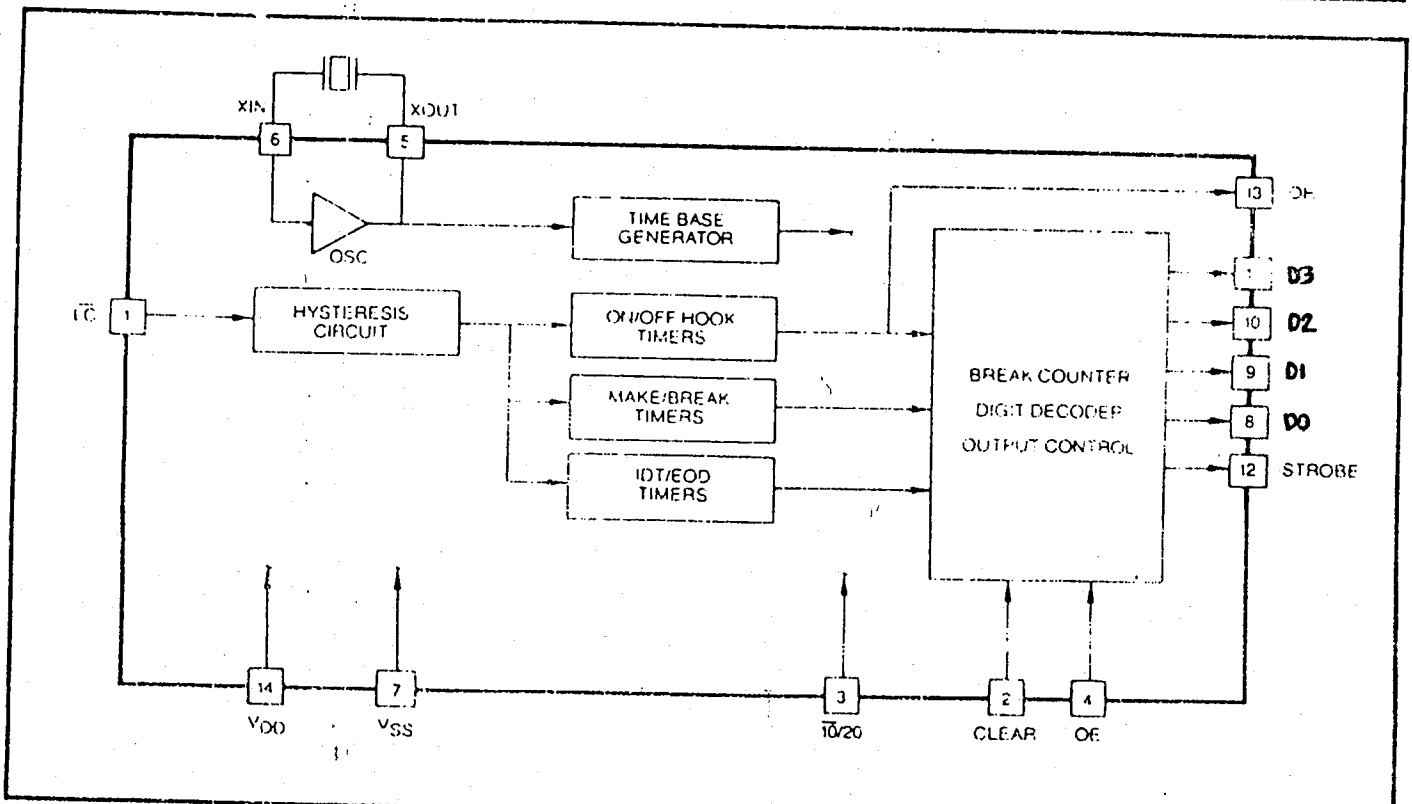


Table 1 Pin Functions

PIN	FUNCTION																																																							
\overline{LC}	Loop Current Input. Signal from phone line to be monitored for dial pulse signalling and hook status. Active low, internally pulled high.																																																							
OH	Off Hook Output. Hook status of phone line. Active (off hook) high.																																																							
$\overline{10/20}$	Pulse Speed Input. Low for 10 pulse per second, high for 20 pulse per second. Internally pulled low.																																																							
D3-D0	Data Outputs. Binary decoded rotary dialed digit. Active during valid digit time (strobe high). low at any other time.																																																							
	<table border="1"> <thead> <tr> <th>Digit Dialed</th> <th>D3</th> <th>D2</th> <th>D1</th> <th>D0</th> </tr> </thead> <tbody> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>2</td><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>3</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>4</td><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>5</td><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>6</td><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>7</td><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>8</td><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>9</td><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td></tr> </tbody> </table>	Digit Dialed	D3	D2	D1	D0	1	0	0	0	1	2	0	0	1	0	3	0	0	1	1	4	0	1	0	0	5	0	1	0	1	6	0	1	1	0	7	0	1	1	1	8	1	0	0	0	9	1	0	0	1	0	1	0	1	0
Digit Dialed	D3	D2	D1	D0																																																				
1	0	0	0	1																																																				
2	0	0	1	0																																																				
3	0	0	1	1																																																				
4	0	1	0	0																																																				
5	0	1	0	1																																																				
6	0	1	1	0																																																				
7	0	1	1	1																																																				
8	1	0	0	0																																																				
9	1	0	0	1																																																				
0	1	0	1	0																																																				
OE	Output Enable Input. Active high, a logic low tri-states D3 through D0 outputs. Internally pulled high.																																																							
XIN	Crystal Oscillator Input.																																																							
XOUT	Crystal Oscillator Output.																																																							
CLEAR	Strobe Control Input. Momentary high during digit valid time resets STROBE latch output low until next valid digit is received. Internally pulled low.																																																							
STROBE	Digit Valid Output. Indicates valid digit data present on D3 through D0. Active high.																																																							

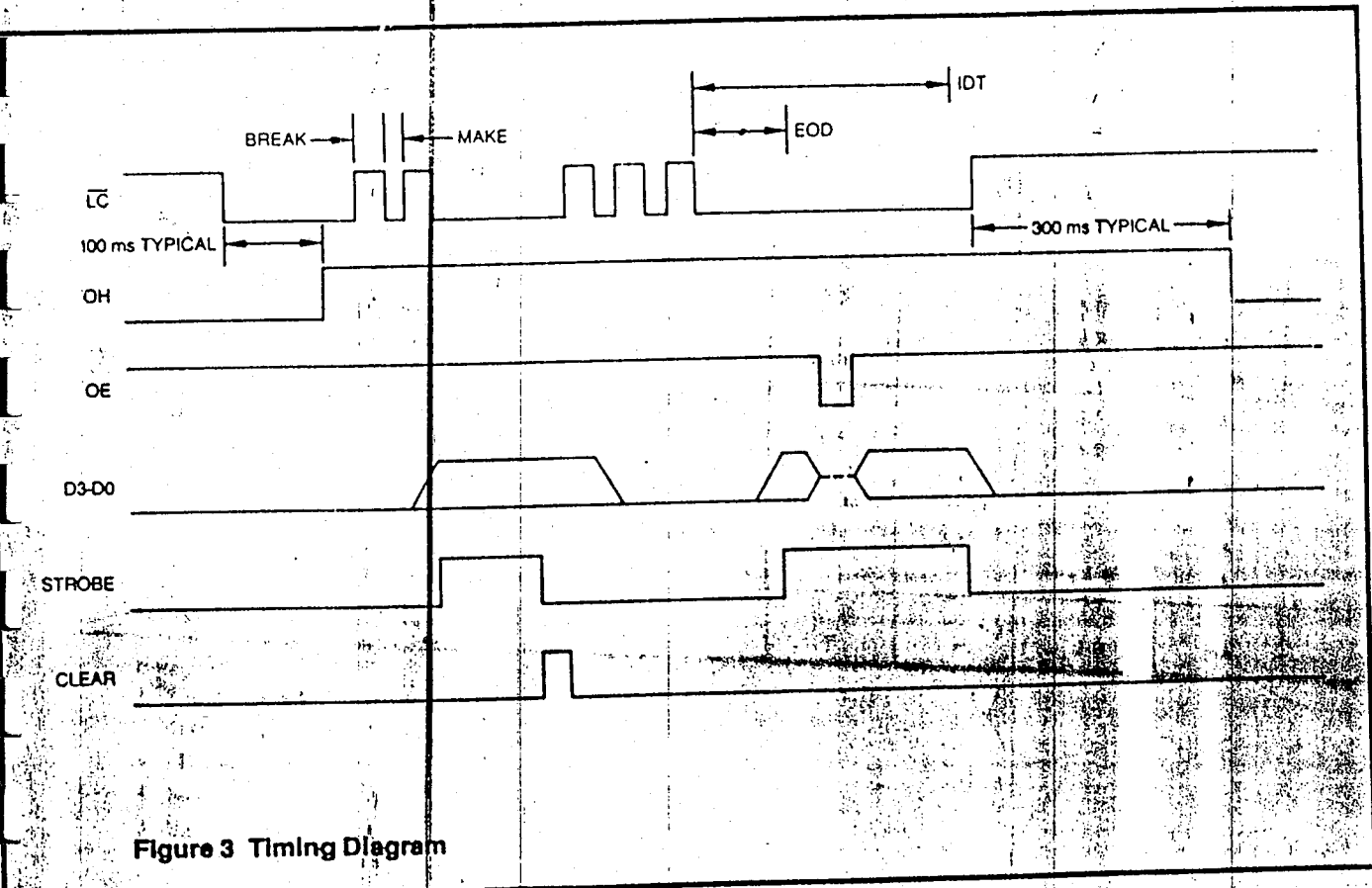


Figure 3 Timing Diagram

Table 2 Absolute Maximum Ratings (Note 1)

198

DC Supply Voltage	6.0V
Any Input Voltage Relative to V_{DD}	+0.3V
Any Input Voltage Relative to V_{SS}	-0.3V
Operating Temperature Range	-40° to +85°C
Storage Temperature Range	-55° to +125°C

Note:

- Exceeding these ratings may permanently damage the M-959.

Table 3 Specifications
 $V_{DD} - V_{SS} = 2.5$ through 6.0V unless otherwise noted.

	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS	NOTES
Signal Timing	Break Recognition	10 PPS	45		85	ms	
		20 PPS	30		40	ms	
	Spurious Break Rejection		0		10	ms	
	Make Recognition	10 PPS	30		65	ms	
		20 PPS	15		24	ms	
	Interdigit Time (IDT)	10 PPS	285	300	315	ms	
		20 PPS	142.5	150	157.5	ms	
	Off-Hook Delay		95	100	105	ms	
	On-Hook Delay		285	300	315	ms	
	I.C Hysteresis		1	1.5	2	ms	
	EOD (End of Digit) Recognition	10 PPS	95	100	105	ms	
		20 PPS	47.5	50	52.5	ms	
STROBE Active	10 PPS	190	200	210	ms		
	20 PPS	95	100	105	ms		
Data Change Before STROBE Active		1.0	1.5	2.0	ms		
Logic Input Requirements	Input Voltages	Logic 0	0.0	2.25	1.5	V	1.2
		Logic 1	3.5	2.75	5.0	V	1.3
	Input Current				±30	μA	
	Pull Up/Down Resistance		200K			ohms	
Logic Output Characteristics	Output Voltages	Logic 0	0.0		0.5	V	1.4
		Logic 1	4.5		5.0	V	1.4
	Output Currents	$V_{out} = 2.5V$	-2.1	-4.2		mA	1
		$V_{out} = 4.6V$	-0.44	-0.88		mA	1
		$V_{out} = 0.4V$	0.44	0.88		mA	1
Tri-State Leakage				±1.0	μA		
Power Requirement	Supply Current				2.0	mA	

Notes

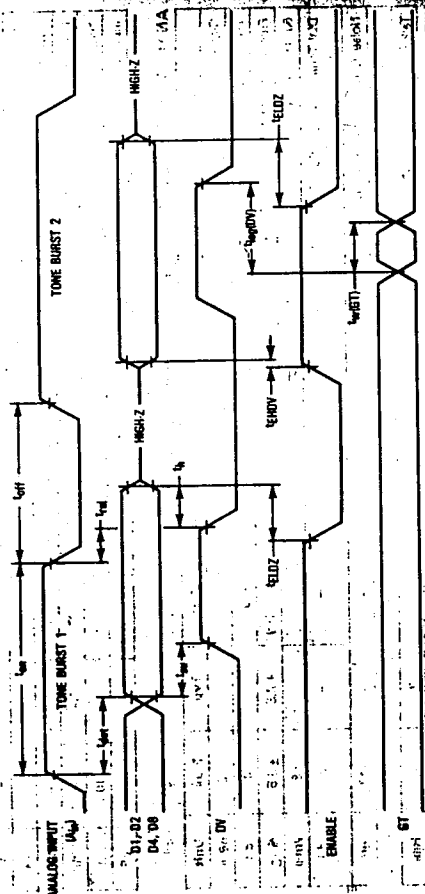
- $V_{DD} - V_{SS} = 5.0V$.
- Maximum is 30% of $V_{DD} - V_{SS}$.
- Minimum is 70% of $V_{DD} - V_{SS}$.
- No load.
- Typical column for reference only.

c) Detector de Discado Multifrecuencial MC145436

AC CHARACTERISTICS (V_{DD} = 5.0 V ± 10%, T_A = -40 to +85°C)

Characteristic	Symbol	Min	Typ	Max	Unit
Tone On Time For Detection For Rejection	Ton _{on}	40	—	20	ms
Pause Time For Detection For Rejection	Ton _{off}	40	—	20	ms
Detect Time	t _{det}	22	—	40	ms
	t _{det}	32	—	50	ms
Release Time	t _{rel}	28	—	40	ms
	t _{rel}	18	—	30	ms
Data Setup Time	t _{su}	7	—	—	μs
Data Hold Time	t _h	4.2	4.6	5	ms
Pulse Width	t _w (GT)	18	—	—	μs
DV Reset Lag Time	t _{lag} (DV)	—	—	5	ms
Enable High to Output Data Valid	t _{EH} DV	—	200	—	ns
Enable Low to Output High-Z	t _{EL} DZ	—	150	—	ns

TIMING



PIN DESCRIPTION

D1, D2, D4, D8—DATA OUTPUT
These digital outputs provide the hexadecimal codes corresponding to the detected digit (see Table 1). The digital outputs become valid after a tone pair has been detected, and are cleared when a valid pause is timed. These output pins are high impedance when Enable is at a logic 0.

EN—ENABLE

Outputs D1, D2, D4, D8 are enabled when EN is at a logic 1, and high impedances (disabled) when EN is at a logic 0.

GT—GUARD TIME

The Guard Time control input provides two sets of detected time and release time, both within the allowed ranges of tone on and tone off. A longer tone detect time rejects signals too short to be considered valid. With GT = 1, talk off performance is improved, since it reduces the probability that tones simulated by speech will maintain signal conditions long enough to be accepted. In addition, a shorter release time reduces the probability that a pause simulated by an interruption in speech will be detected as a valid pause. On the other hand, a shorter tone detect time with a long release time would be appropriate for an extremely noisy environment where fast acquisition time and immunity to drop-outs would be required. In general, the tone signal time generated by a telephone is 100 ms, nominal, followed by a pause of about 100 ms. A high-to-low, or low-to-high transition on the GT pin resets the internal logic, and the MC145436 is immediately ready to accept a new tone input.

X_{en}—OSCILLATOR ENABLE

A logic 1 on X_{en} enables the on-chip crystal oscillator. When using alternate time base from the ATB pin, X_{en} should be tied to GND.

A_{in}—ANALOG INPUT

This pin accepts the analog input, and is internally biased so that the input signal may be ac coupled. The input may be dc coupled so long as it does not exceed the positive supply. (See Figure 1.)

X_{in}/X_{out}—OSCILLATOR IN AND OSCILLATOR OUT

These pins connect to an internal crystal oscillator. In operation, a parallel resonant crystal is connected from X_{in} to X_{out}, as well as a 1 MΩ resistor in parallel with the crystal. When using the alternate clock source from ATB, X_{in} should be tied to V_{DD}.

ATB—ALTERNATE TIME BASE

This pin serves as a frequency reference when more than one MC145436 is used, so that only one crystal is required for multiple MC145436s. In this case, all ATB pins should be tied together as shown in Figure 2. When only one MC145436 is used, this pin should be left unconnected. The output frequency of ATB is 447.4 kHz.

DV—DATA VALID

DV signals a detection by going high after a valid tone pair is sensed and decoded at output pins D1, D2, D4, D8. DV remains high until a loss of the current DTMF signal occurs, or until a transition in GT occurs.

V_{DD}—POSITIVE POWER SUPPLY

The digital supply pin, which is connected to the positive side of the power supply.

GND—GROUND

Ground return pin is typically connected to the system ground.

OPERATIONAL INFORMATION

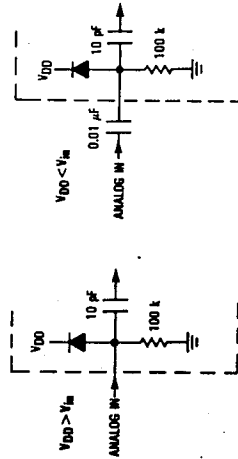


Figure 1. Analog Input

Table 1. Hexadecimal Codes

Digit	Output Code			
	D8	D4	D2	D1
1	0	0	0	1
2	0	0	1	0
3	0	1	0	1
4	0	1	1	0
5	1	0	0	1
6	1	0	1	0
7	1	1	0	1
8	1	1	1	0
*	1	0	1	1
#	1	1	0	0
A	1	1	1	1
B	1	1	1	0
C	1	1	1	1
D	0	0	0	0

Advance Information

Encoder/Decoder (Transcoder)
For Transmission Applications

The MC145439 and MC142103 are high speed CMOS integrated circuits designed to perform the coding translation of clocked serial data into two streams of return to zero (RZ) digital pulses, which are externally mixed to form either AMI, HDB3, B6ZS, or B8ZS (MC142103—AMI or HDB3 only) ternary signals for driving transmission lines. They perform the reverse operation by translating two streams of clocked pulses (which have been derived from an incoming AMI, HDB3, B6ZS, or B8ZS (MC142103—AMI or HDB3 only) ternary encoded signal) into a single stream of clocked binary data. They also feature loopback and error monitoring functions. The coding and decoding functions perform independently at clock rates from zero (dc) to 9 megabits per second. The HDB3 coding and decoding are performed in a manner consistent with the CCITT G.703 recommendations.

Both Devices:

- Low Power CMOS Operation
- Single 5 Volt Power Supply Operation
- Error Monitor Functions Provided
- Loopback Feature Provided
- Encode and Decode Clock Rates to 9 Megabits per Second
- Pin Selectable Modes of Operation
- TTL Compatible Inputs and Outputs

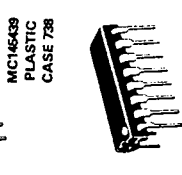
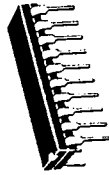
MC145439 Only:

- 20-Pin Package
- NRZ to AMI, HDB3, B6ZS, B8ZS; AMI, HDB3, B6ZS, B8ZS to NRZ
- Force Alarm and Output Enable Function
- Pin Compatible with HC-5560

MC142103 Only:

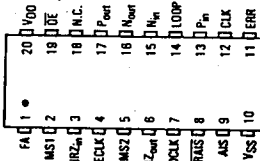
- 16-Pin Package
- NRZ to AMI, HDB3; AMI, HDB3 to NRZ
- Pin Selectable HDB3 or AMI Operation
- Pin Compatible with CD22103 and MJ1471

MC145439
MC142103



MC142103
PLASTIC
CASE 648

PIN ASSIGNMENTS



PIN ASSIGNMENTS

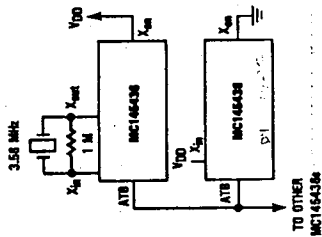
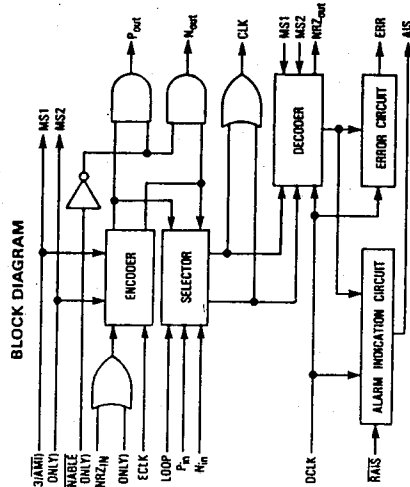
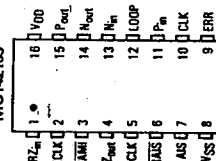


Figure 2. Multiple MC145439s

STD DTMF P40	COL 1	COL 2	COL 3	COL 4	ROW 1
897	1	2	3	A	ROW 1
770	4	5	6	B	ROW 2
852	7	8	9	C	ROW 3
841	0	*	#	D	ROW 4
	1209	1336	1477	1633	
	STD DTMF P40				

Figure 3. 4 x 4 Keyboard Matrix

d) Detector de repique PSB-6620

Ringling Detector

PSB 6620

Preliminary Information

General Description

The PSB 6620 is designed to detect a telephone call signal (AC voltage).

The supply voltage of the IC is derived from the AC input voltage (call signal). During the active state of the device a regulated 5-V DC voltage and a TTL/CMOS logic level is available at the outputs. The high threshold activation voltage provides good immunity against noise (e. g. dialing signals, charge indicator) and a built-in voltage hysteresis guarantees stable operation.

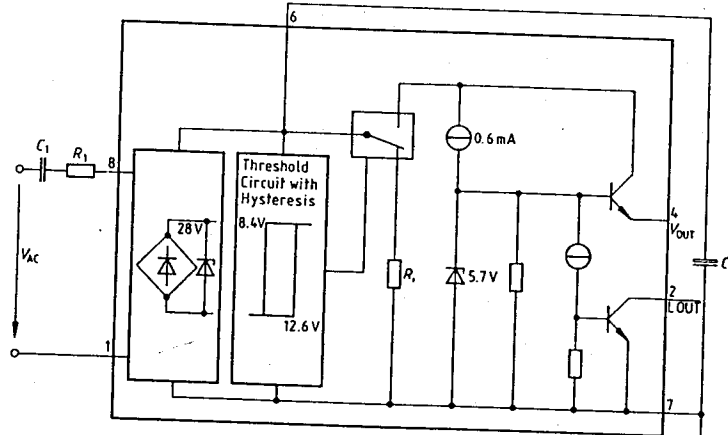
The regulated 5-V voltage (pin 4) of the PSB 6620 enables other devices to be supplied. The PSB 6620 is not limited to telephone applications. For example the device can be used to build up an inexpensive AC voltage detector.

Type	Ordering code	Package
PSB 6620	Q 67 000-A 2498	DIP 8

Features

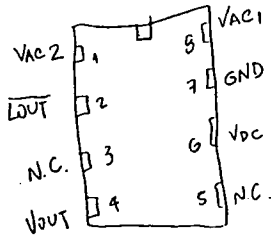
- Integrated bridge rectifier allows direct connection to an AC voltage (e. g. a telephone call signal)
- Low current consumption
- High "tapping" (noise) immunity
- Built-in hysteresis for stable operation
- Only three external components necessary
- Regulated 5-V output voltage
- Logic TTL/CMOS output signal (*open collector*)
- Overvoltage protection in accordance with VDE 0433 (2 kV-10 μ s/700 μ s)
- Pin-compatible with TCM 1520 (Texas Instruments)

Block Diagram

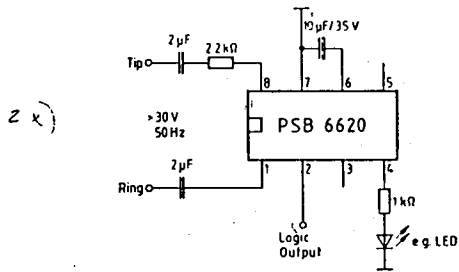


Ringin9 Detector

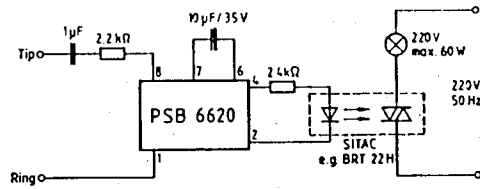
PSB 6620



Application Examples



Call Signal Indicator



Optical Call Signal Indicator

8) GLOSARIO DE TELEFONIA GENERAL

- **Abonado:** Cliente de un sistema telefónico a quien se le proporciona servicio mediante convenio o contrato.

- **Alimentación por Bateria Central (B.C.):** Es aquel sistema telefónico en el cual la energía necesaria para el funcionamiento de todos los teléfonos del sistema, es suministrada por un acumulador común, el cual va conectado a la línea de los abonados a través de choques de alimentación. Llámase también alimentación por Bateria Común.

- **Alimentación por Bateria Local (B.C.):** En este tipo de sistema se encuentra una batería propia en el circuito del micrófono de cada teléfono. Por la línea de enlace no circula ninguna corriente continua, sino la corriente alterna audio-frecuente. Este sistema está en la actualidad prácticamente en desuso.

- **Bucle:** Llamado también lazo, línea de abonado, línea urbana, es el circuito que se establece con un par de conductores (hilos a y b) entre el teléfono del abonado y el equipo de conmutación de la central..

- **Central Telefónica:** Es un equipo de conmutación al que se conectan teléfonos y/o líneas troncales, con el que se logra establecer comunicaciones entre 2 o más personas.

- **Central Telefónica Pública:** Es el equipo al cual se conectan cada uno de los suscriptores de la red telefónica pública. Realiza funciones de supervisión y direccionamiento de las líneas urbanas, permitiendo la comunicación entre abonados.

- **Central Telefonica Privada:** Es una central telefónica destinada a dar servicio a una sola organización, y presta servicios adicionales a sus extensiones. Comúnmente a estos se les llama por una de las siglas siguientes: PABX (Private Access Branch EXchange) o también como CPA (Central privada automática).

- **Clase de Servicio (COS):** Categorización del servicio telefónico que prestan las centrales, de acuerdo con el uso específico de cada abonado. De esta manera, se pueden hacer distintos tipos de combinaciones de facilidades y niveles de acceso.

- **Discado Decádico o por pulsos:** Forma de discado en la cual los números del abonado que se desea llamar, se transmiten hacia la central telefónica pública, en formas de pulso que se logran mediante aperturas y cierres del lazo de corriente.

- **Discado DTMF o Multifrecuencial:** Es la forma de marcación en la que se emiten un par de frecuencias, en rango vocal de ocho posibles, cada vez que el abonado pulsa uno de los dígitos del teclado. Las ocho frecuencias utilizadas son 697 Hz, 770 Hz, 852 Hz, 941 Hz, 1209 Hz, 1339 Hz, 1477 Hz y 1633 Hz.

- **Discado Directo Entrante (D.D.E.):** Es una facilidad que tiene una PABX que permite realizar solo llamadas entrantes, es decir, que tienen origen en la red pública y tienen como destino una extensión de la PABX, pero sin necesidad de pasar por una operadora y sin necesidad de asignar de asignar líneas dedicadas exclusivamente a los usuarios del sistema.

- **Disco:** Dispositivo de marcar que al ser manipulado genera los impulsos necesarios para establecer las conexiones.

- **Distribuidor:** Estructura diseñada para la terminación del cableado permanente de una central y para facilitar el cambio de conexiones entre ellos mediante enlaces por cables o "jumpers".

- **Extensión:** Persona o teléfono conectado a una PABX.

- **Facilidad:** Servicio que presta una central telefónica privada a sus extensiones. Entre las facilidades mas comunes que presta una PABX están: llamada en espera, desviación de llamada, conferencia, No molestar, transferencia, consulta, etc.

- **Horquilla (Hook switch):** Conmutador ubicado en el interior del aparato telefónico donde se apoya el microteléfono. Cuando se levanta el microteléfono para hablar el conmutador se cierra y establece el circuito con la central.

- **Liberar:** Significa desenganchar el equipo empleado en una conversación telefónica y restaurarlo a su posición cuando no esta en uso.

- **Microteléfono:** Es la parte suelta del teléfono que se retira de la horquilla para la comunicación. Contiene un dispositivo de transmisión y otro de recepción de señales. Otros nombres que comúnmente se le da son: Auricular, Bocina, etc.

- **Registrador:** Es un dispositivo de las centrales telefónicas automáticas destinado al almacenamiento de información numérica . El registrador recibe la información de enrutamiento de la llamada y los almacena para posible conversión o traspaso. Este dispositivo generalmente funciona en combinación con un emisor.

- **Ring:** Hilo B, hilo de anillo o hilo de batería. Es uno de los cables que forma el bucle telefónico. Su nombre se deriva de que este hilo se conecta en las comunicaciones realizadas por servicio manual, al anillo (del inglés ring) del enchufe macho de conexión (plug) .

Ring Back

- **Señalización:** Transmisión de señales por impulsos eléctricos que permiten el funcionamiento y la vigilancia de los circuitos de comunicación telefónica. Por ejemplo, con la señalización se establecen las conexiones entre abonados, se anuncian las llamadas y se indica si la línea está libre o ocupada. Existen dos clases enteramente de señalización: una forma que se emplea en los circuitos entre los abonados y la central (Señalización de Línea) y otra utilizada en las líneas de enlace entre centrales (Señalización entre Registradores).

- **Señalización E y M:** Disposición de las líneas de transmisión que se caracteriza por el empleo de vías independientes para los tonos de señalización y para señales de voz. Las señales se transmiten al punto distante por el conductor M y se reciben por el E. La abreviatura de boca en inglés (Mouse) ^{← Mouth} le da el nombre al hilo M, que representa hablar o transmitir, mientras que E se deriva de ear (oído) que representa escuchar o recibir.

- **Teléfono descolgado ("Off hook"):** Condición del aparato telefónico de un abonado que se produce cuando el auricular se ha levantado de la horquilla y se ha establecido un circuito con la central. Esta condición ocasiona que cuando alguien se desee comunicar con este abonado que descolgó, la central pública mande la señal de ocupación al abonado llamante.

- **Teléfono colgado (On The hook):** Condición en la que el microteléfono se encuentra en su posición normal, es decir, sobre el gancho conmutador del teléfono. En este estado se dice que la línea del abonado está en reposo, por lo que, cuando alguien intente llamar a este abonado, la central pública emitirá al abonado llamante la señal de línea de libre.

- **Tip:** Conductor de la línea telefónica, comúnmente denominado como el hilo "a". Al medir con respecto al hilo "Ring" se observa un potencial negativo de -48 Voltios DC. El nombre de "Tip" se deriva de que este conductor era conectado a la punta de la clavija o "plug" de las comunicaciones realizadas por operadores.

- **Troncal:** Es una línea de enlace o circuito establecido entre dos centrales telefónicas (a diferencia de las líneas urbanas o de abonados donde la conexión es entre la central y la estación del teléfono).

- **Velocidad de discado:** Es el número de impulsos de llamada que puede transmitir el disco giratorio del teléfono en un período de tiempo determinado. La velocidad de discado común del disco es de 10 pps (pulsos por segundo).

9. FUNDAMENTOS DE LA TELEFONÍA

A. Reseña histórica de la telefonía

Cuando Alexander Graham Bell llenó su aplicación para la patente del **teléfono** el 14 de Febrero de 1876, el aparato que él había inventado era extremadamente simple. Este consistía en un transmisor y un receptor electromagnéticos conectados junto a una batería, y producía sonidos muy débiles para aplicaciones generales en un sistema telefónico práctico. Sin embargo, el teléfono de Bell es básicamente igual al auricular de un teléfono convencional en la actualidad, con la diferencia de que en vez de emplear un electroimán para generar el campo magnético se utiliza un imán permanente.

Con el desarrollo del **micrófono de carbón** de contactos variables, debido a contribuciones de Berliner (1877), Hughes (1878) y Hunnings (1878), se mejoró la falta de sensibilidad del teléfono de Bell como instrumento para convertir las señales sonoras en señales eléctricas. Esta mejora logró un substancial incremento en las corrientes a las frecuencias de audio que alimentaban al receptor, resultando en un nivel de volumen satisfactorio. Hasta nuestros días, el micrófono de carbón es utilizado en los sistemas de telefonía para comunicaciones de voz.

Antes de que el teléfono pudiera ser de uso práctico para comunicaciones por cables con miles de kilómetros de longitud, algún método tenía que ser usado para controlar automáticamente el volumen del sonido de manera que no fuese muy alto para teléfonos ubicados uno muy cerca del otro, ni muy bajo para teléfonos ubicados muy lejos entre sí. Este problema fue resuelto con el uso de una resistencia variable, llamada **varistor**, la cual decrece su resistencia en la medida en que el voltaje aumenta. Por lo tanto un varistor, colocado a través del receptor del teléfono va automáticamente a regular la corriente de audio para mantener el volumen en el receptor razonablemente constante sin tener en cuenta la distancia entre los teléfonos.

El desarrollo de las comunicaciones "duplex"(comunicaciones en ambos sentidos por la misma vía) se debe a la incorporación de los **transformadores híbridos** en los circuitos de los teléfonos, en los cuales la mayoría de la energía eléctrica producida por el micrófono del emisor es dirigida hacia la estación remota con un mínimo de energía que entra al receptor del auricular. Con el uso de los transformadores diferenciales, como también se les conoce, se consiguió lograr la atenuación del efecto local, la cual tiene su origen en que la voz del emisor se transmite con excesivo volumen al propio auricular. A este efecto de realimentación acústica o "feedback", se le conoce como el Efecto Larsen.

En un principio los teléfonos se usaban por parejas y eran instalados por los propios interesados. Cuando el número de personas que tenían teléfonos aumentó se vio la necesidad de incluir en los teléfonos un conmutador que permitiera seleccionar entre varias líneas. En el año de 1878, se instala la primera **central telefónica**, con el propósito de reducir el número de líneas entre abonados. Luego, con las subsiguientes instalaciones de centrales se crearon compañías comerciales que explotaron el servicio de telefonía. Eventualmente la compleja administración de la red condujo a incluir en el sistema los órganos necesarios para la **tarificación**.

En la segunda década de este siglo, se produjo la invención por parte de Lee Forest, de las válvulas termoiónicas o **triodos**. Con este dispositivo se desarrollaron las técnicas de modulación utilizadas en la multiplexación por distribución de frecuencia (MDF), técnicas estas que permitieron la fabricación de equipos multicanales, aumentando así, el número de abonados a los que se podía atender.

El desarrollo del **transistor** sustituye a la válvula en casi todas sus aplicaciones, lográndose de esta manera disminuir sustancialmente el tamaño de los equipos. Por otra parte, con el avance en la tecnología de integración de grandes cantidades de transistores en pequeñas unidades conocidas como **chips**, fue posible el desarrollo de las técnicas de transmisión digital y de las técnicas de multiplexación de canales en tiempo (TDM).

Se prevé que en un futuro próximo, todo tipo de información que se transmita a través de las redes de comunicación lo hará en forma digital. Los medios de transmisión serán la **fibra**

óptica mayoritariamente y el espacio radioeléctrico para facilitar las rutas alternativas. Las líneas por conductores de cobre quedarán relegadas al bucle del abonado y a algunas zonas rurales.

B.El teléfono

El teléfono deriva su nombre de las palabras griegas "tele" y "fono" que significan *sonido o voz a distancia*. Si bien el teléfono ha experimentado notables modificaciones y perfeccionamientos desde su invención, su uso básico de enviar la voz a distancia permanece invariable.

El aparato telefónico es un dispositivo en el cual se realiza la conversión de energía acústica en eléctrica y viceversa. La conversión de las ondas sonoras de la voz en señales electromagnéticas las efectúa el **micrófono o transmisor**, mientras que el proceso contrario lo realiza la **corneta o receptor**. Esta conversión es necesaria para que la voz pueda llegar a su destino, ya que la energía acústica no puede ser transmitida como tal a grandes distancias mientras que la transmisión de energía eléctrica sí puede lograrse en forma eficaz.

Además de su función primordial que es la de permitir el intercambio de mensajes verbales entre dos usuarios del sistema, el teléfono debe permitir la realización de las funciones de **direccionamiento, recepción de llamadas y señalización**, por lo que se requieren de circuitos adicionales.

La circuitería adicional del teléfono, es encapsulada hoy en día. Un ejemplo de ello es el encapsulado K500, en el que se logra integrar en una sola unidad al **transformador híbrido, la red de balance, 2 varistores y un condensador**. En la Figura 55 se puede apreciar este encapsulado encerrado entre líneas punteadas.

Los **varistores** que se encuentran dentro del encapsulado sirven para reducir el nivel del sonido cuando el teléfono llamado está muy cerca, de manera de mantener un nivel de sonido satisfactorio. Su funcionamiento se basa en la corriente continua que circula por el bucle. Cuando la corriente es pequeña, producto de una línea larga, los varistores presentan una resistencia elevada y no afectan al circuito, mientras que en una línea corta la corriente es de magnitud

elevada y los varistores presentan una baja resistencia ohmica, imponiendo una carga al circuito y obteniéndose así, un aumento de la atenuación.

El **capacitor** sirve para bloquear la corriente continua, que se encuentra siempre presente a través de la línea telefónica y además en conjunto con un **par de bobinas** forma el circuito del repique del teléfono. La combinación del condensador y la inductancia es resonante a 25 Hz, razón por la cual el teléfono sonará con un mínimo de señal de repique presente en la línea.

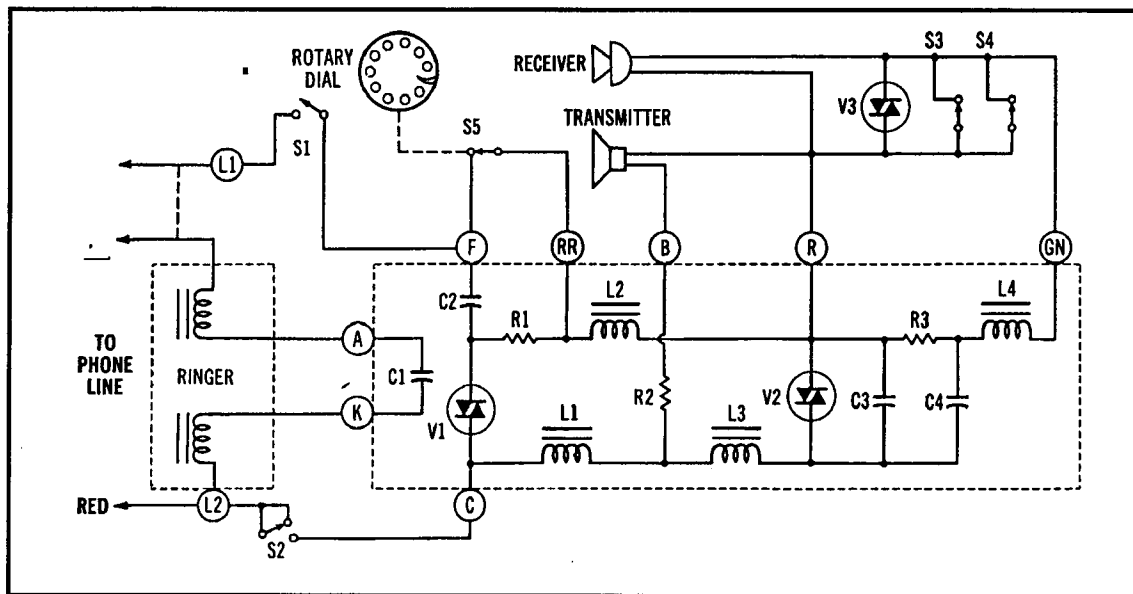


Fig. 55 Circuito básico del teléfono

El **interruptor de descolgado** esta compuesto de tres secciones. Estas son ilustradas en la Fig.55 como S1, S2, S3. Cuando el teléfono no está en uso (colgado) S1 y S2 están abiertos y S3 está cerrado. Esto desconecta por completo el circuito del teléfono de la línea telefónica; con la excepción de las bobinas del repique y el capacitor C1. En esta situación el teléfono está preparado y esperando por una llamada telefónica entrante.

Quando el teléfono suena y el auricular es levantado, S3 permanece cerrado hasta que S1 y S2 hayan conectado la circuitería del teléfono a la línea telefónica. Entonces, S3 se abre, y el teléfono está listo para comunicaciones de voz. Esta característica previene de interferencias "clicks" en el receptor cuando el circuito es conectado a la línea telefónica. Un segundo

interruptor colocado a través del receptor(S4), lo cortocircuita cada vez que el dial es rotado, eliminando los ruidos molestos originados por el discado.

Con el **disco rotativo**, se le dá al usuario del teléfono la capacidad de seleccionar a la persona con la que se desea comunicar. Este dispositivo, permite la transmisión de cifras en forma de trenes de impulsos en donde cada cifra está representada por un número correspondiente de impulsos. El mecanismo está construido de tal manera que asegura un espacio suficiente entre dos cifras consecutivas. La identificación de cifras y ciertas funciones de conexión, son efectuadas en los intervalos de tiempo que separan las cifras.

El disco controla al interruptor S5, el cual realiza la apertura y cierre del lazo. Una resistencia y un condensador, en paralelo a este contacto, forman un circuito denominado "**antichasquido**", que impide la formación de chispas al interrumpirse la corriente.

C. Centrales telefónicas

Las centrales telefónicas se pueden clasificar dependiendo de su pertenencia, su capacidad, y su jerarquía.

- Según su capacidad: pequeñas, medianas y grandes
- Según su pertenencia: públicas y privadas
- Según su jerarquía: locales, de grupo, de zona y de región

Existen, en todo el mundo, mas de trescientos millones de abonados telefónicos conectados a centrales con posibilidad de comunicarse entre sí. O sea el número de conexiones potenciales es de 4.5×10^{16} .

Si no existieran las centrales la forma mas simple de poder conectarlos sería que a cada abonado le llegaran 300 millones de pares de cobre, situación que se presenta en forma muy simplificada en la Figura 56 (a).

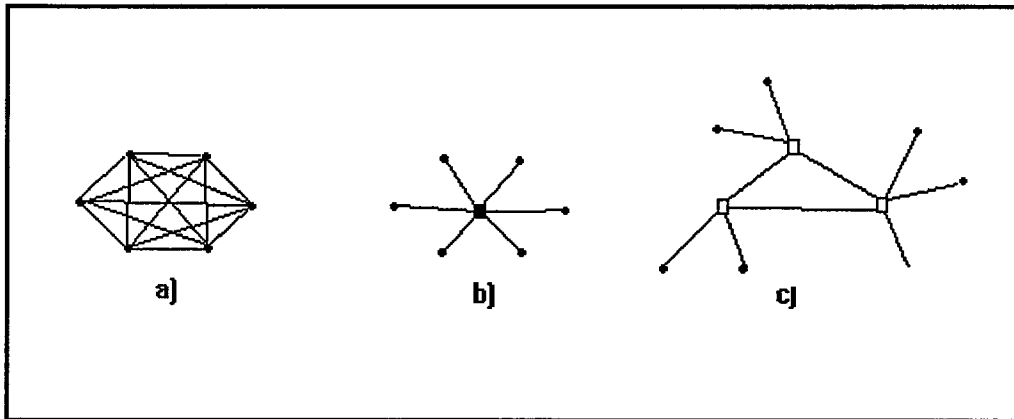


Fig. 56 Conexión entre Centrales Telefónicas. a) completa. b) estrella. c) combinada.

En la Figura 56 (b) se presenta la posibilidad de conectar en una sola super-central a todos los abonados, situación utópica por las dificultades de poder construir, operar y mantener una central de esa naturaleza.

La Figura 56 (c) muestra el caso de la realidad donde el sistema telefónico se organiza en forma redundante y con varios niveles de jerarquías.

Una descripción muy simplificada de este sistema sería: cada abonado está conectado con un par de hilos de cobre a la central local mas cercana de la compañía de teléfonos, según se indica en la Figura 56 (b). El bucle del abonado, o sea, el par de hilos mencionado oscila entre 1 y 10 kilometros, siendo el promedio mas bajo para ciudades que para áreas rurales. La sumatoria total de bucles de abonados del mundo permitiría ir y volver a la luna mas de 1000 veces.

Típicamente una central local se identifica por la numeración asociada al código de área y a los tres primeros dígitos de un número cualquiera. En un país como los Estados Unidos puede haber mas de 20.000 centrales locales y en Venezuela mas de 400. Una llamada entre dos abonados de una misma central local establece, durante todo el tiempo de duración, una conexión galvánica entre ambos bucles y equipos de la central.

Cuando se trata de una llamada a un abonado que pertenece a otra central, entonces se utiliza un enlace denominado troncal que conecta a ambas centrales; pero si la central no es vecina se requiere ir a una central de grupo o zona denominada central tandem quien puede estar

conectada directamente con la central de destino, o podría también a través de troncales entre tandems conectarse a otra tandem de igual jerarquía y así llegar a la central de destino.

En la medida que el abonado que llama intenta comunicarse con otro fuera de su ciudad, región o país. El establecimiento de cada conexión exigirá utilizar eventualmente centrales tandem urbanas, y de otras jerarquías como las centrales de larga distancia nacional, y las centrales de larga distancia internacional.

El número de centrales de diferente jerarquía varía según la densidad telefónica de la región o país. Los enlaces entre centrales varían desde cables multipares de cobre, cables coaxiales, sistemas de microondas, sistemas de fibras ópticas e incluso medios satelitales.

D.Red Pública de Telefonía

La red pública de Telefonía esta ordenada jerárquicamente ya que las centrales que la conforman, tienen distintas funciones. En la Figura 57 ilustra una parte de la red telefónica venezolana en la que se puede observar las distintas jerarquías de las centrales.

La red local o red urbana esta constituida por el conjunto de líneas de abonado de una misma ciudad conectadas a una o varias centrales de grupo. Las centrales locales en el área se interconectan por medio de cables troncales.

Para los fines de la telefonía interurbana automática el país se ha dividido en regiones, las regiones en zonas y estas en grupos. Como se ve en la Figura 57 existen centrales de regiones que se conectan con la central principal de Caracas, centrales de zona que se conectan a la central de región, centrales de grupo que se conectan a las de zona y por ultimo centrales terminales que se conectan a las de grupo . Los abonados estan conectados a las centrales terminales.

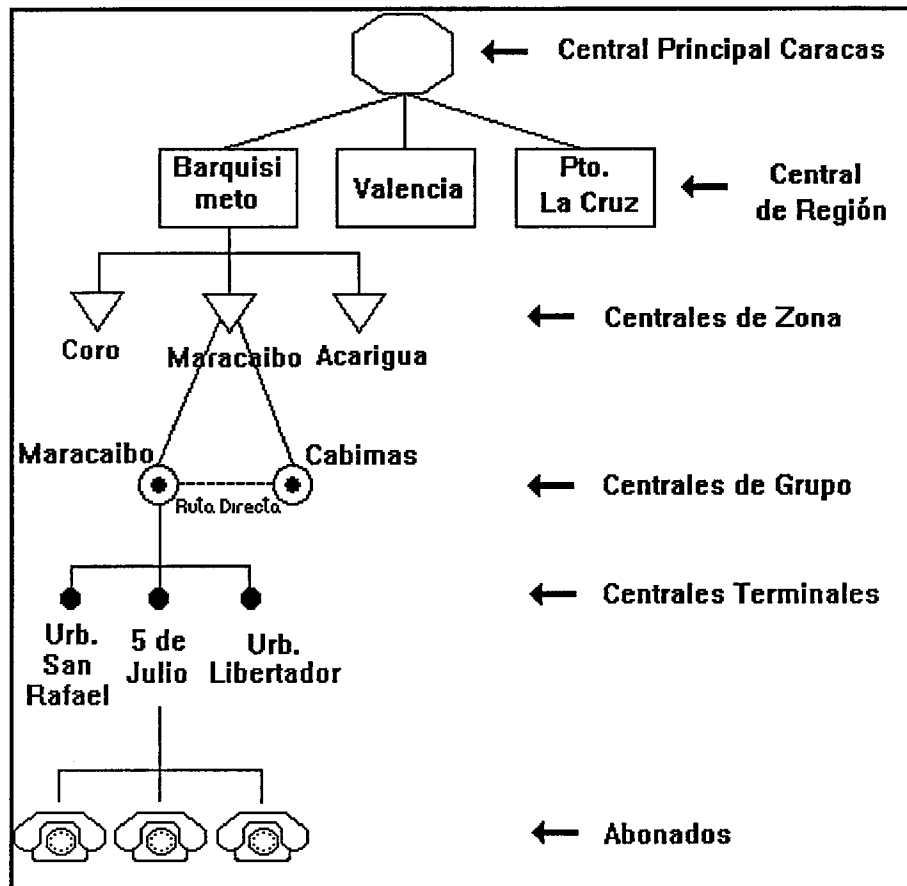


Fig. 57 Segmento de la red de telefonía pública en Venezuela.

Si las centrales locales o **terminales** no estuvieran conectadas entre sí, solamente podrían comunicarse telefónicamente aquellos abonados pertenecientes a la misma área local. Es necesario, por lo tanto conectar entre sí las centrales. Pero el número de centrales es excesivamente alto para conectarlas a todas entre sí. Por esta razón, se hace imprescindible la existencia de una central de rango superior a la terminal que conecte entre sí las centrales locales. Esta central se denomina central de **Grupo**.

Las **centrales terminales** solamente conmutan el tráfico proveniente de los abonados. Las **centrales de grupo** se destinan para conmutar el tráfico proveniente de las centrales terminales, mientras que la de las **centrales de zona** pueden manejar el tráfico proveniente de las centrales de grupo y centrales terminales.

La **central de región** puede conmutar el tráfico de **centrales de Zona**, de Grupo y Terminales. Por último, la **Central Principal** es la única en la cual concurren todas las centrales de región y puede conmutar cualquiera de las centrales de categoría inferior conectadas.

Cada central estará conectada al menos con una de las de orden superior, sin embargo pueden existir **rutas directas** entre dos centrales que permitan su comunicación sin la necesidad de pasar por un enlace de grado superior. (ver Figura 57)

Si las Centrales de origen y destino no tienen una ruta directa que las una o esta ruta se encuentra ocupada, entonces se probarán las diferentes posibilidades de **enrutamiento** a través de los centros de tránsito de la red de modo que priven las comunicaciones que tengan un menor número de puntos de tránsito para completar la llamada sobre lo que introducen mayor número.

E.Evolución de la señalización telefónica

A través de la historia de la telefonía, pasando de la central telefónica manual a las centrales telefónicas automáticas, diferentes principios y cantidad de métodos han sido usados.

En las centrales telefónicas antiguas, las comunicaciones eran conmutadas a través de operadores. Las señales necesarias en esta época eran solamente: llamada y desconexión. Dichas señales eran producidas en el mismo aparato telefónico del abonado, por medio de la acción de un magneto con una manivela.

Posteriormente estos sistemas en donde la alimentación era provista localmente (alimentación por batería local) fueron reemplazados por los sistemas de **batería central (BC)**, en los que el teléfono del abonado es alimentado por la central. Luego de este avance, la señalización se comenzó a emitir automáticamente al levantar y colgar el teléfono.

En los sistemas manuales, el usuario debía comunicar oralmente a la operadora el número del abonado con el que deseaba establecer una conversación, en cambio en las centrales automáticas actuales, el número del destinatario se transmite por medio de señales que son

emitidas por un disco o un teclado, recibidas y transmitidas por un órgano interno denominado registro, que más adelante explicaremos, todo ello en ausencia de la operadora.

Los sistemas **paso a paso**, que realizaban la conmutación de las llamadas automáticamente basados en la recepción de los impulsos generados por el disco rotatorio de los teléfonos, fueron desplazados por el avance de la tecnología electrónica. Estos conmutadores mecánicos presentaban muchos inconvenientes, especialmente cuando había que introducirlos en las centrales. Por otro lado, eran utilizados fijos por cada línea por lo que a nivel integral de la central se podría decir que muchos de los equipos y partes eran subutilizados.

Se comenzó entonces a introducir en muchos países un nuevo elemento: el **registro**, el cual tiene como tarea recibir los impulsos del disco dactilar del abonado, de almacenar las cifras y de dirigir la conexión hacia los siguientes órganos de la central y del sistema telefónico.

La señalización ha debido ser adaptada a ciertas propiedades de sistemas de centrales existentes y a las exigencias específicas concernientes a las impedancias de líneas de transmisión. Aún se piensa y se procura normalizar el sistema de señalización, sin embargo, las empresas que han desarrollado la telefonía se han regido por sus propios estándares, provocando una falta de homogeneidad en cuanto a la señalización que usan los diferentes modelos de centrales.

Es por esta razón que existen organismos internacionales que se encargan de establecer **normas estándares**. En telefonía, el organismo internacional normativo es el CCITT (Comité Consultivo Internacional de Telegrafía y Telefonía). Sus recomendaciones se encuentran condensadas en el famoso Libro Azul. Estas recomendaciones abarcan entre otras cosas: las especificaciones de las funciones de las señales, los sistemas de transmisión de las señales, tratamiento de casos de conexión anormales, etc.

10. TABLA DE COSTOS

RESUMEN

- **COSTO TOTAL DEL EMULADOR DE IMPULSOS DE COMPUTO (EIC)**
 - COSTO DEL EIC PARA 8 LINEAS TELEFONICAS = Bs 25.295
 - COSTO DEL EIC PARA 16 LINEAS TELEFONICAS = Bs 41.434
 - COSTO DEL EIC PARA 24 LINEAS TELEFONICAS = Bs 57.573

- **COSTO TOTAL DEL TARIFICADOR TELEFONICO CASERO (TTC) = Bs 12.691**

- **COSTO TOTAL DEL PROTOTIPO MONTADO = Bs 27.133**

COSTO DEL EMULADOR DE IMPULSOS DE CÓMPUTO

TARJETA CONTROLADORA

CPU

Cantidad	Serial	Descripción	Costo Unit.	Total
1	8031	Microprocesador	375	375
1	74LS373	"Latch" de 8 bits	55	55
1/4	74LS08	Compuertas AND	20	5
2	18nF	Condensador 12V	10	20
1	10uF	Condensador 12V	25	25
1	-	Interruptor SPST	55	55
1	4,7K	Resistencia 1/4 w	1	1
1	10K	Resistencia 1/4 w	1	1
1	Cristal	12Mhz	190	195
1	-	Pulsador	65	65
				797

DECODIFICACIÓN

2	74LS139	Decodificador de 3 a 8	38	76
3/4	74LS08	Compuertas AND	20	15
1/6	74LS14	Compuerta NOT	30	5
				96

RELOJ CALENDARIO

1	MM58167A	Reloj calendario	227	227
1/6	74LS14	Compuertas Not	30	5
1	20M	Resistencia 1/4 w	1	1
1	Cristal	32,768 Khz	120	120
2	18pF	Condensador de 12 V.	15	30
				383

OSCILADOR DE 42,5 KHZ

1/6	74LS14	Compuerta Not	30	5
1	5K	Potenciómetro de precisión	220	220
1	1nF	Condensador 12 V	15	15
				240

REFERENCIA DE 2.5 V

1	LM336	Referencia de 2,5 V	7	7
1	1,5K	Resistencia 1/4 w	4	4
				840

MEMORIAS

1	27C256	Memoria Eprom	350	350
1	62256	Memoria Ram	380	380
				730

3086

MÓDULO DE INTERFASES

INTERFASE SERIAL

1	LM311	Comparador	40	40
1	1N751	Diodo Zener 5,1V	25	25
1	4,7K	Resistencia 1/4 w	4	4
2	1M	Resistencia 1/4 w	4	8
1	1,5K	Resistencia 1/4 w	4	4
1	74LS14	Compuerta Not	30	5
				86

INTERFASE VISUAL

1	HD44780	Controlador display LCD	1600	1600
3/4	74LS32	Compuertas EXOR	40	30
2/6	74LS04	Compuertas NOT	30	5
1	10K	Potenciómetro	54	54
				1689
				1775

TRES TARJETAS DETECTORAS

INTERFAZ DIGITAL

24	TIL113	Ópticoacoplador	41	984
24	75K	Resistencia 1/4 w	1	24
24	33K	Resistencia 1/4 w	1	24
				1032

DETECTOR DE DISCADO DECÁDICO

24	M-959	Detector de discado decádico	497	11928
24	Cristal	3,58 Mhz	65	1560
				13488

DETECTOR DE REPIQUE

24	PSB 6620	Detector de repique	200	4800
24	10uF	Condensador de 35 V	30	720
48	2,2uF	Condensador de 100 V	35	1680
48	2,2K	Resistencia 1/4 w	1	48
24	CNY 171	Ópticoacoplador	41	984
24	5,1K	Resistencia de 1/4w	1	24
				8256

INTERFAZ ANALÓGICA

24	--	Transformador de audio	250	6000
24	47K	Resistencia 1/4 w	1	24
24	0,1uF	Condensador 12 V	15	360
				6384

ACOPLE DE IMPULSOS

24	--	Relé DPDT 5 V	150	3600
24	2N3904	Transistor NPN	10	240
24	27K	Resistencia de 1/4 w	1	24
3	ULN 2803	Driver de corriente	65	195
3	74LS377	Registro de 8 bits	90	270
				4329

FILTRO PASABANDA $F_0 = 425$ HZ

12	MF 10	Filtro activo	360	4320
24	10K	Resistencia 1/4 w	1	24
24	10K	Resistencia 1/4 w	1	24
24	1M	Resistencia 1/4 w	1	24
				4392

ATENUADOR

24	300K	Resistencia 1/4 w	1	24
24	100K	Resistencia 1/4 w	1	24
				48

DETECTOR DE TONO DE 425 HZ

24	LM567	Detector de tono	59	1416
24	4,7uF	Condensador 12 V	15	360
24	1uF	Condensador 12 V	15	360
24	0,1uF	Condensador 12 V	15	360
24	20K	Resistencia 1/4 w	1	24
24	5K	Potenciómetro de precisión	220	5280
24	4,7K	Resistencia 1/4 w	1	24
24	10nF	Condensador 12 V	15	360
				8184

INTEGRADOR

24/2	MC1458	Operacional	70	420
48	1N4148	Diodo	2	96
24	0,1uF	Condensador 12 V	15	360
24	1M	Resistencia 1/4 w	1	24
24	10K	Resistencia 1/4 w	1	24
24	2K	Resistencia 1/4 w	1	24
24	47K	Resistencia 1/4 w	1	24
				972

COMPARADOR

24/2	MC1458	Operacional	70	420
				420

RECTIFICADOR

24	1N4148	Diodo	2	48
24	5,1K	Resistencia	1	24
				72

INTERFASE HACIA EL CPU

24	74LS244	"Buffer tri-state"	35	840
				840
				48417

CONECTORES

CONECTOR PARA LA INTERFAZ SERIAL

1	--	1/2 m de cable plano 24 hilos	225	225
1	--	Conector DB25 hembra	150	150
1	--	Conector para cable plano tipo "chip"	180	180
				555

CONECTORES PARA CUATRO LÍNEAS URBANAS

1	--	Conector Amphenol de 25 pares	980	980
4	--	Puertos de dos pines verdes	25	100
				1080

CONECTOR PARA EL DISPLAY

1	--	Conector para cable plano tipo "chip"	120	120
1	--	1/2 m de cable plano de 14 hilos	180	180
1	--	Para cable plano de 14 pines rectan.	120	120
				420
				2055

EXTERNOS

1	--	Caja de metal	1200	1200
4	--	Bases de goma	80	320
12	--	Tornillos	60	720
				2240
				2240

TOTAL	57573
-------	-------

COSTO DEL TARIFICADOR TELEFÓNICO CASERO

MÓDULO CONTROL

CPU

Cantidad	Serial	Descripción	Costo Unit.	Total
1	8031	Microprocesador	375	375
1	74LS373	"Latch" de 8 bits	55	55
1/4	74LS08	Compuertas AND	20	5
2	18nF	Condensador 12V	10	20
1	10uF	Condensador 12V	25	25
1	--	Interruptor SPST	55	55
1	4,7K	Resistencia 1/4 w	1	1
1	10K	Resistencia 1/4 w	1	1
1	Cristal	12Mhz	190	195
1	--	Pulsador	65	65
				797

DECODIFICACIÓN

2	74LS139	Decodificador de 3 a 8	38	76
3/4	74LS08	Compuertas AND	20	15
1/6	74LS14	Compuerta NOT	30	5
				96

RELOJ CALENDARIO

1	MM58167A	Reloj calendario	227	227
1/6	74LS14	Compuertas Not	30	5
1	20M	Resistencia 1/4 w	1	1
1	Cristal	32,768 Khz	120	120
2	18pF	Condensador de 12 V.	15	30
				383

MEMORIAS

1	27C256	Memoria Eprom	350	350
1	62256	Memoria Ram	380	380
				730

2006

MÓDULO DE INTERFASES

INTERFASE SERIAL

1	LM311	Comparador	40	40
1	1N751	Diodo Zener 5,1V	25	25
1	4,7K	Resistencia 1/4 w	4	4
2	1M	Resistencia 1/4 w	4	8
1	1,5K	Resistencia 1/4 w	4	4
1	--	Cable plano 24 hilos	225	225
1	--	Conector DB25 hembra	150	150
1	--	Conector para cable plano tipo "chip"	180	180
1	74LS14	Compuerta Not	30	5
				641

INTERFASE VISUAL

1	HD44780	Controlador display LCD	1600	1600
3/4	74LS32	Compuertas EXOR	40	30
2/6	74LS04	Compuertas NOT	30	5
1	10K	Potenciómetro	54	54
				1689

2330

MÓDULO FUENTE**FUENTE DE ALIMENTACIÓN**

1	--	Interrupptor SPST	120	120
1	--	Transformador Center Tap	800	800
1	--	Fusible 12,5 A	15	15
1	--	Puente rectificador	80	80
2	2200uF	Condensadores 25 V	55	110
1	LM7805	Regulador de voltaje 5 V	70	70
1	LM7905	Regulador de voltaje -5V	70	70
2	10nF	Condesador de 12V	15	30
				1295

RESPALDO DE BATERÍAS

1	--	Bateria Nicd 3,6 V	450	450
1	510 OHM	Resistencia 1/4 w de precisión	1	1
1	180 OHM	Resistencia 1/4 w de precisión	1	1
1	1N4148	Diodo rectificador, Si	2	2
1	1N277	Diodo rectificador, Ge	2	2
1	1N4001	Diodo de Señal, Si 1A	6	6
				462
				1757

MODULO DETECTORES**INTERFAZ DIGITAL**

1	TIL113	Óptocoplador	41	41
1	75K	Resistencia 1/4 w	1	1
1	33K	Resistencia 1/4 w	1	1
				43

DETECTOR DE DISCADO DECÁDICO

1	M-959	Detector de discado decádico	497	497
1	Cristal	3,58 Mhz	65	65
				562

DETECTOR DE REPIQUE

1	PSB 6620	Detector de repique	200	200
1	10uF	Condensador de 35 V	30	30
2	2,2uF	Condensador de 100 V	35	70
2	2,2K	Resistencia 1/4 w	1	2
1	CNY 171	Óptocoplador	41	41
1	5,1K	Resistencia de 1/4w	1	1
				344

INTERFAZ ANALÓGICA

1	--	Transformador de audio	250	250
1	47K	Resistencia 1/4 w	1	1
1	0,1uF	Condensador 12 V	15	15
				266

FILTRO PASABANDA $F_0 = 425$ HZ

1	MF 10	Filtro activo	360	360
1	10K	Resistencia 1/4 w	1	1
1	10K	Resistencia 1/4 w	1	1
1	1M	Resistencia 1/4 w	1	1
				363

OSCILADOR DE 42,5 KHZ

1/6	74LS14	Compuerta Not	30	5
1	5K	Potenciómetro de precisión	220	220
1	1nF	Condensador 12 V	15	15
				240

ATENUADOR

1	300K	Resistencia 1/4 w	1	1
1	100K	Resistencia 1/4 w	1	1
				2

DETECTOR DE TONO DE 425 HZ

1	LM567	Detector de tono	59	59
1	4,7uF	Condensador 12 V	15	15
1	1uF	Condensador 12 V	15	15
1	0,1uF	Condensador 12 V	15	15
1	20K	Resistencia 1/4 w	1	1
1	5K	Potenciómetro de precisión	220	220
1	4,7K	Resistencia 1/4 w	1	1
1	10nF	Condensador 12 V	15	15
				341

INTEGRADOR

1/2	MC1458	Operacional	70	35
2	1N4148	Diodo	2	4
1	0,1uF	Condensador 12 V	15	15
1	1M	Resistencia 1/4 w	1	1
1	10K	Resistencia 1/4 w	1	1
1	2K	Resistencia 1/4 w	1	1
1	47K	Resistencia 1/4 w	1	1
				58

COMPARADOR

1/2	MC1458	Operacional	70	35
1	LM336	Referencia de 2,5 V	7	7
1	1,5K	Resistencia 1/4 w	4	4
				46

RECTIFICADOR

1	1N4148	Diodo	2	2
1	5,1K	Resistencia 1/4w	1	1
				3

INTERFASE HACIA EL CPU

1	74LS244	"Buffer tri-state"	35	35
				35
				2303

CONECTORES

CONECTOR PARA LA INTERFAZ SERIAL

1	--	1/2 m de cable plano 24 hilos	225	225
1	--	Conector DB25 hembra	150	150
1	--	Conector para cable plano tipo "chip"	180	180
				555

CONECTORES PARA CUATRO LÍNEAS URBANAS

1	--	Conector Amphenol de 25 pares	980	980
4	--	Puertos de dos pines verdes	25	100
				1080

CONECTOR PARA EL DISPLAY

1	--	Conector para cable plano tipo "chip"	120	120
1	--	1/2 m de cable plano de 14 hilos	180	180
1	--	Para cable plano de 14 pines rectan.	120	120
				420
				2055

EXTERNOS

1	--	Caja de metal	1200	1200
4	--	Bases de goma	80	320
12	--	Tornillos	60	720
				2240
				2240

TOTAL	12691
-------	-------

COSTO DEL PROTOTIPO MONTADO

MÓDULO CONTROL

CPU

Cantidad	Serial	Descripción	Costo Unit. (Bs)	Total (Bs)
1	8031	Microprocesador	375	375
1	74LS373	"Latch" de 8 bits	55	55
1/4	74LS08	Compuertas AND	20	5
2	18nF	Condensador 12V	10	20
1	10uF	Condensador 12V	25	25
1	--	Interruptor SPST	55	55
1	4,7K	Resistencia 1/4 w	1	1
1	10K	Resistencia 1/4 w	1	1
1	Cristal	12Mhz	190	195
1	--	Pulsador	65	65
				797

DECODIFICACIÓN

2	74LS139	Decodificador de 3 a 8	38	76
3/4	74LS08	Compuertas AND	20	15
1/6	74LS14	Compuerta NOT	30	5
				96

RELOJ CALENDARIO

1	MM58167A	Reloj calendario	227	227
1/6	74LS14	Compuertas Not	30	5
1	20M	Resistencia 1/4 w	1	1
1	Cristal	32,768 KHz	120	120
2	18pF	Condensador	15	30
				383

OSCILADOR DE 42,5 KHZ

1/6	74LS14	Compuerta Not	30	5
1	5K	Potenciómetro de precisión	220	220
1	1nF	Condensador 12 V	15	15
				240

REFERENCIA DE 2,5 V

1	LM338	Referencia de 2,5 V	7	7
1	1,5K	Resistencia 1/4 w	4	4
				11

MEMORIAS

1	27C256	Memoria Eprom	350	350
1	62256	Memoria Ram	380	380
				730
				2257

MÓDULO DE INTERFASES

INTERFASE SERIAL

1	LM311	Comparador	40	40
1	1N751	Diodo Zener 5,1V	25	25
1	4,7K	Resistencia 1/4 w	4	4
2	1M	Resistencia 1/4 w	4	8
1	1,5K	Resistencia 1/4 w	4	4
1/6	74LS14	Compuerta Not	30	5
				86

INTERFASE VISUAL

1	HD44780	Controlador display LCD	1600	1600
3/4	74LS32	Compuertas EXOR	40	40
2/6	74LS14	Compuertas NOT	30	5
1	10K	Potenciómetro	54	54
				1699
				1785

MÓDULO FUENTE**FUENTE DE ALIMENTACIÓN**

1	--	Interruptor SPST	120	120
1	--	Transformador Center Tap	800	800
1	--	Fusible 1,25 A	15	15
1	--	Puente rectificador	80	80
2	2200uF	Condensadores 25 V	55	110
1	LM7805	Regulador de voltaje 5 V	70	70
1	LM7905	Regulador de voltaje -5V	70	70
2	10nF	Condensador de 12V	15	30
				1295

RESPALDO DE BATERÍAS

1	--	Batería Nicd 3,6 V	450	450
1	510 OHM	Resistencia 1/4 w de precisión	1	1
1	180 OHM	Resistencia 1/4 w de precisión	1	1
1	1N4148	Diodo rectificador, Si	2	2
1	1N277	Diodo rectificador, Ge	2	2
1	1N4001	Diodo de Señal, Si 1A	6	6
				462
				1757

MÓDULO DETECTORES**INTERFAZ DIGITAL**

4	TIL113	Optoacoplador	41	164
4	75K	Resistencia 1/4 w	1	4
4	33K	Resistencia 1/4 w	1	4
				172

DETECTOR DE DISCADO DECÁDICO

2	M-959	Detector de discado decádico	497	994
1	Cristal	3,58 Mhz	65	65
				1059

DETECTOR DE DISCADO MULTIFRECUENCIAL

2	MC 145436	Detec. de discado multifrecuencial	550	1100
1	CRISTAL	3,58 Mhz	65	65
				1165

DETECTOR DE REPIQUE

4	PSB 6620	Detector de repique	75	300
4	10uF	Condensador de 35 V	30	120
8	2,2uF	Condensador de 100 V	35	280
8	2,2K	Resistencia 1/4 w	1	8
4	CNY 171	Ópticoacoplador	41	164
4	5,1K	Resistencia de 1/4w	1	4
				876

INTERFAZ ANALÓGICA

4	--	Transformador de audio	250	1000
4	47K	Resistencia 1/4 w	1	4
4	0,1uF	Condensador 12 V	15	60
				1064

FILTRO PASABANDA Fo=425 HZ

2	MF 10	Filtro activo	360	720
8	10K	Resistencia 1/4 w	1	8
4	1M	Resistencia 1/4 w	1	4
				732

ATENUADOR

4	300K	Resistencia 1/4 w	1	4
4	100K	Resistencia 1/4 w	1	4
				8

DETECTOR DE TONO DE 425 HZ

4	LM567	Detector de tono	59	238
4	4,7uF	Condensador 12 V	15	60
4	1uF	Condensador 12 V	15	60
4	0,1uF	Condensador 12 V	15	60
4	20K	Resistencia 1/4 w	1	4
4	5K	Potenciómetro de precisión	220	880
4	4,7K	Resistencia 1/4 w	1	4
4	10nF	Condensador 12 V	15	60
				1364

INTEGRADOR

4	MC1458	Operacional	70	280
8	1N4148	Diodo	2	16
4	0,1uF	Condensador 12 V	15	60
4	1M	Resistencia 1/4 w	1	4
4	10K	Resistencia 1/4 w	1	4
4	2K	Resistencia 1/4 w	1	4
4	47K	Resistencia 1/4 w	1	4
				372

COMPARADOR

4/2	MC1458	Operacional	70	35
				35

RECTIFICADOR

4	1N4148	Diodo	2	8
4	5,1K	Resistencia	1	4
				12

INTERFASE HACIA EL CPU

4	74LS244	"Buffer tri-state"	35	140
				140
				6999

CONECTORES

CONECTOR PARA LA INTERFAZ SERIAL

1	--	1/2 m de cable plano 24 hilos	225	225
1	--	Conector DB25 hembra	150	150
1	--	Conector para cable plano tipo "chip"	180	180
				555

CONECTORES PARA CUATRO LÍNEAS URBANAS

1	--	Conector Amphenol de 25 pares	980	980
4	--	Puertos de dos pines verdes	25	100
				1080

CONECTOR PARA EL DISPLAY

1	--	Conector para cable plano tipo "chip"	120	120
1	--	1/2 m de cable plano de 14 hilos	180	180
1	--	Para cable plano de 14 pines rectan.	120	120
				420
				2055

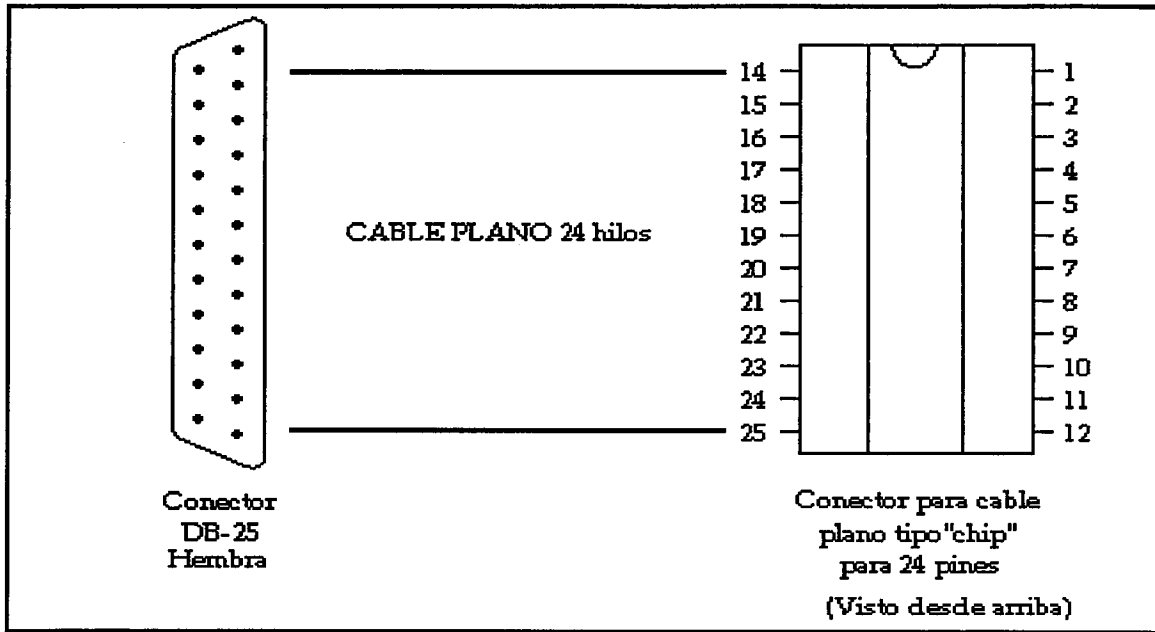
WIPE-RAP

2	--	Bases perforadas	2500	5000
1	--	Base de 24 pines	220	220
2	--	Base de 28 pines	170	340
5	--	Base de 24 pines	150	750
10	--	Base de 20 pines	120	1200
10	--	Base de 16 pines	110	1100
10	--	Base de 14 pines	95	950
6	--	Base de 8 pines	80	480
				10040
				10040

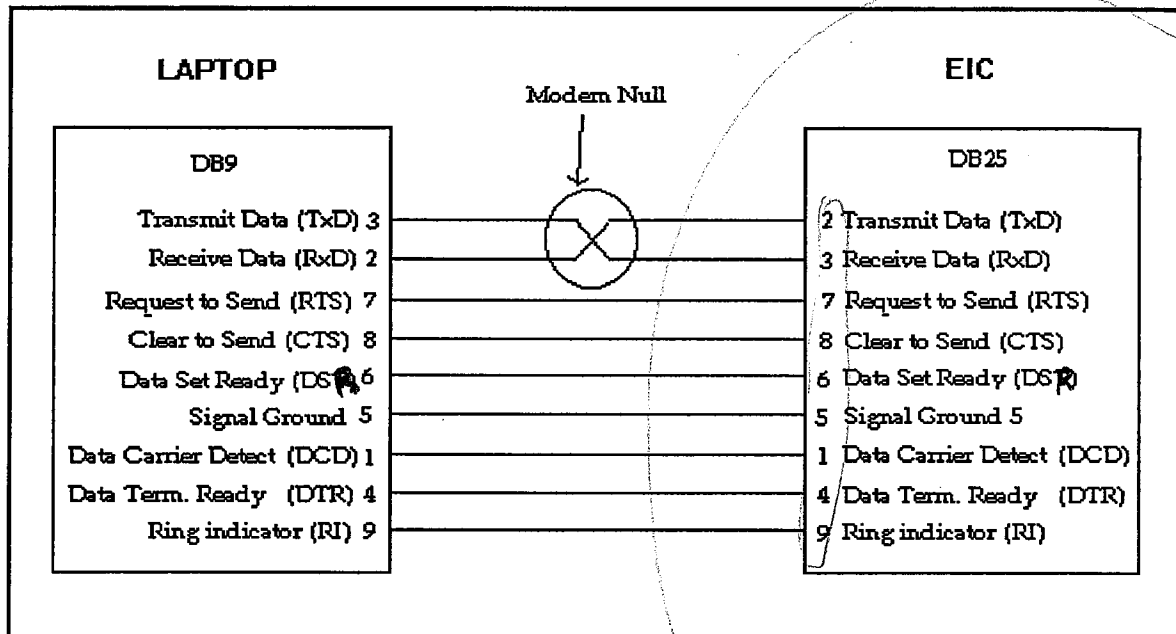
EXTERNOS

1	--	Caja de metal	1200	1200	
4	--	Bases de goma	80	320	
12	--	Tornillos	80	720	
				2240	2240

TOTAL	27133
-------	-------



Cable del puerto serial del EIC a la Tarjeta Controladora.



Cable de conversión DB9 a DB25

12. ANALIZADOR DE BASES DE DATOS PARA EXCEL

En el transcurso de la pasantía se realizó una versión preliminar de un macro para EXCEL que analizara la base de datos emitida por el prototipo montado. Sin embargo, este macro se pudo desarrollar muy poco.

La idea sería tener una ventana, que permitiera seleccionar distintas opciones de análisis de la tabla de datos. Y luego poder presentar gráficos ilustrativos de los tipos de llamadas realizados.

En la Figura 62 se muestra la ventana realizada con los macros de EXCEL.

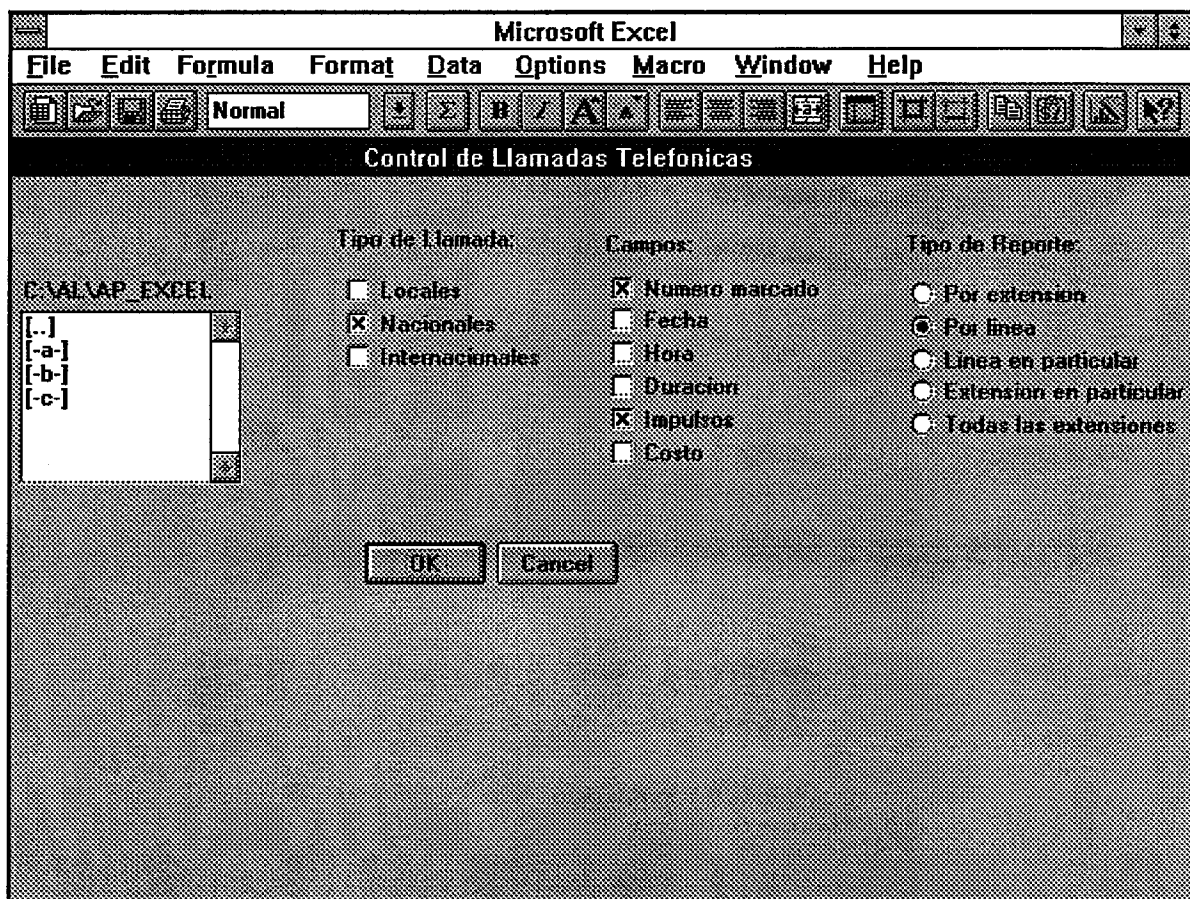


Fig. 62 Macro para EXCEL.

A continuación se muestra el código que se pudo desarrollar.

names	commands	comments
-------	----------	----------

Summary Information	
Title:	v1.0
Version:	Alejandro Avella
Author:	SIEMENS S.A.
Corporation:	Monday, October 16, 1993
Creation Date:	

getinfo	
WChan	=INITIATE("winword", "c:\excel\examples\ale1.doc") =REQUEST(WChan, "BMk1") =TERMINATE(WChan) =RETURN()

Ventana	
Ventana	=DIALOG.BOX(control) =IF(Opcion=3) = OPEN(GET.DOCUMENT(2)&"\"&GET.DOCUMENT(1)) =END.IF() =RETURN()

DIALOGS

type	x	y	wide	high	text	init/result	names
------	---	---	------	------	------	-------------	-------

			1100	1100	Control de Llamadas Telefonicas		
13	407	50			Locales	FALSE	
13					Nacionales	TRUE	
13					Internacionales	FALSE	
13	600	48			Numero marcado	TRUE	
13					Fecha	FALSE	
13					Hora	FALSE	
13					Duracion	FALSE	
13					Impulsos	TRUE	
13					Costo	FALSE	
11						2	
12	818	51			Por extension		
12					Por linea		
12					Linea en particular		
12					Extension en particular		
12					Todas las extensiones		
5	400	25			Tipo de Llamada:		
5	596	28			Campos:		
5	814	29			Tipo de Reporte:		
1	421	183	88		OK		
2	517	183	88		Cancel		
6	7	45	160			amortize.xls	
18	7	68	160	84			
19	174	68	160	84			
20	176	51	160	12	Text		

13. ENSAMBLAJE EN UN PROGRAMA EN ASSEMBLER

Los nemónicos en assembler del programa deben ser escritos en un editor de texto. Algunas de los más son: QEDIT, WORDSTAR, EDLIN, etc. Cualquiera de ellos sirve, con la única condición de que el archivo, generado sea escrito por completo en mayúsculas y que la extensión del archivo sea ".asm". Esto es:

nombre.asm

Con el programa AVMAC51.EXE se traducen los némonicos de las intrucciones, las pseudo-operaciones y las direcciones simbólicas que se encuentran en el archivo *nombre.asm*, en los "opcodes" y direcciones numéricas que el procesador entiende.

Este programa genera un archivo con los números de línea del programa y los errores cometidos (*nombre.prn*) y otro archivo con el programa objeto (*nombre.obj*).

El programa AVLINK.EXE genera un archivo ".hex" tomando como base los ".obj" generados por AVMAC51.EXE. Este archivo tiene el formato INTEL hexadecimal, el cual tiene la siguiente estructura que se observa en la Figura 63

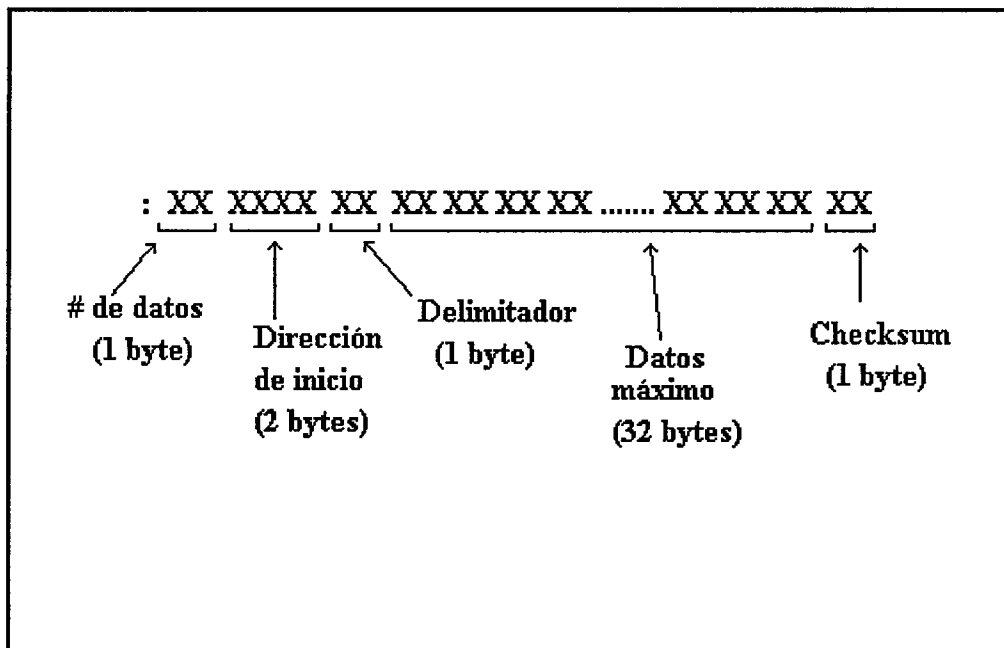


Fig. 63 Formato Intel Hexadecimal.

El programa HEXFORM.EXE toma el archivo ".hex" y lo convierte en el formato binario ".bin", el cual es el que puede ser grabado en EPROM. Sin embargo, si el grabador de EPROM lo permite se puede usar directamente el archivo ".hex".

Para lograr ensamblar un programa por módulos se debe crear un archivo principal el cual incluya los demás archivos con la pseudo-operación %include <nombre.asm> .

Al principio del programa se define un segmento, mediante la pseudo-operación:

```
DEFSEGNOMBRE_SEG,ABSOLUTE
```

```
SEGNOMBRE_SEG
```

con lo que se define un segmento con las direcciones absolutas. También se pueden definir segmentos con código relocizable.

La Figura 64 es un esquema de los pasos que se deben seguir para ensamblar un programa.

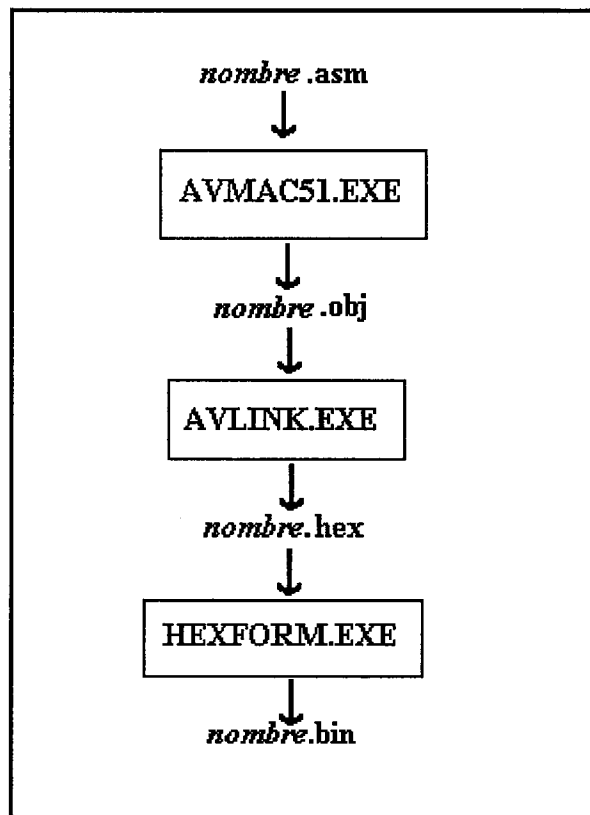


Fig. 64 Ensamblaje de un programa en ensamblador.

14. LISTADO DEL PROGRAMA EN BORLANDC++ PARA INTERFAZ CON EL USUARIO

```

/*****
/* Funcion =      InitInstance                               */
/* Descripción=  Inicializa la Instancia de una aplicación  */
/* Parametros   HINSTANCE hInst = Handle de la Instancia  */
/*              INT       iCmdShow = Bandera de visulización */
/*              de la ventana                               */
/* Devuelve     BOOL = True si la instancia pudo ser registrada */
/*              False en caso contrario                       */
*****/

```

```

BOOL

```

```

InitInstance(HINSTANCE hInst,int iCmdShow)

```

```

{
    hInstance=hInst;

    hWnd = CreateWindow("Proyecto",
                        "Tarificacion de Llamadas Telefónicas",
                        WS_OVERLAPPEDWINDOW|
                        WS_MAXIMIZE,
                        0,
                        0,
                        CW_USEDEFAULT,
                        CW_USEDEFAULT,
                        0,0,hInst,NULL);

    if (hWnd == NULL)
    {
        _InitEasyWin();
        perror("La Ventana no pudo ser creada");
        return FALSE;
    }

    ShowWindow(hWnd,iCmdShow|SW_SHOWMAXIMIZED);
    UpdateWindow(hWnd);

    return TRUE;
}

```

```
/* **** */
/* MODULO      = Definición                               */
/* NOMBRE ARCHIVO = Proyecto.def                          */
/* DESCRIPCION  = Definición de parámetros del programa  */
/* **** */
```

```
NAME      proyecto
DESCRIPTION 'Tarificación'
EXETYPE   WINDOWS
STUB      'WINSTUB.EXE'
CODE      PRELOAD MOVEABLE DISCARDABLE
DATA      PRELOAD MOVEABLE MULTIPLE
```

```
HEAPSIZE  1024
STACKSIZE 5120
```

```
EXPORTS
  MainWndProc      @1
  Config
  Tar_Urb
```

```

        NacInfo=MakeProcInstance((DLGPROC)CelInfo,
hInstance);
        DialogBox(hInstance,"TAR_CELINFO",hCodTar,
NacInfo);
        FreeProcInstance((FARPROC)NacInfo);
    }
    break;
} /* End del Switch del WM_COMMAND */
} /* End del switch de Mensaje */
return FALSE;
} /* Fin de la funcion */

```

```

LRESULT WINAPI Tar_Urb (HWND hTar,UINT Mensaje,WPARAM wParam,LPARAM lParam
)
{
    switch (Mensaje)
    {
        case WM_INITDIALOG:
            break;

        case WM_COMMAND:
            switch (wParam)
            {
                case IDCANCEL:
                    EndDialog(hTar,FALSE);
                    break;

            } /* End del switch de WM_COMMAND */
            break; /* Break de WM_COMMAND */

    } /*End del Swicht de Mensaje */
return FALSE;
}

```

```

LRESULT WINAPI Tar_NacDDN(HWND hTarNac,UINT Mensaje,WPARAM wParam,LPARAM l
Param)
{
    HWND vScroll;
    switch (Mensaje)
    {
        case WM_INITDIALOG:
            {

```

```

int cont;
FILE* f1;
char Aux[4];
struct Cantv
{
    char ciudad[25];
    int  codigo;
    int  tarifa;
}cantv;

f1=fopen("tarifas.dat","r+b");
if (f1==NULL)
    MessageBox(hTarNac,"!!!! No se pudo abrir archivo
!!!!","Mensaje",MB_OK);

for(cont = 0;cont<=MAX_CIUADAD; cont++)
{
    fseek(f1,cont*sizeof(struct Cantv),SEEK_SET);
    fread(&cantv,sizeof(struct Cantv),1,f1);
    SendDlgItemMessage(hTarNac, IDM_NACDDNCITY, LB_ADDST
RING, 0, (LPARAM) cantv.ciudad);
    sprintf(Aux, "%d", cantv.codigo);
    SendDlgItemMessage(hTarNac, IDM_NACDDNAREA, LB_ADDST
RING, 0, (LPARAM) Aux);
    sprintf(Aux, "%d", cantv.tarifa);
    SendDlgItemMessage(hTarNac, IDM_NACDDNTAR, LB_ADDSTR
ING, 0, (LPARAM) Aux);
}
fclose(f1);
}
break;

case WM_VSCROLL:
    vScroll=GetDlgItem(hTarNac, IDM_TARNACSCROLL);
    SetScrollRange(vScroll, SB_CTL, 1, MAX_CIUADAD, TRUE);
    switch (wParam)
    {
        int aux;
        case SB_LINEUP:
            aux=SendDlgItemMessage(hTarNac, IDM_NACDDNC
ITY, LB_GETCURSEL, 0, 0L);
            SendDlgItemMessage(hTarNac, IDM_NACDDNAREA,
LB_SETCURSEL, (WPARAM) aux-1, 0);
            SendDlgItemMessage(hTarNac, IDM_NACDDNTAR, L
B_SETCURSEL, (WPARAM) aux-1, 0);
            SendDlgItemMessage(hTarNac, IDM_NACDDNCITY,
LB_SETCURSEL, (WPARAM) aux-1, 0);
            SetScrollPos(vScroll, SB_CTL, aux-1, TRUE);

```



```

case IDM_NACDDNTAR:

    if (HIWORD(lParam) == LBN_SELCHANGE)
    {
        int aux;
        aux=SendDlgItemMessage(hTarNac, IDM
_NACDDNTAR, LB_GETCURSEL, 0, 0);
        SendDlgItemMessage(hTarNac, IDM_NAC
DDNAREA, LB_SETCURSEL, (WPARAM) aux, 0);
        SendDlgItemMessage(hTarNac, IDM_NAC
DDNCITY, LB_SETCURSEL, (WPARAM) aux, 0);
        if (!SetScrollPos(vScroll, SB_CTL, a
ux, TRUE))
            MessageBox(hTarNac, "Esto no sirve"
, "Mensaje 1", MB_OK);
    }

    break;
} /* End del Switch del WM_COMMAND */

} /* End del switch de Mensaje */

return FALSE;
} /* Fin de la funcion */

```

```

LRESULT WINAPI Tar_NacOper(HWND hTarNac, UINT Mensaje, WPARAM wParam, LPARAM
lParam)
{
    switch (Mensaje)
    {
        case WM_COMMAND:
            switch (wParam)
            {
                case IDCANCEL:
                    EndDialog(hTarNac, FALSE);
                    break;

                case IDOK:
                    {
                    }
                    break;

                case IDM_TARNACINFO:
                    {
                        DLGPROC NacInfo;

```

```

        NacInfo=MakeProcInstance((DLGPROC) CelInfo,
hInstance);
        DialogBox(hInstance,"TAR_CELINFO",hTarNac,
NacInfo);
        FreeProcInstance((FARPROC)NacInfo);
    }
} /* End del Switch del WM_COMMAND */

} /* End del switch de Mensaje */

return FALSE;
} /* Fin de la funcion */

```

```

LRESULT WINAPI Tar_Inter(HWND hTarInter,UINT Mensaje,WPARAM wParam,LPARAM
lParam)
{
    switch (Mensaje)
    {
        case WM_COMMAND:
            switch (wParam)
            {
                case IDCANCEL:
                    EndDialog(hTarInter,FALSE);
                    break;
            } /* End del Switch del WM_COMMAND */
    } /* End del switch de Mensaje */

    return FALSE;
} /* Fin de la funcion */

```

```

LRESULT WINAPI Tar_Cel(HWND hTarCel,UINT Mensaje,WPARAM wParam,LPARAM lPar
am)
{
    switch (Mensaje)
    {
        case WM_COMMAND:
            switch (wParam)
            {
                case IDCANCEL:
                    EndDialog(hTarCel,FALSE);
                    break;

                case IDM_TARCELINFO:
                    {
                        DLGPROC TarCelInfo;

```

```

        TarCelInfo = (DLGPROC)MakeProcInstance((FA
RPROC)CelInfo,hInstance);
        DialogBox(hInstance,"TAR_CELINFO",hTarCel,
TarCelInfo);
        FreeProcInstance((FARPROC)TarCelInfo);
    }
    break;
} /* End del Switch del WM_COMMAND */

} /* End del switch de Mensaje */

return FALSE;
} /* Fin de la funcion */

LRESULT WINAPI Config(HWND hConfig,UINT Mensaje,WPARAM wParam,LPARAM lParam)
{
    switch (Mensaje)
    {
        case WM_INITDIALOG:
            SendDlgItemMessage(hConfig,ID_CFG_FREQ,CB_ADDSTRING,0,(LPA
RAM)"50 Hz");
            SendDlgItemMessage(hConfig,ID_CFG_FREQ,CB_ADDSTRING,0,(LPA
RAM)"16 KHz");
            SendDlgItemMessage(hConfig,ID_CFG_FREQ,CB_ADDSTRING,0,(LPA
RAM)"20 KHz");
            SendDlgItemMessage(hConfig,ID_CFG_NOLIN,CB_ADDSTRING,0,(LP
ARAM)"8 Lineas");
            SendDlgItemMessage(hConfig,ID_CFG_NOLIN,CB_ADDSTRING,0,(LP
ARAM)"16 Lineas");
            SendDlgItemMessage(hConfig,ID_CFG_NOLIN,CB_ADDSTRING,0,(LP
ARAM)"24 Lineas");
            SendDlgItemMessage(hConfig,ID_CFG_FREQ,CB_SELECTSTRING,-1,
(LPARAM)"50 Hz");
            SendDlgItemMessage(hConfig,ID_CFG_NOLIN,CB_SELECTSTRING,-1
,(LPARAM)"8 Lineas");
            break;

        case WM_COMMAND:
            switch (wParam)
            {
                case IDCANCEL:
                    EndDialog(hConfig,FALSE);
                    break;

                case IDM_TARURB:
                    {
                        DLGPROC ProcTar;
                        ProcTar = (DLGPROC)MakeProcInstance((FARPROC)Tar_U
rb,hInstance);
                        DialogBox(hInstance,"TAR_URB",hConfig,ProcTar);
                    }
            }
    }
}

```

```

        FreeProcInstance((FARPROC)ProcTar);
    }
    break;

    case IDM_TARNAC:
    {
        int Bandera;
        DLGPROC ProcTar,Auxiliar;
        Auxiliar = (DLGPROC)MakeProcInstance((FARPROC)Aux_
Proc,hInstance);
        Bandera=DialogBox(hInstance,"TARNAC_CHOICE",hConfi
g,Auxiliar);
        FreeProcInstance((FARPROC)Auxiliar);
        if (Bandera == IDM_DDN)
        {
            ProcTar = (DLGPROC)MakeProcInstance((FARPR
OC)Tar_NacDDN,hInstance);
            DialogBox(hInstance,"TAR_NAC",hConfig,Proc
Tar);
            FreeProcInstance((FARPROC)ProcTar);
        }
        if (Bandera == IDM_OPERADORA)
        {
            ProcTar = (DLGPROC)MakeProcInstance((FARPR
OC)Tar_NacOper,hInstance);
            DialogBox(hInstance,"TARNAC_OPERADOR",hCon
fig,ProcTar);
            FreeProcInstance((FARPROC)ProcTar);
        }
    }
    break;

    case IDM_TARINTER:
    {
        DLGPROC ProcTar;
        ProcTar = (DLGPROC)MakeProcInstance((FARPROC)Tar_I
nter,hInstance);
        DialogBox(hInstance,"TAR_INTER",hConfig,ProcTar);
        FreeProcInstance((FARPROC)ProcTar);
    }
    break;

    case IDM_TARCEL:
    {
        DLGPROC ProcTar;
        ProcTar = (DLGPROC)MakeProcInstance((FARPROC)Tar_C
el,hInstance);
        DialogBox(hInstance,"TAR_CEL",hConfig,ProcTar);
        FreeProcInstance((FARPROC)ProcTar);
    }
    break;

```

```
        } /* End del switch de WM_COMMAND */  
break; /* Break de WM_COMMAND */  
    } /*End del Swicht de Mensaje */  
return FALSE;  
}
```

```

/*****
/* MODULO          = Principal
/* NOMBRE ARCHIVO = Proyecto.c
/* DESCRIPCION    = Inicialización Aplicación e Instancia
/*****

```

```

#include <windows.h>
#include "proyecto.h"

```

```

/*****
/* Funcion =      WinMain
/* Descripción= Punto de entrada a la aplicación
/* Parametros  HINSTANCE hInst =      Handle de la Instancia
/*
/*              HINSTANCE hPrevInst = Handle de la Instancia
/*              previa
/*              LPSTR lpCmdLine =      Apuntador a una linea
/*              de comando.
/*              INT iCmdShow =        Bandera de
/*              visualización.
/* Devuelve     INT PASCAL =          No es usado por Windows
/*              Actualmente
/*****

```

```

int PASCAL WinMain(HINSTANCE hInst,HINSTANCE hPrevInst,LPSTR lpCmdLine,
                  int iCmdShow)

```

```

{
MSG msg;
/* Si es la primera aplicación que esta corriendo (hPrevInst = 0)
se registran las clases (InitApplication, en caso de error en
la inicialización se devuelve FALSE*/
if (!hPrevInst)
if (!InitApplication(hInst))
return 0;

/* Realiza las inicializaciones correspondiente a cada ocurrencia
de una aplicación. */

if (!InitInstance(hInst,iCmdShow))
return 0;

while (GetMessage(&msg,0,0,0))
{
TranslateMessage(&msg);
DispatchMessage(&msg);
};

return msg.wParam;
}

```

```

/*****
/* MODULO          = Mensajes          */
/* NOMBRE ARCHIVO = Proy_men.c          */
/* DESCRIPCION    = Manejo de mensajes de la Ventana principal*/
*****/

#include <windows.h>
#include <commdlg.h>
#include <string.h>
#include "proyecto.h"

extern hInstance;

/*****
/* Funcion =      MainWndProc          */
/* Descripción=  Manejo de Mensajes de la Ventana Principal          */
/* Parametros   HWND   hWnd           = Handle de la Ventana Principal          */
/*              UINT   Message        = True si la clase pudo ser registrada          */
/*              WPARAM wParam         = Parametro que trae informacion extra          */
/*              LPARAM lParam         = Parametro que trae informacion extra          */
/* Devuelve     LRESULT WINAPI retorna FALSE en caso de exito          */
*****/

/*****
/* Funcion de Ventana          */
*****/
LRESULT WINAPI MainWndProc(HWND hWnd,UINT Message,WPARAM wParam,
                          LPARAM lParam)
{
    switch (Message)
    {
        case WM_DESTROY:
            PostQuitMessage(0);
            break;

        case WM_COMMAND:
            switch (wParam)
            {
                case IDM_MNUCONFIG:
                    DLGPROC ProcConfig;

```

```

ProcConfig =(DLGPROC)MakeProcInstance((FARPROC)Config,hInst
nce);

DialogBox(hInstance,"CONFIG",hWnd,ProcConfig);
FreeProcInstance((FARPROC)ProcConfig);
}
break;

case IDM_MNUABRIR:
{
OPENFILENAME ofn;
char szFile[256], szFileTitle[256];
UINT i, cbString;
char chReplace; /* string separator for szFilter */
char szFilter[256];
HFILE hf;

szFile[0] = '\0';

if ((cbString = LoadString(hInstance, IDS_FILTRO,
szFilter, sizeof(szFilter))) == 0)
return 0L;

chReplace = szFilter[cbString - 1]; /* retrieve wildcard */

for (i = 0; szFilter[i] != '\0'; i++)
{
if (szFilter[i] == chReplace)
szFilter[i] = '\0';
}

/* Set all structure members to zero. */
memset(&ofn, 0, sizeof(OPENFILENAME));

ofn.lStructSize = sizeof(OPENFILENAME);
ofn.hwndOwner = hWnd;
ofn.lpstrFilter = szFilter;
ofn.nFilterIndex = 1;
ofn.lpstrFile = szFile;
ofn.nMaxFile = sizeof(szFile);
ofn.lpstrFileTitle = szFileTitle;
ofn.nMaxFileTitle = sizeof(szFileTitle);
ofn.lpstrInitialDir = NULL;
ofn.Flags = OFN_SHOWHELP | OFN_PATHMUSTEXIST | OFN_FILEMUSTE

ofn.lpstrTitle = "Abrir";

```

IST;


```

        if (GetOpenFileName(&ofn))
        {
            HMENU hMenu,hSubMenu;
            hMenu=GetMenu(hWnd);
            hSubMenu=GetSubMenu(hMenu,0);
            EnableMenuItem(hSubMenu, IDM_MNUGUARDAR, MF_ENABLED|MF_BYC
MMAND);
            EnableMenuItem(hSubMenu, IDM_MNUIMPRIMIR, MF_ENABLED|MF_BY
OMMAND);
            EnableMenuItem(hMenu, IDM_MNUCONFIG, MF_ENABLED|MF_BYCOMMA
D);
            DrawMenuBar(hWnd);
            hf = _lopen(ofn.lpstrFile, OF_READ);

                /*COLOCAR LO QUE SE QUIERA HACER ON EL ARCHIVO*/

        }
        else
            ;

    }
    break;

    case IDM_MNUSALIR:
        DestroyWindow(hWnd);
        break;

    case IDM_MNURECUPERAR:
        {
            DLGPROC Recuperar;
            Recuperar = (DLGPROC)MakeProcInstance((FARPROC)Rec_Proc,hIn
tance);
            DialogBox(hInstance,"BASE_D",hWnd,Recuperar);
            FreeProcInstance((FARPROC)Recuperar);
        }
        break;
    } /*End del switch de WM_COMMAND */

    default:
        return DefWindowProc(hWnd,Message,wParam,lParam);

}; /*End del switch de Message */

return FALSE;
}

```

```

/*****
/* MODULO           = Recursos
/* NOMBRE ARCHIVO   = Proyecto.rc
/* DESCRIPCION      = Definición de recursos
*****/

#include "proyecto.h"

CONFIG DIALOG 5, 15, 225, 138
STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION
CAPTION "Configuración"
BEGIN
    CONTROL "", ID_CFG_FREQ, "COMBOBOX", CBS_DROPDOWNLIST | WS
    _CHILD | WS_VISIBLE | WS_TABSTOP, 100, 6, 48, 36
    CONTROL "", ID_CFG_NOLIN, "COMBOBOX", CBS_DROPDOWNLIST | W
    S_CHILD | WS_VISIBLE | WS_TABSTOP, 75, 25, 52, 40
    CONTROL "12", ID_CFG_HORA, "EDIT", ES_LEFT | WS_CHILD | WS
    _VISIBLE | WS_TABSTOP, 40, 48, 8, 12
    CONTROL "00", ID_CFG_MIN, "EDIT", ES_LEFT | WS_CHILD | WS_
    VISIBLE | WS_TABSTOP, 52, 48, 8, 12
    CONTROL "00", ID_CFG_SEG, "EDIT", ES_LEFT | WS_CHILD | WS_
    VISIBLE | WS_TABSTOP, 64, 48, 8, 12
    CONTROL "01", ID_CFG_DIA, "EDIT", ES_LEFT | WS_CHILD | WS_
    VISIBLE | WS_TABSTOP, 38, 76, 8, 12
    CONTROL "01", ID_CFG_DIA, "EDIT", ES_LEFT | WS_CHILD | WS_
    VISIBLE | WS_TABSTOP, 50, 76, 8, 12
    CONTROL "94", ID_CFG_DIA, "EDIT", ES_LEFT | WS_CHILD | WS_
    VISIBLE | WS_TABSTOP, 62, 76, 8, 12
    CONTROL "Button", IDOK, "BorBtn", BS_PUSHBUTTON | WS_CHILD
    | WS_VISIBLE, 28, 100, 32, 20
    CONTROL "Button", IDCANCEL, "BorBtn", BS_PUSHBUTTON | WS_C
    HILD | WS_VISIBLE, 88, 100, 32, 20
    LTEXT "Frecuencia de los impulsos:", -1, 2, 7, 93, 8, WS_C
    HILD | WS_VISIBLE | WS_GROUP
    LTEXT "Número de líneas:", -1, 4, 24, 60, 7, WS_CHILD | WS
    _VISIBLE | WS_GROUP
    LTEXT "Hora:", -1, 5, 46, 19, 8, WS_CHILD | WS_VISIBLE | W
    S_GROUP
    LTEXT "Fecha:", -1, 4, 75, 22, 9, WS_CHILD | WS_VISIBLE |
    WS_GROUP
    LTEXT "/", -1, 47, 76, 3, 8, WS_CHILD | WS_VISIBLE | WS_GR
    OUP
    LTEXT "/", -1, 59, 76, 3, 8, WS_CHILD | WS_VISIBLE | WS_GR
    OUP
    PUSHBUTTON "Urbana", IDM_TARURB, 156, 52, 52, 12, WS_CHILD
    | WS_VISIBLE | WS_TABSTOP
    LTEXT ":", -1, 49, 48, 3, 8, WS_CHILD | WS_VISIBLE | WS_GR
    OUP
    LTEXT ":", -1, 61, 48, 3, 8, WS_CHILD | WS_VISIBLE | WS_GR
    OUP

```

```

CONTROL "Tarifas", -1, "STATIC", SS_LEFTNOWORDWRAP | WS_CH
ILD | WS_VISIBLE | WS_GROUP, 168, 36, 24, 8
PUSHBUTTON "Nacional", IDM_TARNAC, 156, 64, 52, 12, WS_CHI
LD | WS_VISIBLE | WS_TABSTOP
PUSHBUTTON "Internacional", IDM_TARINTER, 156, 76, 52, 12,
WS_CHILD | WS_VISIBLE | WS_TABSTOP
PUSHBUTTON "Celular", IDM_TARCEL, 156, 88, 52, 12, WS_CHIL
D | WS_VISIBLE | WS_TABSTOP
CONTROL "", -1, "static", SS_BLACKFRAME | WS_CHILD | WS_VI
SIBLE, 148, 44, 68, 60
END

```

```

MENU_1 MENU
BEGIN

```

```

POPUP "&Archivo"
BEGIN

```

```

MENUITEM "&Abrir", IDM_MNUABRIR
MENUITEM "&Guardar", IDM_MNUGUARDAR, GRAYED
MENUITEM SEPARATOR
MENUITEM "&Imprimir", IDM_MNUIMPRIMIR, GRAYED
MENUITEM SEPARATOR
MENUITEM "&Salir", IDM_MNUSALIR

```

```

END

```

```

MENUITEM "&Configurar", IDM_MNUCONFIG, GRAYED
MENUITEM "&Recuperar", IDM_MNURECUPERAR
POPUP "A&yuda"
BEGIN

```

```

MENUITEM "&Instrucciones", IDM_MNUINST
MENUITEM SEPARATOR
MENUITEM "Acerca de ...", IDM_MNUABOUT

```

```

END

```

```

END

```

```

BASE_D DIALOG 6, 15, 300, 178

```

```

STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION

```

```

CAPTION "Base de Datos"

```

```

BEGIN

```

```

CTEXT "Hora", -1, 17, 5, 16, 8, WS_CHILD | WS_VISIBLE | WS
_GROUP

```

```

CTEXT "Fecha", -1, 46, 5, 33, 8, WS_CHILD | WS_VISIBLE | W
S_GROUP

```

```

CTEXT "Duracion", -1, 85, 5, 47, 8, WS_CHILD | WS_VISIBLE
| WS_GROUP

```

```

CTEXT "Costo", -1, 150, 5, 25, 7, WS_CHILD | WS_VISIBLE |
WS_GROUP

```

```

CTEXT "Numero Destino", -1, 190, 5, 64, 8, WS_CHILD | WS_V
ISIBLE | WS_GROUP

```

```

CONTROL "", -1, "static", SS_BLACKRECT | WS_CHILD | WS_VI
SIBLE, 1, 16, 256, 1

```

```

CONTROL "", 101, "LISTBOX", LBS_NOTIFY | LBS_MULTIPLESEL |
LBS_NOINTEGRALHEIGHT | WS_CHILD | WS_VISIBLE | WS_BORDER, 5, 24,
35, 108
CONTROL "", 102, "LISTBOX", LBS_NOTIFY | LBS_MULTIPLESEL |
LBS_NOINTEGRALHEIGHT | WS_CHILD | WS_VISIBLE | WS_BORDER, 45, 24,
40, 108
CONTROL "", 102, "LISTBOX", LBS_NOTIFY | LBS_MULTIPLESEL |
LBS_NOINTEGRALHEIGHT | WS_CHILD | WS_VISIBLE | WS_BORDER, 90, 24,
45, 108
CONTROL "", 102, "LISTBOX", LBS_NOTIFY | LBS_MULTIPLESEL |
LBS_NOINTEGRALHEIGHT | WS_CHILD | WS_VISIBLE | WS_BORDER | WS_VSC
ROLL, 193, 24, 70, 108
CONTROL "", 102, "LISTBOX", LBS_NOTIFY | LBS_MULTIPLESEL |
LBS_NOINTEGRALHEIGHT | WS_CHILD | WS_VISIBLE | WS_BORDER, 140, 24
, 45, 108
PUSHBUTTON "Estadísticas...", IDM_DBEST, 176, 148, 56, 16,
WS_CHILD | WS_VISIBLE | WS_TABSTOP
CONTROL "Button", IDOK, "BorBtn", BS_PUSHBUTTON | WS_CHILD
| WS_VISIBLE, 76, 148, 32, 20
END
TAR_URB DIALOG 64, 35, 156, 117
STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION
CAPTION "Tarifa Urbana"
BEGIN
LTEXT "Costo Impulso:", -1, 4, 3, 52, 8, WS_CHILD | WS_VIS
IBLE | WS_GROUP
LTEXT "Residencial", -1, 12, 24, 40, 8, WS_CHILD | WS_VISI
BLE | WS_GROUP
LTEXT "No Residencial", -1, 12, 44, 56, 8, WS_CHILD | WS_V
ISIBLE | WS_GROUP
CONTROL "1,18", 101, "EDIT", ES_LEFT | WS_CHILD | WS_VISIB
LE | WS_BORDER | WS_TABSTOP, 80, 20, 20, 12
CONTROL "1,56", 102, "EDIT", ES_LEFT | WS_CHILD | WS_VISIB
LE | WS_BORDER | WS_TABSTOP, 80, 40, 20, 12
LTEXT "Frecuencia de los Impulsos :", -1, 4, 60, 52, 16, W
S_CHILD | WS_VISIBLE | WS_GROUP
LTEXT "90 seg", -1, 80, 64, 36, 8, WS_CHILD | WS_VISIBLE |
WS_GROUP
CONTROL "Button", IDOK, "BorBtn", BS_PUSHBUTTON | WS_CHILD
| WS_VISIBLE, 36, 88, 32, 20
CONTROL "Button", IDCANCEL, "BorBtn", BS_PUSHBUTTON | WS_C
HILD | WS_VISIBLE, 99, 88, 32, 20
LTEXT "Bs", -1, 104, 23, 16, 8, WS_CHILD | WS_VISIBLE | WS
_GROUP
LTEXT "Bs", -1, 103, 44, 16, 8, WS_CHILD | WS_VISIBLE | WS
_GROUP
END
TAR_NAC DIALOG 29, 23, 254, 169
STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION
CAPTION "Tarifas Nacionales"
BEGIN

```

```

CONTROL "Ciudad", -1, "STATIC", WS_CHILD | WS_VISIBLE, 42,
12, 24, 8
CONTROL "Codigo de Area", -1, "STATIC", WS_CHILD | WS_VISIBLE, 108, 8, 28, 16
CONTROL "Codigo de Tarifa", -1, "STATIC", WS_CHILD | WS_VISIBLE, 143, 8, 32, 16
CONTROL "", -1, "static", SS_BLACKRECT | WS_CHILD | WS_VISIBLE, 4, 24, 188, 1
CONTROL "", IDM_NACDDNCITY, "LISTBOX", LBS_NOTIFY | LBS_NOINTEGRALHEIGHT | WS_CHILD | WS_VISIBLE | WS_BORDER | WS_GROUP, 8, 28, 92, 88
CONTROL "", IDM_NACDDNAREA, "LISTBOX", LBS_NOTIFY | LBS_NOINTEGRALHEIGHT | WS_CHILD | WS_VISIBLE | WS_BORDER | WS_GROUP, 112, 28, 20, 88
CONTROL "", IDM_NACDDNTAR, "LISTBOX", LBS_NOTIFY | LBS_NOINTEGRALHEIGHT | WS_CHILD | WS_VISIBLE | WS_BORDER | WS_GROUP, 149, 28, 20, 88
CONTROL "Costo Impulso:", -1, "STATIC", WS_CHILD | WS_VISIBLE, 16, 124, 52, 8
CONTROL "Residencial", -1, "STATIC", WS_CHILD | WS_VISIBLE, 28, 136, 40, 8
CONTROL "No Residencial", -1, "STATIC", WS_CHILD | WS_VISIBLE, 28, 152, 56, 8
CONTROL "1,18", 101, "EDIT", ES_LEFT | WS_CHILD | WS_VISIBLE | WS_BORDER | WS_TABSTOP, 88, 136, 20, 12
CONTROL "1,56", 102, "EDIT", ES_LEFT | WS_CHILD | WS_VISIBLE | WS_BORDER | WS_TABSTOP, 88, 152, 20, 12
PUSHBUTTON "Codigo de Tarifa...", IDM_CODTAR, 147, 134, 68, 15, WS_CHILD | WS_VISIBLE | WS_TABSTOP
CONTROL "Button", IDOK, "BorBtn", BS_PUSHBUTTON | WS_CHILD | WS_VISIBLE, 204, 36, 32, 20
CONTROL "Button", IDCANCEL, "BorBtn", BS_PUSHBUTTON | WS_CHILD | WS_VISIBLE, 204, 72, 32, 20
CONTROL "Bs", -1, "STATIC", SS_LEFT | WS_CHILD | WS_VISIBLE, 109, 138, 12, 8
CONTROL "Bs", -1, "STATIC", WS_CHILD | WS_VISIBLE, 109, 154, 12, 8
SCROLLBAR IDM_TARNACSCROLL, 177, 28, 9, 88, SBS_VERT | WS_CHILD | WS_VISIBLE
CONTROL "", 103, "BorShade", BSS_GROUP | WS_CHILD | WS_VISIBLE, 0, 0, 256, 168
CONTROL "", 104, "BorShade", BSS_GROUP | WS_CHILD | WS_VISIBLE, 13, 122, 117, 43
END

STRINGTABLE
BEGIN
IDS_FILTRO, "Bases de Datos(*.DAT)|*.dat|Todos(*.*)|*.*|"
END

TAR_CEL DIALOG 55, 33, 204, 142

```

```

STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION
CAPTION "Tarifa Celular"
BEGIN
    LTEXT "Costo Impulso:", -1, 14, 13, 52, 8, WS_CHILD | WS_V
ISIBLE | WS_GROUP
    LTEXT "Residencial", -1, 22, 34, 40, 8, WS_CHILD | WS_VISI
BLE | WS_GROUP
    LTEXT "No Residencial", -1, 22, 54, 56, 8, WS_CHILD | WS_V
ISIBLE | WS_GROUP
    CONTROL "1,18", 101, "EDIT", ES_LEFT | WS_CHILD | WS_VISIB
LE | WS_BORDER | WS_TABSTOP, 86, 30, 19, 12
    CONTROL "1,56", 102, "EDIT", ES_LEFT | WS_CHILD | WS_VISIB
LE | WS_BORDER | WS_TABSTOP, 86, 50, 19, 12
    LTEXT "Tarifa Normal:", -1, 8, 83, 52, 7, WS_CHILD | WS_VI
SIBLE | WS_GROUP
    LTEXT "1 Impulso cada 2 seg", -1, 80, 83, 79, 8, WS_CHILD
| WS_VISIBLE | WS_GROUP
    CONTROL "Button", IDOK, "BorBtn", BS_PUSHBUTTON | WS_CHILD
| WS_VISIBLE, 58, 114, 32, 20
    CONTROL "Button", IDCANCEL, "BorBtn", BS_PUSHBUTTON | WS_C
HILD | WS_VISIBLE, 119, 114, 32, 20
    LTEXT "Tarifa Especial:", -1, 8, 97, 52, 7, WS_CHILD | WS_
VISIBLE | WS_GROUP
    LTEXT "1 Impulso cada 2,5 seg", -1, 80, 96, 79, 8, WS_CHIL
D | WS_VISIBLE | WS_GROUP
    PUSHBUTTON "Informacion...", IDM_TARCELINFO, 140, 59, 56,
18, WS_CHILD | WS_VISIBLE | WS_TABSTOP
    LTEXT "Bs.", -1, 108, 34, 16, 8, WS_CHILD | WS_VISIBLE | W
S_GROUP
    LTEXT "Bs.", -1, 108, 54, 16, 8, WS_CHILD | WS_VISIBLE | W
S_GROUP
END
TAR_CELINFO DIALOG 55, 25, 153, 135
STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION
CAPTION "Informacion"
BEGIN
    LTEXT "Tarifa Normal: Se aplica los dias laborales entre 7
.00 am y 7.00 pm", -1, 15, 13, 122, 22
    LTEXT "Tarifa Especial: Se aplica en los dias laborales de
7.00 pm a 7.00 am, los dias sábados y domingos, y los dias feriad
os las 24 horas del día", -1, 18, 48, 116, 40, WS_CHILD | WS_VISI
BLE | WS_GROUP
    CONTROL "", -1, "static", SS_BLACKFRAME | WS_CHILD | WS_VI
SIBLE, 11, 9, 131, 86
    CONTROL "Button", IDOK, "BorBtn", BS_PUSHBUTTON | WS_CHILD
| WS_VISIBLE, 60, 108, 32, 20
    CONTROL "", 101, "BorShade", BSS_GROUP | WS_CHILD | WS_VIS
IBLE, 0, 0, 152, 136
END
TAR_INTER DIALOG 12, 14, 290, 189
STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION

```

CAPTION "Tarifa Internacional"

BEGIN

```

CONTROL "", 101, "LISTBOX", LBS_NOTIFY | WS_CHILD | WS_VISIBLE |
WS_BORDER, 8, 36, 68, 120
CONTROL "", 102, "LISTBOX", LBS_NOTIFY | WS_CHILD | WS_VISIBLE |
WS_BORDER, 83, 36, 64, 120
CONTROL "", 103, "LISTBOX", LBS_NOTIFY | WS_CHILD | WS_VISIBLE |
WS_BORDER, 156, 36, 24, 120
CONTROL "", 104, "LISTBOX", LBS_NOTIFY | WS_CHILD | WS_VISIBLE |
WS_BORDER, 188, 36, 24, 120
LISTBOX 105, 252, 36, 32, 120
CONTROL "", 106, "LISTBOX", LBS_NOTIFY | WS_CHILD | WS_VISIBLE |
WS_BORDER, 220, 36, 24, 120
LTEXT "Continente", -1, 23, 12, 36, 8, WS_CHILD | WS_VISIBLE |
WS_GROUP
LTEXT "Pais", -1, 108, 12, 16, 8
LTEXT "Codigo", -1, 156, 14, 24, 8
CTEXT "Tarifa Normal", -1, 188, 6, 28, 20, WS_CHILD | WS_VISIBLE |
WS_GROUP
CTEXT "Tarifa Reducida", -1, 216, 6, 32, 20, WS_CHILD | WS_VISIBLE |
WS_GROUP
CTEXT "Tarifa Economica", -1, 248, 6, 40, 20, WS_CHILD | WS_VISIBLE |
WS_GROUP
CONTROL "", -1, "static", SS_BLACKFRAME | WS_CHILD | WS_VISIBLE, 5, 2, 284, 28
CONTROL "Button", IDOK, "BorBtn", BS_PUSHBUTTON | WS_CHILD |
WS_VISIBLE, 44, 164, 32, 20
CONTROL "Button", IDCANCEL, "BorBtn", BS_PUSHBUTTON | WS_CHILD |
WS_VISIBLE, 105, 164, 32, 20
PUSHBUTTON "Informacion", IDM_TARINTERINFO, 196, 164, 72, 20,
WS_CHILD | WS_VISIBLE | WS_TABSTOP
END

```

FERIADOS DIALOG 37, 13, 175, 187

STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION

CAPTION "Dias Feriados"

BEGIN

```

CONTROL "1", 1, "BorBtn", BBS_BITMAP | WS_CHILD | WS_VISIBLE, 76, 166, 12, 12
CONTROL "Año Nuevo", 101, "EDIT", ES_LEFT | WS_CHILD | WS_VISIBLE |
WS_BORDER | WS_GROUP | WS_TABSTOP, 8, 8, 93, 12
CONTROL "Carnaval", 101, "EDIT", ES_LEFT | WS_CHILD | WS_VISIBLE |
WS_BORDER | WS_GROUP | WS_TABSTOP, 8, 22, 93, 12
CONTROL "Jueves Santo", 101, "EDIT", ES_LEFT | WS_CHILD | WS_VISIBLE |
WS_BORDER | WS_TABSTOP, 8, 36, 93, 12
CONTROL "Viernes Santo", 101, "EDIT", ES_LEFT | WS_CHILD | WS_VISIBLE |
WS_BORDER | WS_TABSTOP, 8, 50, 93, 12
CONTROL "Independencia", 101, "EDIT", ES_LEFT | WS_CHILD | WS_VISIBLE |
WS_BORDER | WS_TABSTOP, 8, 64, 93, 12
CONTROL "Dia del Trabajo", 101, "EDIT", ES_LEFT | WS_CHILD | WS_VISIBLE |
WS_BORDER | WS_TABSTOP, 8, 78, 93, 12

```

```

CONTROL "Batalla de Carabobo", 101, "EDIT", ES_LEFT | WS_C
HILD | WS_VISIBLE | WS_BORDER | WS_TABSTOP, 8, 92, 93, 12
CONTROL "Independencia", 101, "EDIT", ES_LEFT | WS_CHILD |
WS_VISIBLE | WS_BORDER | WS_TABSTOP, 8, 106, 93, 12
CONTROL "Natalicio de Simon Bolivar", 101, "EDIT", ES_LEFT
| WS_CHILD | WS_VISIBLE | WS_BORDER | WS_TABSTOP, 8, 120, 93, 12
CONTROL "Dia de la Raza", 101, "EDIT", ES_LEFT | WS_CHILD
| WS_VISIBLE | WS_BORDER | WS_TABSTOP, 8, 134, 93, 12
CONTROL "1 Enero", 101, "EDIT", ES_LEFT | WS_CHILD | WS_VI
SIBLE | WS_BORDER | WS_TABSTOP, 108, 8, 60, 12
CONTROL "14-15 Febrero", 101, "EDIT", ES_LEFT | WS_CHILD |
WS_VISIBLE | WS_BORDER | WS_TABSTOP, 108, 22, 60, 12
CONTROL "31 Marzo", 101, "EDIT", ES_LEFT | WS_CHILD | WS_V
ISIBLE | WS_BORDER | WS_TABSTOP, 108, 36, 60, 12
CONTROL "1 Abril", 101, "EDIT", ES_LEFT | WS_CHILD | WS_VI
SIBLE | WS_BORDER | WS_TABSTOP, 108, 50, 60, 12
CONTROL "19 Abril", 101, "EDIT", ES_LEFT | WS_CHILD | WS_V
ISIBLE | WS_BORDER | WS_TABSTOP, 108, 64, 60, 12
CONTROL "1 Mayo", 101, "EDIT", ES_LEFT | WS_CHILD | WS_VIS
IBLE | WS_BORDER | WS_TABSTOP, 108, 78, 60, 12
CONTROL "24 Junio", 101, "EDIT", ES_LEFT | WS_CHILD | WS_V
ISIBLE | WS_BORDER | WS_TABSTOP, 108, 92, 60, 12
CONTROL "5 Julio", 101, "EDIT", ES_LEFT | WS_CHILD | WS_VI
SIBLE | WS_BORDER | WS_TABSTOP, 108, 106, 60, 12
CONTROL "24 Julio", 101, "EDIT", ES_LEFT | WS_CHILD | WS_V
ISIBLE | WS_BORDER | WS_TABSTOP, 108, 120, 60, 12
CONTROL "12 Octubre", 101, "EDIT", ES_LEFT | WS_CHILD | WS
_VISIBLE | WS_BORDER | WS_TABSTOP, 108, 134, 60, 12
CONTROL "Navidad", 101, "EDIT", ES_LEFT | WS_CHILD | WS_VI
SIBLE | WS_BORDER | WS_TABSTOP, 8, 150, 93, 12
CONTROL "25 Diciembre", 101, "EDIT", ES_LEFT | WS_CHILD |
WS_VISIBLE | WS_BORDER | WS_TABSTOP, 108, 150, 60, 12
END

```

```
FRAN HOR DIALOG 20, 17, 228, 158
```

```
STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION
```

```
CAPTION "Franjas Horarias para Tarifas Internacionales"
```

```
BEGIN
```

```
CONTROL "Tarifa Reducida", 101, "EDIT", ES_LEFT | ES_READO
NLY | WS_CHILD | WS_VISIBLE | WS_DISABLED | WS_BORDER | WS_TABSTOP
, 4, 6, 80, 96
```

```
CONTROL "Tarifa Economica", 101, "EDIT", ES_LEFT | WS_CHIL
D | WS_VISIBLE | WS_DISABLED | WS_BORDER | WS_TABSTOP, 4, 102, 80,
24
```

```
CONTROL "America del Norte", 102, "EDIT", ES_LEFT | WS_CHI
LD | WS_VISIBLE | WS_DISABLED | WS_BORDER | WS_TABSTOP, 84, 6, 80,
12
```

```
CONTROL "America del Sur", 102, "EDIT", ES_LEFT | WS_CHILD
| WS_VISIBLE | WS_DISABLED | WS_BORDER | WS_TABSTOP, 84, 18, 80,
12
```

```
CONTROL "Antillas", 102, "EDIT", ES_LEFT | WS_CHILD | WS_V
```



```

VISIBLE | WS_DISABLED | WS_BORDER | WS_TABSTOP, 84, 30, 80, 12
CONTROL "Asia Occidental", 102, "EDIT", ES_LEFT | WS_CHILD
| WS_VISIBLE | WS_DISABLED | WS_BORDER | WS_TABSTOP, 84, 42, 80,
12
CONTROL "Resto del Mundo", 102, "EDIT", ES_LEFT | WS_CHILD
| WS_VISIBLE | WS_DISABLED | WS_BORDER | WS_TABSTOP, 84, 54, 80,
12
CONTROL "Asia Oriental", 102, "EDIT", ES_LEFT | WS_CHILD |
WS_VISIBLE | WS_DISABLED | WS_BORDER | WS_TABSTOP, 84, 66, 80, 12
CONTROL "America Central", 102, "EDIT", ES_LEFT | WS_CHILD
| WS_VISIBLE | WS_DISABLED | WS_BORDER | WS_TABSTOP, 84, 78, 80,
12
CONTROL "Europa", 102, "EDIT", ES_LEFT | WS_CHILD | WS_VIS
IBLE | WS_DISABLED | WS_BORDER | WS_TABSTOP, 84, 90, 80, 12
CONTROL "Europa", 102, "EDIT", ES_LEFT | WS_CHILD | WS_VIS
IBLE | WS_DISABLED | WS_BORDER | WS_TABSTOP, 84, 102, 80, 12
CONTROL "Resto del Mundo", 102, "EDIT", ES_LEFT | WS_CHILD
| WS_VISIBLE | WS_DISABLED | WS_BORDER | WS_TABSTOP, 84, 114, 80,
12
CONTROL "18h-24h", 103, "EDIT", ES_LEFT | WS_CHILD | WS_VI
SIBLE | WS_DISABLED | WS_BORDER | WS_TABSTOP, 164, 6, 56, 12
CONTROL "18h-24h", 103, "EDIT", ES_LEFT | WS_CHILD | WS_VI
SIBLE | WS_DISABLED | WS_BORDER | WS_TABSTOP, 164, 18, 56, 12
CONTROL "18h-24h", 103, "EDIT", ES_LEFT | WS_CHILD | WS_VI
SIBLE | WS_DISABLED | WS_BORDER | WS_TABSTOP, 164, 30, 56, 12
CONTROL "18h-7h", 103, "EDIT", ES_LEFT | WS_CHILD | WS_VIS
IBLE | WS_DISABLED | WS_BORDER | WS_TABSTOP, 164, 42, 56, 12
CONTROL "18h-7h", 103, "EDIT", ES_LEFT | WS_CHILD | WS_VIS
IBLE | WS_DISABLED | WS_BORDER | WS_TABSTOP, 164, 54, 56, 12
CONTROL "22h-7h", 103, "EDIT", ES_LEFT | WS_CHILD | WS_VIS
IBLE | WS_DISABLED | WS_BORDER | WS_TABSTOP, 164, 66, 56, 12
CONTROL "16h-24h", 103, "EDIT", ES_LEFT | WS_CHILD | WS_VI
SIBLE | WS_DISABLED | WS_BORDER | WS_TABSTOP, 164, 78, 56, 12
CONTROL "16h-20h", 103, "EDIT", ES_LEFT | WS_CHILD | WS_VI
SIBLE | WS_DISABLED | WS_BORDER | WS_TABSTOP, 164, 90, 56, 12
CONTROL "20h-7h", 103, "EDIT", ES_LEFT | WS_CHILD | WS_VIS
IBLE | WS_DISABLED | WS_BORDER | WS_TABSTOP, 164, 102, 56, 12
CONTROL "00h-7h", 103, "EDIT", ES_LEFT | WS_CHILD | WS_VIS
IBLE | WS_DISABLED | WS_BORDER | WS_TABSTOP, 164, 114, 56, 12
CONTROL "Button", IDOK, "BorBtn", BS_PUSHBUTTON | WS_CHILD
| WS_VISIBLE, 152, 135, 32, 20
CONTROL "Button", IDCANCEL, "BorBtn", BS_PUSHBUTTON | WS_C
HILD | WS_VISIBLE, 186, 135, 32, 20
END

```

```
TARNAC INFO DIALOG 17, 15, 263, 187
```

```
STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION
```

```
CAPTION "Llamadas a través de Discado Directo Nacional (DDN)"
```

```
BEGIN
```

```
CONTROL "Cadencia :", -1, "STATIC", SS_CENTER | SS_NOPREFI
X | WS_CHILD | WS_VISIBLE | WS_BORDER, 8, 3, 41, 10
```

```

    CTEXT "Codigo de Tarifa", -1, 8, 16, 38, 18, WS_CHILD | WS
_VISIBLE | WS_GROUP
    CTEXT "Distancia (Kms)", -1, 75, 16, 32, 16, WS_CHILD | WS
_VISIBLE | WS_GROUP
    CTEXT "Tarifa Normal 1 Impulso cada", -1, 136, 16, 50, 17
, WS_CHILD | WS_VISIBLE | WS_GROUP
    CTEXT "Tarifa Especial 1 Impulso cada", -1, 204, 16, 52, 1
6, WS_CHILD | WS_VISIBLE | WS_GROUP
    CONTROL "", -1, "static", SS_BLACKFRAME | WS_CHILD | WS_VI
SIBLE, 4, 36, 252, 1
    CONTROL "", -1, "static", SS_BLACKFRAME | WS_CHILD | WS_VI
SIBLE, 8, 39, 244, 16
    CONTROL "", -1, "static", SS_BLACKFRAME | WS_CHILD | WS_VI
SIBLE, 8, 60, 244, 16
    CONTROL "", -1, "static", SS_BLACKFRAME | WS_CHILD | WS_VI
SIBLE, 7, 80, 244, 16
    CONTROL "", -1, "static", SS_BLACKFRAME | WS_CHILD | WS_VI
SIBLE, 8, 100, 244, 16
    CONTROL "", -1, "static", SS_BLACKFRAME | WS_CHILD | WS_VI
SIBLE, 8, 120, 244, 16
    CONTROL "", -1, "static", SS_BLACKFRAME | WS_CHILD | WS_VI
SIBLE, 8, 140, 244, 16
    CONTROL "", -1, "static", SS_BLACKRECT | WS_CHILD | WS_VIS
IBLE, 56, 39, 1, 117
    CONTROL "", -1, "static", SS_BLACKRECT | WS_CHILD | WS_VIS
IBLE, 124, 39, 1, 117
    CONTROL "", -1, "static", SS_BLACKRECT | WS_CHILD | WS_VIS
IBLE, 195, 39, 1, 116
    CTEXT "1", -1, 19, 43, 16, 8, WS_CHILD | WS_VISIBLE | WS_G
ROUP
    CTEXT "2", -1, 19, 65, 16, 8, WS_CHILD | WS_VISIBLE | WS_G
ROUP
    CTEXT "3", -1, 19, 84, 16, 8, WS_CHILD | WS_VISIBLE | WS_G
ROUP
    CTEXT "4", -1, 19, 104, 16, 8, WS_CHILD | WS_VISIBLE | WS_
GROUP
    CTEXT "5", -1, 19, 125, 16, 8, WS_CHILD | WS_VISIBLE | WS_
GROUP
    CTEXT "6", -1, 19, 144, 16, 8, WS_CHILD | WS_VISIBLE | WS_
GROUP
    CTEXT "0 - 40", -1, 76, 44, 30, 9, WS_CHILD | WS_VISIBLE |
WS_GROUP
    CTEXT "501 en adelante", -1, 63, 144, 56, 8, WS_CHILD | WS
_VISIBLE | WS_GROUP
    CTEXT "41-90", -1, 80, 65, 22, 9, WS_CHILD | WS_VISIBLE |
WS_GROUP
    CTEXT "91-170", -1, 77, 85, 27, 8, WS_CHILD | WS_VISIBLE |
WS_GROUP
    CTEXT "171-300", -1, 77, 105, 28, 8, WS_CHILD | WS_VISIBLE
| WS_GROUP
    CTEXT "301-500", -1, 76, 125, 30, 8, WS_CHILD | WS_VISIBLE

```



```

CONTROL "169,5", 102, "EDIT", ES_LEFT | WS_CHILD | WS_VISI
BLE | WS_BORDER | WS_TABSTOP, 132, 152, 25, 12
CONTROL "15,5", 102, "EDIT", ES_LEFT | WS_CHILD | WS_VISI
LE | WS_BORDER | WS_TABSTOP, 166, 51, 25, 12
CONTROL "23,5", 102, "EDIT", ES_LEFT | WS_CHILD | WS_VISI
LE | WS_BORDER | WS_GROUP | WS_TABSTOP, 166, 72, 25, 12
CONTROL "29,5", 102, "EDIT", ES_LEFT | WS_CHILD | WS_VISI
LE | WS_BORDER | WS_TABSTOP, 166, 92, 25, 12
CONTROL "40,5", 102, "EDIT", ES_LEFT | WS_CHILD | WS_VISI
LE | WS_BORDER | WS_TABSTOP, 166, 112, 25, 12
CONTROL "49,5", 102, "EDIT", ES_LEFT | WS_CHILD | WS_VISI
LE | WS_BORDER | WS_TABSTOP, 166, 132, 25, 12
CONTROL "56,5", 102, "EDIT", ES_LEFT | WS_CHILD | WS_VISI
LE | WS_BORDER | WS_TABSTOP, 166, 152, 25, 12
CONTROL "28,5", 102, "EDIT", ES_LEFT | WS_CHILD | WS_VISI
LE | WS_BORDER | WS_TABSTOP, 204, 51, 25, 12
CONTROL "44,25", 102, "EDIT", ES_LEFT | WS_CHILD | WS_VISI
BLE | WS_BORDER | WS_TABSTOP, 204, 72, 25, 12
CONTROL "57,0", 102, "EDIT", ES_LEFT | WS_CHILD | WS_VISI
LE | WS_BORDER | WS_TABSTOP, 204, 92, 25, 12
CONTROL "78,75", 102, "EDIT", ES_LEFT | WS_CHILD | WS_VISI
BLE | WS_BORDER | WS_TABSTOP, 204, 112, 25, 12
CONTROL "98,25", 102, "EDIT", ES_LEFT | WS_CHILD | WS_VISI
BLE | WS_BORDER | WS_TABSTOP, 204, 132, 25, 12
CONTROL "104,25", 102, "EDIT", ES_LEFT | WS_CHILD | WS_VISI
IBLE | WS_BORDER | WS_TABSTOP, 204, 152, 25, 12
CONTROL "9,5", 102, "EDIT", ES_LEFT | WS_CHILD | WS_VISIBL
E | WS_BORDER | WS_TABSTOP, 244, 51, 25, 12
CONTROL "14,75", 102, "EDIT", ES_LEFT | WS_CHILD | WS_VISI
BLE | WS_BORDER | WS_TABSTOP, 243, 72, 25, 12
CONTROL "19,0", 102, "EDIT", ES_LEFT | WS_CHILD | WS_VISI
LE | WS_BORDER | WS_TABSTOP, 244, 92, 25, 12
CONTROL "26,25", 102, "EDIT", ES_LEFT | WS_CHILD | WS_VISI
BLE | WS_BORDER | WS_TABSTOP, 244, 112, 25, 12
CONTROL "32,75", 102, "EDIT", ES_LEFT | WS_CHILD | WS_VISI
BLE | WS_BORDER | WS_TABSTOP, 244, 132, 25, 12
CONTROL "34,75", 102, "EDIT", ES_LEFT | WS_CHILD | WS_VISI
BLE | WS_BORDER | WS_TABSTOP, 244, 152, 25, 12
CONTROL "", IDOK, "BorBtn", BS_PUSHBUTTON | WS_CHILD | WS_
VISIBLE, 57, 169, 32, 20
CONTROL "Button", IDCANCEL, "BorBtn", BS_PUSHBUTTON | WS_C
HILD | WS_VISIBLE, 112, 169, 32, 19
CONTROL "Cadencia :", -1, "STATIC", SS_LEFT | WS_CHILD | W
S_VISIBLE | WS_BORDER | WS_GROUP, 10, 7, 44, 9
CONTROL "Codigo de Tarifa", -1, "STATIC", SS_CENTER | WS_C
HILD | WS_VISIBLE | WS_BORDER | WS_GROUP, 12, 24, 38, 18
CTEXT "Distancia (Kms)", -1, 80, 26, 32, 16, WS_CHILD | WS
_VISIBLE | WS_GROUP
LTEXT "Tarifa Normal", -1, 139, 4, 46, 8, WS_CHILD | WS_VI
SIBLE | WS_GROUP
LTEXT "Tarifa Especial", -1, 221, 4, 51, 9, WS_CHILD | WS_

```

```

VISIBLE | WS_GROUP
CONTROL "", -1, "static", SS_BLACKFRAME | WS_CHILD | WS_VI
SIBLE, 8, 46, 278, 1
CONTROL "", -1, "static", SS_BLACKFRAME | WS_CHILD | WS_VI
SIBLE, 12, 49, 260, 16
CONTROL "", -1, "static", SS_BLACKFRAME | WS_CHILD | WS_VI
SIBLE | WS_GROUP, 12, 70, 260, 16
CONTROL "", -1, "static", SS_BLACKFRAME | WS_CHILD | WS_VI
SIBLE | WS_GROUP, 12, 90, 260, 16
CONTROL "", -1, "static", SS_BLACKFRAME | WS_CHILD | WS_VI
SIBLE, 12, 110, 260, 16
CONTROL "", -1, "static", SS_BLACKFRAME | WS_CHILD | WS_VI
SIBLE, 12, 130, 260, 16
CONTROL "", -1, "static", SS_BLACKFRAME | WS_CHILD | WS_VI
SIBLE, 12, 150, 260, 16
CONTROL "", -1, "static", SS_BLACKRECT | WS_CHILD | WS_VIS
IBLE, 60, 49, 1, 117
CONTROL "", -1, "static", SS_BLACKRECT | WS_CHILD | WS_VIS
IBLE, 128, 49, 1, 117
CONTROL "", -1, "static", SS_BLACKRECT | WS_CHILD | WS_VIS
IBLE, 199, 49, 1, 116
LTEXT "1", -1, 24, 53, 16, 8, WS_CHILD | WS_VISIBLE | WS_G
ROUP
LTEXT "2", -1, 24, 75, 16, 8, WS_CHILD | WS_VISIBLE | WS_G
ROUP
LTEXT "3", -1, 24, 94, 16, 8, WS_CHILD | WS_VISIBLE | WS_G
ROUP
LTEXT "4", -1, 24, 114, 16, 8, WS_CHILD | WS_VISIBLE | WS_
GROUP
LTEXT "5", -1, 24, 135, 16, 8, WS_CHILD | WS_VISIBLE | WS_
GROUP
LTEXT "6", -1, 24, 154, 16, 8, WS_CHILD | WS_VISIBLE | WS_
GROUP
LTEXT "0 - 40", -1, 80, 54, 30, 9, WS_CHILD | WS_VISIBLE |
WS_GROUP
LTEXT "501 en adelante", -1, 67, 154, 56, 8, WS_CHILD | WS
_VISIBLE | WS_GROUP
LTEXT "41-90", -1, 80, 75, 22, 9, WS_CHILD | WS_VISIBLE |
WS_GROUP
LTEXT "91-170", -1, 80, 95, 27, 8, WS_CHILD | WS_VISIBLE |
WS_GROUP
LTEXT "171-300", -1, 80, 115, 28, 8, WS_CHILD | WS_VISIBLE
| WS_GROUP
LTEXT "301-500", -1, 80, 135, 30, 8, WS_CHILD | WS_VISIBLE
| WS_GROUP
PUSHBUTTON "Información...", IDM_TARNACINFO, 189, 171, 56,
16, WS_CHILD | WS_VISIBLE | WS_TABSTOP
CTEXT "Conexion 3 minutos (Bs)", -1, 127, 18, 35, 26, WS_C
HILD | WS_VISIBLE | WS_GROUP
CTEXT "Minuto Adicional (Bs)", -1, 168, 18, 30, 25, WS_CHI
LD | WS_VISIBLE | WS_GROUP

```

```

        CTEXT "Conexion 3 minutos (Bs)", -1, 204, 18, 35, 26, WS_C
HILD | WS_VISIBLE | WS_GROUP
        CTEXT "Minuto Adicional (Bs)", -1, 247, 18, 30, 25, WS_CHI
LD | WS_VISIBLE | WS_GROUP
        CONTROL "", 103, "BorShade", 1 | WS_CHILD | WS_VISIBLE | W
S_GROUP, 0, 2, 298, 189
        CONTROL "", 104, "BorShade", BSS_GROUP | WS_CHILD | WS_VIS
IBLE, 76, 24, 40, 20
        CONTROL "", 105, "BorShade", BSS_GROUP | WS_CHILD | WS_VIS
IBLE, 8, 22, 47, 22
        CONTROL "", 106, "BorShade", BSS_GROUP | WS_CHILD | WS_VIS
IBLE, 123, 17, 161, 28
        CONTROL "", 107, "BorShade", BSS_GROUP | WS_CHILD | WS_VIS
IBLE, 6, 46, 279, 122
END

```

```

TARNAC_CHOICE_DIALOG 18, 18, 155, 102
STYLE DS_MODALFRAME | WS_POPUP | WS_CAPTION
CAPTION "Tarifa Nacional"
BEGIN
        CONTROL "Discado Directo Nacional", IDM_DDN, "BUTTON", BS_
AUTORADIOBUTTON | WS_CHILD | WS_VISIBLE | WS_TABSTOP, 16, 16, 92,
12
        CONTROL "Llamada por Operadora", IDM_OPERADORA, "BUTTON",
BS_AUTORADIOBUTTON | WS_CHILD | WS_VISIBLE | WS_TABSTOP, 16, 32, 8
8, 12
        CONTROL "Button", IDCANCEL, "BorBtn", BS_PUSHBUTTON | WS_C
HILD | WS_VISIBLE, 92, 68, 32, 20
        CONTROL "Button", IDOK, "BorBtn", BS_PUSHBUTTON | WS_CHILD
| WS_VISIBLE, 36, 68, 32, 20
        CONTROL "", -1, "static", SS_BLACKFRAME | WS_CHILD | WS_VI
SIBLE, 10, 8, 104, 44
END

```

```

/*****
/* MODULO           = Librería
/* NOMBRE ARCHIVO   = Proyecto.h
/* DESCRIPCION      = Definicion de constantes
/*
/* Definicion de Prototipo
*****/

/* Constantes */

//#ifndef __PROYECTO_H

#define __PROYECTO_H

#define MAX_CIUADAD           36
#define MENÜ_1                100
#define IDM_DBEST             MENU_1+1
#define IDM_BDCERRAR          IDM_DBEST+1
#define IDM_TARURB            IDM_BDCERRAR+1
#define IDM_TARNAC            IDM_TARURB+1
#define IDM_TARINTER          IDM_TARNAC+1
#define IDM_TARCEL            IDM_TARINTER+1
#define IDM_MNUDB             IDM_TARCEL+1
#define IDM_MNUTAR            IDM_MNUDB+1
#define IDM_MNUDBG            IDM_MNUTAR+1
#define IDM_MNUDBA            IDM_MNUDBG+1
#define IDM_MNUTARA           IDM_MNUDBA+1
#define IDM_MNUTARG           IDM_MNUTARA+1
#define IDM_MNUABRIR          IDM_MNUTARG+1
#define IDM_MNUGUARDAR        IDM_MNUABRIR+1
#define IDM_MNUIMPRIMIR       IDM_MNUGUARDAR+1
#define IDM_MNUSALIR          IDM_MNUIMPRIMIR+1
#define IDM_MNUCONFIG         IDM_MNUSALIR+1
#define IDM_MNURECUPERAR     IDM_MNUCONFIG+1
#define IDM_MNUINST           IDM_MNURECUPERAR+1
#define IDM_MNUABOUT         IDM_MNUINST+1
#define IDM_CFGOK             IDM_MNUABOUT+1
#define IDM_CFGCANCEL        IDM_CFGOK+1
#define IDM_CFGTAR            IDM_CFGCANCEL+1
#define ID_CFG_OK             IDM_CFGTAR+1
#define ID_CFG_CANCEL        ID_CFG_OK+1
#define ID_CFG_TARIFAS       ID_CFG_CANCEL+1
#define ID_CFG_DIA            ID_CFG_TARIFAS+1
#define ID_CFG_MES            ID_CFG_DIA+1
#define ID_CFG_ANO            ID_CFG_MES+1
#define ID_CFG_HORA           ID_CFG_ANO+1
#define ID_CFG_MIN            ID_CFG_HORA+1
#define ID_CFG_SEG            ID_CFG_MIN+1
#define ID_CFG_NOLIN          ID_CFG_SEG+1
#define ID_CFG_FREC           ID_CFG_NOLIN+1
#define IDS_FILTRO            ID_CFG_FREC+1

```

```

#define IDM_CODTAR                IDS_FILTRO+1
#define IDM_TARNACINFO            IDM_CODTAR+1
#define IDM_TARINTERINFO         IDM_TARNACINFO+1
#define IDM_TARCELINFO           IDM_TARINTERINFO+1
#define IDM_DDN                  IDM_TARCELINFO+1
#define IDM_OPERADORA            IDM_DDN+1
#define IDM_NACDDNCITY           IDM_OPERADORA+1
#define IDM_NACDDNAREA          IDM_NACDDNCITY+1
#define IDM_NACDDNTAR           IDM_NACDDNAREA+1
#define IDM_TARNACSCROLL         IDM_NACDDNTAR+1

```

```

/* Definiciones de Prototipo */

```

```

// Funciones Internas

```

```

BOOL InitApplication(HINSTANCE hInst);

```

```

BOOL InitInstance(HINSTANCE hInst, int iCmdShow);

```

```

// Funciones Externas

```

```

LRESULT WINAPI MainWndProc(HWND hWnd, UINT Message, WPARAM wParam,
                          LPARAM lParam);

```

```

LRESULT WINAPI Config(HWND hConfig, UINT Mensaje, WPARAM wParam,
                     LPARAM lParam);

```

```

LRESULT WINAPI Tar_Urb (HWND hTar, UINT Mensaje, WPARAM wParam,
                      LPARAM lParam);

```

```

LRESULT WINAPI CelInfo(HWND hTarCelInfo, UINT Mensaje,
                      WPARAM wParam, LPARAM lParam);

```

```

LRESULT WINAPI Rec_Proc(HWND hRec, UINT Mensaje, WPARAM wParam,
                      LPARAM lParam);

```

```

//#endif

```