

73  
24



UNIVERSIDAD NACIONAL AUTONOMA  
DE MEXICO

FACULTAD DE INGENIERIA

DISEÑO Y CONSTRUCCION DE UN SISTEMA QUE  
OPTIMICE LA CAPTURA Y PROCESAMIENTO DE  
DATOS EN CADENAS DE RESTAURANTES

T E S I S

QUE PARA OBTENER EL TITULO DE:  
INGENIERO MECANICO  
ELECTRICISTA  
PRESENTAN:  
EDGAR CORONADO LOPEZ  
SERGIO DE LA FUENTE TORRES

DIRECTOR DE TESIS: ING. JOSE FRANCISCO ESCAMILLA GUZMAN  
CODIRECTOR DE TESIS: ING. LARRY HIPOLITO ESCOBAR SALGUERO



MEXICO, D. F.

1996

TESIS CON  
FALLA DE ORIGEN

TESIS CON  
FALLA DE ORIGEN



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

## AGRADECIMIENTOS

A la Universidad Nacional Autónoma de México, por haber hecho posible nuestra formación profesional.

A todas las personas que de una u otra manera ayudaron a la elaboración de esta tesis. En especial al:

Ing. Francisco Escamilla Guzmán y al  
Ing. Larry Hipólito Escobar Salguero.

## DEDICATORIAS

A mi padre Sr. G. Antonio Coronado, y a mi madre Sra. Ma. Dalila Irma López, quienes confiaron en mí y me ayudaron a formarme como persona y como profesionista con su amor, paciencia y apoyo.

A mi hermana Irma Lina, y a mis hermanos Juan Antonio y Oscar Arael, por todas las experiencias vividas a través de todos estos años.

Edgar.

## DEDICATORIAS

A mis padres, Margarita y Pablo  
que con su cariño, apoyo y plena  
confianza hicieron posible mi  
formación como profesionista.

A mis hermanos, Pablo, Agustín,  
Margarita y Mónica por todo lo  
que ustedes significan para mi.

Sergio.

DISEÑO Y CONSTRUCCION  
DE UN SISTEMA QUE  
OPTIMICE LA CAPTURA Y  
PROCESAMIENTO DE DATOS  
EN CADENAS DE  
RESTAURANTES

## INDICE

### INTRODUCCION

Página

<b>1 FUNDAMENTOS TEORICOS.....</b>	<b>1</b>
1.1 Modulación.....	1
1.1.1 Modulación analógica.....	1
1.1.2 Modulación de señales digitales.....	3
1.2 Osciladores.....	5
1.2.1 Oscilador LC.....	5
1.2.2 Oscilador Hartley.....	6
1.2.3 Oscilador Colpitts.....	7
1.2.4 Oscilador Clapp.....	8
1.2.5 Oscilador controlado por cristal.....	8
1.3 Amplificadores.....	9
1.3.1 Clasificación de amplificadores.....	10
1.4 Mezcladores.....	13
1.5 Filtros.....	13
1.5.1 Análisis de tipos de filtros.....	14
1.6 Arquitectura de una computadora personal (PC).....	16
1.6.1 Unidad central de procesamiento (CPU).....	17
1.6.2 Unidad de control.....	17
1.6.3 Unidad aritmética lógica (ALU).....	18
1.6.4 Registros.....	18
1.6.5 Sistema operativo.....	21
1.6.6 Dispositivos de entrada/salida (I/O).....	21
1.6.7 Slot de la computadora personal.....	23
1.7 Métodos de transmisión.....	26
1.7.1 Transmisión de caracteres.....	27
1.7.2 Sincronización por caracter.....	28
1.7.3 Protocolos.....	29
<b>2 HERRAMIENTAS DEL PROYECTO.....</b>	<b>31</b>
2.1 La familia del microcontrolador TMS370.....	31
2.1.1 Aplicaciones comunes.....	31
2.1.2 Elementos de la familia TMS370.....	31
2.1.3 Componentes comunes en la arquitectura de la familia TMS370.....	32
2.1.4 Diagrama a bloques del TMS370Cx2x.....	35

2.2	Lenguajes de programación.....	36
2.2.1	Lenguajes de alto nivel.....	37
2.2.2	Compiladores.....	38
2.2.3	Lenguaje C.....	39
2.3	Lenguaje ensamblador de la familia TMS370.....	39
2.3.1	Modos de direccionamiento.....	40
<b>3</b>	<b>ORGANIZACION DEL SISTEMA.....</b>	<b>42</b>
3.1	Diagrama a bloques del sistema.....	42
3.2	Terminal de captura inalámbrica, (TCI).....	43
3.2.1	Teclado.....	44
3.2.2	Pantalla.....	44
3.2.3	Microcontrolador (MC).....	45
3.2.4	Modulador-demodulador, (Modem).....	45
3.2.5	Transmisor.....	45
3.2.6	Receptor.....	46
3.2.7	Antena.....	47
3.2.8	Circuitos de apoyo.....	47
3.3	Módulo central.....	48
3.3.1	Módulo de Comunicaciones de la Interfaz, (MCI).....	48
3.3.2	Interfaz de la Computadora Personal, (IPC).....	49
<b>4</b>	<b>DISEÑO DE TERMINALES DE CAPTURA INALAMBRICA.....</b>	<b>51</b>
4.1	Circuito de la pantalla.....	51
4.2	Circuito del teclado.....	52
4.3	Circuito del microcontrolador MC.....	53
4.4	Circuitos de radio-frecuencia.....	55
4.4.1	Circuito de transmisión.....	56
4.4.2	Circuito de recepción.....	58
4.5	Circuito de modulación <i>FSK</i> .....	59
4.6	Circuitos de apoyo.....	61
4.6.1	Circuito de encendido y apagado de iluminación de teclado y pantalla.....	61
4.6.2	Circuito de sonido.....	63
4.6.3	Circuito indicador de nivel de batería.....	65
4.6.4	Circuito de alimentación.....	67
4.7	Integración de los circuitos que forman la TCI.....	67



5 DISEÑO DEL MODULO CENTRAL.....	74
5.1 Circuitos del módulo de comunicaciones de la interfaz, MCI..	74
5.2 Integración de los circuitos que forman el MCI.....	74
5.3 Circuitos de la interfaz de la PC, IPC.....	78
5.3.1 Circuito del modem.....	78
5.3.2 Circuitos de interfaz microcontrolador-bus-PC.....	79
5.4 Integración de los circuitos que forman la IPC.....	86

## 6 PROGRAMACION DEL SISTEMA.

6.1 Programación del microcontrolador de terminal inalámbrica..	91
6.2 Programación del microcontrolador de la interfaz de la PC..	118
6.3 Programa de la PC.....	126

## RESULTADOS Y CONCLUSIONES.

### APENDICE.A HOJAS DE ESPECIFICACIONES TECNICAS

### APENDICE B LISTADO DEL PROGRAMA DEL MICROCONTROLADOR DEL MODULO CENTRAL Y DEL PROGRAMA DE LA PC.

## BIBLIOGRAFIA

# INTRODUCCION

## INTRODUCCION

La forma tradicional de operación de los restaurantes, en la cual los meseros toman la orden de los clientes en papel, para posteriormente llevarla a la cocina, así como los métodos que se utilizan para el control de almacén estadísticas de ventas, ocasiona con mucha frecuencia pérdida de tiempo, dinero y lo más importante el cliente.

Partiendo de estas necesidades se plantea diseñar un sistema, que controlará estos factores para eficientizar la operación de los restaurantes.

Debido a lo anterior, el objetivo de esta tesis es el diseño e implementación de un sistema de captura, comunicación y procesamiento de datos controlado por un MCU TMS370C622 y una computadora personal (PC).

El sistema que se propone consiste de terminales portátiles, con las cuales el mesero tome las órdenes en el área de mesas para que éstas sean transmitidas vía radio frecuencia (RF) hacia una computadora personal (PC) que manejará y enviará la información directamente a la cocina para su atención.

En la PC estará un programa de monitoreo con el cual se podrá observar el funcionamiento del restaurante.

El empleo de este sistema aportará los siguientes beneficios:

- Al procesarse electrónicamente la petición del cliente no habrá error en el pedido.
- Se llevará un control exacto de las órdenes que levanta cada mesero para la asignación de propinas e incentivos.
- Con el programa de la PC se podrá llevar un control de almacén y estadística de venta, debido a que en la PC se almacena la orden pedida, la identificación del mesero que atiende y la mesa atendida. Es por eso, que se pueden realizar operaciones tales como: descontar los ingredientes empleados en el platillo, de la existencia en almacén, lo que nos dará una información real de las existencias, se conocerá qué platillos son más solicitados y en qué época para afrontar estas curvas de demanda.

El sistema tendrá las siguientes características:

Los equipos portátiles utilizarán una etapa de radio frecuencia (RF) para la comunicación de datos, para lo cual se emplearán las bandas de 46 y 49 Mhz. Contará con un teclado para la introducción de información y un display para la visualización de ésta y la enviada por la PC. Como

dispositivos de control se empleará el microcontrolador (MC) marca Texas Instruments (TMS370622).

La PC llevará una tarjeta que se adicionará en uno de sus *slot's*, la cual servirá como una interfaz entre los equipos portátiles y el *bus* de la PC. El dispositivo de control será también el MC TMS370C622. El programa de la PC se realizará en lenguaje C.

# CAPITULO I

## FUNDAMENTOS TEORICOS

El objetivo de este capítulo es presentar los conceptos básicos de los elementos que conforman un sistema de comunicación via radio frecuencia (RF), tales como; osciladores, mezcladores, amplificadores y filtros. Y los principales tipos de modulación de señales que existen. También en el presente capítulo se describirá la organización de un sistema de cómputo. Se mencionarán los dispositivos necesarios para la introducción y extracción de información en una computadora personal (PC). Se describirán los métodos de transmisión y recepción más comunes y los tipos de protocolos necesarios para establecer una comunicación.

## 1.1 MODULACION.

La modulación es el proceso mediante el cual se altera una onda portadora variando uno de los parámetros de la señal: amplitud, frecuencia o fase, de acuerdo a la señal de entrada (señal moduladora); mientras que la demodulación es el proceso inverso, es decir, se obtiene la señal original a partir de la señal modulada.

Muchos tipos de señales no pueden ser enviadas directamente por un canal de transmisión debido a la insuficiente inmunidad al ruido eléctrico, limitaciones de ancho de banda, y principalmente debido a que las frecuencias de las señales de interés por lo general se encuentran dentro de un intervalo de frecuencias muy bajas, por lo que sus longitudes de onda son bastante largas y en consecuencia la antena transmisora que se tendría que utilizar para convertir las señales en radiación electromagnética sería muy grande. Por estas razones, es necesario utilizar una forma de onda "portadora" con longitud de onda más corta y cuyas propiedades se adapten mejor al medio de transmisión que se esté empleando. Debido a esto, la antena transmisora que se utilice tendrá una longitud moderada.

En general las señales se clasifican en dos tipos, aquellas que pueden tomar cualquier valor de amplitud dentro de un rango y que son llamadas analógicas y las que solo pueden tener un número finito de valores de amplitud llamados digitales. Debido a esto, es posible identificar dos técnicas de modulación, las cuales se discuten a continuación.

### 1.1.1 MODULACION ANALOGICA.

En este tipo de modulación, la señal de información es analógica y la onda portadora la constituye una señal senoidal de mayor frecuencia; la cual dependiendo del parámetro que le sea variado, ya sea la amplitud, fase o frecuencia, será el tipo de modulación empleada. Los tipos de modulación para este caso son:

- Amplitud Modulada (AM).
- Frecuencia Modulada (FM).
- Modulación de Fase (MF).

**AMPLITUD MODULADA (AM)**. En este tipo de modulación, la frecuencia y la fase de la onda portadora se mantiene constantes; no así su amplitud, la cual varía de acuerdo a la señal de entrada (moduladora).

No obstante, este proceso es muy sensible al ruido, por lo que su empleo sólo resulta conveniente en aquellos sistemas en los que no afecta que la señal recuperada venga mezclada con ruido.

En la fig. 1.1.1. se muestra un ejemplo de la modulación AM.

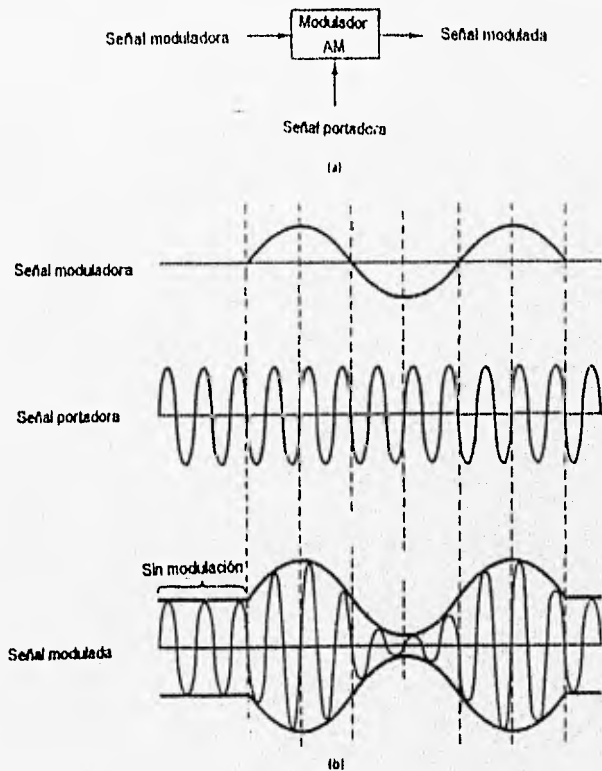


Fig. 1.1.1 Modulación en AM: a) modulador AM, b) proceso de modulación en AM.

**FRECUENCIA MODULADA (FM).** En este caso, la amplitud y la fase de la onda portadora se mantiene constante; no así su frecuencia, la cual varía de acuerdo a la señal de entrada.

**MODULACION DE FASE (MF).** Este proceso a diferencia de los anteriores ocasiona que la fase de la onda portadora, varíe de acuerdo a la señal de entrada. De tal forma que cada cambio en la fase represente la información deseada. En tales sistemas, el cambio de fase es referenciado a la fase de la señal actual y no a una posición predeterminada.

En la fig. 1.1.2 se muestra un ejemplo de la modulación FM y MF.

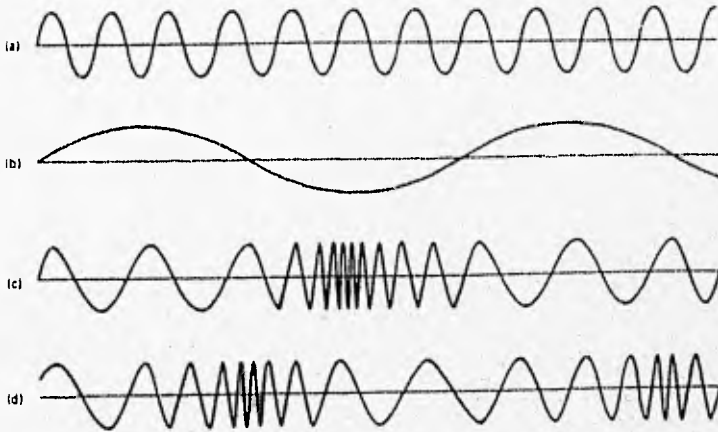


Fig. 1.1.2 Modulación en FM y MF: a) señal portadora, b) señal moduladora, c) señal modulada en FM, d) señal modulada en MF.

### 1.1.2 MODULACION DE SEÑALES DIGITALES.

Para éste tipo de modulación la señal de entrada es una señal digital y la onda portadora, es una onda senoidal de mayor frecuencia, la cual dependiendo del parámetro variado; ya sea amplitud, fase o frecuencia, será el tipo de modulación empleado:

- Modulación por cambio de amplitud ASK (*Amplitud Shift Keyed*).
- Modulación por cambio de frecuencia FSK (*Frequency Shift Keyed*).
- Modulación por cambio de fase PSK (*Phase Shift Keyed*).



**MODULACION POR CAMBIO DE AMPLITUD (ASK).** En este tipo de modulación la amplitud de la señal portadora se alterna entre dos o más valores, por lo general en 0 lógico y 1 lógico, correspondiente a las señales binarias. En la fig. 1.1.3b se muestra un ejemplo de ASK en el cual la onda portadora es modulada por cambio de amplitud de acuerdo al mensaje binario 101101.

**MODULACION POR CAMBIO DE FASE (PSK).** En la modulación por cambio de fase la modulación ocurre cuando la fase de la onda portadora varía tantas veces como lo hace la señal de entrada (señal moduladora), en donde cada cambio en la fase representa un dígito binario. En la fig. 1.1.3c se muestra un ejemplo de PSK en el cual la onda portadora es modulada por cambio de fase de acuerdo al mensaje binario 101101.

**MODULACION POR CAMBIO DE FRECUENCIA (FSK).** La modulación FSK es parecida a la modulación FM, excepto que como ya se mencionó, la señal moduladora es un tren de pulsos binario que varía entre dos niveles de voltaje en comparación a la forma de onda continua y variante en el tiempo de la señal moduladora para el caso de la modulación en FM. En la fig. 1.1.3d se muestra un ejemplo de FSK en el cual la onda portadora es modulada por cambio de frecuencia de acuerdo al mensaje binario 101101.

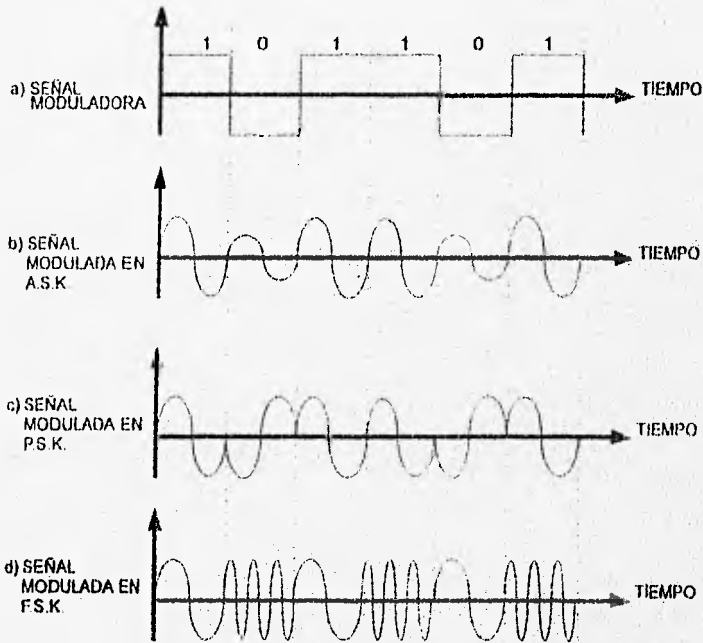


Fig. 1.1.3 Modulación digital a) señal moduladora, b) modulación ASK, c) modulación PSK, d) modulación FSK.

## 1.2 OSCILADORES.

Un oscilador es un dispositivo electrónico que proporciona una señal de salida, sin contar con una señal de entrada, mas que sólo una fuente de alimentación. La señal de salida debe ser periódica, usualmente con una frecuencia fija, y una forma de onda en particular. La forma de onda de la salida puede ser senoidal, cuadrada, diente sierra, triangular, etc.

Los osciladores para radio frecuencia RF proveen una salida senoidal, estos se utilizan en la mayoría de los receptores, como los de AM, FM, y receptores de Televisión, en estaciones de radio como parte del transmisor, en los sistemas de radar, hornos de microondas, etc.

Existen varios tipos de osciladores de onda senoidal disponibles para circuitos electrónicos. El criterio par seleccionar un tipo de oscilador se basa en la siguiente lista:

- Frecuencia de salida requerida.
- Estabilidad de frecuencia requerida.
- Intervalos de frecuencia, si la frecuencia es variable.
- Distorsión permisible en la forma de onda.
- Potencia de salida requerida.

Estas consideraciones de comportamiento combinadas con los factores económicos, determinan la forma del oscilador a ser usado en una aplicación.

### 1.2.1 OSCILADOR LC.

La acción de cargar el capacitor de la fig. 1.2.1a a un determinado voltaje y el cierre del interruptor genera la forma de onda de la fig.1.2.1b. El interruptor permite el flujo de corriente conforme el capacitor empieza a descargarse a través del inductor. El inductor se opone al cambio del flujo de la corriente y genera una forma de onda gradual senoidal de corriente que alcanza su máximo cuando el capacitor se descarga completamente. Ya que en este punto el voltaje del capacitor es cero, el flujo de corriente es máxima, por lo que la energía del campo magnético alrededor del inductor es máxima. El campo magnético en el inductor se empieza a colapsar debido al voltaje del capacitor, provocando que su fuerza electromotriz (FEM), mantenga el flujo de corriente en la misma dirección hacia el capacitor, lográndose así que el capacitor se cargue con la polaridad opuesta de su carga original. Este cambio repetitivo de la energía se le llama efecto volante (*flywheel*). Las pérdidas del circuito (principalmente en la resistencia de cd del embobinado del inductor) provocan que  $V_c(t)$  sea gradualmente más pequeña conforme este proceso se repite después del colapso completo del campo magnético. La forma de onda resultante, de la fig. 1.2.1b, es una onda senoidal amortiguada.

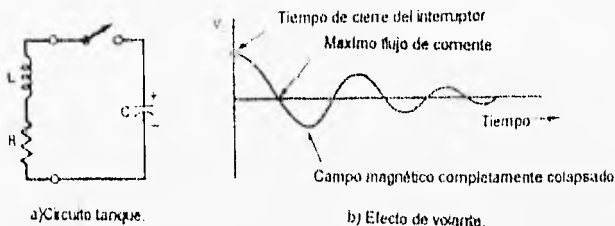


Fig. 1.2.1 Efecto de volante del circuito tanque.

De lo anterior se concluye que la energía del campo magnético se convierte en la energía del campo eléctrico del capacitor y viceversa. El proceso se repite a la frecuencia de resonancia,  $f_r$ , dada por la ecuación 1.2.1:

$$f_r = \frac{1}{2\pi\sqrt{LC}} \quad \text{ec. 1.2.1}$$

### 1.2.2 OSCILADOR HARTLEY.

El oscilador Hartley en forma simplificada se muestra en la fig.1.2.2. Los inductores  $L_1$  y  $L_2$  son parte de una bobina con toma central. La realimentación positiva se obtiene por efectos de la inductancia mutua entre las dos bobinas  $L_1$  y  $L_2$ , con  $L_1$  en la salida del circuito del transistor, y  $L_2$  entre la base y el emisor del circuito. Una porción de la señal del amplificador en el circuito del colector ( $L_1$ ) es realimentada a la base del circuito debido al acoplamiento inductivo de  $L_1$  a  $L_2$ . Como siempre en un circuito emisor-común, los voltajes de colector y base están  $180^\circ$  fuera de fase. Otra fase de  $180^\circ$  contraría entre estos dos voltajes ocurre debido a que son tomadas de las terminales opuestas de la toma central del inductor el cual está unido a la terminal común del transistor (el emisor). Así la realimentación en fase requerida se logra y la ganancia de realimentación es determinada por Q1. La frecuencia de oscilación es:

$$f_r = \frac{1}{2\pi\sqrt{(L_1+L_2)C}} \quad \text{ec. 1.2.2}$$

y es influenciada por los parámetros del transistor y el acoplamiento entre  $L_1$  y  $L_2$ .

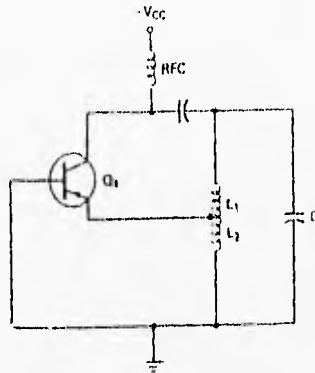


Fig. 1.2.2 Oscilador Hartley simplificado.

### 1.2.3 OSCILADOR COLPITTS.

La fig. 1.2.3 muestra un oscilador Colpitts, es muy similar al oscilador Hartley excepto que los elementos del circuito tanque tienen intercambiadas sus funciones. El capacitor es ahora el que se divide en dos y el inductor no tiene toma central. Los detalles de operación del circuito son idénticos al del circuito Hartley. La frecuencia de oscilación está determinada aproximadamente por la frecuencia de resonancia de  $L_1$  y  $C_1$  en serie con  $C_2$  del circuito tanque:

$$f_r = \frac{1}{2\pi\sqrt{(C_1 C_2 / (C_1 + C_2)) L_1}} \quad \text{ec. 1.2.3}$$

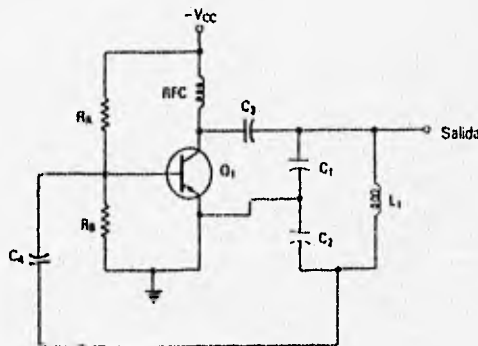


Fig. 1.2.3 Oscilador Colpitts.

#### 1.2.4 OSCILADOR CLAPP.

Una variación del oscilador Colpitts se muestra en la fig. 1.2.4. El oscilador Clapp tiene un capacitor  $C_3$  en serie con el inductor del circuito tanque. Si  $C_1$  y  $C_2$  son bastante grandes eliminarán las capacitancias inherentes de juntura del transistor, anulando las variaciones del transistor y los cambios de capacitancia de juntura con la temperatura. La frecuencia de oscilación se expresa en la ec. 1.2.4.

$$f_r = \frac{1}{2\pi\sqrt{L_1 C_3}} \quad \text{ec. 1.2.4}$$

este es un oscilador con mejor estabilidad en frecuencia que el Hartley y el Colpitts. Sin embargo el oscilador Clapp no tiene mucho ajuste de intervalo de frecuencia.

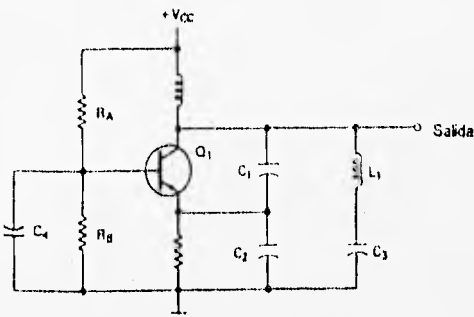
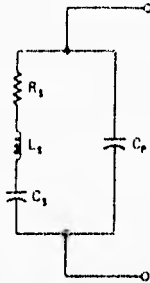


Fig. 1.2.4 Oscilador Clapp.

#### 1.2.5 OSCILADOR CONTROLADO POR CRISTAL.

Cuando es necesaria una mayor estabilidad de frecuencia, se utilizan osciladores controlados por cristal. Este es el que utiliza un cristal piezoeléctrico, al igual que un circuito LC utiliza un elemento inductivo. El Cristal (usualmente de cuarzo), tiene también una frecuencia de resonancia, pero se obtiene un mejor comportamiento cuando se acopla con una capacitancia externa.

El circuito equivalente de un cristal se muestra en la fig. 1.2.5. El cual representa al cristal como un circuito de resonancia en serie (con pérdidas resistivas) en paralelo con una capacitancia  $C_p$ . Las frecuencias de resonancia de estos circuitos (en serie y en paralelo) se encuentran muy cercanas por lo cual la impedancia del cristal varía dentro de un intervalo de frecuencia muy estrecho. Esto es equivalente a un circuito con un factor  $Q$  muy alto. Por esta razón y debido a la buenas características de estabilidad de tiempo y temperatura del cristal de cuarzo, los cristales son capaces de mantener una frecuencia a una variación de  $\pm 0.001\%$  en un gran intervalo de temperatura.



1.2.5 Circuito eléctrico equivalente de un cristal.

El circuito típico para un oscilador controlado por cristal se muestra en la fig 1.2.6 el cual se le conoce como circuito oscilador Pierce.

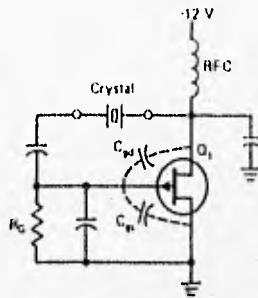


Fig. 1.2.6 Oscilador Pierce.

### 1.3 AMPLIFICADORES.

La necesidad de amplificar señales se presenta en virtud de que los transductores proporcionan señales que se les considera débiles o pequeñas, es decir del orden de los microvolts ( $\mu V$ ) o milivolts (mV), y con poca energía.

Una red de dos puertos operando como amplificador puede dar a la carga una potencia mayor que la que se obtiene de la fuente de señal. Los amplificadores necesitan fuentes de voltaje de dc (Vdc) para su operación, estas Vdc proporcionan la potencia extra que se entrega a la carga, así como cualquier potencia que pueda disiparse en la red misma de dos puertos.

Cuando una señal se amplifica, debe tenerse cuidado para que la información contenida en la señal no cambie, ni se introduzca nueva información. Así, cuando se alimenta una señal a un amplificador, es

deseable que la señal de salida del amplificador sea una réplica exacta de la de entrada, excepto, que tendrá una mayor amplitud. Es decir, las variaciones en la forma de la onda de salida deben de ser idénticas a las de la onda de entrada. Cualquier cambio en la forma de onda se considera distorsión.

Un amplificador que preserve los detalles de la forma de onda de la señal se caracteriza por la relación

$$V_o(t) = AV_i(t)$$

ec. 1.3.1

donde  $V_i$  y  $V_o$  son las señales de entrada y salida respectivamente,  $A$  es una constante que representa la magnitud de la amplificación y se le conoce como la ganancia del amplificador.

Hasta ahora, se ha supuesto que los amplificadores operan con señales de entrada muy pequeñas. Su propósito es hacer más grande la amplitud de la señal y por tanto se consideran como amplificadores de voltaje o de tensión.

Otro tipo de señal de amplificador es el de potencia. Este tipo de amplificador otorga poca o ninguna ganancia de voltaje, pero si una sustancial ganancia de corriente.

### 1.3.1 CLASIFICACION DE AMPLIFICADORES.

Hay muchas clases y subclases de amplificadores electrónicos. Por lo que es difícil proponer una clasificación completa. Cuando había solamente amplificadores con tubo de vacío, estos se clasificaban como amplificadores de voltaje o de potencia. Aún hoy, los amplificadores se clasifican en dos categorías.

- a.1) Una de las categorías incluye amplificadores de voltaje. Estos amplificadores se diseñan de tal forma que pueden levantar el nivel de voltaje o de corriente de señal. Desde luego que los voltajes y corrientes no son en general cantidades independientes, pero conviene usar un modelo de amplificación de voltaje o uno de corriente.
- a.2) La otra categoría general la constituyen los amplificadores de potencia, donde el rendimiento es una consideración importante y los niveles de la señal son generalmente altos.

Los amplificadores también se pueden clasificar como sintonizados y no sintonizados.

- b.1) Los sintonizados llevan circuitos resonantes sintonizados y generalmente operan en un intervalo de frecuencias, por encima de algunos centenares de kHz. Estos amplificadores son generalmente de banda angosta se diseñan con el propósito de selección de frecuencias específicas.
- b.2) Los amplificadores no sintonizados son en general de banda ancha.

Los amplificadores también se clasifican de acuerdo a la posición del punto de operación sobre sus curvas características.

- c.1) El amplificador clase A se polariza, de tal manera que la magnitud de la corriente de colector sea mayor que cero, todo el tiempo y que la señal pueda variar igualmente por encima y por debajo del punto de operación. Los amplificadores no sintonizados de salida única (no balanceados con respecto a tierra), generalmente operan como clase A.
- c.2) El amplificador clase B se polariza de tal manera que la corriente promedio de colector, es casi cero (corte). Las corrientes de colector aumentan en magnitud, a medida que la señal de entrada aumenta en magnitud.
- c.3) El amplificador clase AB tiene el punto de operación entre los de clase A y la clase B. La magnitud de la corriente de colector debe estar muy por encima del corte.
- c.4) El amplificador clase C tiene el punto de operación por debajo del corte. La magnitud de la corriente de colector es mayor que cero, durante menos de la mitad de un ciclo del voltaje de la señal de entrada. Los amplificadores clase C son generalmente amplificadores sintonizados.
- d) Los amplificadores también se clasifican, de acuerdo a su aplicación, algunos se diseñan para niveles de señal de entrada muy bajos y por tanto se llaman preamplificadores. El nombre dado a una categoría de amplificadores, generalmente indica el servicio a que se destinan, por ejemplo, amplificador para línea, amplificador de separación, amplificador monitor, amplificador de distribución, amplificador de radio frecuencia (RF), amplificador de frecuencia intermedia (IF) etc.

Un amplificador de RF es un amplificador sintonizado de alta ganancia y bajo ruido y es la primera etapa en un sistema de recepción. El propósito principal del amplificador de RF es lograr la selectividad, la amplificación, y la sensibilidad de una señal.

El término de RF simplemente significa que la frecuencia es bastante grande para ser eficientemente radiada por una antena y propagarse a través del espacio en forma de una onda electromagnética. La RF para la banda de emisión en AM se encuentra entre 535 y 1605 kHz, para la banda de FM entre 88 y 108 MHz, y para la radiación de microondas a partir de 1 GHz. Una frecuencia de IF común en receptores para la banda de emisión en FM, es 10.7 MHz y para los receptores de AM es de 455 kHz.

La RF es simplemente, la frecuencia radiada o recibida, y la IF es una frecuencia intermedia dentro de un transmisor o un receptor. Por lo que, muchas de las consideraciones para los amplificadores de RF también son aplicados a los amplificadores de IF tales como, el filtrado y el acoplamiento de impedancias.

La figura 1.3.1 muestra un diagrama esquemático para un amplificador bipolar de RF. Los elementos Ca, Cb, Cc, y Ll forman el circuito de acoplamiento de la antena. El transistor Q1 es de polarización de clase A para reducir la distorsión. El circuito del colector es un transformador



acoplado al mezclador a través de  $T_1$ , el cual está sintonizado a la frecuencia de operación para lograr mayor selectividad. Los capacitores  $C_x$  y  $C_y$  se utilizan para desviación o derivación paralela (de *bypass*). Sus símbolos indican que son capacitores de alimentación de paso especialmente contruidos. Estos capacitores ofrecen menos inductancia, lo cual previene la radiación de sus terminales,  $C_n$  es un capacitor de neutralización. Una parte de la señal del colector es realimentada a la base del circuito para neutralizar la señal realimentada a través de las terminales de capacitancia entre colector y base del transistor para prevenir oscilaciones.  $C_f$ , conjuntamente con  $C_n$ , forman un divisor de voltaje ca para la señal de realimentación. Esta configuración de neutralización es llamada neutralización de tierra-fuera.

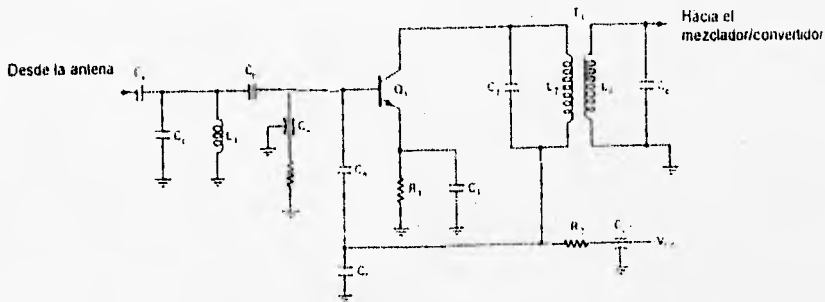


Fig 1 3 1 Amplificador bipolar de RF

#### 1.4 MEZCLADORES.

Para lograr la mezcla de dos señales de diferente frecuencia con un amplificador, se requiere que el amplificador opere en la región no lineal de las curvas características.

Los mezcladores reciben las señales de RF y del oscilador local *LO* (*Local Oscillator*) y amplifica la diferencia de estas señales, esta diferencia pasa a través del amplificador de frecuencia intermedia *IF*. La señal del *LO* es parte integral del mezclador. Por eso la única señal externa recibida es la de RF.

Los circuitos *LO* comúnmente utilizados en los mezcladores son los del tipo Hartley, Colpitts, y variaciones de cada uno. El oscilador Hartley se utiliza en la banda baja de RF, y el Colpitts en ultra alta frecuencia (*UHF*).

Con el avance tecnológico de los circuitos integrados (*IC's*), actualmente se cuenta con componentes para alta frecuencia que son el equivalente de transistores individuales acoplados juntos a un tercer transistor, esta técnica provee un aislamiento completo entre la entrada de RF y el circuito *LO*.

#### 1.5 FILTROS.

Considerando que un filtro tiene el propósito de dejar pasar señales que tienen un espectro de frecuencias específico y suprime señales cuyo espectro de frecuencias difieren de éste en especial, se puede decir que los filtros son circuitos selectivos de frecuencias. Aunque se usan en casi todos los sistemas electrónicos su uso más extensivo se encuentra en los sistemas de comunicaciones.

De acuerdo a las bandas de frecuencias que se dejan pasar o suprimen, existen cuatro tipos de filtros: paso bajas, paso altas, paso banda, y supresor de banda.

**-FILTRO PASO BAJAS.** El filtro paso bajas ideal, es aquel que proporciona una salida constante a partir de  $\omega$  hasta una frecuencia de corte  $f_c$  y después de esta frecuencia no deja pasar ninguna señal.

En la fig. 1.5.1a se muestra la respuesta en frecuencia de un filtro paso bajas.

**-FILTRO PASO ALTAS.** El filtro paso altas ideal, es aquel que solo permite el paso de señales arriba de la frecuencia de corte  $f_c$  y anula todas las señales que estén por debajo de esta frecuencia.

En la fig. 1.5.1b se muestra la respuesta en frecuencia de un filtro paso altas.

**-FILTRO PASO BANDA.** Cuando el circuito del filtro pasa señales que se encuentran arriba de una frecuencia de corte  $f_1$  y por debajo de una segunda frecuencia de corte  $f_2$  es un filtro paso banda ideal.

En la fig. 1.5.1c se muestra la respuesta en frecuencia de un filtro paso bandas.

**-FILTRO SUPRESOR DE BANDA.** Cuando el circuito del filtro deja pasar señales que se encuentran debajo de una frecuencia de corte  $f_1$  y por arriba de una segunda frecuencia de corte  $f_2$  es un filtro supresor de banda ideal.

En la fig. 1.5.1d se muestra la respuesta en frecuencia de un filtro supresor de banda.

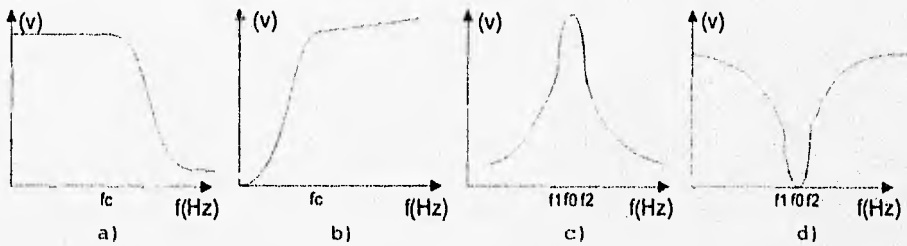


Fig. 1.5.1 Respuesta en frecuencia de un filtro: a) Paso baja, b) Paso altas, c) Paso bandas y d) Supresor de banda.

### 1.5.1 ANALISIS DE TIPOS DE FILTROS.

Los filtros consisten primordialmente de reactancias, las cuales pueden adoptar las formas físicas de bobinas, capacitores, cristales y resonadores mecánicos. Los filtros que utilizan amplificadores en combinación con resistores y capacitores se denominan filtros activos.

**-FILTROS LC.** Estos filtros constan de inductores y capacitores. Debido a las capacitancias e inductancias parásitas de sus elementos, su aplicación en bandas de ultra alta frecuencia (UHF) no es práctica. Los filtros de frecuencias bajas (LF) usan valores altos de inductancia y capacitancia que a su vez exigen componentes de tamaño grande. Por lo que se considera que la gama óptima de frecuencias de operación de los filtros LC para evitar los inconvenientes anteriores, es aproximadamente de 100 Hz a 300 MHz.

**-FILTROS ACTIVOS.** Como se mencionó este tipo de filtros emplean amplificadores operacionales, resistores y capacitores, y su gama de frecuencias de operación aproximadamente es de frecuencias bajas a 500kHz. Estos filtros pueden ser diseñados para operar a frecuencias muy bajas lográndose resultados satisfactorios.

Se pueden obtener valores del factor de calidad (Q) de unos cuantos centenares en la gama inferior del espectro de frecuencias de operación, donde los amplificadores tienen una ganancia muy alta de lazo abierto. La reducción de la ganancia de lazo abierto limita los Q obtenibles a las frecuencias más altas.

El diseño de filtros activos, permite establecer una impedancia deseada de entrada y salida, independiente de la frecuencia, además de que se dispone de ganancia de voltaje.

**-FILTROS DE CRISTAL.** Los resonadores de cristal de cuarzo cuyo circuito equivalente es el que se muestra en la fig. 1.2.5 (del tema de oscilador controlado por cristal) tienen un factor Q de hasta 1 000 000, por lo que se consideran elementos de filtros casi perfectos. Los filtros de cristal tienen una alta estabilidad, ya que los parámetros eléctricos del cuarzo permanecen esencialmente constantes con el tiempo y la temperatura.

Debido a la configuración del circuito equivalente, a las limitaciones de los valores de sus elementos y a factores económicos, se considera que los cristales son convenientes como elementos de filtros sólo cuando se requieren valores muy altos de Q y una alta estabilidad en filtros paso banda con ancho de banda muy estrecho.

**-FILTROS MECANICOS.** Este tipo de filtro, recibe una señal eléctrica, la convierte en vibraciones mecánicas con un transductor, aplica esas vibraciones a una serie de discos interconectados y vuelve a convertir las vibraciones resultantes en señales eléctricas de salida. Mediante el diseño apropiado de esos discos metálicos se pueden obtener resonancias mecánicas de un factor Q alto, de modo que cada disco sea el equivalente mecánico de un circuito resonante eléctrico en paralelo. Puesto que esos discos se acoplan mecánicamente, la señal de entrada se ve afectada por la respuesta de cada disco al pasar entre transductores de entrada y salida.

Los filtros mecánicos son más apropiados para filtros pasa banda de ancho de banda estrecho, en la gama de frecuencias de 50 a 500 kHz. Se pueden obtener valores de Q de paso bandas de hasta 1000, con una buena estabilidad de frecuencia.

Uno de los inconvenientes principales de este tipo de filtros es la elevada pérdida por inserción. Esto se debe principalmente, a la ineficiencia de los transductores de entrada y salida.

La fig. 1.5.2 muestra las gamas de frecuencias operacionales para los diferentes tipos de filtros existentes.

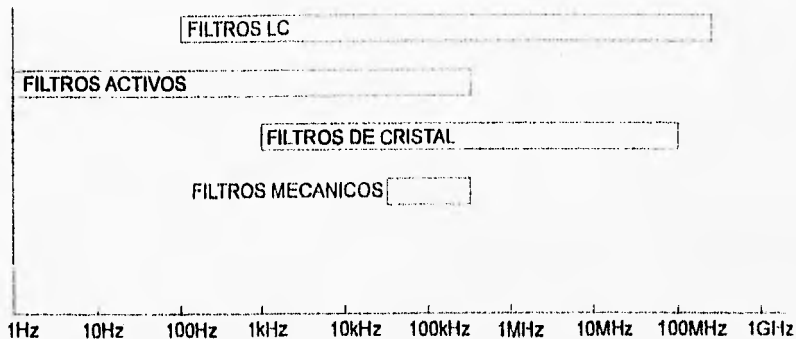


Fig. 1.5.2 Gamas de frecuencias de filtros.

#### 1.6. ARQUITECTURA DE UNA COMPUTADORA PERSONAL.

Una computadora personal *PC* (*Personal computer*) es un sistema digital, que ejecuta una secuencia de operaciones, sobre datos que se expresan en forma binaria.

Una computadora puede ser un conjunto de elementos individuales que tienen relación entre ellos para lograr un objetivo común. Una *PC* se divide en dos subsistemas llamados *hardware* y *software*.

El *hardware* es la parte eléctrica, electrónica y mecánica del sistema; es decir, involucra todos los componentes físicos de la *PC*.

El *hardware* se divide en: unidad de procesamiento central, memorias y dispositivos de entrada/salida. fig.1.6.1.

El *hardware* requiere de otro subsistema para funcionar, es decir, del *software*, que tiene los elementos necesarios para el control del sistema, es decir: programa de aplicación, algoritmo y procedimientos para solucionar problemas. El *software* es el término genérico de todos los programas de un sistema de procesamiento de datos.

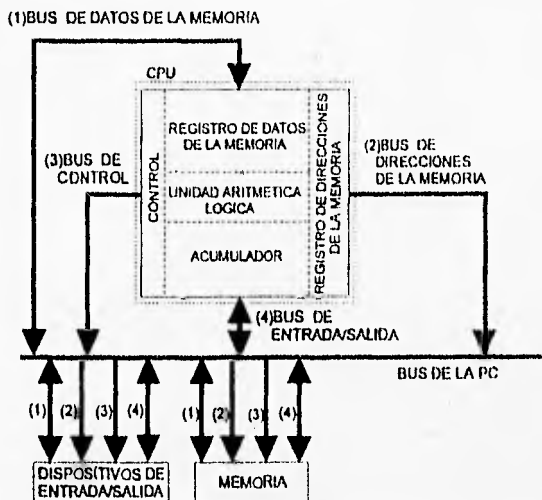


FIG. 1.6.1 Organización básica del hardware de una PC.

### 1.6.1 UNIDAD CENTRAL DE PROCESAMIENTO (CPU).

La Unidad Central de Procesamiento CPU (Central Processing Unit) de la fig. 1.6.1 es el cerebro de la computadora. Ejecuta los programas almacenados en la memoria principal a través de las secuencias de operaciones básicas, con la búsqueda (*fetching*) de sus instrucciones, examinándolas y ejecutándolas. La CPU tiene la configuración de la fig. 1.6.1.

### 1.6.2 UNIDAD DE CONTROL

Esta unidad tiene el papel supervisor para toda la máquina, adquiere las instrucciones de la memoria en el orden apropiado, interpreta las instrucciones y hace que los componentes apropiados de la máquina efectúen las operaciones especificadas por las instrucciones. El resultado de las operaciones aritmético lógicas en la unidad de procesamiento ALU, puede seleccionar diferentes alternativas en las instrucciones. Esta unidad es la parte más importante del *hardware* de la computadora, ya que interpreta los comandos del usuario dirigidos al sistema de programas.

La unidad de control ejecuta dos tareas:

- Controla los eventos en el procesador central, como la transferencia de datos de una localidad de memoria a otra, o de una localidad de memoria a la unidad aritmético-lógica.

- Controla el flujo de datos entre el procesador central y los dispositivos periféricos. Para esto usa las líneas de dirección y control.

La unidad de control coordina todas las unidades funcionales de la PC, con una secuencia lógica en un tiempo correcto. Un reloj central controlado genera los pulsos de reloj. El resto de los circuitos están sincronizados con estos pulsos. La unidad de control recibe las instrucciones del programa almacenándolas en la memoria principal, después las decodifica y dirige al resto de unidades funcionales para ejecutarlas.

### 1.6.3 UNIDAD ARITMETICA-LOGICA (ALU)

La Unidad Aritmética Lógica ALU (*Arithmetic-Logical Unit*) ejecuta operaciones para cálculo de datos numéricos, tales como adición, resta, etc., en seguida almacena el resultado en un registro de salida y de allí regresa a la memoria si así se desea. La unidad aritmética-lógica también toma decisiones lógicas, es decir, puede comparar un par de datos para determinar si un dato es mayor que, menor que o igual que otro.

### 1.6.4 REGISTROS.

La CPU contiene una memoria pequeña de alta velocidad usada para almacenar resultados temporales y cierta información de control. Esta memoria consiste de un número de registros, cada uno de ellos con una función especial.

Los registros más importantes son:

**ACUMULADOR.** Es el registro principal de trabajo del microprocesador. En muchos microcontroladores los resultados de las operaciones aritméticas o lógicas realizadas en la ALU se transfieren y almacenan en el acumulador. El acumulador se considera un registro de propósito general que almacena uno de los operandos que usa la ALU, al realizar operaciones aritméticas o lógicas. El acumulador puede ser registro fuente o registro destino.

**CONTADOR DE PROGRAMA PC (*Program Counter*).** En un programa las instrucciones están almacenadas en posiciones sucesivas de la memoria de programa, que puede consistir de varios circuitos integrados de memoria ROM (*Read Only Memory*). A cada posición de memoria se le asigna un número o código único, llamado dirección. Para ejecutar el programa en la secuencia correcta, el CPU debe conocer en qué posición de memoria buscar la próxima instrucción. El *program counter*, es el registro que apunta a la siguiente instrucción que va a ser buscada, y por esta razón al *program counter* se le considera como el registro más importante de la CPU.

**REGISTRO DE DIRECCIONES.** Es un dispositivo de almacenamiento temporal, que mantiene la dirección de lectura o escritura de datos en la memoria de datos. En algunos microprocesadores más sofisticados, el registro de direcciones es programable y el programa puede modificar su contenido mediante las instrucciones adecuadas.

**REGISTRO DE INSTRUCCIONES (IR).** La búsqueda de instrucciones de la memoria de programa requiere dos operaciones independientes. Primero, la CPU transmite la dirección de la instrucción del contador del programa a la memoria. La memoria entonces transmite el contenido de la posición de memoria direccionada a la CPU, donde se almacena temporalmente en un registro especial llamado registro de instrucción.

**MEMORIA.** En una computadora el sistema de memoria se usa principalmente para dos propósitos:

- Para disponer de un sistema de almacenamiento de datos u operandos.
- Para disponer de un medio para almacenar el programa (un grupo de comandos e instrucciones)

Existen dos tipos de memorias en la computadora: memoria principal y memoria secundaria.

**MEMORIA PRINCIPAL.** Es la parte de la computadora donde se almacenan programas y datos.

La porción de memoria destinada al almacenamiento de operandos se llama memoria de datos. Esta memoria almacena los datos que la computadora usa durante la ejecución del programa. La memoria de datos (es del tipo) memoria de acceso aleatorio *RAM (Random Access Memory)*.

La parte de la memoria que almacena los comandos o instrucciones se llama memoria de programa. Cada instrucción de esta memoria es suministrada a la computadora en una cierta secuencia preestablecida. La computadora decodifica cada instrucción e inicia el proceso específico ordenado por ésta. La memoria de programa es del tipo sólo lectura (*ROM*).

**MEMORIA SECUNDARIA.** Ya que cada palabra en la memoria principal puede ser directamente accesada en un tiempo muy corto, la memoria principal tiene un costo más alto que la memoria secundaria. En consecuencia, muchas computadoras tienen memorias alternas conocidas como memorias secundarias, que son más lentas, baratas y de mayor capacidad. La memoria secundaria se usa para mantener datos en un lugar lejano con respecto a la memoria principal. En seguida se describen algunos tipos de memoria secundaria más utilizados:

**CINTA MAGNETICA.** Las cintas magnéticas fueron de los primeros tipos de memoria secundaria. Un manejador de cinta de computadora es similar a una grabadora casera, con un cassette de 2400 pies de largo. Se graba la información haciendo variar la corriente en la cabeza de escritura.

Los dispositivos de cinta magnética son apropiados cuando se van a acceder datos secuencialmente.



DISCOS MAGNETICOS. Es una pieza de metal circular de alrededor de 5 a 10 pulgadas de diámetro, la cual tiene una cubierta magnetizable puesta directamente en el proceso de manufactura, generalmente en ambos lados. La información se graba en círculos concéntricos, llamados *tracks* o pistas.

Los discos tienen típicamente entre 40 y algunos cientos de pistas por superficie. Cada manejador de disco (*drive*) tiene una cabeza movable que se desplaza del centro hacia afuera. La cabeza tiene la anchura exacta para leer o escribir información solamente en una pista.

Las pistas se dividen en sectores, normalmente entre 10 y 100 sectores por pista. Un sector consiste de un cierto número de *bytes*, usualmente 512.

Casi todas las computadoras usan discos múltiples para su registro principal de datos. Estos se llaman discos duros (*hard disks*). El tipo más común es el *winchester disk*. Su capacidad es del orden de 100 *mega-bytes* en computadoras personales y alrededor de 10 *giga-bytes* en grandes *mainframes*.

DISCOS FLOPPY. Con el advenimiento de *PC's*, la solución a la distribución del *software* se hace con los *diskettes* o discos flexibles (*floppy*), un pequeño medio removible, llamado así por que es físicamente flexible. Este fue inventado por IBM (*International Buisness Machines*) para grabar información de mantenimiento para *mainframes*, pero fue rápidamente adquirido por los fabricantes de computadoras personales para distribuir la venta de *software*.

En la actualidad se usan dos tamaños, 5.25 y 3.5 pulgadas. Cada una con versión en alta y baja densidad. Los *diskettes* de 3.5 pulgadas vienen en una cubierta rígida para protección, así que no son realmente flexibles (*floppy*). Los discos de 3.5 pulgadas registran más datos y están mejor protegidos. Por lo que están reemplazando a los de 5.25 pulgadas.

DISCOS OPTICOS. En años recientes se empezó a comercializar los discos ópticos. A diferencia de los magnéticos tienen mayor densidad de grabación que los discos magnéticos convencionales. Los discos ópticos sirvieron para grabar programas de televisión, pero además se usan para grabar datos.

Estos discos se basan en la misma tecnología de los *Compact Disk* y se llaman *CD ROMs* (*Compact Disk Read Only Memory*). Los *CD ROM's* se leen con dispositivos similares a los reproductores de *CD* de audio.

El *CD ROM* se imprime mejor que el grabado de los discos *floppy*, éstos pueden producirse en masa con una máquina automatizada a bajo costo. El plástico que cubre el impreso de aluminio de los *CD ROM's* no siempre es perfecto, lo que tiene como consecuencias algunos errores en la información digital.

La información en un *CD ROM* se escribe como una espiral continua simple, a diferencia de los discos magnéticos, con sus cilindros discretos y sus pistas. Cada *CD ROM* contiene 270,000 bloques de datos, para una capacidad total de 553 *megabytes*.

La desventaja de este enorme potencial, es que en los *CD ROM's*, hasta hoy no se puede borrar, lo cual limita su uso como dispositivo de registro de una computadora. El disco *WORM (Write Once Read Memory)* permite escribir información en los discos ópticos, sólo una vez.

#### 1.6.5 SISTEMA OPERATIVO.

El sistema operativo es un programa que se encarga de administrar los recursos de la computadora.

Dentro de las ventajas del uso de los sistemas operativos se tienen las siguientes:

- Permite aumentar la velocidad de proceso, ejecutando programas independientes en forma consecutiva y sin interrupción.
- Incluye compiladores y ensambladores que facilitan la programación.
- La participación del operador se reduce al mínimo.
- Incluye programas de definición para ayudar al programador a descubrir y corregir errores de programación.
- Permite reducir el tiempo ocioso del *CPU* al atender por separado la ejecución de trabajos en el *CPU* y la ejecución de entradas y salidas.

El sistema operativo tiene el control de todos los programas. Esta es la tarea más importante del sistema.

#### 1.6.6 DISPOSITIVOS DE ENTRADA/SALIDA (I/O).

Es el medio que usa la computadora para transferir información.

No todas las entradas o salidas van hacia el operador, ya que si conectamos la computadora como un operador, la información puede llegar de otros medios, como son los sensores, o llegar hacia ellos.

Los dispositivos de entrada/salida más comunes en las computadoras son: el teclado, el *mouse*, la impresora y el monitor, aunque hay muchos más dispositivos de *I/O* disponibles hoy en día.

El Teclado es un conjunto ordenado de teclas que al pulsarse realizan la comunicación del usuario con la máquina. Existe a la venta una gran variedad de ellos.

El Monitor es un tubo de rayos catódicos *CRT*, (*cathode ray tube*) y su fuente de poder. El *CRT* tiene un cañón de electrones que son dirigidos

hacia una pantalla fosforescente (para las pantallas de color se tienen tres cañones de electrones, una para cada color: rojo, verde y azul).

El Mouse (ratón) es una pequeña caja de plástico que se coloca sobre la mesa de la computadora. Cuando este se mueve sobre la mesa, el cursor también se mueve en la pantalla. El ratón tiene uno, dos o tres botones para seleccionar fácilmente las opciones del menú.

Las computadoras personales (PC's) tienen una estructura física sencilla. Muchas de estas máquinas tienen una caja con una tarjeta de circuito impreso en la parte superior, llamada "tarjeta madre" (Mother Board). Esta contiene el CI de la CPU, algo de memoria y varios CI's de soporte. También contiene un bus y ranuras en las que se conecta memoria adicional y tarjetas de I/O.

La estructura lógica de una PC es mostrada en la fig. 1.6.2. Muchas PCs tienen sólo un bus para conectar la CPU, la memoria y los dispositivos de I/O. Cada dispositivo de I/O, consiste de dos partes, una que contiene la parte electrónica, llamada el "controlador", y otra que contiene el dispositivo de I/O.

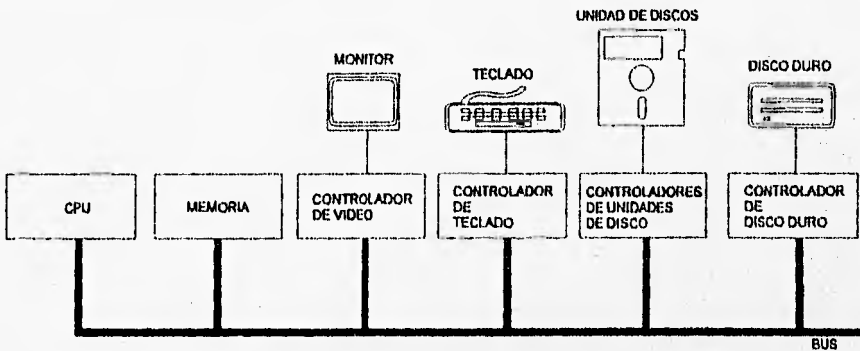


Fig. 1.6.2 Estructura lógica de una computadora personal.

Los controladores están conectados generalmente en el bus de la PC excepto para aquellos controladores que no son opcionales (como el teclado y el monitor), los cuales están en la tarjeta madre.

El trabajo de un controlador es manejar su dispositivo de I/O y manejar el bus de acceso hacia él. Cuando un programa quiere datos del disco, manda un comando al controlador del disco, el cual inicia una búsqueda. Cuando se localizan la pista y el sector, el drive comienza a enviar los datos como un flujo de bits en serie. Cuando el controlador lee o escribe un bloque de datos a otra memoria sin la intervención de la CPU, se dice que está desempeñando un acceso directo a memoria (DMA Direct Memory Access).

Si la CPU y algún controlador quieren usar el bus al mismo tiempo, existe un circuito que decide quién usa el bus primero y quién después. En general, los dispositivos de I/O tienen preferencia sobre la CPU, ya que los dispositivos de movimiento como los discos no se pueden parar, y forzarlos a esperar podría resultar en una pérdida de datos.

La computadora tiene como funciones principales los siguientes puntos:

- Almacenamiento de los datos.
- Procesamiento para obtener información adicional.
- Despliegue de los datos en la pantalla.

### 1.6.7 SLOT DE LA COMPUTADORA PERSONAL.

El bus de la PC/XT consta de líneas para la fuente de poder y líneas de acceso directo a memoria. El bus contiene 8 líneas para datos, 20 líneas de direcciones, 6 líneas de interrupciones, líneas de control para memoria y los puertos e/s (entrada/salida), relojes, 3 canales de DMA, etc. Estas funciones se encuentran en un conector de 62 contactos.

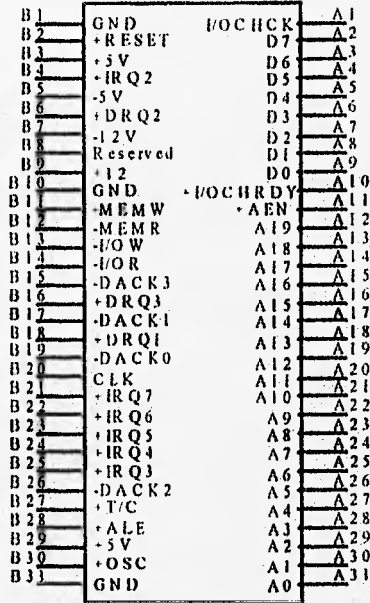


Fig. 1.6.3 Slot de la PC/XT a 8 bits.

Se tiene disponible una señal de "ready" para la operación de tarjetas de expansión lentas, ya sea de e/s o de memoria. Si esta señal de "ready" no se activa, todos los ciclos de lectura o escritura a memoria generados por el procesador, duran 4 pulsos de reloj del sistema es decir 840 ns/byte.

Los ciclos de lectura o escritura a los puertos de e/s generados por el procesador, requieren 5 pulsos de reloj para completar un tiempo de 1.05 us por byte. Los ciclos de refresco ocurren uno cada 72 pulsos de reloj (aproximadamente 15 us) y requieren de 4 ciclos de reloj, es decir ocupan aproximadamente el 7% del ancho de banda del bus.

En el bus se encuentran 512 puertos de entrada/salida para el uso de las tarjetas de expansión, a pesar de que el 8088 podría manejar hasta 64 k. Existe una línea llamada "channel check" que reporta condiciones de error al procesador. Al activar esta línea se genera una señal NMI (interrupción no enmascarable) para el 8088. Las tarjetas para expansión de memoria generalmente usan esta línea para reportar errores de paridad.

El slot-bus está diseñado para dar suficiente alimentación a todas las ranuras de expansión, por lo que soporta hasta 2 cargas del tipo LS (low-power shottky) por ranura máximo. Por lo general las tarjetas de buena calidad sólo tienen una carga LS.

La fig. 1.6.3 muestra la distribución de los contactos del bus de la PC.

A continuación en la tabla 1.6.1 de la siguiente página, se da una explicación más detallada de cada una de las señales del bus:

Tabla 1.6.1

Señal	E/S	Utilización
OSC	S	Oscilador: Reloj de alta velocidad con un periodo de 70 ns. Tiene un ciclo de trabajo del 50 %.
CLK	S	Reloj del sistema: Opera a la tercera parte de la frecuencia del oscilador y tiene un periodo de 210 ns. (4.77 Mhz) Tiene un ciclo de trabajo del 33%.
RESET	S	Esta línea se usa para inicializar la lógica del sistema al encender la máquina o durante una baja en el voltaje de alimentación. Está sincronizada con el flanco de bajada del reloj y es activo alto.
A0-A19	S	Bits 0 al 19 de las direcciones: Estas líneas se usan para manejar la memoria y los dispositivos de E/S del sistema.
D0-D7	E/S	Bits de Datos 0 de 7: Es el bus de datos.
ALE	S	Habilita Direcciones. Esta línea es generada por el controlador del bus (8288) y se usa para indicar cuando existen direcciones válidas dadas por el procesador. Está disponible en el bus de la PC, y se usa en general en conjunto con la señal AEN. Activo alto.
I/O CHK	E	Revisión del canal: Da la información sobre errores en las tarjetas del bus de la PC y se usa generalmente errores de paridad. Activo bajo.
I/O RDY	E	Canal Listo: Esta línea, activo alto, se coloca en cero por alguna tarjeta cuando esta necesita tener un ciclo de lectura o escritura más largo, ya sea en memoria o en E/S, No puede tener abajo más de 10 ciclos de reloj.
IRQ2.. IRQ7	E	Pedidos de Interrupción 2 a 7: Se usan para señalar al procesador que un dispositivo de E/S requiere atención. Están dispuestas por prioridades, teniendo IRQ2 la máxima prioridad e IRQ7 la mínima. Una interrupción se genera levantando una línea de IRQ (de 0 a 1) y manteniéndola en alto hasta recibir el aviso de que se ha reconocido la interrupción.
IOR	S	Lectura del Dispositivo de E/S: Esta línea se usa para indicar al dispositivo, que debe poner un dato en el bus del procesador. Puede ser generada por el procesador o por el controlador de DMA. Esta señal es activo bajo.
IOW	S	Escritura a dispositivo de E/S: Esta línea se usa para indicar al dispositivo que debe leer el dato que el procesador colocó en el bus. Puede ser generada por el procesador o por el controlador DMA. Activo bajo.
MEMR	S	Lectura de Memoria: Indica a la sección de memoria que debe poner un dato en el bus. Puede ser generada por el procesador o por el controlador DMA. Activo bajo.
MEMW	S	Escritura a Memoria: Indica a la sección de memoria que debe leer el dato que el procesador ha puesto en el bus. Puede ser generada por el procesador o por el controlador de DMA. Activo bajo.
DRQ1.. DRQ2	E	Pedido de DMA 1 a 3: Son pedidos asíncronos usados por dispositivos periféricos para lograr acceso directo a memoria.
DACK0 DACK3	S	Reconocimiento de DMA: Estas líneas se usan para avisar que la petición a DMA ha sido concedida (para drq1-drq3) y para el refresco de la memoria dinámica del sistema (DACK0). Son de activo bajo.
AEN	S	Habilita Dirección: Esta línea se usa para desconectar al procesador y otros dispositivos del bus de la PC para permitir el acceso directo a memoria. Cuando esta línea se coloca en activo alto, quiere decir que el controlador de DMA tiene el control del bus de datos, y de direcciones de las líneas de escritura y lectura tanto a memoria como a E/S.
T/C	S	Cuenta Terminal: Esta línea provee un pulso cuando el DMA de alguno de los canales debe terminar. Es activo alto.

## 1.7. METODOS DE TRANSMISION.

Los métodos de transmisión más comunes son:

*Simplex*, *Half-duplex* y *Full-duplex* a continuación explicaremos cada uno de ellos:

**SIMPLEX.** Un enlace en modo *simplex* lleva información en una sola dirección, esto es, que sólo transmite o sólo recibe información. Un enlace en modo *simplex* es la elección apropiada cuando sólo se necesita enviar una información de una terminal a otra y nunca se necesita recibir información.

Una de sus ventajas es su bajo costo de instalación y es simple en el *software*. Sin embargo, como no puede enviar información en las dos direcciones, sus aplicaciones son limitadas.

**HALF-DUPLEX.** Para muchas aplicaciones es necesario comunicarse en ambas direcciones (transmitir y recibir). La forma menos cara de conseguir esto es con *half-duplex*, en la que la línea lleva la información en una dirección a la vez.

En un sistema *half-duplex*, las dos terminales de la línea pueden comunicar no sólo información sino también sus caracteres de información.

Suponiendo que la información viaja de una terminal A a una terminal B de izquierda a derecha. La terminal A es la que envía y la B es la que recibe. Cuando la terminal A termina de enviar sus datos, la terminal B reconoce la recepción de datos y entonces pide la línea. Cuando la terminal A termina de enviar y comienza a recibir, se tiene una conmutación de línea. Cuando la terminal B termina de transmitir, la terminal A puede pedir la línea y conmutará de nuevo.

Un sistema *half-duplex* requiere un circuito para realizar la conmutación de la línea. El tiempo que el sistema invierte cambiando la línea es tiempo perdido. Sin embargo, *half-duplex* es menos caro que el sistema *full-duplex* porque el primero necesita sólo dos alambres para efectuar la transmisión de datos.

**FULL-DUPLEX.** En un sistema *full-duplex* la información viaja en ambas direcciones (transmite y recibe) a la vez.

Puesto que los datos son transmitidos como señales analógicas, se debe considerar el ancho de banda. Cuando se transmite en una dirección a la vez con *half-duplex* o *simplex*, se puede usar la banda entera, o el rango de frecuencias que la línea tiene disponible. Si se trata de transmitir en ambas direcciones a la vez, cada dirección puede usar sólo la mitad de la banda. Si se usan cuatro alambres en lugar de dos, que son los usuales, entonces se le puede dar rapidez al sistema *full-duplex*.

### 1.7.1 TRANSMISION DE CARACTERES.

Los *bits* de un caracter pueden ser transmitidos usando transmisión en paralelo o transmisión serial.

**TRANSMISION EN PARALELO.** En la transmisión en paralelo un caracter completo (de 8 *bits*) es transmitido a la vez: cada *bit* tiene su propio cable por lo que cada *bit* está moviéndose paralelamente a los otros *bits*. Las posiciones de los *bits* en los cables no varían.

Sin embargo sólo podemos usar transmisión en paralelo sobre distancias menores a 1.52 m . Más allá de esta distancia, la interferencia por radio frecuencia se convierte en un problema y la señal de datos se vuelve muy débil.

Los periféricos de un sistema de cómputo, por ejemplo, una impresora frecuentemente usan una conexión en paralelo; en general, cuando la distancia entre los dispositivos es pequeña, la transmisión en paralelo es la más acertada.

**TRANSMISION SERIAL.** Con la transmisión serial, un caracter se mueve en un cable a la rapidez de un *bit* por unidad de tiempo. Se puede usar transmisión serial sobre distancias grandes o cortas, lo que hace el método más común para transmitir datos.

A continuación se hablará de cómo los datos seriales son transmitidos de un lugar a otro, utilizando el concepto de sincronización.

Primeramente el receptor debe saber cuando iniciar la búsqueda del primer *bit* en el primer caracter. Después de esto, si el reloj de las dos terminales está corriendo a la misma velocidad, el transmisor y el receptor permanecerán sincronizados.

Podemos especificar la velocidad de la transmisión de datos en *bauds* o *bits por segundo (bps)*. Estos términos están muy relacionados pero no son idénticos, aunque la mayoría de los especialistas en comunicación de datos los utilizan de manera indistinta.

Un *baud* es la unidad de velocidad de modulación, se refiere al lado analógico, y se define como el número de cambios de estado de la señal analógica en un segundo, un cambio de estado puede ser un cambio de amplitud, frecuencia, fase o una combinación de estas.

Un *bps* es una unidad de información se utiliza para expresar la velocidad de transmisión de la información, se refiere al lado digital y es la cantidad de *bits* transmitidos en un segundo.



### 1.7.2 SINCRONIZACION POR CARACTER.

Los sistemas seriales tienen dos alternativas de comunicación: Comunicación sincrónica y asíncrona.

**TRANSMISION SINCRONA.** En un bloque de transmisión sincrónica, los caracteres de datos son organizados en bloques de longitudes estándar (frecuentemente 256 caracteres) y son las unidades de transmisión. Cada bloque está formado por un carácter de inicio de texto (STX) y un de fin de texto (ETX). (fig. 1.7.1).



Fig. 1.7.1 Bloque de transmisión sincrónico.

En este método de transmisión tanto el dispositivo emisor como el receptor operan simultáneamente y se resincronizan después de transmitirse algunos miles de *bits* de señal de datos.

No se requieren *bits* de inicio/parada por cada carácter. La sincronización se establece y mantiene cuando no se están transmitiendo señales o justo antes de la transmisión de una señal de datos. Esta sincronización se establece enviando un grupo predeterminado de caracteres de sincronización entre los dispositivos transmisor y receptor.

Durante una sesión de transmisión, el receptor comenzará por ver dos o más caracteres de espacio (*SPC*) de aviso de comienzo de transmisión de un bloque de datos, o de separación entre bloques de datos. El carácter *SYN* deja al receptor sincronizado. Ahora el receptor sabe qué *bit* es el primer *bit* en el primer carácter, y puede leer los caracteres en el bloque que sigue. Después del carácter *SYN*, el receptor verá un carácter *STX*. Todo lo que sigue, hasta que el receptor vea un *ETX*, son datos. (fig. 1.7.2).



Fig. 1.7.2 Secuencia de transmisión de un bloque típico.

La transmisión sincrónica se utiliza cuando es necesario un gran volumen y alta velocidad en la transmisión de datos. Debido a que los datos son empaquetados en bloques, esta transmisión de datos puede ser muy eficiente. En el mercado hay disponible *software* de comunicación sincrónica para terminales inteligentes y microcomputadoras.

La transmisión sincrónica es más eficiente en el sentido de que hay menos *bits* de control respecto al número total de *bits* transmitidos. La

sincronización puede necesitar sólo de 16 a 32 bits, en tanto que la secuencia de bits para la señal de datos puede tener varios miles.

**TRANSMISION ASINCRONA.** Al modo de transmisión asincrónico se le conoce como transmisión de arranque y parada, porque el dispositivo transmisor puede transmitir un carácter en cualquier momento que sea conveniente y el dispositivo receptor lo acepta, por lo que es posible enviar caracteres a intervalos irregulares.

Para este tipo de transmisión, cada carácter debe permanecer independiente. El receptor debe sincronizarse a cada carácter independiente (fig. 1.7.3).



Fig. 1.7.3 Transmisión asincrónica de caracteres.

En una transmisión asincrónica, cada carácter está formado por bits de inicio y bits de parada (fig. 1.7.4). El bit de inicio le dice al receptor que un carácter está llegando y el bit de parada le dice al receptor que espere otro bit de inicio antes de muestrear. El receptor sabe cuantos bits deberían ir dentro de un carácter. Este toma ese número de bits y lo guarda en un espacio del carácter. La mayoría de los sistemas usan un bit de inicio y dos bits de parada. En la transmisión asincrónica se usa la verificación del carácter por paridad, el cual detecta errores en la información.



Fig. 1.7.4 Estructura de un carácter asincrónico.

### 1.7.3 PROTOCOLOS.

Para que dos entidades puedan comunicarse, deben "hablar el mismo lenguaje". Qué, cómo y cuándo se comunican estas entidades debe conformar un arreglo de convenciones mutuas. Un protocolo es un conjunto de reglas y procedimientos que proporcionan una técnica uniforme para gobernar el intercambio de información entre dos o más entidades.

En el caso de la comunicación de datos, estas reglas están diseñadas para resolver los problemas de operación en las siguientes áreas:

-Detección: Es determinar del grupo de ocho bits cuáles son caracteres, y lo más importante, qué grupos de caracteres constituyen mensajes.

-Control de errores: La detección de errores en la aceptación de mensajes y la solicitud para aceptar mensajes defectuosos, se efectúan mediante varias técnicas, tales como verificación longitudinal, vertical y redundancia cíclica.

-Control de frecuencia: Es la enumeración de los mensajes para eliminar aquellos que son repetidos.

-Transparencia: El usuario debe ser capaz de elegir el código o patrones de bits a usar en la transferencia de datos.

-Control de línea: Es la determinación de cual estación va a transmitir y cual(es) estación(es) va(n) a recibir la información correspondiente. En caso de línea *duplex* o multipunto.

-Casos especiales: Resuelve el problema de qué envía el transmisor cuando no hay un dato para transmitir.

-Control de tiempo muerto: Resuelve el problema de qué hacer cuando el flujo de transmisión cesa por completo.

-Control de inicio: El proceso de obtención de transmisiones iniciadas en un sistema de comunicación deficiente.

Los protocolos de enlaces de datos se clasifican en dos categorías: protocolos de bit orientado (*BOP*) y protocolos de byte controlado (*BCP*). Los protocolos de byte controlado usan caracteres de control específicos para informar al receptor cuando está mandando direcciones o datos y donde empiezan y terminan los bits de dato. El protocolo de bit orientado, depende de la posición de los bits dentro de un campo o un bloque.

## CAPITULO II

# HERRAMIENTAS DEL PROYECTO

En este capítulo se describirán las características principales de la familia del microcontrolador TMS370, y sus aplicaciones más comunes. También se cubrirá en este capítulo una introducción a los lenguajes de alto nivel, los compiladores y ensambladores, así como una clasificación de los lenguajes de alto nivel, explicando brevemente los lenguajes C y C++, con el objeto de conocer las herramientas de programación utilizadas en el proyecto.

## 2.1 LA FAMILIA DEL MICROCONTROLADOR TMS370.

La familia TMS370 es una serie de microcontroladores CMOS de circuitos integrados de muy grande escala VLSI (*Very Large Scale Integrated*), de 8-bits con una memoria EEPROM de almacenamiento y funciones de soporte periférico. Estos dispositivos ofrecen un desempeño superior en aplicaciones complejas y de control en tiempo real en ambientes de mucha demanda y cuentan con memorias ROM programables y EPROM. Además, se cuenta con una gran cantidad de opciones para elegir de acuerdo a las necesidades de diseño y capacidad económica.

### 2.1.1 APLICACIONES COMUNES.

La familia TMS370 de dispositivos es una elección ideal para aplicaciones como:

Modos automáticos.- En sistemas de control de acondicionamiento de ambiente, control de cruce, sistemas de entretenimiento, instrumentación, sistemas de navegación, control de máquinas, etc.

Computadoras.- En teclados, "control de interfaces periféricas", controladores de discos, terminales.

Industria.- Controladores de motores, controladores de termómetros, control de procesos, control de mediciones, instrumentación médica, sistemas de seguridad.

Telecomunicaciones.- Modems, teléfonos inteligentes, control de tarjetas telefónicas, telecopiadoras, tarjetas de crédito.

### 2.1.2 ELEMENTOS DE LA FAMILIA TMS370.

La familia TMS370 tiene las siguientes características:

-Software compatible entre los elementos de la familia.

-Tecnología CMOS EPROM que contiene EPROM para reprogramar, memoria programable de una oportunidad para programar OTP (*One Time Programmable*) para ser usada como prototipos y para volúmenes pequeños de producción, respectivamente.

- Tecnología *CMOS EEPROM* que contiene programación de *EEPROM* con una alimentación de 5 Volts.
- Tecnología *A/D* que convierte señales analógicas a digitales.
- Registros *RAM* estáticos que ofrecen numerosas opciones de memoria.
- Características de operación flexibles:
  - Reducción de potencia en estado de espera y en modos de interrupción.
  - Temperatura de operación de -40°C a 85°C.
  - Frecuencia de reloj de entrada de 2 MHz a 20 MHz.
  - Voltaje de operación = 5V ± 10 %
- Manejo de interrupción flexible para la versatilidad en el diseño:
  - Dos niveles de interrupción programables.
  - Programación de detección de límites superiores y/o inferiores.
- Características para la integridad del sistema que incrementa la flexibilidad durante la fase del desarrollo de *software* con:
  - Un oscilador de detección de falla.
  - Modo de protección privilegiado.
  - Contador *watchdog*.
- Puertos mapeados en memoria para direccionar fácilmente.
- Una librería modular para cambiar rápidamente a configuraciones diferentes.
- Catorce modos de direccionamiento que usan ocho formatos, incluyendo:
  - Aritmética de registro a registro.
  - Direccionamiento indirecto.
  - Salto y accesos directos e indexados.
- 250 mA de corriente típica estable a 25°C

### 2.1.3 COMPONENTES COMUNES EN LA ARQUITECTURA DE LA FAMILIA TMS370.

Además de los elementos listados anteriormente, los miembros de la familia TMS370 tienen los siguientes elementos de arquitectura, de acuerdo a la categoría del dispositivo.

**CPU.** La CPU de 8 bits de la TMS370 tiene un registro de estado, un registro de contador de programa (CP), y el *stack pointer* (SP). La CPU usa el archivo de registro como registros de trabajo, accedidos en el bus interno a un ciclo de bus. Los dispositivos TMS370Cx5x permiten una expansión de memoria externa a través de los puertos A, B, C, y D.

**AREA DE REGISTROS.** Estos registros se ubican al principio del mapa de memoria de la TMS370. Cuando este sector del mapa de memoria es usado como una *RAM* de propósito general las instrucciones de acceso a registro, permiten el acceso a cualquiera de los primeros 256 registros en un ciclo de *bus*. Esta área de registros también puede ser empleada como *stack*.

**MEMORIA RAM.** Los otros módulos de memoria *RAM* que no están en el archivo de registro se encuentran en otra área de memoria. La familia TMS370 accesa esta *RAM* en dos ciclos de reloj del sistema.

**DATA EEPROM.** Los módulos de *EEPROM* de datos proporcionan programabilidad y retención de datos en el modo apagado. Los módulos contienen 256 o 512 bytes de *EEPROM*. Esta memoria es útil para constantes y variables sin cambios continuos requeridos por el programa de aplicación. La *EEPROM* puede ser programada y borrada con programadores de *EEPROM* o con el mismo TMS370 bajo el control de programa.

**MEMORIA DE PROGRAMA.** La memoria de programa provee de alternativas para respaldar las necesidades de una aplicación. Los módulos de memoria de programa de acuerdo a la categoría del dispositivo contienen 2k, 4k, 8k, 16k, o 32k bytes de memoria. La memoria de programa de los, TMS370C6xx y TMS370C7xx es una *EPROM*. La *EPROM* puede ser programada, borrada y reprogramada para prototipos (con cubierta cerámica). Los dispositivos *EPROM* que no tienen ventana (con cubierta de plástico) son dispositivos para programar sólo una vez (*OTP*). En los TMS370C0xx y TMS370C3xx, el programa de memoria es una *ROM* con máscara programada por el fabricante.

**PUERTOS DE ENTRADA/SALIDA.** Los TMS370Cx1x tienen dos puertos: el puerto A y el puerto D. El puerto A es un puerto de 8 bits, mientras que el puerto D es de 5 bits. Ambos puertos pueden ser programados, bit por bit, para trabajar como entrada o salida digital.

Los TMS370Cx2x tienen cuatro puertos: puertos A, B, C, y D. Los puertos A y B son de 8 bits, el C es de 1 bit, el D es de 5 bits. Cada uno de estos puertos puede ser programado para trabajar como entrada o salida digital. Los TMS370Cx3x tienen dos puertos: el A y el D. El A es de 8 bits y el D es de 4 bits. Ambos pueden trabajar como entrada o salida digital.

Los TMS370Cx4x tienen tres puertos: el A, B, y D. El A es de 8 bits, el B de 3 bits, y el D es de 5 bits. También pueden funcionar como entrada o salida digital. Los TMS370Cx5x tienen cuatro puertos de 8 bits: el A, B, C, y (el puerto D de los dispositivos de 64 pines es de 6 bits). Estos puertos pueden ser configurados para trabajar como líneas de datos, de control, y de dirección para memorias externas.

**TIMER 1 Y TIMER 2.** Los timer 1 y 2 son de 16 bits que pueden ser configurados de las siguientes formas:

-Prescalador programable de 8 bits que determina las fuentes independientes de reloj para el timer de propósito general y el timer watchdog.

-Timer de evento de 16 bits, para llevar el contéo de transiciones.

-Acumulador de pulso de 16 bits, para medir el ancho de pulso a la entrada.

-Función de captura a la entrada de 16 bits, que guarda el valor del contador al existir una entrada externa.

-Dos registros de 16 bits de comparación que se disparan cuando el contador iguala los contenidos de un registro de comparación.

-Función de control de salida de mismo contenido de modulación por ancho de pulso PWM (Pulse Width Modulated).

El resultado de estas operaciones puede ocasionar interrupciones a la CPU, encender bits de bandera, resetear el contador timer, cambiar una línea entrada/salida, generar salidas PWM. Estos timers pueden tener una resolución de 200 ns a 20 MHz.

**TIMER WATCHDOG.** El timer watchdog ayuda a asegurar la integridad del sistema. Puede ser programado para generar un reset de hardware al acabar su tiempo. Esta función genera un monitoreo del software para evitar pérdidas del programa. Si no se necesita como watchdog, este también puede ser usado como timer de propósito general.

**TIMER PROGRAMABLE DE ADQUISICION Y CONTROL PACT** (Programmable Acquisition and Control). El módulo PACT es un módulo de tiempo programable que utiliza la memoria RAM para guardar sus comandos así como los valores del timer. Este módulo ofrece las siguientes opciones:

-Captura de entrada de 6 pines, de los cuales 4 pines pueden tener un preescalador.

-Un pin de entrada de captura puede manejar un contador de eventos de 8 bits.

-8 salidas para trabajar como timers manejables.

-Capacidad de timer de 20 bits.

-Interacción del contador de eventos y la actividad del timer.

-18 vectores de interrupción independientes para permitir un mejor servicio de eventos.

-Un watchdog con un periodo selectivo de tiempo-fuera.



-Una interfaz serial de comunicaciones *SCI* (*Serial Communications Interface*), que trabaja como un Receptor Transmisor Universal Asíncrono *UART* (*Universal Asynchronous Receiver Transmitter*) *full-duplex*.

Una vez configurado, el *PACT* ya no requiere de la supervisión del *CPU* excepto para servicios de interrupción.

La interfaz *SCI*. Es un módulo que ofrece las siguientes características:

- Se puede programar para trabajar en modo asíncrono (a 156 kbits/seg) o modo síncrono (a 2.5 Mbits/seg).
- Comunicación *full-duplex* con doble *buffer* para Tx y Rx.
- Formato programable con capacidad para revisión de errores.

El módulo *SCI* programa y controla los factores de tiempo, formato y protocolo de datos. La *CPU* no toma parte en la comunicación serial excepto para escribir datos transmitidos en los registros de la *SCI* y para leer datos recibidos de los registros de la *SCI* cuando está interrumpido.

**INTERFAZ PERIFERICA SERIAL *SPI*** (*Serial Peripheral Interface*). El módulo *SPI* facilita la comunicación entre las *CPU* y los dispositivos periféricos externos. Proporciona transmisión síncrona de datos a 2.5 Mbits/seg. Al igual que la *SCI*, es configurada por *software*. Después de eso, la *CPU* no toma parte en el tiempo, formato, y protocolo de los datos. También, al igual que la *SCI*, la *CPU* lee y escribe en la memoria mapeada de los registros para recibir y transmitir datos. Una interrupción de la *SPI* avisa a la *CPU* cuando los datos recibidos están listos.

**CONVERTIDOR ANALOGICO DIGITAL A/D** (*Analog to Digital*). El módulo del convertidor A/D de 8 bits desarrolla aproximaciones sucesivas. La fuente de referencia y el canal de entrada son selectivos. Las líneas de entrada que no se necesitan para la conversión A/D se pueden programar para ser líneas de entrada digitales.

#### 2.1.4 DIAGRAMA A BLOQUES DEL TMS370Cx2x.

La familia TMS370 se basa en una arquitectura de registro a registro, la cual permite el acceso a un archivo de registro (de 256 bytes) en un sólo ciclo de *bus*. El circuito integrado de memoria incluye la memoria de programa (ROM con máscara o EPROM), RAM estática, y EEPROM de datos.

Las funciones periféricas incluyen una interfaz de comunicación serial, una interfaz periférica serial, tres diferentes módulos *tímer* y 34 *pin*s de entrada/salida digital. El diagrama a bloques del TMS370Cx2x se muestra en la fig. 2.1.1.

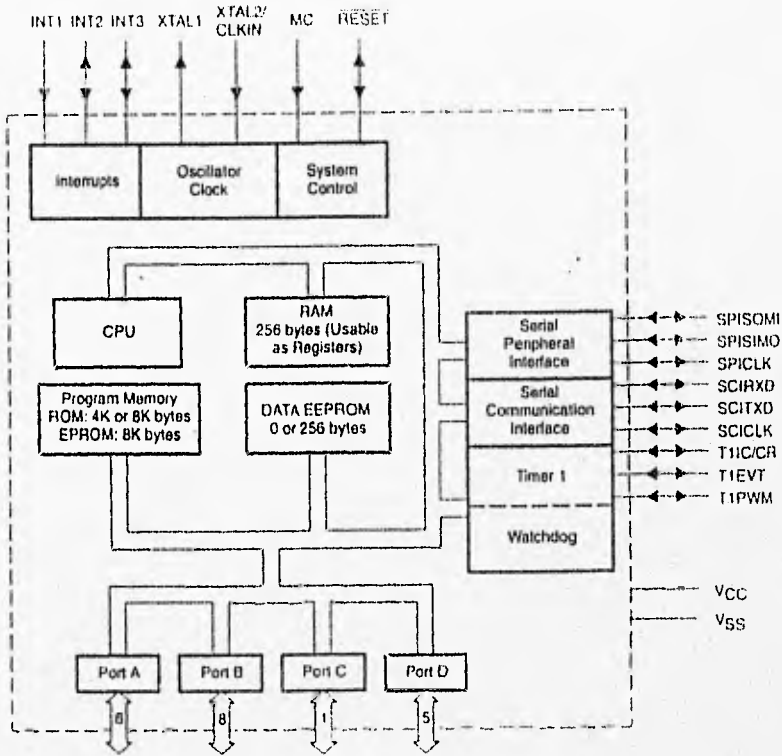


Fig. 2.1.1 Diagrama a bloques del TMS370Cx2x

## 2.2 LENGUAJES DE PROGRAMACION.

Los lenguajes de programación son utilizados para implementar algoritmos; esto es, ejecutar secuencias o pasos para la solución de problemas. Un lenguaje de programación tiene que definir la acción de las trayectorias dinámicas de los algoritmos. No solo debe dar un significado para especificar las operaciones básicas de una computadora, sino que también debe permitir al usuario conocer la secuencia a seguir de los pasos para resolver un problema particular.

### 2.2.1 LENGUAJES DE ALTO NIVEL.

Los lenguajes de programación de cómputo pueden ser clasificados en dos grandes grupos: lenguajes de bajo nivel y lenguajes de alto nivel. Los lenguajes de bajo nivel son parecidos al lenguaje de máquina y tienen una fuerte correspondencia entre las operaciones implementadas por el lenguaje y las operaciones del *hardware*. Los lenguajes de alto nivel, por otro lado, son parecidos a los lenguajes usados por el humano para expresar problemas y algoritmos. Cada instrucción en un lenguaje de alto nivel puede ser equivalente a muchas instrucciones en un lenguaje de bajo nivel.

Podemos decir en forma concreta, que un lenguaje de programación de alto nivel, es aquel que permite con una sola instrucción, hacer una serie de operaciones.

El intervalo de operaciones directamente implementado por el *hardware* de una computadora es usualmente muy limitado; por lo que, la codificación de algoritmos complejos usando solo estas operaciones es un problema grande. La codificación puede ser tediosa y se pueden cometer errores. Ese es el obstáculo a vencer de los lenguajes de programación, especialmente de los lenguajes de alto nivel, dar al programador herramientas para que, los problemas de codificación de los algoritmos sean realmente simplificados. El algoritmo, una vez codificado en el lenguaje de programación elegido, es traducido, normalmente sin la intervención humana, a una instrucción básica de máquina. Estas instrucciones son ejecutadas por el *hardware*, y si todo está bien, el efecto deseado del algoritmo se produce.

En resumen las ventajas de los lenguajes de alto nivel incluyen:

- Respaldo en las ideas de abstracción ya que así el programador puede concentrarse en la solución del problema, y no en los detalles de bajo nivel en la representación de datos.
- Facilidad en su aprendizaje.
- Rapidez en la solución de problemas.
- Portabilidad y depuración simplificado, facilidad de modificación y mantenimiento.
- Bajo costo de *software* y mayor confiabilidad.

El cómo hacer que una computadora lleve a cabo la tarea de cálculo que deseamos, se logra mediante una serie de instrucciones que obedecen ciertas reglas y métodos utilizando por llamarle de alguna manera, un vocabulario propio.

De acuerdo al problema que se tenga (científico, técnico, administrativo) se empleará el lenguaje más apropiado.

El programa se debe realizar de acuerdo a las reglas del lenguaje para transmitir a la computadora las instrucciones que debe realizar exactamente, este conjunto de instrucciones alimentan a la computadora a través de la unidad de entrada, las cuales son traducidas por el compilador al lenguaje de máquina. Esto trae como consecuencia que para cada lenguaje debe haber un traductor o compilador que transforme nuestras instrucciones al lenguaje de máquina. El compilador es un programa suministrado generalmente por el fabricante del equipo de cómputo.

Decidir qué lenguaje de programación es conveniente utilizar depende de algunos factores tales como características del problema a resolver, memoria disponible de la computadora en que se va a procesar, compiladores o traductores de que dispone dicha máquina, etc.

Por lo general es posible distinguir dos grupos de problemas, los que son técnicos o científicos y los que son administrativos.

En el grupo de los técnicos o científicos se observa que intervienen muchos cálculos y sin embargo, en el reporte de salida hay poca información. Por el contrario en el grupo de los administrativos, existen pocos cálculos y una amplia variedad de reportes de salida.

En el grupo de problemas técnicos y científicos los lenguajes comúnmente utilizados son: C, C++, FORTRAN, ALGOL, BASIC Y PASCAL.

En el grupo de problemas administrativos, el lenguaje más utilizado es el COBOL y para reportes el RPG.

### 2.2.2 COMPILADORES.

Un lenguaje de programación de alto nivel es traducido a su equivalente código de máquina por medio de un programa llamado compilador.

Los compiladores son programas largos y complejos, posiblemente envolviendo decenas o cientos de miles de líneas de código. Un compilador no se puede analizar considerándolo como una estructura monolítica; por lo que es necesario descomponerlo en módulos más simples.

El proceso de compilación se divide en dos subtarear:

- El análisis del programa fuente.
- La síntesis del programa objeto.

### 2.2.3 LENGUAJE C.

El lenguaje C es un lenguaje de programación de propósito general y ha sido utilizado en una gran gama de aplicaciones. Su aplicación más famosa fué en la creación del sistema operativo UNIX, donde el lenguaje C ha sido utilizado no solamente para la codificación del sistema operativo sino que también para sus utilidades asociadas.

Una característica poderosa de C es el uso de variables con direcciones (apuntadores). Se pueden definir arreglos y estructuras. También, se pueden manejar uniones para que las variables contengan valores de diferentes tipos.

El lenguaje C estándar es relativamente portable a pesar de las operaciones de bajo nivel que maneja. Las funciones del lenguaje C pueden operar en la mayoría de las computadoras y sistemas operativos. El lenguaje C es un lenguaje popular, y debido a que está muy asociado con el sistema operativo UNIX, probablemente continuará siendo popular por algún tiempo.

**LENGUAJE C++.** Una interesante evolución del lenguaje C es el lenguaje C++. El lenguaje C++ es una mejora del lenguaje C (con muy pocas excepciones).

El lenguaje C++ fué desarrollado a principios de los 80's en los Laboratorios AT & T Bell. El propósito era perfeccionar el lenguaje C para que generara un lenguaje, que con estructuras incluidas, ayudara en el desarrollo de sistemas de *software* extensos, manteniendo las ventajas del C de control de bajo nivel y formato conciso.

Con el lenguaje C++, los programas en lenguaje C necesitan muy poca modificación para correr en el ambiente C++. El lenguaje C++ es capaz de efectuar una compilación eficiente.

### 2.3 LENGUAJE ENSAMBLADOR DE LA FAMILIA TMS370.

Un lenguaje ensamblador es un lenguaje simbólico, que muestra de forma más legible los códigos de máquina binarios, por lo que se puede decir que el lenguaje ensamblador es un lenguaje intermedio entre un lenguaje de alto nivel y el código de máquina. La familia TMS370 consta de 73 instrucciones de ensamblador con una gran variedad de modos de direccionamiento.

El grupo de instrucciones del lenguaje ensamblador, provee un método conveniente para programar al MC. Cada instrucción consiste de los siguientes elementos.

-Un **mnemónico de función**. El mnemónico especifica el tipo de operación del MC.

-De cero a tres operandos. Los operandos indican donde el MC puede encontrar o almacenar datos durante la ejecución de una instrucción. El tipo y combinación de operandos determina el código de operación actual para una instrucción. La instrucción MOV, por ejemplo, tiene 27 opciones diferentes, cada una con su propio código de operación.

Una instrucción típica de dos operandos se muestra a continuación:

MNEMONICO	FUENTE	DESTINO
ADD	#9,	R3

El ejemplo anterior se lee de la siguiente forma: suma el valor 9 al contenido del registro número 3 y coloca el resultado en el mismo registro número 3. El registro destino sirve como una segunda fuente y como dirección final del resultado; los registros pueden ser manipulados sin utilizar registros intermedios.

El siguiente ejemplo muestra como la instrucción anterior puede aparecer en una línea de programa completa.

ETIQUETA	INTRUCCION	OPERANDOS	COMENTARIO
XXXXX	ADD	#9,R3	;comentario

Debe existir al menos un espacio entre cada tipo de entrada. La entrada de la etiqueta y el comentario son opcionales.

Las 73 instrucciones son respaldadas por 246 códigos de operación que proveen un control flexible del flujo del programa del MC. Algunas instrucciones tales como CLRC y TEST A, comparten el mismo código de operación con el objeto de ayudar al programador a entender todas las funciones de un código de operación. Algunas instrucciones utilizan 16 códigos de operación de 16 bits, dependiendo del tipo de instrucción y modo de direccionamiento empleado. El ensamblador forma varias manipulaciones de instrucciones de bit de otras instrucciones para simplificar la escritura y lectura del programa.

### 2.3.1 MODOS DE DIRECCIONAMIENTO.

Cada instrucción del lenguaje ensamblador de la familia TMS370 tiene de cero a tres operandos. Cada operando tiene un modo de direccionamiento. El modo de direccionamiento especifica la forma en que el MC calcula la dirección del dato requerido por la instrucción. La potencialidad de la familia TMS370 está basada por el gran número de modos de direccionamiento disponibles.

Los 14 modos de direccionamiento se dividen en dos clases:

- General, el cual utiliza un intervalo de direccionamiento de 8 bits.
- Extendido, que utiliza un intervalo de direccionamiento de 16 bits.

La división de estas dos clases de direccionamiento se muestra en la tabla 2.3.1.

MODO DE DIRECCIONAMIENTO	
GENERAL	IMPLICADO
	REGISTRO
	PERIFERICO
	INMEDIATO
	CONTADOR DE PROGRAMA RELATIVO
	STACK POINTER RELATIVO
EXTENDIDO	DIRECTO ABSOLUTO
	INDEXADO ABSOLUTO
	INDIRECTO ABSOLUTO
	INDIRECTO DE OFFSET ABSOLUTO
	DIRECTO RELATIVO
	INDEXADO RELATIVO
	INDIRECTO RELATIVO
	INDIRECTO DE OFFSET RELATIVO

Tabla 2.3.1 Modos de direccionamiento.

En el apéndice A se incluye la lista del formato de todas las instrucciones, códigos de operación, longitudes de byte, ciclos/instrucción, operandos, registro de estados y una descripción de la instrucción.

## CAPITULO III

# ORGANIZACION DEL SISTEMA



El objetivo del proyecto es crear un sistema para lograr la comunicación inalámbrica entre una PC y una o varias unidades remotas de captura. El sistema propuesto, podrá emplearse en el monitoreo, captura y control de datos en cadenas de restaurantes. En este capítulo se presenta una descripción general del sistema propuesto a manera de diagramas a bloques. Este capítulo se ha dividido en tres temas:

- Diagrama a bloques del sistema.
- Terminal de captura inalámbrica.
- Módulo central.

### 3.1 DIAGRAMA A BLOQUES DEL SISTEMA.

El sistema está constituido por Terminales de Captura Inalámbricas (TCI's), las cuales tienen comunicación con una Computadora Personal (PC) por medio de dos etapas: el Módulo de Comunicaciones de la Interfaz (MCI), y la Interfaz de la PC (IPC); estas dos últimas etapas junto con la PC forman el Módulo Central del sistema.

Las TCI's tienen como finalidad capturar, desplegar y enviar órdenes introducidas por un operador así como recibir y desplegar la información enviada por una PC. La PC se encargará de almacenar la información proveniente de las TCI's para ser procesada por un programa elaborado en lenguaje C++.

En conjunto, tanto las TCI's como el Módulo Central, forman un sistema de adquisición y control de datos. El diagrama a bloques del sistema se muestra en la fig. 3.1.1.

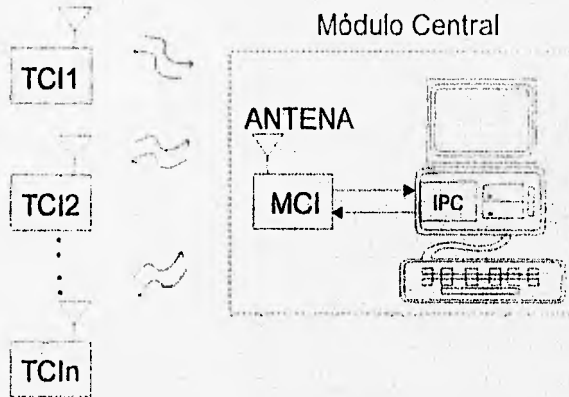


Fig. 3.1.1 Diagrama a bloques del sistema.

### 3.2 TERMINAL DE CAPTURA INALAMBRICA, (TCI).

En esta sección se describen las características de una terminal de captura inalámbrica, TCI, la cual como se mencionó anteriormente tiene las siguientes funciones:

- Captura de información.
- Despliegue de información.
- Acondicionamiento de señales para su procesamiento.
- Comunicación con el Módulo Central.

La Terminal de Captura Inalámbrica TCI, es un dispositivo transmisor-receptor de datos, que tiene como unidad de procesamiento un microcontrolador TMS370, el cual se encarga de interpretar la información introducida en un teclado, esta información es desplegada para su visualización en una pantalla y codificada para ser enviada posteriormente a un modulador-demodulador (modem), éste le da la forma necesaria a la señal para ser emitida por un circuito transmisor de RF y ser enviada hacia el Módulo Central.

La TCI también recibe señales de RF transmitidas desde el Módulo Central, mediante un circuito receptor, esta señal es enviada al modem para su digitalización, de esta forma la señal pueda ser procesada por el microcontrolador, y visualizada en la pantalla.

El diagrama a bloques de la fig. 3.2.1, muestra los elementos que componen una terminal de captura inalámbrica: pantalla, microcontrolador, teclado, modem, transmisor, receptor, antena y circuitos de apoyo. A continuación se describe cada elemento mencionado.

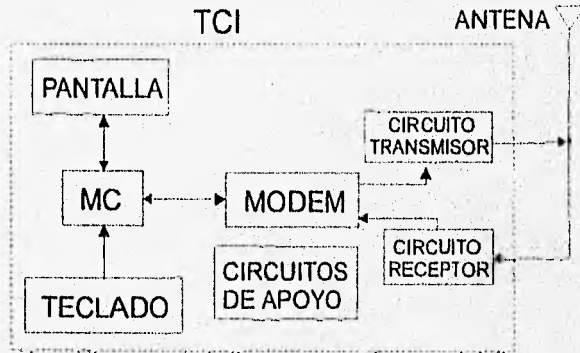


Fig. 3.2.1 Diagrama a bloques de la TCI.

### 3.2.1. TECLADO.

El teclado de la TCI, es el medio con el cual se introduce información a la terminal. Para cumplir el objetivo del sistema de optimizar la captura de datos en restaurantes, la información a capturar se dividió de la siguiente forma:

- Identificación del mesero que toma la orden.
- Mesa en la que se toma la orden.
- Orden tomada por el mesero (platos, cuenta, etc.).

De acuerdo a estas necesidades las funciones del teclado se clasifican de la siguiente forma:

- Digitos del 0 al 9.
- Funciones de operación: transmitir, cancelar, borrar y función alterna.
- Funciones de aplicación: platos, cuenta, mesero y mesa.

De acuerdo a las necesidades detectadas en los restaurantes, se concluyó que el número de teclas que cubrían la optimización de la captura de datos, fuera de 25. La organización del teclado se muestra en la tabla 3.2.1

Tabla 3.2.1.

MESERO	MESA	CUENTA	BORRAR	FUNC. ALTERNA
PLATILLO 1	PLATILLO 5	1	2	3
PLATILLO 2	PLATILLO 6	4	5	6
PLATILLO 3	PLATILLO 7	7	8	9
PLATILLO 4	PLATILLO 8	TRANSMITIR	0	CANCELAR

### 3.2.2 PANTALLA.

La pantalla es el dispositivo en el que el operador de la TCI visualizará tanto la información que es capturada mediante el teclado como la enviada por el Módulo Central.

### 3.2.3 MICROCONTROLADOR (MC).

El MC, es el encargado de procesar toda la información que se maneja en la TCI, tanto la información capturada en la misma, como la información recibida desde el Módulo Central. El MC empleado pertenece a la familia TMS370 de la marca Texas Instruments. En el capítulo 2 se explicaron las características de esta familia de microcontroladores.

En el sistema propuesto, el microcontrolador se encargará de realizar los siguientes procesos:

- Leer e interpretar la información introducida a través del teclado.
- Codificar la información para ser transmitida.
- Decodificar la información procedente del Módulo Central.
- Enviar la información a la pantalla para su visualización.

### 3.2.4 MODULADOR-DEMULADOR (MODEM).

En la TCI se incluye un modulador-demodulador (modem) el cual se encarga de modular la señal binaria entregada por el microcontrolador para obtener una señal analógica. Esta señal analógica de baja frecuencia puede ser manejada por una etapa de RF y ser transmitida al Módulo Central. Por otro lado, una señal de RF radiada desde el Módulo Central, después de pasar por una etapa demoduladora para obtener una señal analógica de baja frecuencia, es de nuevo demodulada por el modem para obtener una señal digital, la cual puede ser decodificada por el microcontrolador de la TCI.

### 3.2.5 TRANSMISOR.

El hecho de elegir una comunicación inalámbrica entre la TCI y el Módulo Central, involucra el empleo de una etapa de alta frecuencia para que las señales puedan ser emitidas por medio de una antena. La señal analógica entregada por el modem anteriormente mencionado, será la señal de información que se desea transmitir. Esta señal de baja frecuencia no puede ser emitida directamente a través de una antena, por lo que es necesario el empleo de un proceso de modulación en el que se utiliza una señal portadora con las características necesarias para ser emitida a través de dicha antena. Debido a lo anterior, la señal de información proveniente del modem pasará por un circuito transmisor cuyo diagrama a bloques se muestra en la fig. 3.2.2.

A continuación se explica cada una de las etapas del diagrama a bloques del transmisor.

La señal de información del modem pasa por una etapa de amplificación para proporcionarle el nivel requerido por la etapa moduladora. La etapa del oscilador de RF define la frecuencia de la señal portadora o un submúltiplo de esta. La etapa del multiplicador, el cual sintoniza una armónica de la frecuencia del oscilador, provoca que esta frecuencia se multiplique, lográndose así, que la frecuencia final de la portadora sea un múltiplo de la frecuencia del oscilador. La etapa de amplificación de RF del oscilador incrementa el nivel de éste para ser entregada al modulador analógico. El modulador analógico de la fig. 3.2.2 combina los componentes de las frecuencias tanto de la señal de información, como de la portadora para generar la modulación. Se hace necesario el uso de un amplificador adicional después de la etapa de modulación, para aumentar la potencia de la señal entregada a la antena. La antena transmisora convierte la señal de RF en una onda electromagnética y la radia hacia el receptor del Módulo Central.

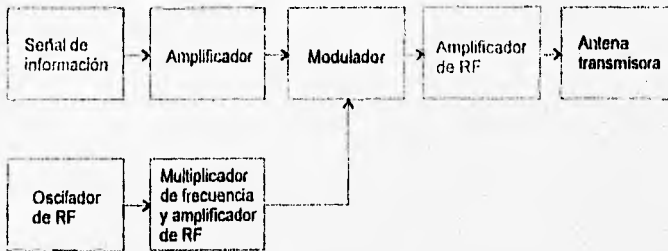


Fig. 3.2.2 Diagrama a bloques de la etapa de transmisión

### 3.2.6 RECEPTOR.

El receptor, es la etapa que se encargará de demodular la señal de RF proveniente del Módulo Central, obteniendo la señal analógica de baja frecuencia que posteriormente se envía al modem para que la demodule en una señal digital. La fig. 3.2.3 muestra el diagrama a bloques de la etapa de recepción, cuya dinámica se describe a continuación.

La señal de RF radiada por la antena del Módulo Central, es captada por la antena de la TCI. La etapa de amplificación de RF aumenta el nivel de esta señal de alta frecuencia para aplicarse a la entrada del mezclador. El oscilador local genera una frecuencia definida por la suma o diferencia de la frecuencia de la señal de RF y la frecuencia intermedia IF. El mezclador es un dispositivo que convierte la frecuencia de la señal RF de recepción en la frecuencia intermedia IF. El amplificador de IF aumenta el nivel de señal para su manipulación. El detector recobra la señal mensaje original de la señal de entrada IF modulada. El amplificador de la última etapa, aumenta el nivel de la señal de la salida del detector. La salida del amplificador de la última etapa es la señal de baja frecuencia, la cual se envía directamente al modem para su demodulación a señal digital.

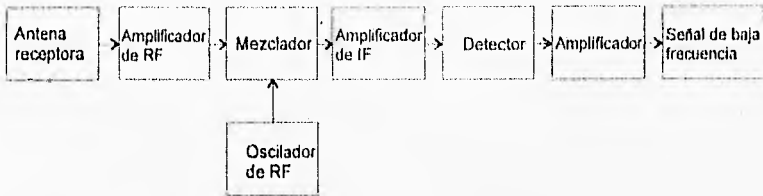


Fig. 3.2.3 Diagrama a bloques de la etapa de recepción

### 3.2.7 ANTENA.

Debido a que la comunicación entre TCI's y Módulo Central se realiza via RF, es necesario el empleo de una antena que se encargue de realizar dos funciones:

- Radiar las señales de RF de transmisión.
- Ser el medio en el que se induzca las señales de RF de recepción.

Por lo tanto, la antena de la TCI convertirá la señal de RF de transmisión, en una onda electromagnética y la radiará hacia el receptor del Módulo Central. Esta misma antena también operará como receptora cuando se induzca en ella una señal de RF radiada desde el Módulo Central.

### 3.2.8 CIRCUITOS DE APOYO.

Para asegurar el funcionamiento óptimo de la TCI, se hace necesario la utilización de elementos que respalden la operación de los bloques que forman la TCI. Estos elementos atenderán los siguientes aspectos:

- Iluminación de la TCI en lugares oscuros.
- Generación de una señal audible para alertar al operador.
- Detección de nivel de batería.
- Acondicionadores de voltajes para polarización.

El empleo de las TCI's en lugares poco iluminados, hace necesario el contar con una fuente de iluminación para el teclado y la pantalla que son los elementos con los que el operador tiene contacto visual, así mismo se requiere en la TCI una señal audible para que el operador atienda alguna información urgente mostrada en la pantalla.

Debido a que la TCI será un dispositivo portátil que empleará una batería para su alimentación, se incluye una etapa de detección del nivel bajo de voltaje de la batería previo a un nivel que haga inoperable los circuitos de la TCI.

El último elemento de apoyo de la TCI son los circuitos que se encargaran de proporcionar el nivel de voltaje requerido por cada uno de los elementos de la TCI.

### 3.3 MODULO CENTRAL.

La finalidad del Módulo Central es la de coordinar la información, tanto la procesada en las TCI's como la generada en el mismo Módulo Central. Como ya se ha mencionado, este módulo contiene tres bloques, el Módulo de Comunicaciones de la Interfaz (MCI), la Interfaz de la PC personal (IPC) y la PC, los cuales realizan las operaciones necesarias para lograr la comunicación con las TCI's. El bloque MCI se encarga de la etapa de RF, la IPC adecuará y administrará el flujo de la información, y la PC almacena esta información para ser utilizada por un programa residente en la misma.

#### 3.3.1 MODULO DE COMUNICACIONES DE LA INTERFAZ, (MCI).

Debido a que la comunicación entre la TCI y el Módulo Central se realiza via RF es necesario que ambos bloques contengan etapas de transmisión y recepción para establecer así el enlace de RF entre ellos. El MCI es la etapa de RF del Módulo Central y su objetivo es lograr la comunicación con la etapa de RF de la TCI.

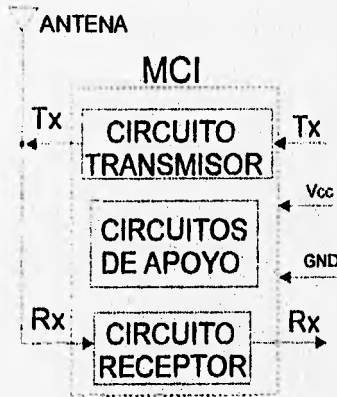


Fig. 3.3.1 Diagrama a bloques del MCI.

A continuación se describen los elementos del MCI mostrados en el diagrama a bloques de la fig. 3.3.1:

**ANTENA.** Realiza las mismas funciones efectuadas por la antena de la TCI.

**TRANSMISOR DEL MODULO CENTRAL.** Esta etapa tiene la misma función que el transmisor de la TCI, se emplea para generar señales de RF y su configuración es similar a la descrita en el punto 3.2.6.

**RECEPTOR DEL MODULO CENTRAL.** Esta etapa tiene la misma función que el receptor de la TCI, se emplea para recibir las señales de RF provenientes de la TCI y su configuración es similar a la descrita en el punto 3.2.7.

**CIRCUITOS DE APOYO.** Su función, como en el caso de las TCI's, es la de asegurar el funcionamiento óptimo del MCI, por lo que se debe utilizar elementos que respalden la operación de los circuitos que forman el MCI. Estos elementos atenderán los siguientes aspectos:

- Acondicionamiento de voltajes para polarización.
- Indicador de polarización.

El primer punto mencionado, tiene la misma función que para el caso de la TCI. El circuito indicador de polarización, muestra si el MCI se encuentra energizado.

### 3.3.2 INTERFAZ DE LA COMPUTADORA PERSONAL, (IPC).

Para que dos entidades se comuniquen, es necesario que exista una compatibilidad entre ellas. Si no existe esta compatibilidad, es necesario emplear un dispositivo que acople estas entidades y establezca la comunicación. En el caso del sistema presentado, estas dos entidades son el MCI y la PC, y el dispositivo acoplador es la interfaz de la computadora personal, IPC.

Como se describió en el punto 1.6.8, la comunicación con el slot de la PC se lleva a través de líneas de datos, direcciones y control. Por lo tanto, para establecer la comunicación es necesario una etapa de aislamiento con las líneas de datos del *bus*, así como una etapa de codificación para las líneas de direcciones y control. Es por eso, que la IPC, consta de un modem, un microcontrolador (MC), un *buffer*, y un codificador. A continuación se describen los elementos contenidos en la IPC, los cuales se muestran en el diagrama a bloques de la fig. 3.3.2:



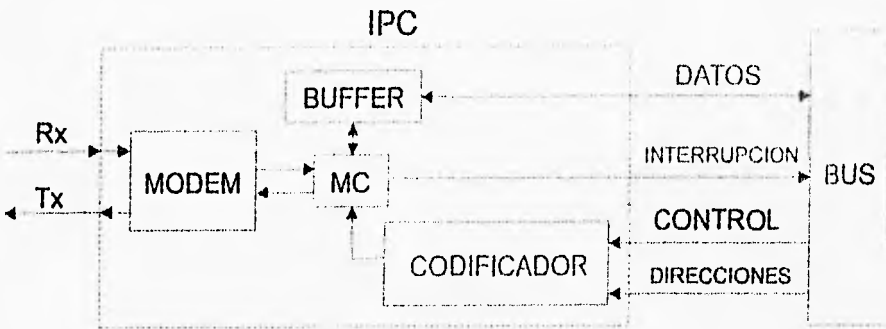


Fig. 3.3.2 Diagrama a bloques de la IPC.

**MODEM.** La función y objetivo de esta etapa, se discutieron en el punto 3.2.4.

**MICROCONTROLADOR (MC).** Es el dispositivo que se encarga de procesar la información en la IPC, y tiene como funciones principales:

- Solicitar información a las TCI's.
- Decodificar la información proveniente de las TCI's.
- Realizar la comunicación con el bus de la PC.
- Administrar el flujo de información entre las TCI's y la PC.

**BUFFER.** El objetivo de esta etapa es aislar las líneas de datos del bus de la PC, del puerto del MC encargado de manejar dichas líneas. Lo anterior evita conflictos de comunicación con los periféricos que comparten el bus de datos de la PC, ya que no se puede acceder dos periféricos simultáneamente.

**CODIFICADOR.** Para conocer el momento en que la PC está atendiendo a la IPC, es necesario que el MC monitoree tanto las líneas de direcciones como de control del bus de la PC. Estas líneas son demasiadas como para ser atendidas por el microcontrolador, por lo que se debe implementar un codificador que minimize dichas líneas.

## CAPITULO IV

### DISEÑO DE LA TERMINAL DE CAPTURA INALAMBRICA

En el presente capítulo, se describe el proceso de selección, descripción y acondicionamiento de cada una de las etapas que constituyen la TCI. Los circuitos que conforman la TCI, son los siguientes:

- Circuito de la pantalla.
- Circuito del teclado.
- Circuito del microcontrolador.
- Circuitos de RF.
- Circuito del modem.
- Circuitos de apoyo.

A continuación se presenta el diseño de las etapas mencionadas anteriormente.

#### 4.1 CIRCUITO DE LA PANTALLA.

Este circuito sirve para visualizar tanto la información capturada en la TCI, como la recibida desde el Módulo Central. Los requisitos que debe cumplir esta etapa son:

- Capacidad para desplegar caracteres alfanuméricos.
- Despliegue de un mínimo 20 caracteres simultáneamente.
- Tamaño compacto.
- Posibilidad de visualización en la oscuridad.
- Voltaje de alimentación menor o igual a 7.2 V.

El circuito seleccionado para esta etapa es el AND491ST-LED de la marca AND, el cual cumple con los requisitos mencionados. Este circuito tiene las siguientes características:

- Módulo LCD (*Liquid Crystal Display*).
- Presentación de caracteres en forma de matriz de puntos.
- Formato de carácter de 5 x 7 puntos más línea de cursor.
- Voltaje de operación de +5 V.
- Interfaz directa para CPU de 4 u 8 bits.
- Led para iluminación de la pantalla.

Una de las características que influyeron en la elección de este dispositivo, es que se conecta directamente al microcontrolador y únicamente requiere una fuente de polarización de +5 V para su funcionamiento. El módulo cuenta con una terminal para control de contraste la cual en el diseño es conectada a 0 V, debido a que con este nivel se tiene una visibilidad adecuada. Para iluminar el módulo es necesario aplicar un nivel de voltaje entre +3.8 y +4.4 V en el pin LA. El diagrama de conexión de este dispositivo se muestra en la fig. 4.1.1. Las hojas de especificaciones de la pantalla se muestran en el apéndice A.

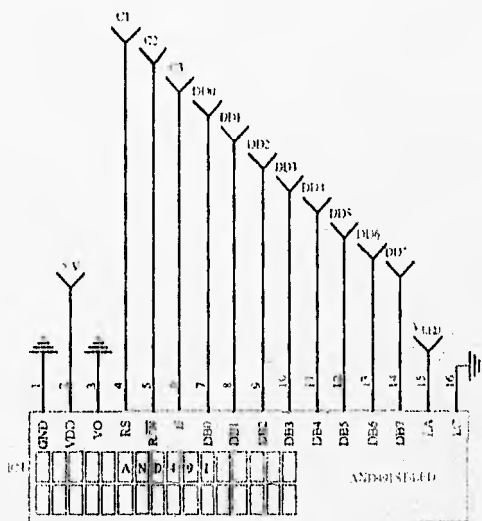


Fig. 4.1.1 Circuito de la pantalla.

#### 4.2 CIRCUITO DEL TECLADO.

Como se mencionó en el capítulo 3, la captura de datos en la TCI se realiza mediante un teclado que consta de 25 teclas. Para esta función se implementa un arreglo de interruptores en forma de matriz de 5 x 5, de tal forma que para conocer qué interruptor se encuentra cerrado, se enviará un voltaje a través de cada una de las líneas del arreglo matricial, y se buscará a través de las columnas qué interruptor está cerrado, lo cual indicará la coordenada de la tecla oprimida.

Para la implementación del teclado se utilizan interruptores del tipo "contacto momentaneo normalmente abierto", el material de estos interruptores son de hule de silicón con contactos de carbón, los cuales actúan sobre una tarjeta de circuito impreso, donde se encuentran las terminales normalmente abiertas. Este tipo de teclado se eligió debido a su bajo costo y durabilidad. En el diagrama eléctrico de la fig. 4.2.1 se muestra la conexión del teclado.

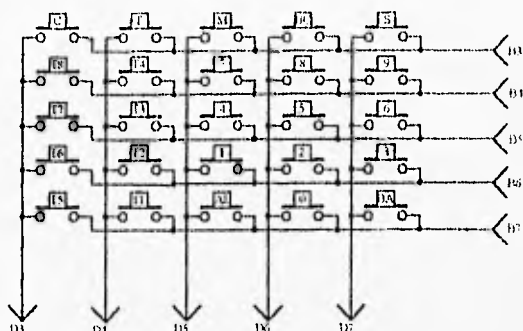


Fig. 4.2.1 Circuito del teclado.

#### 4.3 CIRCUITO DEL MICROCONTROLADOR MC.

El microcontrolador utilizado para el diseño pertenece a la familia de microcontroladores TMS370 de Texas Instruments, ya descrita en el capítulo 2. De esta familia de microcontroladores se eligió el dispositivo TMS370C622. Este dispositivo entre los microcontroladores existentes en el mercado cubre las necesidades del diseño, tales como:

- Un puerto serial para la transmisión de información al Módulo Central.
- 30 terminales de entrada y salida, de las cuales 10 se emplearán para la lectura del teclado, 11 para el despliegado de información en la pantalla, y 3 para el control de circuitos periféricos.

Para lograr el adecuado funcionamiento del microcontrolador, se requiere satisfacer las siguientes condiciones de operación:

- Polarizar al dispositivo con +5 Vdc.
- Poner un nivel de 0 V en el pin 34, para que el dispositivo funcione en el modo simple, en el cual toda la memoria (de programa y datos) y los periféricos se encuentran dentro del circuito integrado.
- Conectar un cristal para establecer la frecuencia de operación. En el caso del sistema propuesto se emplea un cristal de 20 MHz. El cristal requiere de dos capacitores conectados a tierra de un valor recomendado por el fabricante de 15 pF.
- Contar con un circuito para reinicializar externamente el microcontrolador. Este circuito se debe conectar al pin 6 del MC. Un pulso de 0 V inicia la secuencia de reset del sistema. El sistema puede detectar pulsos de reset de duración corta de algunos nanosegundos, sin embargo se necesita de un pulso de 0 V de por lo menos un ciclo, para garantizar la inicialización del sistema. El sistema se encuentra en

reset hasta que el *pin* de reset se desactiva, es decir, vuelve a un nivel de 5 V. La acción de reset debe de generar un pulso de 0 V en el *pin* de reset. Este *pin* debe de permanecer con un nivel de 0 V, el tiempo suficiente para inicializar los diferentes dispositivos periféricos existentes. Para el caso del presente diseño, el circuito que necesita de un tiempo específico para reinicializarse es la pantalla, este tiempo requerido es de un mínimo de 4.2 ms. El circuito empleado para realizar la acción de reset se muestra en la fig. 4.3.1. Los valores de los elementos del circuito de reset se calculan a partir de la ecuación 4.3.1.

$$\tau = RC \quad \text{ec. 4.3.1}$$

Por lo que:

$$C = \tau/R$$

Fijando el valor de  $R_{26} = 10 \text{ k}\Omega$  se calcula el valor de C.

$$C_5 = 4.2 \text{ ms}/10 \text{ k}\Omega$$

$$C_5 = .42 \text{ }\mu\text{F}$$

por tanto, el valor comercial empleado es  $C_5 = .47 \text{ }\mu\text{F}$

Debido a que la corriente máxima en el *pin* de reset es de 2 mA, se requiere una resistencia que limite la corriente, el valor de esta resistencia se determina con la siguiente ecuación:

$$\begin{aligned} R &= V/I && \text{ec. 4.3.2} \\ &= (5 \text{ V})/(2 \text{ mA}) \\ &= 2.5 \text{ k}\Omega \end{aligned}$$

por tanto  $R_{32} = 2.7 \text{ k}\Omega$

Se incluye un diodo D1, que permite al capacitor descargarse rápidamente durante una despolarización repentina del sistema. El diagrama eléctrico del microcontrolador se muestra en la fig. 4.3.1.

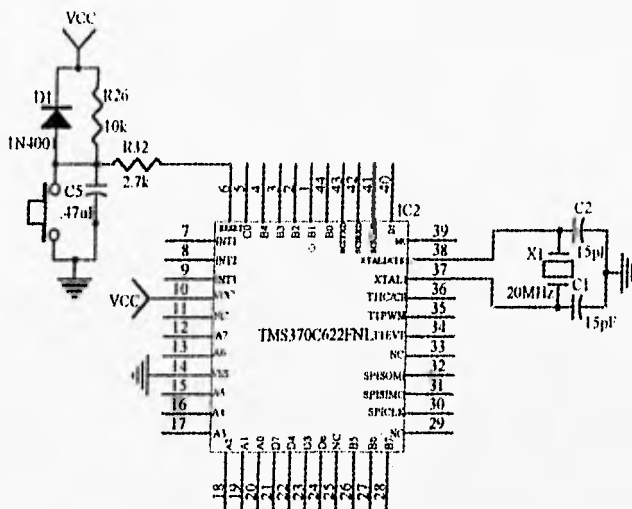


Fig. 4.3.1 Circuito del microcontrolador.

**4.4 CIRCUITOS DE RADIO-FRECUENCIA.**

De acuerdo a lo expuesto en el capítulo 3, se estableció que la comunicación entre las TCIs y el Módulo Central se realiza en forma inalámbrica, lo que implica el uso del espectro electromagnético. El empleo del espacio aereo como medio de transmisión está regulado por la Secretaría de Comunicaciones y Transportes (SCT), la cual tiene disponible la banda de 46 y 49 MHz para sistemas como el expuesto en el presente trabajo; dicha banda está dividida en 12 canales como se muestra a continuación:

Numero de canal.	Frecuencia de transmisión del Módulo Central (Mhz).	Frecuencia de transmisión de la TCI (MHz).
1	46.610	49.830
2	46.630	49.845
3	46.710	49.875
4	46.730	49.890
5	46.870	49.930
6	46.930	49.970
7	46.970	49.990
8	46.720	49.720
9	46.740	49.740
10	46.760	49.760
11	46.790	49.790
12	46.810	49.810

Para la realización de la etapa de comunicación se emplean dos circuitos integrados de la marca Motorola, ya que cuentan internamente con los bloques necesarios para la implementación de un transmisor y un receptor de FM de baja potencia. Estos circuitos integrados son el MC2833, el cual es un sistema transmisor de baja potencia y el MC3362, el cual es un receptor de FM de baja potencia. Estos circuitos integrados fueron elegidos por las siguientes razones:

- Requieren de bajos voltajes para su operación (3 ~ 8 V).
- Operan con un bajo consumo de corriente.
- Requieren de un mínimo número de componentes externos para su operación.
- Tienen un tamaño reducido.
- Estos circuitos están fabricados para operar dentro de las bandas de frecuencias de 46 y 49 MHz.

De la lista de los canales de transmisión anterior, se emplea el canal 2, debido a que ya se contaba con los componentes externos principales que requieren los circuitos integrados MC2833 y MC3362 para operar en las frecuencias de este canal.

A continuación se presenta la descripción de las etapas de transmisión y recepción de la TCI.

#### 4.4.1 CIRCUITO DE TRANSMISION.

Como ya se mencionó, para la etapa de transmisión de la TCI, se emplea el circuito integrado MC2833, cuya configuración para operar en la frecuencia de 49.845 MHz recomendada por Motorola se muestra en la fig. 4.4.1.

Las etapas implementadas en el circuito transmisor son las siguientes:

- Modulador de FM.
- Triplicador de frecuencia.
- Amplificador de RF.

La señal moduladora es aplicada en el *pin* 3 del circuito integrado, esta señal realiza la modulación en FM en la señal portadora generada por el oscilador de RF, cuya frecuencia de operación es fijada externamente por un cristal piezoeléctrico cuyo valor se evalúa mediante la ecuación 4.4.1.

$$F_{xtal} \approx F_{op}/3$$

ec. 4.4.1





las TCIs se active únicamente cuando se envíen datos a la PC y que dicha activación se realice en forma secuencial. El encendido y apagado del circuito de transmisión es controlado por el microcontrolador.

#### 4.4.2 CIRCUITO DE RECEPCION.

Para la etapa de recepción de la TCI, como ya se mencionó, se emplea el circuito integrado MC3362, cuya configuración para funcionar en la frecuencia de 46.63 MHz, es recomendada por Motorola.

A continuación se describe la función de la etapa de recepción de la TCI, mostrada en la fig. 4.4.2:

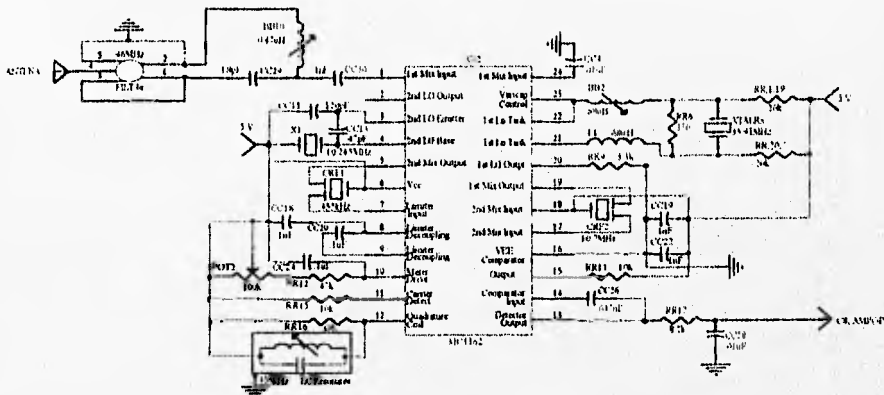


Fig. 4.4.2 Circuito de recepción.

La señal de recepción proveniente de la antena es filtrada para minimizar las señales que pueden afectar la calidad de la comunicación, el filtro que se emplea es el KAF-46NRME marca Kyocera del tipo SAW (Surface Acoustic Wave) cuya frecuencia central es de 46.5 MHz y un ancho de banda de 3 MHz. La señal proveniente del filtro se envía al pin de entrada del circuito integrado a través de una etapa que acopla las impedancias del filtro y del circuito integrado. La primera etapa del circuito integrado es un mezclador usado para convertir la entrada de RF a la primera etapa de frecuencia intermedia de 10.7 MHz, para lograr esto se emplea el primer oscilador local cuya frecuencia se define por un cristal piezoeléctrico externo, dicha frecuencia se evalúa con la ec. 4.4.4.

$$\begin{aligned}
 f_{xtali} &= f_{Rk} - 10.7 \text{ MHz} \\
 f_{xtali} &= 46.63 \text{ MHz} - 10.7 \text{ MHz} \\
 f_{xtali} &= 35.93 \text{ MHz}
 \end{aligned}
 \tag{ec. 4.4.4}$$

La señal de frecuencia intermedia de 10.7 MHz pasa por un filtro cerámico del mismo valor, para luego ser aplicada al segundo mezclador, la cual convertirá la frecuencia de esta señal en una frecuencia de 455 kHz

para posteriormente ser filtrado nuevamente por un filtro cerámico de 455 kHz para esta segunda frecuencia intermedia, la señal de 455 kHz se aplica al amplificador limitador de frecuencia intermedia antes de ser dirigido al circuito detector de cuadratura, que es la última etapa del circuito integrado.

Se incluye como última etapa del receptor de la TCI, un amplificador que proporciona un nivel adecuado a la señal de salida del circuito receptor, para ser manejado por las etapas posteriores. El circuito amplificador es el circuito integrado MC34119, cuyas hojas de especificaciones se muestran en el apéndice A.

Los capacitores C6 y C7 de los pines 2 y 3, controlan la alimentación y el rechazo de ruido del IC3.

La configuración interna del circuito integrado, consiste de dos amplificadores idénticos. El amplificador 1 tiene una ganancia de malla abierta de más de 80 dB, esta ganancia se calcula con la ec. 4.4.5.

$$G = 2R_f/R_i \quad \text{ec. 4.4.5.}$$

La señal entregada por el MC3362 es aproximadamente 100 mV, por lo tanto para tener aproximadamente 4 V en el circuito modem se propone una ganancia de  $G = 40$  y  $R_f = R_{34} = 68 \text{ k}\Omega$  y despejando  $R_i$  de la ecuación 4.4.5,

$$\begin{aligned} R_i &= 2R_f/G \\ R_i &= (2 \times 68 \text{ k}\Omega)/40 \\ R_i &= 4 \text{ k}\Omega \end{aligned}$$

por lo tanto:  $R_i = R_{30} = 3.9 \text{ k}\Omega$

El diagrama de conexión del MC34119 se muestra en la fig. 4.4.3.

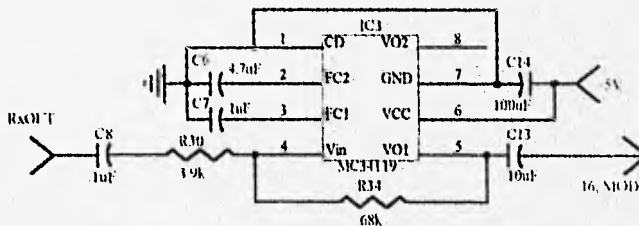


Fig. 4.4.3 Circuito amplificador de baja frecuencia.

#### 4.5 CIRCUITO DE MODULACION FSK.

Debido a que la comunicación de la TCI con el Módulo Central se efectúa vía RF, las señales de información en forma digital generadas o recibidas por el MC, requieren ser convertidas en señales analógicas, por lo que se requiere de una modulación y demodulación digital. Para esta

etapa se empleó el CI MC145442 de Motorola, el cual es un modem que tiene integrados un modulador-demodulador de manipulación de cambio de frecuencia FSK. Este dispositivo puede establecer una comunicación *full duplex* o *half duplex* de 300 baud. También incluye un circuito detector de portadora para la etapa de demodulación. En la fig. 4.5.1, se muestra el diagrama eléctrico del CI del modem en la TCI.

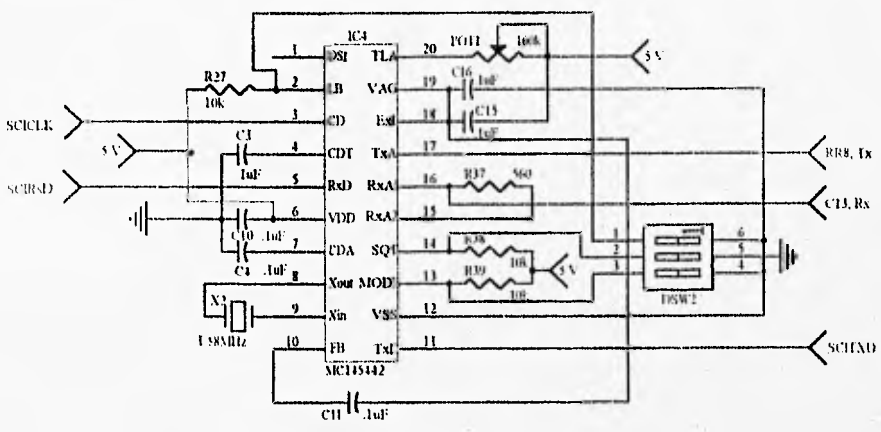


Fig. 4.5.1 Circuito del modem.

El modo de operación del circuito integrado determina las frecuencias de transmisión y recepción con las que opera el dispositivo. El modo de operación del dispositivo se programa mediante los niveles de voltaje aplicados a sus *pin*s 2, 13, y 14. El modo de operación elegido fué el modo respuesta, por lo que los *pin*s 2, 13 y 14 se conectan a tierra. Las frecuencias de transmisión y recepción del modem se especifican en la tabla 4.5.1.

Tabla 4.5.1

Dato	Transmisión	Recepción
Espacio	1850 Hz	1180 Hz
Marca	1650 Hz	980 Hz

La señal digital de transmisión procedente del MC es aplicada al *pin* 11 del modem. Los datos son modulados en FSK y enviados a las etapas de filtrado y amplificación. La señal analógica entregada por el *pin* 17 del CI del modem, se aplica a la etapa de transmisión para su modulación en FM.

La señal entregada por el amplificador MC34119 se aplica al *pin* 16 del CI del modem, internamente pasa por un duplicador, seguido de un filtro llamado anti-alias y un circuito muestreador-retenedor, después pasa por una etapa de filtrado y amplificación para finalmente pasar por un

demodulador. La señal digital resultante sale por el pin 5 del modem para ser enviada al MC.

#### 4.6 CIRCUITOS DE APOYO

Los circuitos de apoyo con los que cuenta la TCI, tienen la finalidad de ayudar al sistema a tener un funcionamiento más confiable, optimizando el consumo de energía, y en general monitoreando funciones de operación. Por lo tanto, el sistema TCI cuenta con los siguientes circuitos de apoyo:

- Circuito de encendido y apagado de iluminación para teclado y pantalla.
- Circuito de sonido.
- Circuito indicador de nivel de batería.
- Circuito de alimentación.

##### 4.6.1 CIRCUITO DE ENCENDIDO Y APAGADO DE ILUMINACION DE TECALDO Y PANTALLA.

Como ya se mencionó en el punto 3.2.9, la TCI puede emplearse en lugares poco iluminados, por lo que es necesario incluir una etapa que permita mantener iluminadas las áreas de la TCI con las cuales se tiene contacto visual, es decir el pantalla y el teclado, en la implementación de esta etapa se considera lo siguiente:

- La iluminación de la pantalla se efectúa con el led contenido en el propio dispositivo.
- Debido a que el teclado está fabricado con un material translúcido, este será iluminado por un arreglo de leds ubicados en la placa de su circuito impreso.

Con el objeto de economizar la energía de la TCI proveniente de una batería, el encendido de la iluminación la realiza el microcontrolador, únicamente cuando sea necesario. Debido a que el microcontrolador no puede proporcionar directamente el voltaje y la corriente demandada por los leds del teclado y de la pantalla, se emplea una etapa que a partir de los voltajes de 5 y 0 V realice el encendido y apagado de los dispositivos de iluminación. La alimentación para la iluminación, se interrumpe simultáneamente en el teclado y la pantalla mediante la operación en las regiones de corte y activa directa de los transistores T1 y T2 mostrados en la fig. 4.6.1. El empleo de dos transistores, se debe a que los niveles de voltaje de encendido son diferentes en el teclado y la pantalla. La iluminación del teclado se realiza mediante 14 leds, conectados en paralelo, con su ánodo conectado al emisor de T1, y el cátodo conectado a tierra. La pantalla cuenta para su iluminación, con dos terminales, ánodo y cátodo. La terminal denominada ánodo se conecta al emisor de T2, y la terminal denominada cátodo va conectada a tierra. La acción de encendido de la iluminación se efectúa aplicando un nivel de +5 V en las resistencias de base de T1 y T2, para que estos pasen de la región de corte, a la

región de activa directa, encendiéndose así la iluminación, esta señal es enviada por el microcontrolador. El apagado de la iluminación se efectúa con un nivel de 0 V aplicado en las resistencias de base de los transistores T1 y T2, provocando que pasen a la región de corte, lográndose así el apagado de la iluminación.

A continuación se presenta el cálculo de los elementos de la etapa de control de encendido y apagado de la iluminación del teclado y pantalla, mostrados en la fig. 4.6.1.

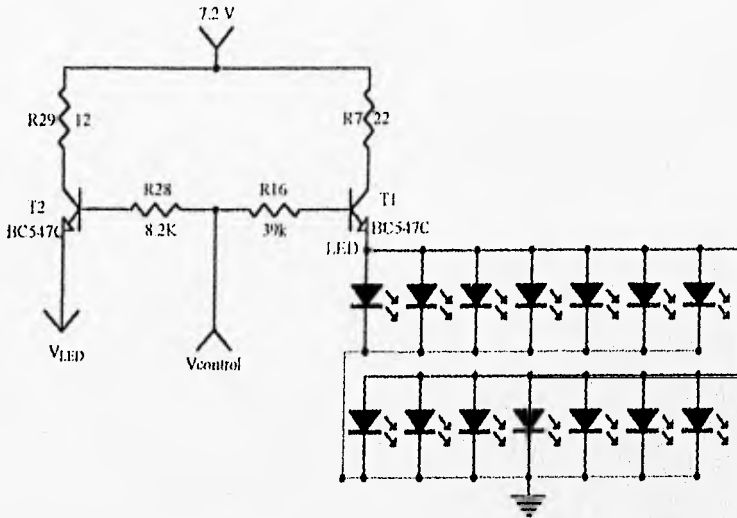


Fig. 4.6.1 Circuito de encendido y apagado de iluminación de teclado y pantalla.

Para que exista una buena iluminación en el teclado, se estableció que los leds operen con los siguientes valores de polarización:

$$V_{LED} = 2.2 \text{ V}$$

$$I_{LED} = 20 \text{ mA} = I_E$$

Como se mencionó, para encender la iluminación se requiere que el transistor T1 opere en la región de activa directa. En esta etapa se emplea el transistor BC547C, que tiene una ganancia de corriente típica de 400 y un voltaje de encendido de la unión base-emisor de  $V_{BE} = .7 \text{ V}$ .

Considerando  $V_{CE} = 4.5 \text{ V}$ , para que exista una caída de voltaje en la resistencia de colector muy pequeña.

De las ecuaciones de malla obtenemos:

$$V_{control} = R_{16}I_B + V_{BE} + V_{LED} \quad \text{ec.4.6.1}$$

$$V_{cc} = R7I_c + V_{ce} + V_{LED}$$

ec. 4.6.2

despejando R16 de la ec. 4.6.1.

$$\begin{aligned} R16 &= (V_{control} - V_{BE} - V_{LED}) / (I_E / 401) \\ &= (5V - 0.7V - 2.2V) / (20mA / 401) = 42 \text{ k}\Omega \end{aligned}$$

el valor comercial que se utiliza es: R16 = 39 k $\Omega$

despejando R7 de la ec. 4.6.2.

$$R7 = (V_{cc} - V_{ce} - V_{LED}) / I_c = (7.2V - 4.5V - 2.2V) / (20mA) = 25 \Omega$$

el valor comercial que se empleará para R7 = 22  $\Omega$ .

En el caso de la pantalla, para que exista una buena iluminación, se requieren de las siguientes características:

$$\begin{aligned} V_{LED} &= 4 \text{ V} \\ I_{LED} &= 15 \text{ mA} = I_E \\ V_{ce} &= 3 \text{ V} \end{aligned}$$

empleando las ecuaciones 4.6.1 y 4.6.2, se calcula los valores de R28 y R29:

$$\begin{aligned} R28 &= (V_{control} - V_{BE} - V_{LED}) / I_B \\ &= (5V - 0.7V - 4V) / (15mA / 401) = 8.02 \text{ k}\Omega \end{aligned}$$

el valor comercial que se utiliza es: R28 = 8.2 k $\Omega$

para el cálculo de R29,

$$R29 = (V_{cc} - V_{ce} - V_{LED}) / I_c = (7.2V - 3V - 4V) / (15mA) = 13.33 \Omega$$

el valor comercial que se utiliza es: R29 = 12  $\Omega$

#### 4.6.2 CIRCUITO DE SONIDO.

Esta etapa es incluida en la TCI, para realizar las siguientes funciones:

- Generar una señal audible al oprimir cada tecla.
- Generar una señal audible cuando se quiera que el operador de la TCI, atienda algun aviso mostrado en la pantalla.

Esta etapa es implementada con un multivibrador astable, la cual genera una señal que al ser aplicada a un buzzer emite un sonido audible.

El multivibrador astable de la fig. 4.6.2, se implementa en forma discreta, empleandose dos transistores BC547C. La frecuencia de oscilación elegida para el circuito es de 1.25 kHz. La activación de la señal audible la realiza el microcontrolador enviando un pulso de +5 Vdc.

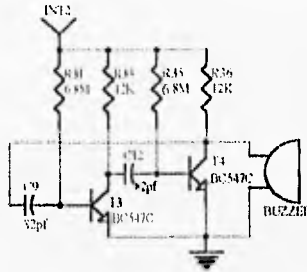


Fig. 4.6.2 Circuito de sonido.

Considerando los siguientes parámetros de operación:

$$V_{enc} = +5 \text{ Vdc}$$

$$I_{enc} = .5 \text{ mA}$$

$$V_{BE} = 2 \text{ V}$$

$$I_{CT4} = I_{CT1} = I_{enc}/2 = .25 \text{ mA}$$

entonces, de la ecuación de malla de colector de T4:

$$V_{enc} = R36 \times I_{CT4} - V_{CE} \quad \text{ec.4.6.3}$$

despejando R36 de la ecuación anterior y sustituyendo valores:

$$\begin{aligned} R36 &= (V_{enc} - V_{CE}) / I_{CT4} \\ &= (5 \text{ V} - 2 \text{ V}) / .25 \text{ mA} \\ &= 12 \text{ k}\Omega \end{aligned}$$

por lo tanto  $R33 = R36 = 12 \text{ k}\Omega$ .

De la ecuación de malla de base de T4:

$$V_{enc} = R35 \times I_B - V_{BE}$$

despejando R35 de la ecuación anterior y sustituyendo valores:

$$\begin{aligned} R35 &= (V_{enc} - V_{BE}) / I_B \\ &= (5 \text{ V} - .7 \text{ V}) / (.25 \text{ mA}/400) = 6.88 \text{ M}\Omega \end{aligned}$$

por lo tanto  $R35 = R31 = 6.8 \text{ M}\Omega$

La frecuencia de oscilación del circuito está dada por la ecuación 4.6.4, de la cual se obtienen los valores de C9 y C12:

$$f = 1.4 / (R31 \times C9 + R35 \times C12) \quad \text{ec.4.6.4}$$

$$\begin{aligned} C9 = C12 &= 1.4 / (2 f R) \\ &= 1.4 / (2 \times 1.25 \text{ kHz} \times 6.8 \text{ M}\Omega) \\ &= 82 \text{ pF.} \end{aligned}$$



#### 4.6.3 CIRCUITO INDICADOR DE NIVEL DE BATERIA.

Como ya se menciono en el capitulo 3, es necesario incluir un circuito que alerte al operador de la TCI, que el voltaje proporcionado por la batería está cerca de ser insuficiente para que la TCI funcione adecuadamente.

Este circuito consta principalmente de un diodo zener, un transistor, y un led. El diodo zener D2 de la fig. 4.6.3, está conectado directamente a la batería de 7.2 V. De esta forma, D2 se encuentra conduciendo y manteniendo la base del transistor T5 con un nivel alto de voltaje, esto provoca que T5 opere en el estado de saturación. Lo anterior evita que el led encienda, ya que el voltaje de saturación de T5 es menor al voltaje de encendido del led. En el momento en que el nivel de voltaje aplicado a D2, cae por debajo de su nivel de encendido, D2 deja de conducir, provocando a su vez que T5 cambie a estado de corte. Es entonces cuando toda la corriente de R6, fluye por el led, el cual se enciende e indica así un nivel bajo de batería.

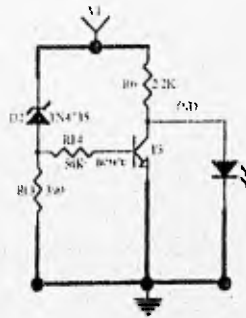


Fig. 4.6.3 Circuito indicador de nivel de batería.

El diodo zener empleado es el 1N4735, que opera con las siguientes características:

$$\begin{aligned} -V_Z &= 6 \text{ V} \\ -I_Z &= 3 \text{ mA.} \end{aligned}$$

El transistor empleado es el BC547C, que opera con las siguientes características de operación en la región de saturación:

$$\begin{aligned} -V_{CEsat} &= .2 \text{ V} \\ -V_{BEsat} &= .7 \text{ V.} \end{aligned}$$

Para que el led tenga una iluminación adecuada en el momento de un nivel bajo del voltaje de batería, se requiere que consuma un mínimo de corriente de 2 mA y 3 V de alimentación. Por lo tanto el valor de R6, para cumplir con estas características, se obtiene a partir de la ecuación 4.6.3, con  $V_1 = 7.2 \text{ V}$ ,  $V_{led} = 3 \text{ V}$  e  $I_{led} = 2 \text{ mA}$ .

$$\begin{aligned}
 R6 &= (V_1 - V_{cesat})/I_{csat} && \text{ec. 4.6.3} \\
 &= (7.2 \text{ V} - 3\text{V})/2 \text{ mA} \\
 &= 2.1 \text{ k}\Omega
 \end{aligned}$$

por tanto el valor comercial de  $R6 = 2.2 \text{ k}\Omega$ .

De la malla de colector del circuito se obtendrá a continuación la corriente de colector de saturación.

$$V_1 = R6 I_{csat} + V_{cesat} \quad \text{ec. 4.6.4}$$

despejando  $I_{csat}$  de la ecuación anterior y sustituyendo valores:

$$\begin{aligned}
 I_{csat} &= (V_1 - V_{cesat})/R6 \\
 &= (7.2 \text{ V} - .2 \text{ V})/2.2 \text{ k}\Omega \\
 &= 3.18 \text{ mA}
 \end{aligned}$$

La corriente  $I_{basat}$  se obtiene despejándola de la siguiente ecuación:

$$I_c = \beta I_b \quad \text{ec. 4.6.5}$$

entonces

$$\begin{aligned}
 I_b &= 3.18 \text{ mA}/400 \\
 I_b &= 7.95 \text{ }\mu\text{A}
 \end{aligned}$$

La resistencia  $R14$  se obtendrá de la malla de base, como se muestra a continuación.

$$V_1 = V_z + R14 \times I_{basat} + V_{besat} \quad \text{ec. 4.6.6}$$

despejando  $R14$ ,

$$\begin{aligned}
 R14 &= (V_1 - V_z - V_{besat})/I_{basat} \\
 &= (7.2 \text{ V} - 6 \text{ V} - .7 \text{ V})/7.95 \text{ }\mu\text{A} \\
 &= 62.9 \text{ k}\Omega
 \end{aligned}$$

Para asegurar el estado de saturación de  $T5$ , la Corriente de  $I_{basat}$  debe ser mayor a  $7.95 \text{ }\mu\text{A}$ , por lo tanto el valor comercial de  $R14 = 56 \text{ k}\Omega$ .

La resistencia  $R13$  debe asegurar la saturación del transistor  $T5$ , por lo tanto de la ecuación 4.6.7 se obtiene su valor:

$$\begin{aligned}
 R13 &= (V_1 - V_z)/(I_z - I_{basat}) && \text{ec. 4.6.7} \\
 &= (7.2 \text{ V} - 6 \text{ V})/(3 \text{ mA} - 8.93 \text{ }\mu\text{A}) \\
 &= 401 \text{ }\Omega
 \end{aligned}$$

por lo tanto el valor comercial de  $R13 = 390 \text{ }\Omega$ .

#### 4.6.4 CIRCUITO DE ALIMENTACION.

Para realizar la energización de los circuitos que componen la TCI se emplea una batería recargable de níquel-cadmio, cuyas características son las siguientes:

- Voltaje nominal,  $V = 7.2 \text{ V}$
- Capacidad nominal (por 5 horas),  $I = 500 \text{ mA/h.}$
- Carga estándar  $I = 50 \text{ mA}$ ,  $T = 14 \text{ hrs.}$ ,  $V_{\text{max}} = 1.5 \text{ V.}$
- De 500 a 1000 ciclos de carga y descarga.

Para satisfacer las necesidades de los circuitos se requieren de dos niveles de voltaje: 5 V y 7.2 V, el voltaje de 5 V lo obtenemos empleando el regulador de voltaje CI MC78L05C.

Con el objeto de proteger a los circuitos de la TCI de posibles sobrecorrientes, se incluirá un fusible cuyas características se calculan a continuación:

El consumo total de corriente se calcula a partir de los siguientes datos:

CIRCUITO	VOLTAJE (V)	CORRIENTE (mA)
TRANSMISOR	7.2	6.4
RECEPTOR	5.0	5.7
MICROCONTROLADOR	5.0	27.7
MODEM	5.0	5.5
AMPLIFICADOR DE BAJA FRECUENCIA	5.0	3.1
INDICADOR DE NIVEL DE BATERIA	5.0	3.1
DISPLAY	5.0	15
ILUMINACION DE TECLADO	7.2	20
ILUMINACION DE DISPLAY	7.2	3.7

Por lo tanto el consumo total de corriente es  $I_{\text{tot}} = 90.2 \text{ mA}$ . De acuerdo a las necesidades del sistema, el fusible comercial empleado tendrá las siguientes características:

- Tipo europeo.
- $I = .1 \text{ A}$
- $V = 250 \text{ V}$

#### 4.7 INTEGRACION DE LOS CIRCUITOS QUE FORMAN LA TCI.

DIAGRAMA ELECTRICO DE LA TCI. La conexión de los circuitos que forman la TCI, se muestra en la figura 4.7.1.

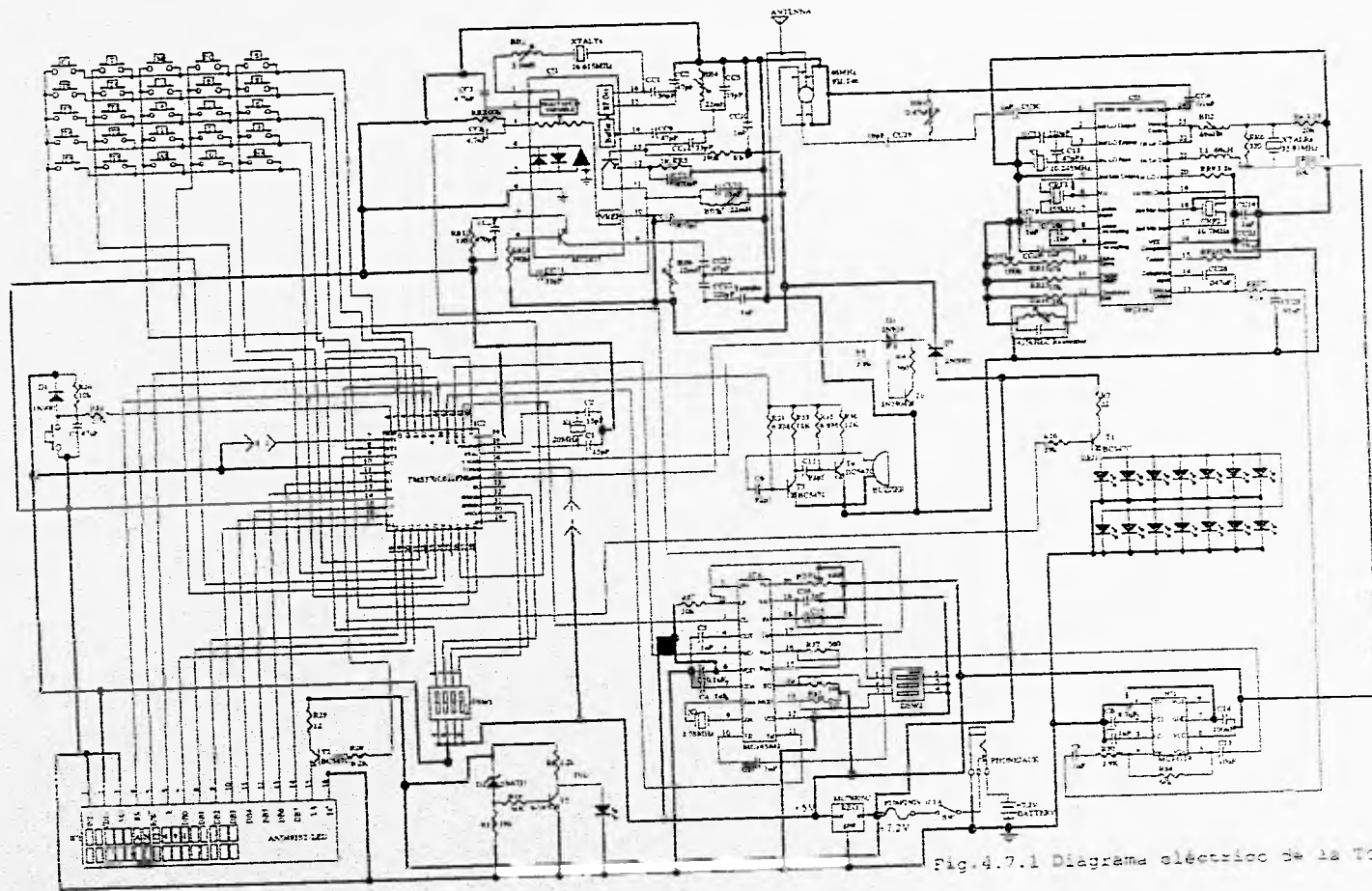


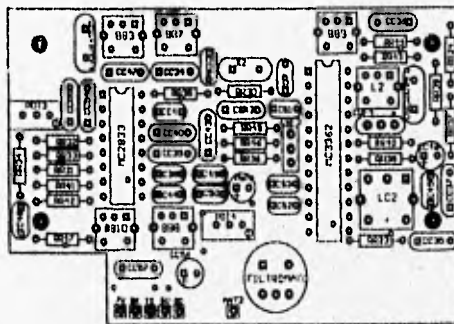
Fig. 4.7.1 Diagrama eléctrico de la VCI.

A continuación se muestra la lista de costos y materiales de la TCI.

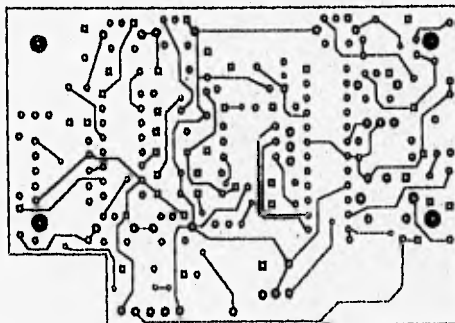
Número	Cantidad	Parte	Abreviaturas	Costo U. N\$	Total N\$
1	1	TMS370C622FHL	IC2	87.15	87.15
2	7	.1uF	C3, C4, C8, C10, C11, C15, C16	0.37	2.59
3	1	.01uF	CC4	0.23	0.23
4	4	.1uF	CC18, CC19, CC20, CC24	0.37	1.40
5	1	.01uF	CC28	0.23	0.23
6	3	.22ml	BB4, BB5, BB6	20.47	61.41
7	1	.47uF	C5	0.32	0.32
8	1	.047uF	CC26	0.16	0.16
9	2	.68uH	BB2, L1	20.47	40.94
10	1	.47uH	BB10	20.47	20.47
11	1	1N914	D3	0.22	0.22
12	1	1N4001	D1	0.14	0.14
13	1	1N4735	D2	0.70	0.70
14	2	1K	R13, R5	0.08	0.16
15	2	1nF	CC10, CC30	0.20	0.40
16	2	1uF	C7, CC22	0.39	0.78
17	1	2.2K	R6	0.08	0.08
18	1	2.7K	R32	0.20	0.20
19	1	2N3904	T6	0.45	0.45
20	1	2N5061	T7	1.90	1.90
21	1	3.3K	RR9	0.20	0.20
22	1	3.3mH	BB1	20.47	20.47
23	2	3.9K	R30, R40	0.20	0.40
24	1	3.58MHz	X2	7.77	7.77
25	2	4.7nF	CC3, CC6	0.23	0.46
26	1	4.7uF	C6	0.32	0.32
27	2	6.8M	R31, R35	0.08	0.16
28	2	8.2K	R28, R17	0.08	0.16
29	1	10.7MHz	CRF2	7.16	7.16
30	1	10.245MHz	X1	7.42	7.42
31	6	10K	R26, R27, R38, R39, RR14, RR15	0.08	0.48
32	1	10uF	C13	0.39	0.39
33	1	12	R29	0.08	0.08
34	2	12K	R33, R36	0.08	0.16
35	2	15pF	C1, C2	0.14	0.28
36	1	16.615MHz	XTALTx	7.42	7.42
37	1	18pF	CC29	0.14	0.14
38	4	10K	RR1, RR19, RR20, RR7	0.08	0.32
39	1	22	R7	0.08	0.08
40	4	33pF	CC5, CC12, CC16, CC23	0.15	0.60
41	1	35.93MHz	XTALRx	7.42	7.42
42	12	39K	R16, RR16	0.08	0.16
43	1	ANTENA		25.00	25.00
44	2	47K	R14, RR13	0.08	0.16
45	4	47pF	CC2, CC9, CC13, CC25	0.45	1.80
46	1	56pF	CC1	0.25	0.25
47	1	68K	R34	0.08	0.08
48	2	82pF	C9, C12	1.00	2.00
49	2	100K	POT1, POT2	6.63	13.26
50	1	PUSHBUTTON		1.63	1.63
51	1	100uF	C14	0.42	0.42
52	1	120pF	CC11	0.48	0.48
53	1	220pF	CC27	0.13	0.13

54	2	330	RR6, RR12	0.08	0.16
55	2	390K	RR4, RR19	0.08	0.16
56	1	455KHz	CRF1	6.88	6.88
57	3	470pF	CC14, CC17, CC21	0.13	0.39
58	2	560	R37, R41	0.08	0.16
59	1	AND491ST-LED	IC1	302.40	302.40
60	1	BATERIA	+7.2V	70.00	70.00
61	5	BC547C	T1, T2, T3, T4, T5	0.42	2.10
62	1	46MHZ	FIL746	13.23	13.23
63	1	250V/0.1A	FUSE	0.26	0.26
64	1	MC70L05C	REG	2.70	2.70
65	1	455KHZ	LC RESONATOR	18.53	18.53
66	1	MC2833	CI1	12.64	12.64
67	1	NC3362	CI2	16.56	16.56
68	1	MC34119	IC3	5.60	5.60
69	1	MC145442	IC4	45.15	45.15
70	1	1uF	TANTALIO	1.60	1.60
71	1	100K	RR2	0.08	0.08
T O T A L					825.87

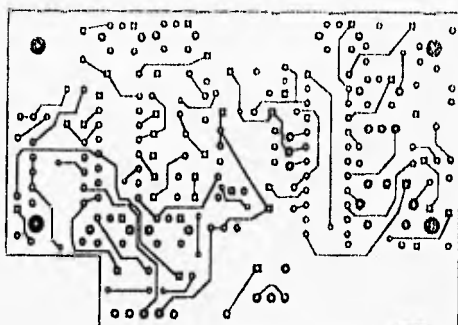
**CIRCUITO IMPRESO DE LA ETAPA DE COMUNICACION DE LA TCI.** La tarjeta del circuito impreso de la TCI mostrada en la figura 4.7.2, contiene los circuitos de transmisión y recepción, y sus componentes externos.



a) Vista de componentes.



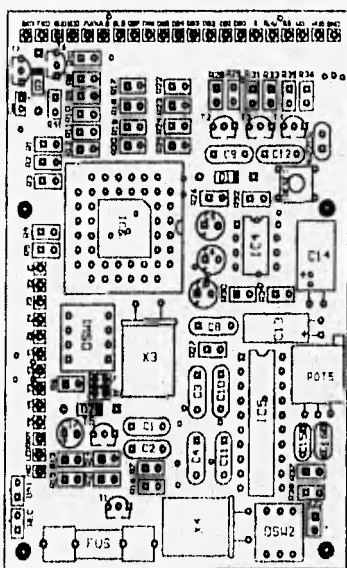
b) Vista superior.



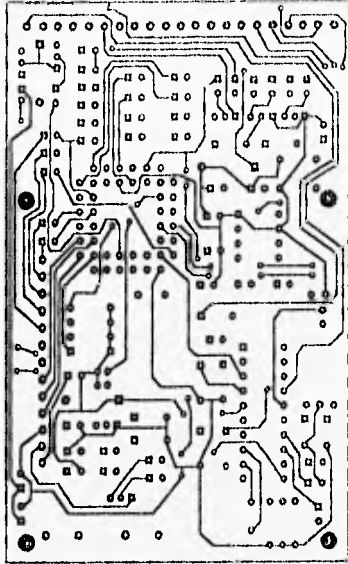
c) Vista interior.

Fig. 4.7.2 Tarjeta del circuito impreso de la comunicacion de la TCI.

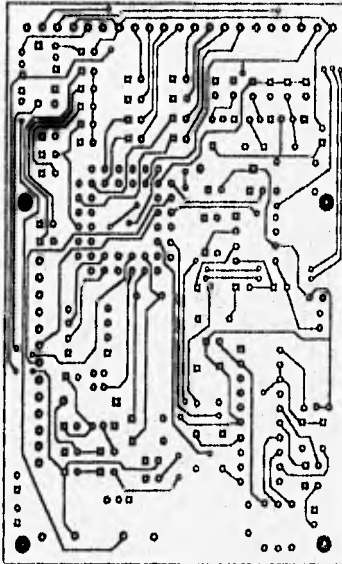
**CIRCUITO IMPRESO DEL MICROCONTROLADOR DE LA TCI.** Esta tarjeta contiene, al MC, al modem, al amplificador para la señal de baja frecuencia, los dispositivos de encendido y apagado de alimentación y a los circuitos de apoyo.



a) Vista de componentes.



b) Vista superior.

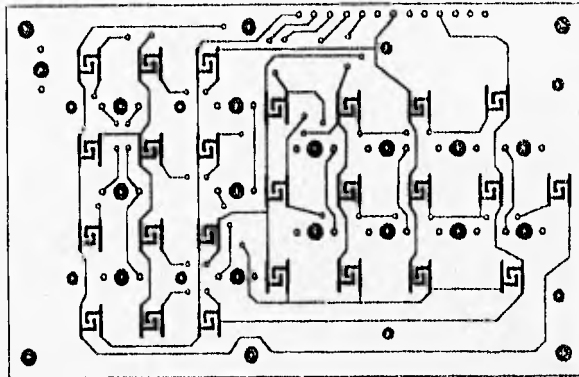


c) Vista inferior.

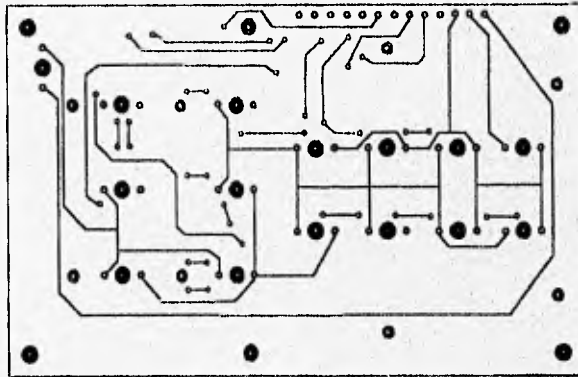
Fig. 4.7.3 Tarjeta del circuito impreso del MC de la TCI.



CIRCUITO IMPRESO DEL TECLADO DE LA TCI. La figura 4.7.4 muestra la distribución de las terminales normalmente abiertas, para el teclado de la TCI. También muestra la distribución de los leds que iluminan al teclado, y la localización del led indicador de nivel de batería.



a) Vista superior.



b) Vista inferior.

Fig. 4.7.4 Tarjeta del circuito impreso del teclado de la TCI.

CAPITULO V

DISEÑO DEL  
MODULO CENTRAL

En este capítulo se explica el desarrollo del diseño del Módulo Central, así como la descripción de la función de los circuitos que lo constituyen. El Módulo Central se divide en dos etapas Módulo de Comunicaciones de la Interfaz (MCI) e Interfaz de la PC (IPC). También se presenta el diagrama eléctrico de los circuitos de las dos etapas que forman el Módulo Central, el listado de componentes y los esquemas de las tarjetas de circuitos impresos para su construcción.

### 5.1 CIRCUITOS DEL MODULO DE COMUNICACIONES DE LA INTERFAZ, MCI.

Esta etapa se encarga de la comunicación vía RF del Módulo Central, y consta de tres circuitos integrados, un circuito transmisor de RF, un circuito receptor de RF y un circuito amplificador de baja frecuencia. También cuenta con una etapa de acondicionamiento de voltajes de alimentación.

Los circuitos integrados de transmisión y recepción, como en el caso de la TCI, son el CI MC2833 y el CI MC3362 respectivamente. Los componentes externos de ambos circuitos integrados, son iguales a los utilizados en la TCI, al igual que la configuración de las etapas que integran estos circuitos. Las frecuencias de operación de los cristales piezoeléctricos, se determinan de la misma forma que para el caso de la TCI. Las frecuencias de operación de el Módulo Central son 46.63 MHz para la transmisión y de 49.845 MHz para la recepción. Para el caso del circuito integrado de transmisión se utiliza un cristal de 15.5433 MHz, y para el circuito integrado de recepción un cristal de 39.145 MHz.

El otro circuito integrado en el MCI, que permanece sin cambio en su configuración eléctrica y en sus componentes externos es el CI MC34119 que como se mencionó, sirve para amplificar la señal entregada por el circuito receptor.

En la etapa de filtrado de las señales de RF, se emplea el filtro KAF-46NRME a la salida del circuito de transmisión. También se emplea un filtro 7KLT-335BBV-0224 de la marca Toko, entre la antena y la entrada del circuito integrado de recepción, para limitar al máximo las señales de interferencia en la recepción. Las hojas de especificaciones de los filtros se encuentran el apéndice A.

Los circuitos de apoyo con los que cuenta el MCI, son dos reguladores de voltaje, el CI MC78L08C y el CI MC78L05C de +8 V y +5 V respectivamente para la alimentación del MCI; además de un circuito indicador de polarización mediante un led.

### 5.2 INTEGRACION DE LOS CIRCUITOS QUE FORMAN EL MCI.

**DIAGRAMA ELECTRICO DEL MCI.** La conexión de los circuitos que forman el MCI, se muestra en la figura 5.2.1.

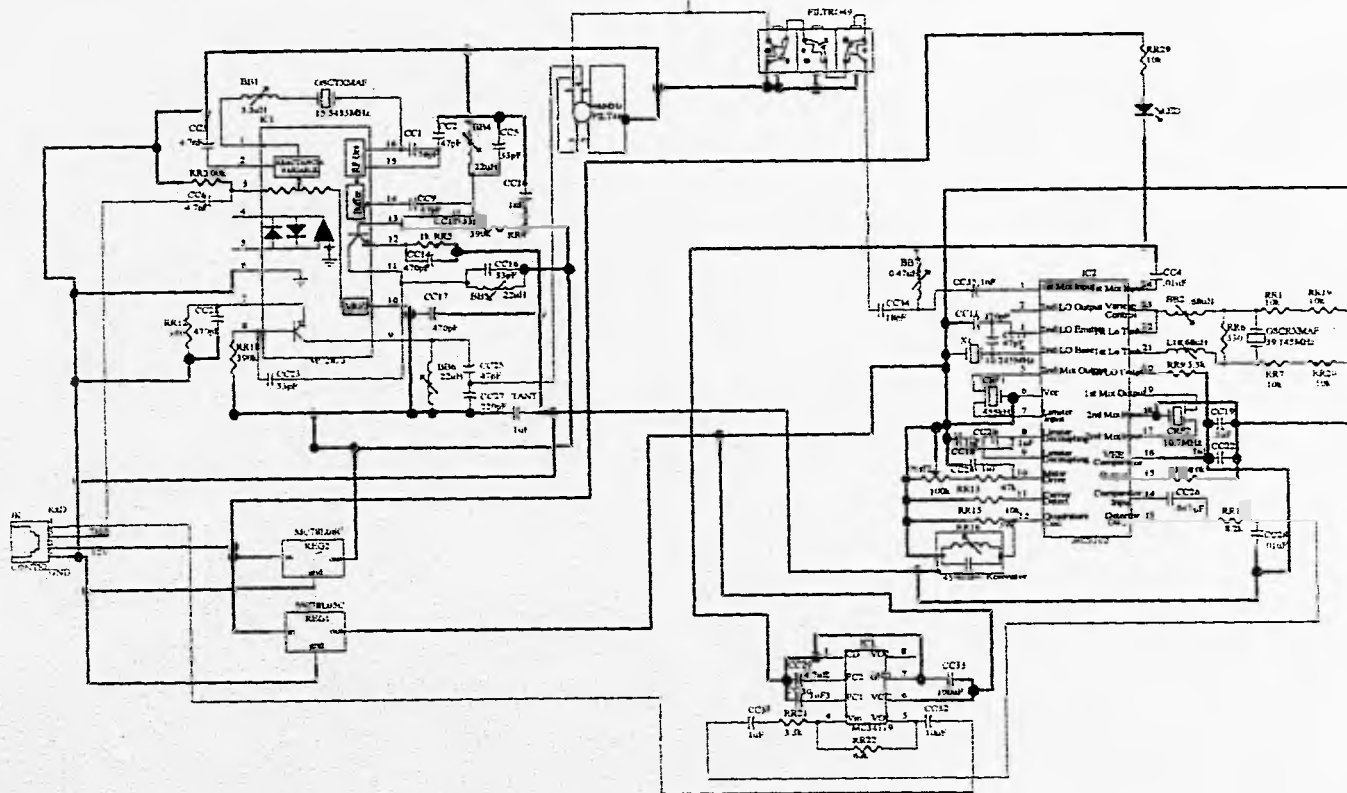
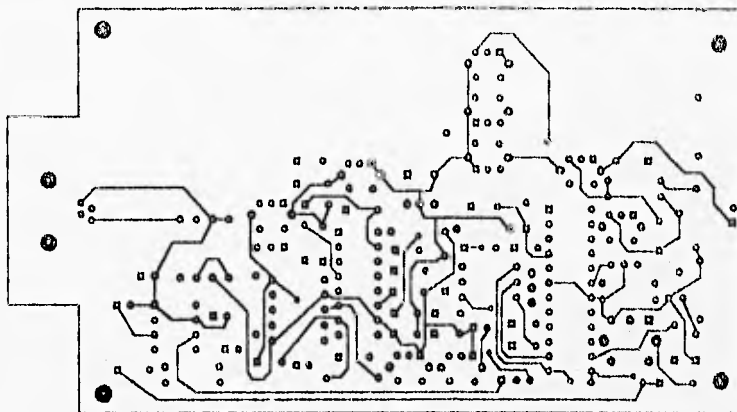


Fig. 5.2.1 Diagrama eléctrico del MCI.

A continuación se muestra la lista de costos y materiales del MCI.

Número	Cantidad	Parte	Abreviatura	Costo U. N\$	Total N\$
1	1	.01uF	CC4	0.23	0.23
2	4	.1uF	CC18, CC19, CC20, CC24	0.37	1.48
3	1	.01uF	CC28	0.23	0.23
4	1	.1uF	CC31	0.37	0.37
5	3	.22uH	BB4, BB5, BB6	20.47	61.41
6	1	.047uF	CC26	0.16	0.16
7	1	ANTENA		25.00	25.00
8	2	.68uH	BB2, L1	20.47	40.94
9	1	.47uH	BB7	20.47	20.47
10	2	1K	RR5, RR11	0.08	0.16
11	2	1nF	CC10, CC35	0.20	0.40
12	3	1uF	CC15, CC22, CC30	0.39	1.17
13	1	2.7K	RR6	0.20	0.20
14	2	3.3K	RR9, RR21	0.20	0.40
15	1	3.3uH	BB1	20.47	20.47
16	1	4.7K	RR10	0.08	0.08
17	2	4.7nF	CC3, CC6	0.23	0.46
18	1	4.7uF	CC29	0.32	0.32
19	1	10K	RR29	0.08	0.08
20	1	6.2K	RR17	0.08	0.08
21	1	10.7MHz	CRF2	7.16	7.16
22	1	10.245MHz	X1	7.42	7.42
23	5	10K	RR1, RR7, RR14, RR19, RR20	0.08	0.40
24	1	10uF	CC32	0.39	0.39
25	1	15.5433MHz	OSCTXMAF	7.42	7.42
26	1	18pF	CC34	0.14	0.14
27	4	33pF	CC5, CC12, CC16, CC23	0.15	0.60
28	1	39.145MHz	OSCRXMAF	7.42	7.42
29	1	39K	RR16	0.08	0.08
30	1	47K	RR13	0.08	0.08
31	4	47pF	CC2, CC9, CC13, CC25	0.45	1.80
32	1	56pF	CC1	0.25	0.25
33	1	68K	RR22	0.08	0.08
34	1	100K	POT2	6.63	6.63
35	1	100uF	CC33	0.42	0.42
36	1	120K	RR3	0.08	0.08
37	1	120pF	CC11	0.40	0.40
38	1	220pF	CC27	0.13	0.13
39	2	330	RR6, RR12	0.08	0.16
40	2	390K	RR4, RR18	0.08	0.16
41	1	455KHz	CRF1	6.08	6.08
42	3	470pF	CC14, CC17, CC21	0.13	0.39
43	1	MC78L05C	REG1	2.70	2.70
44	1	MC78L08C	REG2	3.00	3.00
45	1	46MHz	FILT46	13.23	13.23
46	1	49MHz	FILT49	58.55	58.55
47	1	455KHz	LC RESONATOR	10.53	10.53
48	1	MC2833	IC1	12.64	12.64
49	1	MC3362	IC2	16.56	16.56
50	1	MC34119	IC3	5.60	5.60
51	1	JACK TEL.		17.39	17.39
52	1	1uF	TANTALIO	1.60	1.60
T O T A L					365.12





c) Vista inferior.

Fig. 5.2.2 Tarjeta del circuito impreso del MCI.

### 5.3 CIRCUITOS DE LA INTERFAZ DE LA PC, IPC.

La interfaz de comunicación de la PC se conecta a un slot dentro de la PC y tiene como finalidad realizar las siguientes funciones:

- Modular y demodular la información entre el MC y la MCI.
- Administrar la información entre las TCIs y la PC.
- Acondicionar la información en el formato requerido por el bus de la pc.

En los siguientes puntos se explica el diseño de los módulos que forman la interfaz de la pc.

#### 5.3.1 CIRCUITO DEL MODEM.

El modem empleado al igual que en la TCI, es el CI MC145442 de Motorola, cuyas características ya se mencionaron. En la fig. 5.4.1, se muestra el diagrama eléctrico del CI del modem en la IPC.

El modem opera en el modo origen, donde los pines 2, 13, y 14 tienen conectados los niveles 0, 5, y 0 V, respectivamente en sus terminales. Las frecuencias de transmisión y recepción se muestran en la tabla 5.3.1.

Tabla 5.3.1

Dato	Transmisión	Recepción
Espacio	1180 Hz	1850 Hz
Marca	980 Hz	1650 Hz

### 5.3.2 CIRCUITOS DE INTERFAZ MICROCONTROLADOR-BUS-PC.

**MICROCONTROLADOR.** El microcontrolador MC utilizado para el diseño es el mismo que el empleado en la TCI, las especificaciones y requerimientos para su operación fueron tratados en el capítulo 4. Las funciones del microcontrolador en el Módulo Central son:

-Administrar el flujo de la información entre las TCIs y la PC, es decir, en forma secuencial solicitará a las TCIs la información que se transmitirá a la PC, ya que como se mencionó las TCIs no deben transmitir información simultáneamente al Módulo Central.

-Se comunicará con el bus de la PC manipulando las señales de control del mismo.

**COMUNICACION CON LA PC.** La comunicación con la PC, se realiza como se mencionó, accedendo el bus de datos, direcciones y control disponible en un slot de la PC, cuya asignación de terminales se describió en el capítulo 1.

Para el diseño de esta etapa, se deben considerar los siguientes parámetros:

-Se realizarán operaciones de lectura y escritura en el bus.

-La señal de interrupción así como la dirección que se asigna a la tarjeta deben ser configurables con la finalidad de no utilizar las asignadas a otro dispositivo periférico.

-El MCI se alimenta mediante los voltajes disponibles en el slot.

Con el objeto de cumplir los parámetros anteriores se emplean las siguientes señales del bus:

-GND (B1)

-+5V (B3)

-IRQ2 (B4)

-IRQ3 (B25)

-IRQ4 (B24)

ESTA TESIS NO DEBE  
SALIR DE LA BIBLIOTECA



- IRQ7 (B21)
- I/O R (B14)
- I/O W (B13)
- DO-D7 (A2-A9)
- AO-A9 (A22-A31)
- AEN (A12)

Para que estas señales puedan ser interpretadas por el microcontrolador se requieren dos etapas: un *buffer* para las líneas de datos, y una etapa de multiplexaje que maneje las líneas de direcciones y control, estas etapas se describen a continuación.

**BUFFER.** Como ya se mencionó, el *bus* de datos de la *PC* debe ser ocupado únicamente por el periférico accesado por la *PC*. Esto obliga a incluir un dispositivo que aisle el *bus* de datos de la *PC*, del microcontrolador. El dispositivo empleado para realizar esta función, es el circuito integrado 74HC245, el cual es un *buffer* octal bidireccional con salida de tres estados.

**CODIFICADOR.** Para saber el momento en el que el *bus* de la *PC* lee o escribe datos, es necesario que el microcontrolador monitoree las líneas de direcciones y control, las cuales indicarán qué periférico se está accedendo y qué tipo de operación se realiza. Debido a que el número total de líneas de dirección y control es mayor que las disponibles en el microcontrolador, se implementa una etapa de codificación, la cual minimiza el número de dichas líneas para que puedan ser manejadas por el microcontrolador. Dicha codificación consta de un demultiplexor y un conjunto de compuertas lógicas.

La descripción del diseño de las etapas mencionadas se presenta a continuación.

**DISEÑO DE LAS ETAPAS DEL CODIFICADOR Y BUFFER.** La comunicación entre el microcontrolador y el *bus* de la *PC* se divide en dos partes, transmisión del microcontrolador al *bus* de la *PC*, y transmisión de el *bus* de la *PC* al microcontrolador.

Para realizar una transmisión del microcontrolador al *bus* de la *PC* se deben cumplir las siguientes características:

- El microcontrolador debe activar una línea de interrupción, y mantenerla hasta que la *PC* reconozca la interrupción y lea el dato.

- Cuando el *bus* lee el dato:

Pone la dirección del dispositivo asociado a la interrupción en las líneas de direcciones.

La señal AEN tiene nivel de 0 V.

La señal  $\overline{I/OR}$  cambia su nivel de 5 V a 0 V, para avisar a los periféricos que se va a leer un dato del bus.

El dato colocado en el bus es leído.

Para realizar una transmisión del bus de la PC al microcontrolador se deben cumplir las siguientes características:

-Cuando el bus envía un dato a algún periférico:

Se coloca en las líneas de direcciones, la dirección del periférico al que se envía el dato.

La línea AEN tiene un nivel de 0 V.

La señal  $\overline{I/OW}$  cambia su nivel de 5 V a 0 V, para avisar al dispositivo que lea el dato colocado en el bus.

Codificando las líneas de direcciones del bus de la PC se obtiene una señal, la cual indicará cuándo el bus envía la dirección que accesa al microcontrolador.

Las direcciones disponibles para tarjetas en los slots de una PC se muestran en la tabla 5.3.2.

Tabla 5.3.2

DIR	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9
2E0-736	0	0	0	0	0	1	1	1	0	1
2C0-704	0	0	0	0	0	0	1	1	0	1
2A0-672	0	0	0	0	0	1	0	1	0	1
280-640	0	0	0	0	0	0	0	1	0	1
260-608	0	0	0	0	0	1	1	0	0	1
240-576	0	0	0	0	0	0	1	0	0	1
220-544	0	0	0	0	0	1	0	0	0	1
200-512	0	0	0	0	0	0	0	0	0	1

Para su codificación, las líneas de direcciones se dividen en dos grupos. El primer grupo corresponde a las direcciones de A5 a A9, con las cuales se obtendrá un nivel de 0 V cuando se envíe la dirección para acceder al microcontrolador. De esta forma se tendrá el mismo nivel de voltaje que las líneas de A0 a A4. Se requiere que la dirección sea configurable externamente, por lo tanto se empleará un decodificador/demultiplexor en conjunto con un dispositivo interruptor, para seleccionar alguna dirección que no sea empleada por otro periférico.

Se incluirá en esta etapa la señal AEN para asegurar que se está accediendo un periférico conectado en algún slot, el nivel de voltaje en la terminal AEN al presentarse el caso anterior, es de 0 V. En la tabla 5.3.3 se muestran las señales aplicadas al decodificador/demultiplexor.

Tabla 5.3.3

AEN	A8	A9	A7	A6	A5	DIR
0	0	1	0	0	0	200
0	0	1	0	0	1	220
0	0	1	0	1	0	240
0	0	1	0	1	1	260
0	0	1	1	0	0	280
0	0	1	1	0	1	2A0
0	0	1	1	1	0	2C0
0	0	1	1	1	1	2E0

El decodificador empleado es el 74HC138 de la figura 5.3.1, el cual opera como un demultiplexor de 1 a 8.

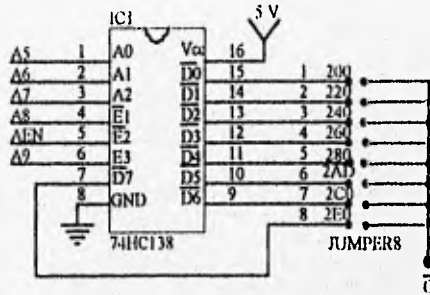


Fig. 5.3.1 Diagrama de conexión del 74HC138.

Con este circuito se logra la tabla de verdad mostrada en la tabla 5.3.4.

Tabla 5.3.4

A8	AEN	A9	A5	A6	A7	$\overline{\text{D0}}$	$\overline{\text{D1}}$	$\overline{\text{D2}}$	$\overline{\text{D3}}$	$\overline{\text{D4}}$	$\overline{\text{D5}}$	$\overline{\text{D6}}$	$\overline{\text{D7}}$
0	0	1	0	0	0	0	1	1	1	1	1	1	1
0	0	1	1	0	0	1	0	1	1	1	1	1	1
0	0	1	0	1	0	1	1	0	1	1	1	1	1
0	0	1	1	1	0	1	1	1	0	1	1	1	1
0	0	1	0	0	1	1	1	1	1	0	1	1	1
0	0	1	1	0	1	1	1	1	1	1	0	1	1
0	0	1	0	1	1	1	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	1	1	1	1	1	0

Por lo tanto cuando se mande al bus, la dirección que corresponde a la línea configurada externamente se tendrá en una salida  $\overline{\text{Dn}}$  del CI 74HC138 un nivel de 0 V, que indicará que la dirección del periférico ha sido enviada. La dirección deseada es configurada por medio de un jumper.

El segundo grupo de direcciones de A0 a A4, tendrán un nivel de 0 V, cuando la dirección enviada sea la deseada, por lo tanto, para lograr un nivel de 5 V al presentarse esta situación, se emplea el circuito lógico de la figura 5.3.2.

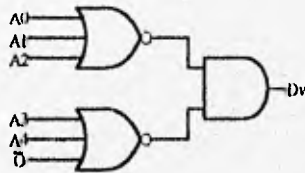


Fig. 5.3.2 Circuito lógico para el segundo grupo de direcciones.

Para saber el momento en que la PC va a realizar una operación de lectura o escritura en el microcontrolador, se empleará la salida del circuito mostrado anteriormente, en conjunto con las señales I/OR e I/OW, como se muestra en la figura 5.3.3.

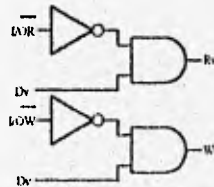


Fig. 5.3.3 Circuito lógico de detección de lectura y escritura de la PC.

Con estas señales se obtendrá un nivel de 5 V, cuando se realice una operación de lectura Rv o escritura Wv, en el microcontrolador.

Para el manejo de las líneas de datos, se empleará un *buffer* que maneje un estado de alta impedancia cuando el microcontrolador no tenga comunicación con el bus de la pc. Para esto se emplea el circuito integrado 74HC245, el cual tiene la tabla de verdad mostrada en la tabla 5.3.5 para su manejo.

Tabla 5.3.5

$\overline{\text{DIR}}$	$\overline{\text{G}}$	FUN
0	0	B→A
0	1	Z
1	0	A→B
1	1	Z

El circuito integrado es conectado al microcontrolador y al bus de la PC como se muestra en la figura 5.3.4.

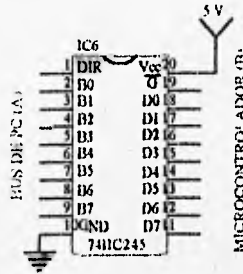


Fig. 5.3.4 Diagrama de conexión para el 74HC245.

Para controlar la dirección del flujo de datos, entre el bus de la PC y el microcontrolador, la terminal  $\overline{\text{DIR}}$  debe tener un nivel de 0 V cuando el bus de la pc lee datos, y un nivel de 5 V cuando el bus de la PC escribe datos. La señal Rv, tiene la lógica inversa a la descrita anteriormente, por lo tanto, se empleará esta señal negada para controlar el flujo de datos, como se muestra en la figura 5.3.5.

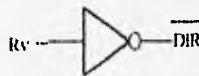


Fig. 5.3.5 Circuito lógico para controlar el sentido de flujo de datos.

El dispositivo *buffer* sólo debe de salir del estado de alta impedancia, cuando se realice un operación de lectura o escritura en el microcontrolador, por lo tanto el control del estado de alta impedancia del circuito se realiza mediante el circuito de la figura 5.3.6.

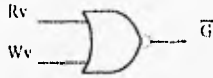


Fig. 5.3.6 Circuito lógico para el control de estado de alta impedancia.

El tiempo en que la PC avisa una operación de lectura o escritura del bus es menor al tiempo en que el microcontrolador podría detectar estas situaciones, es por eso que se requiere del uso de un circuito de memoria que retenga la información al microcontrolador, el tiempo suficiente para que detecte estas situaciones. Por lo tanto, para controlar el circuito de memoria se utilizan los flancos de subida de las líneas  $R_v$  y  $W_v$ . El circuito integrado empleado es el CI 74HC74 de tres estados, el cual tiene la tabla de verdad mostrada en la tabla 5.3.5.

Tabla 5.3.5

SD	$\overline{CD}$	Cp	D	Q
H	L	X	X	L
H	H		H	H
H	H		L	L
H	H	L	X	Q0

Como se mencionó, el microcontrolador deberá cambiar el nivel de voltaje de un línea de interrupción, la cual será seleccionada mediante un *jumper* con el objeto de utilizar una interrupción no ocupada por otro dispositivo periférico, figura 5.3.7.



Fig. 5.3.7 Circuito de configuración para interrupción.

Para eliminar el ruido en la línea de alimentación del bus se emplea una red de 3 capacitores en paralelo con las líneas de alimentación, figura 5.3.8.

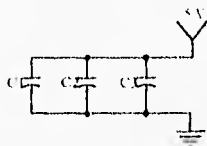


Fig. 5.3.8 Circuito para eliminar ruido en la línea de alimentación del bus.

C1=1 uF cerámico.  
C2= 1 uF tantalio.  
C3= 220 uF electrolítico.

#### 5.4 INTEGRACION DE LOS CIRCUITOS QUE FORMAN LA IPC.

**DIAGRAMA ELECTRICO DE LA IPC.** La conexión de los circuitos que forman la IPC, se muestra en la figura 5.4.1.

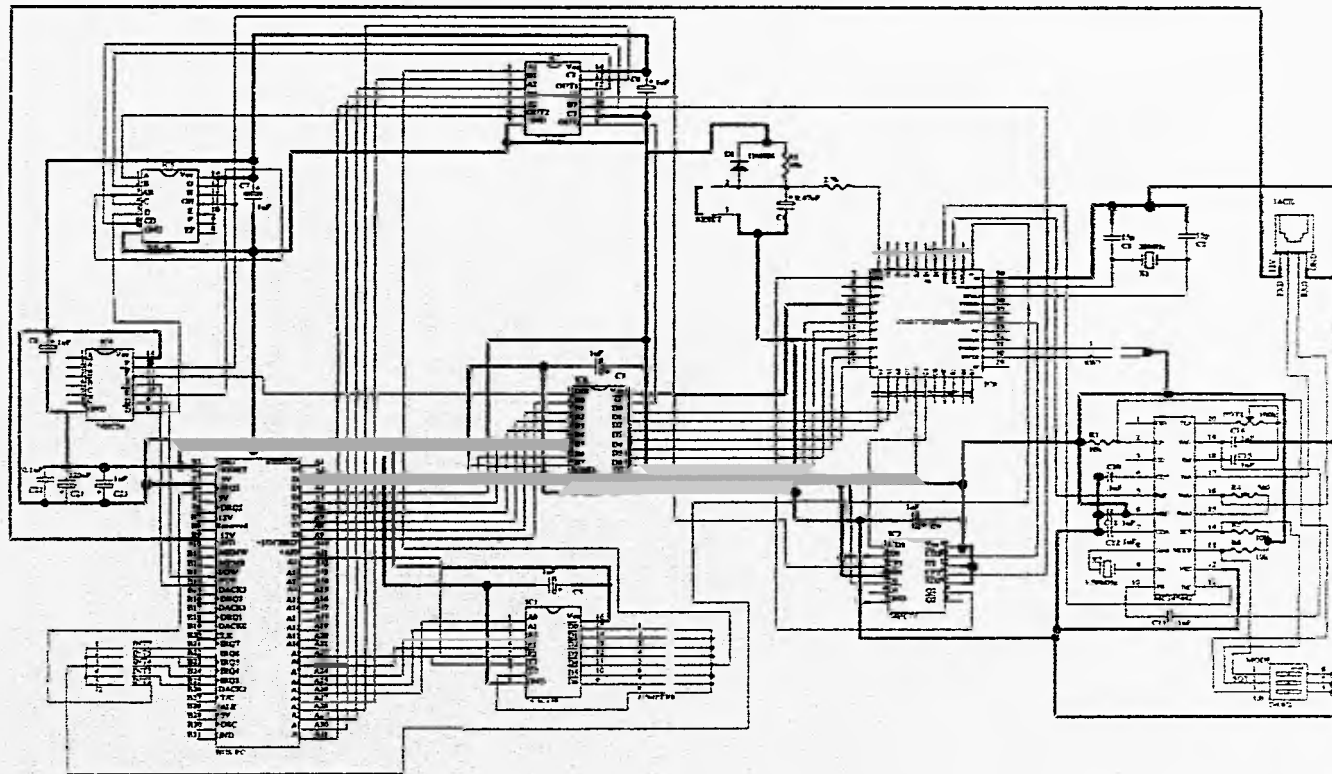
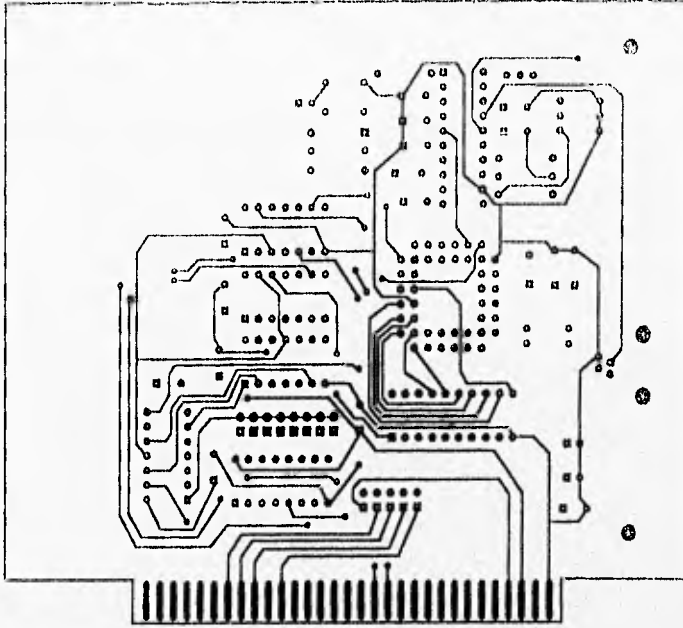


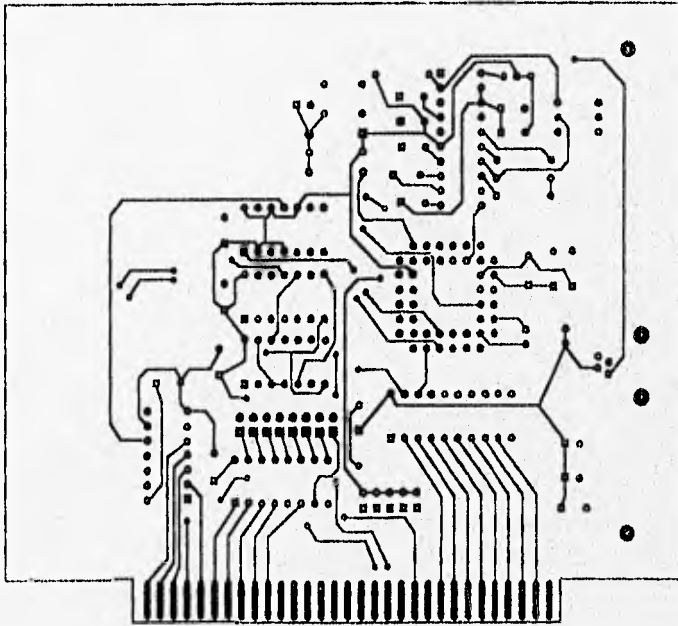
Fig. 5.4.1 Diagrama eléctrico de la IPC.







b) Vista superior.



c) Vista inferior.

Fig. 5.4.2 Tarjeta del circuito impreso de la comunicacion de la TCI.

A continuación se muestra la lista de costos y materiales adicionales del sistema.

Número	Cantidad	Parte	Costo U. N°	Total N°
1	1	TARJETA DE CTO. IMPRESO DE "RF DE TCI"	20.00	20.00
2	1	TARJETA DE CTO. IMPRESO DE "MC DE TCI"	20.00	20.00
3	1	TARJETA DE CTO. IMPRESO DE "TECLADO TCI"	20.00	20.00
4	1	TARJETA DE CTO. IMPRESO DE "MCI"	20.00	20.00
5	1	TARGETA DE CTO. IMPRESO DE "IPC"	20.00	20.00
6	2	BASE DE TMS370C622	14.90	29.80
7	1m	CABLE PLANO	3.50	3.50
8	1	CARCASA DEL MCI	56.52	56.52
9	1	CHASSIS MAF	10.00	10.00
10	2	JUMPERS	0.09	0.18
11	14	LED 3mm	0.45	6.30
12	2	LED 6mm	0.50	1.00
13	2	DIP-SWITCH 4	6.31	12.62
14	2	PARES DE PINES DE 8	0.26	0.52
15	10	TORNILLOS CON TUERCA	0.10	1.00
16	1	PORTA FUSIBLE	0.50	0.50
17	8	SOPORTES TARJETAS	0.67	5.36
18	1	SWITCH AS36S0100	20.00	20.00
19	1	TECLADO (KEY PAD)	10.00	10.00
T O T A L				257.00

CAPITULO VI

PROGRAMACION  
DEL SISTEMA

En este capítulo se desarrolla la programación necesaria para el funcionamiento, tanto de las TCI's como del Módulo Central.

La programación de los microcontroladores de las TCI's y de la interfaz de la PC se realiza en lenguaje de máquina, basándose en el conjunto de instrucciones de la familia TMS370 de Texas Instrument.

La programación de la PC del Módulo Central se efectúa en lenguaje C++, debido a que este lenguaje tiene gran variedad y riqueza de instrucciones. El lenguaje C++ permite la manipulación de bits, bytes y direcciones.

Este capítulo está dividido en tres partes: Programación del microcontrolador de las TCI's, programación del microcontrolador de la IPC y programación de la PC. El código fuente de los programas se anexan en el apéndice B.

## **6.1 PROGRAMACION DEL MICROCONTROLADOR DE LAS TERMINALES DE CAPTURA INALAMBRICAS, (TCI).**

Como se mencionó anteriormente el objetivo del sistema es eficientizar la captura de información en cadenas de restaurantes, en el sistema las TCI's serán el elemento mediante el cual se realiza la captura de información. Las TCI's tienen como dispositivo de control y procesamiento un microcontrolador, el cual tiene las siguientes funciones:

- Interpretar la información introducida mediante el teclado para su procesamiento.
- Enviar en el formato requerido por la pantalla, la información generada localmente en el teclado y la procedente del Módulo Central.
- Codificar la información transmitida al Módulo Central y decodificar la información procedente del mismo.
- Encender el circuito de transmisión cuando se envíe información al Módulo Central.
- Controlar el encendido y apagado de la iluminación del teclado y del display.
- Controlar el encendido y apagado del circuito de sonido.
- Realizar la lectura de identificador de las TCI's.

### **6.1.1 INFORMACION MANEJADA EN LA TCI.**

Las principales funciones que deben ser capturadas por el usuario (mesero) de la TCI son las siguientes:

- Alta y baja de mesero.
- Alta y baja de mesa.
- Alta y baja de ordenes.
- Borrado de información desplegada.
- Mensaje especial.
- Mandar total de cuenta a TCI.
- Mostrar mensaje que llega a la TCI.

Para que esta información sea capturada eficazmente se configuró el teclado y la pantalla de tal forma que las operaciones realizadas sean muy parecidas a las empleadas tradicionalmente en los restaurantes. Estas configuraciones se muestran en las figuras 6.1.1 pantalla y 6.1.2 teclado.

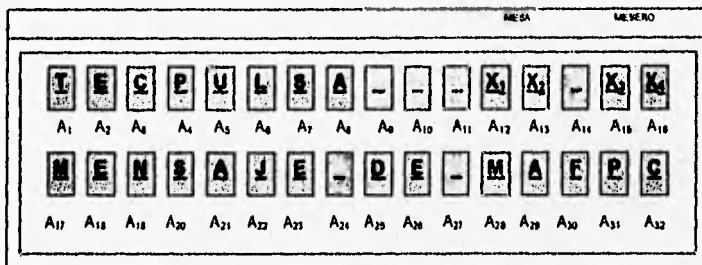


Fig. 6.1.1 Manejo de la pantalla de la TCI.

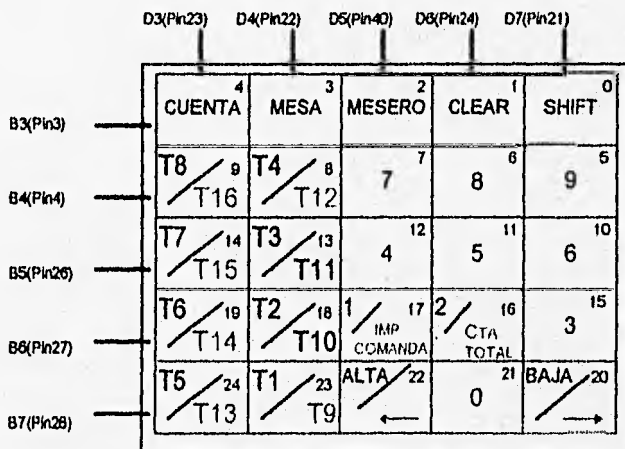


Fig. 6.1.2 Manejo de teclado de la TCI.

### SECUENCIA DE OPERACION.

A continuación se presenta la secuencia de teclas que deben ser pulsadas para realizar la captura de información, así como la información presentada en la pantalla, indicándose la posición de cada letra (A1 - A32) referida en la figura 6.1.1.

#### a) ALTA/BAJA DE MESERO

1° Oprimir tecla

2° Oprimir tecla  y/o

3° Oprimir tecla  /

Mensaje en pantalla: "Mesero#1#2" (A1-A8)  
"Alta" (A17-A20) / "Baja" (A17-A20)

Mensaje de confirmación: "Clave:" (A1-A6)

Oprimir teclas :

Confirmación : Clave : " \* \* \* \*" (A7- A10)

Mensaje en pantalla:

- Si fue BAJA o la clave que se marcó no es correcta: "Cheff System"  
(A19-A30)

- Si la clave fue correcta: "#1 #2" (A15- A16).  
"Alta mesero #1 #2" (A17- A30)

#### b) ALTA/BAJA DE MESA

1° Oprimir tecla

2° Oprimir tecla  y/o

3° Oprimir tecla  /

Mensaje en pantalla: "Mesa #1 #2" (A1 - A6)

"Alta" (A17-A20) / "Baja" (A17-A20)

Confirmación : Alta: "#1 #2" (A12-A13) "L" (A10)

Baja: " " (A12-A13)

"Alta /Baja de Mesa #1 #2"

c) ALTA / BAJA DE COMANDAS (PLATILLOS)

1° Oprimir tecla  y/o

2° Oprimir tecla  y/o

3° Oprimir tecla  y/o

4° Oprimir tecla  /

Mensaje en pantalla: "#1#2 Alta" (A1-A7)  
"Tabla #3 #4" (A17-A32)

Confirmación: "Alta #1#2" (A1-A8) ó "Baja #1 #2" (A1-A8)  
"Platillo" (A17-A31)

d) MENSAJES ESPECIALES

Estos deben aparecer en el área (A17- A31) y se envían del Módulo Central a la TCI.

e) MANDAR TOTAL DE CUENTA A TCI.

1) Oprimir tecla

2) Oprimir tecla

3) Oprimir tecla

Mensaje en pantalla: "Cuenta" (A1-A6)

"Mandar" (A17-A22)

Confirmación : "Cuenta = N\$ #### " (A17-A31)

6.1.2 PARAMETROS PARA LA PROGRAMACION DEL MICROCONTROLADOR DE LA TCI.

A continuación se presenta la descripción de los parámetros de la programación del microcontrolador de la TCI:

- El teclado se configuró de tal forma que quede una matriz de 5 x 5 en la cual se manejarán 5 columnas y 5 renglones las cuales se conectarán directamente al microcontrolador, esta configuración se muestra en la figura 6.1.2.
- Para el adecuado manejo de la información, la pantalla se divide en las áreas mostradas en la figura 6.1.3.



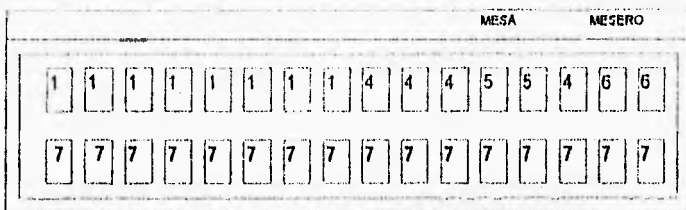


Fig. 6.1.3 Areas de información en la pantalla.

La información es presentada en las áreas de la pantalla de la siguiente forma:

- 1: Area de pantalla en la que se despliega la información introducida a través del teclado para enviarse al Módulo Central.
  - 4: Area de separación de mensajes la cual es dejada en blanco.
  - 5: Area de número de mesa dada de alta en la cual se pueden desplegar dos números decimales.
  - 6: Area de número de mesero dado de alta en la cual se pueden desplegar ya sea dos números decimales o dos letras de identificación del mesero.
  - 7: Area en la cual se despliegan los mensajes de respuesta enviados por el Módulo Central a la TCI; también se utilizan para desplegar las palabras "ALTA", "BAJA", "IMPRIMIR", etc., para confirmar peticiones al Módulo Central.
- La información manejada entre TCI y Módulo Central tienen el siguiente formato:
- a) La información introducida a través del teclado para ser enviada al Módulo Central siempre es de tres bytes y tiene el formato mostrado en la figura 6.1.4.

cantidad / # personas	comando	# de plato, mesa, mesero
1 Byte	1 Byte	1 Byte

Fig. 6.1.4 Formato de información de TCI a Módulo Central.

- b) La información del Módulo Central a las TCI's siempre es enviada en un formato de 16 bytes consecutivos como se muestra en la figura 6.1.5.

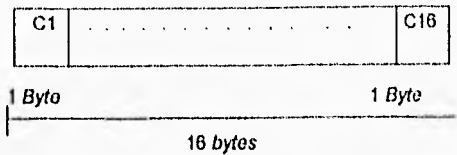


Fig. 6.1.5 Formato de información de Módulo Central a TCI.

- Para manejar la información producida por el teclado se manejan dos números, un número se obtiene a partir de la posición de cada tecla en el arreglo matricial con el cual identifica la tecla pulsada, a partir de este número se asigna otro diferente a cada tecla para facilitar el manejo de despliegado de caracteres así como de formación de mensajes enviados al Módulo Central.

La relación TECLA-NUMERO DE POSICION- NUMERO PARA PROCESAMIENTO (CLAVE) se muestra en la tabla 6.1.1.

TECLA	No. POSICION	No. CLAVE
SHIFT	0	217
CLEAR	1	174
MESERO	2	26
MESA	3	24
CUENTA	4	236
9	5	57
8	6	56
7	7	55
T4 / T12	8	34
T8 / T16	9	42
6	10	54
5	11	53
4	12	52
T3 / T11	13	32
T7 / T15	14	40
3 / CTA. TOTAL	15	51
2	16	50
1	17	49
T2 / T10	18	30
T6 / T14	19	38
BAJA / →	20	201
0	21	48
ALTA / ←	22	200
T1 / T8	23	28
T5 / T13	24	36

Tabla. 6.1.1 Identificadores de teclas.

- Para realizar la comunicación de la TCI con el Módulo Central se manejará un formato de tres bytes como se muestra a continuación:

- 1) El primer byte es un número que indica la cantidad de comandos (plátillos) que se pide o para indicar el número de comensal al que se le está tomando la orden, este número se denomina **CAR1MEN**.
- 2) El segundo byte denominado **CAR2MEN** es un comando que indica que función se debe realizar a petición del usuario de la TCI, la lista de estos comandos así como la forma de obtenerlos al operar diferentes combinaciones de teclas se da en la tabla 6.1.2.
- 3) El tercer byte **CAR3MEN** es una cantidad relacionada al comando antes mencionado por ejemplo número de mesa dada de alta, número de mesero dado de alta, etc.

COMANDO	No.DE C(d)	No.HEX(h)	FORMACION DE COMANDO
MANDAR CTA. TOT. A TCI	204	CC	$SHIFT(D9/217) * 3(33/51) = 204$
ALTA DE TABLA 9	208	DO	$\{SHIFT(D9/217) * (TABLA1(1C/28)) + ALTA (C8/200) = 208$
BAJA DE TABLA 9	209	D1	$\{SHIFT(D9/217) * TABLA1(1C/28) + BAJA (C9/201) = 209$
ALTA DE TABLA 10	210	D2	$\{SHIFT(D9/217) * TABLA2(1E/30) + ALTA (C8/200) = 210$
BAJA DE TABLA 10	211	D3	$\{SHIFT(D9/217) * TABLA2(1E/30) + BAJA (C9/201) = 211$
ALTA DE TABLA 11	212	D4	$\{SHIFT(D9/217) * TABLA3(20/32) + ALTA (C8/200) = 212$
BAJA DE TABLA 11	213	D5	$\{SHIFT(D9/217) * TABLA3(20/32) + BAJA (C9/201) = 213$
ALTA DE TABLA 12	214	D6	$\{SHIFT(D9/217) * TABLA4(22/34) + ALTA (28/200) = 214$
BAJA DE TABLA 12	215	D7	$\{SHIFT(D9/217) * TABLA4(22/34) + BAJA (C9/201) = 215$
ALTA DE TABLA 13	216	D8	$\{SHIFT(D9/217) * TABLA5(24/36) + ALTA (C8/200) = 216$
BAJA DE TABLA 13	217	D9	$\{SHIFT(D9/217) * TABLA5(24/36) + BAJA (C9/201) = 217$
ALTA DE TABLA 14	218	DA	$\{SHIFT(D9/217) * TABLA6(26/38) + ALTA (C8/200) = 218$
BAJA DE TABLA 14	219	DB	$\{SHIFT(D9/217) * TABLA6(26/38) + BAJA (C9/201) = 219$
ALTA DE TABLA 15	220	DC	$\{SHIFT(D9/217) * TABLA7(28/40) + ALTA (C8/200) = 220$
BAJA DE TABLA 15	221	DD	$\{SHIFT(D9/217) * TABLA7(28/40) + BAJA (C9/201) = 221$
LIBRE	222	DE	
LIBRE	223	DF	
ALTA DE MESA	224	EO	$ALTA (C8/200) + MESA (18/24) = 224$
BAJA DE MESA	225	E1	$BAJA (C9/201) + MESA (18/24) = 225$
ALTA DE MESERO	226	E2	$ALTA (C8/200) + MESERO (1A/28) = 226$
BAJA DE MESERO	227	E3	$BAJA (C9/201) + MESERO (1A/28) = 227$
ALTA TABLA 1	228	E4	$ALTA (C8/200) + TABLA 1 (1C/28) = 228$
BAJA TABLA 1	229	E5	$BAJA (C9/201) + TABLA 1 (1C/28) = 229$
ALTA TABLA 2	230	E6	$ALTA (C8/200) + TABLA2 (1E/30) = 230$
BAJA TABLA 2	231	E7	$BAJA (C9/201) + TABLA2 (1E/30) = 231$
ALTA TABLA 3	232	E8	$ALTA (C9/201) + TABLA 3 (20/32) = 232$
BAJA TABLA 3	233	E9	$BAJA (C9/201) + TABLA 3 (20/32) = 233$
ALTA TABLA 4	234	EA	$ALTA (C8/200) + TABLA 4 (22/34) = 234$
BAJA TABLA 4	235	EB	$BAJA (C9/201) + TABLA 4 (22/34) = 235$
ALTA TABLA 5	236	EC	$ALTA (C8/200) + TABLA 5 (24/36) = 236$
BAJA TABLA 5	237	ED	$BAJA (C9/201) + TABLA 5 (24/36) = 237$
ALTA TABLA 6	238	EE	$ALTA (C8/200) + TABLA 6 (26/38) = 238$
BAJA TABLA 6	239	EF	$BAJA (C9/201) + TABLA 6 (26/38) = 239$
ALTA TABLA 7	240	FO	$ALTA (C8/200) + TABLA 7 (28/40) = 240$
BAJA TABLA 7	241	F1	$BAJA (C9/201) + TABLA 7 (28/40) = 241$
ALTA TABLA 8	242	F2	$ALTA (C8/200) + TABLA 8 (2A/42) = 242$
BAJA TABLA 8	243	F3	$BAJA (C9/201) + TABLA 8 (2A/42) = 243$

Tabla 6.1.2. Comandos.

NOTA.- En los comandos anteriores los números entre paréntesis son números asignados a la tecla en formatos hexadecimal y decimal.

Las operaciones realizadas para formar el comando son:

\* = no hay operación aritmética, simplemente se asigna un número al combinarse la secuencia de teclas indicadas.

+ = se realiza la suma aritmética entre los dos números correspondientes a la tecla oprimida.

- Adicionalmente a los comandos listados anteriormente se emplean otros que serán sólo de control y se muestran en la tabla 6.1.3.

CLAVE	
164d	- Este número se transmite del µC del Módulo Central al µC de la TCI para avisarle que le va a mandar un mensaje especial al mesero (mensaje-bip). - Se mandará del µC de la TCI al µC del Módulo Central para confirmarle el mensaje-bip.
166d	- Cuando el Módulo Central pregunta si hay información al µC de la TCI manda este comando para indicar que no hay información para enviar. - Si el Módulo Central quiere mandar un mensaje al µC de la TCI contesta que no lo mande cuando está formando un mensaje.
167d	- Se envía del µC de la TCI al µC del Módulo Central para contestar si hay información para transmitir.
168d	- Lo envía el µC del Módulo Central al µC de la TCI para preguntar si tiene información para transmitir.
169d	- Es enviado mutuamente entre µC de la TCI y µC del Módulo Central para indicar que hubo un error en la comunicación y que vuelva a enviarla.

Tabla 6.1.3 Comandos adicionales de control.

- El formato de los mensajes enviados entre las TCI's y el Módulo Central debe ser el siguiente:

\*ALTA DE MESERO

TCI - Módulo Central

Módulo Central - TCI

CARIMEN Ø Ø	CARZMEN E2	CAR3MEN n1 n2	CO	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15
			A	f	l	a		M	e	s	e	r	o	P	N1	N2	C1	C2

El número E2(h) de CARZMEN es el comando de alta de mesero.

Los números n1 y n2 tienen formato hexadecimal y corresponde al número de mesero dado de alta. (por ejemplo para realizar la operación de alta de mesero 12, se maneja: n1n2=0C).

El mensaje del Módulo Central a la TCI debe ser manejado con las siguientes características:

- 1) Las letras deben ser enviadas con su equivalente en código ASCII (por ejemplo para enviar la letra A, se envía el número 41h).
- 2) Para enviar un espacio en blanco se manda el número 20h.
- 3) Los números N1 y N2 son el número de mesero dado de alta y deben ser enviados con el equivalente en ASCII del número decimal.
- 4) Los números C1 y C2 son la clave codificada del mesero y debe ser enviado en el siguiente formato:

$$\begin{array}{ccc}
 \frac{N_7 N_6 N_5 N_4 N_3 N_2 N_1 N_0}{C1} & & \frac{N_{15} N_{14} N_{13} N_{12} N_{11} N_{10} N_9 N_8}{C2} \\
 N_7 N_6 N_5 N_4 = NC1 & & N_3 N_2 N_1 N_0 = NC2 \\
 N_{15} N_{14} N_{13} N_{12} = NC3 & & N_{11} N_{10} N_9 N_8 = NC4
 \end{array}$$

Donde NC1, NC2, NC3 y NC4 los 4 números que debe teclear el mesero para que se pueda dar de alta. Los números deben ser enviados en el orden que son tecleados. (por ejemplo si la clave es 1728, los números enviados deben ser C1=17h y C2=28h).

**\*BAJA DE MESERO**

TCI - Módulo Central

Módulo Central - TCI

CARIMEN Ø Ø	CARIMEN E3	CARIMEN n1 n2	CO B	C1 a	C2 j	C3 a	C4	C5 M	C6 e	C7 s	C8 e	C9 r	C10 o	C11	C12	C13 N1	C14 N2	C15
----------------	---------------	------------------	---------	---------	---------	---------	----	---------	---------	---------	---------	---------	----------	-----	-----	-----------	-----------	-----

Los datos enviados de la TCI al Módulo Central tienen las mismas características que en ALTA DE MESERO enviando en E3 (comando de baja de mesero). La respuesta enviada del Módulo Central a la TCI, es el mensaje de baja de mesero.

Para dar de baja al mesero la TCI debe verificar si la primera letra es un caracter "A"(41h) en cuyo caso se da de alta al mesero en cualquier otro caso se da de baja al mesero. El caracter C15 debe ser (20h) para manejo de desplegado de mensajes.

**\*ALTA DE MESA**

TCI - Módulo Central

Módulo Central - TCI

CARIMEN Ø Ø	CARIMEN EØ	CARIMEN n1 n2	CO A	C1 l	C2 t	C3 a	C4	C5 d	C6 e	C7	C8 M	C9 e	C10 s	C11 a	C12	C13	C14 Ø	C15 l
----------------	---------------	------------------	---------	---------	---------	---------	----	---------	---------	----	---------	---------	----------	----------	-----	-----	----------	----------

Las características del mensaje de la TCI al Módulo Central y del Módulo Central a la TCI tienen las mismas características de ALTA DE MESERO con las excepciones del carácter 16(C15) en el cual para este caso se envía en el número 20h, y de CAR2MEN en el cual se envía el comando E0 de alta de mesa.

\*BAJA DE MESA

TCI - Módulo Central

Módulo Central - TCI

CAR1MEN Ø Ø	CAR2MEN E1	CAR3MEN n1 n2	CO B	C1 a	C2 j	C3 a	C4	C5 d	C6 e	C7	C8 M	C9 e	C10 s	C11 a	C12	C13 N1	C14 N2	C15
----------------	---------------	------------------	---------	---------	---------	---------	----	---------	---------	----	---------	---------	----------	----------	-----	-----------	-----------	-----

Las características de esta comunicación son similares a las de ALTA DE MESA.

\*MANDAR CUENTA TOTAL A LA TCI.

TCI - Módulo Central

Módulo Central - TCI

CAR1MEN n1n2	CAR2MEN CC	CAR3MEN Ø Ø	CO C	C1 u	C2 e	C3 n	C4 t	C5 a	C6	C7 =	C8	C9 N	C10 \$	C11 7	C12 2	C13 0	C14	C15
-----------------	---------------	----------------	---------	---------	---------	---------	---------	---------	----	---------	----	---------	-----------	----------	----------	----------	-----	-----

En el mensaje de la TCI al Módulo Central se tienen las siguientes características:

- 1) En CAR1MEN se envía el número de comensal del que se desea la cuenta, si este número es ØØ la petición es de el total de la cuenta, si el número es diferente de cero este número indica el comensal del que se debe imprimir la cuenta.
- 2) En CAR2MEN se envía el comando (CC).

El mensaje del Módulo Central a la TCI puede ser el mostrado o cualquier otro que se desee desplegar, tomando en cuenta que cada carácter que sea desplegado debe ir en código ASCII en hexadecimal. Mandar un "blanco" (20h) en C15 para manejo de desplegado de mensajes.

\*ALTA / BAJA DE COMANDAS (PLATILLOS).

TCI - Módulo Central

Módulo Central - TCI

CAR1MEN N1 N2	CAR2MEN C1 C2	CAR3MEN N1 N2	CO Q	C1 u	C2 e	C3 s	C4	C5 d	C6 e	C7	C8 H	C9 u	C10 i	C11 t	C12	C13	C14	C15 L
------------------	------------------	------------------	---------	---------	---------	---------	----	---------	---------	----	---------	---------	----------	----------	-----	-----	-----	----------

La comunicación de TCI a Módulo Central tiene las siguientes características:

En el caracter 1 (CAR1MEN) se envia el número de platillos que se piden.

En el caracter 2 (CAR2MEN) se envia el caracter de control que da la información de qué número de tabla se está accesorando, en este grupo de comandos se incluirán los grupos de menús de cada restaurante (por ejemplo sopas, entradas, bebidas, etc.), la relación de claves se muestra en la tabla 6.1.4.

CLAVE EN CAR2MEN (HEX)	COMANDO
D0	ALTA TABLA 9
D1	BAJA TABLA 9
D2	ALTA TABLA 10
D3	BAJA TABLA 10
D4	ALTA TABLA 11
D5	BAJA TABLA 11
D6	ALTA TABLA 12
D7	BAJA TABLA 12
D8	ALTA TABLA 13
D9	BAJA TABLA 13
DA	ALTA TABLA 14
DB	BAJA TABLA 14
DC	ALTA TABLA 15
DD	BAJA TABLA 15
E4	ALTA TABLA 1
E5	BAJA TABLA 1
E6	ALTA TABLA 2
E7	BAJA TABLA 2
E8	ALTA TABLA 3
E9	BAJA TABLA 3
EA	ALTA TABLA 4
EB	BAJA TABLA 4
EC	ALTA TABLA 5
ED	BAJA TABLA 5
EE	ALTA TABLA 6
EF	BAJA TABLA 6
F0	ALTA TABLA 7
F1	BAJA TABLA 7
F2	ALTA TABLA 8
F3	BAJA TABLA 8

Tabla 6.1.4 Relación de claves de grupos de platillos.

En el caracter 3 (CAR3MEN) se envia la información del platillo de la tabla que se quiere pedir, (por ejemplo si la tabla 1 es CARNE y la CARNE ASADA es el número 10, para pedir este platillo se envia N1 N2 =0A).

La comunicación del Módulo Central a la TCI tiene las siguientes características:

- En los caracteres del 1 al 15 (C0 a C14) puede ir cualquier letra para designar a los platillos enviando la letra que se quiera ver en la pantalla con su equivalente en ASCII.
- En el caracter 16 (C15) se va a mandar codificado la siguiente información: C15 = L<sub>7</sub> L<sub>6</sub> L<sub>5</sub> L<sub>4</sub> L<sub>3</sub> L<sub>2</sub> L<sub>1</sub> L<sub>0</sub>.
- En los dos bytes más significativos se envía la información de la operación realizada, esto es alta, baja o error (se contesta con un error en el caso por ejemplo que quieran dar mas platillos de baja que los que se dieron de alta).

L<sub>7</sub> L<sub>6</sub> = 00 : ERROR EN PETICION  
 L<sub>7</sub> L<sub>6</sub> = 01 : ALTA DE PLATILLOS  
 L<sub>7</sub> L<sub>6</sub> = 11 : BAJA DE PLATILLOS

- En los 6 bits menos significativos se envía la cantidad de platillos que se dan de alta o baja, teniéndose como máximo número de platillos 63. (por ejemplo si se desean dar 14 "ALTAS" de un platillo el caracter 16 (C15) se forma así:

ALTA = 10 b  
 14 h = 001110  
 C15 = ALTA + 14 = 10 1110 = 8E (h)  
 CLAVE #PLATILLOS -----

\*MENSAJES - BIP

Estos mensajes son enviados únicamente de el Módulo Central a la TCI para dar algún aviso al mesero (por ejemplo "platillo listo") con la restricción de que este mensaje sea de 15 caracteres mandando en el caracter 16 el número 20h para manejo de desplegado de mensajes en la TCI.

- Para la elaboración del programa se dividirá la memoria RAM en dos partes una para guardar información que llega de PC en mensajes de 16 caracteres almacenando un total de 5 mensajes y otra para el manejo y control de las rutinas del programa, la distribución de los registros se muestra en las tablas 6.1.5.y 6.1.6:



DIRECCIO N	No.DE REGISTRO	INFORMACION
00h	A	Registro de uso general.
01h	B	Registro de uso general.
02h	R2	Contiene el valor de tecla en rutina de manejo de teclado.
03h	R3	Contiene el valor de tecla anterior en rutina de manejo de teclado.
04h	R4	Contador de número de veces que se lee el teclado para considerar tecla valida.
05h	R5	Registro 1 de banderas para control de rutinas.
06h	R6 (No.TCI)	Contiene el número de TCI programado en <i>dip switch</i> .
07h	R7 (DATDES)	Registro de dato a desplegar en LCD.
08h	R8 (CONTB)	Contador 1 para control de tamaño de frame en comunicación.
09h	R9 (PRUEBDIS)	Contador 2 para control de tamaño de frame en comunicación.
0Ah	R10	Registro 2 de banderas para control de rutinas.
0Bh	R11 (DATOSAL)	Registro de dato a transmitir en SCI (Tx).
0Ch	R12 (DATO)	Registro de dato recibido en SCI (Rx).
0Dh	R13	Registro de valor clave de tecla.
0Eh	R14 (CARIMEN)	Registro de caracter 1 de mensaje a transmitir.
0Fh	R15 (CAR2MEN)	Registro de caracter 2 de mensaje a transmitir.
10h	R16 (CAR3MEN)	Registro de caracter 3 de mensaje a transmitir.
11h	R17	Registro temporal de número para rutinas.
12h	R18	Registro 3 de banderas para control de rutinas.
13h	R19	Contador para controlar borrado de caracteres.
14h	R20	Registro de último dato recibido por SCI.
15h	R21 (DAM)	Dirección actual de memoria: registro que guarda la dirección a la que apunta el último mensaje almacenado en memoria.
16h	R22	Registro para rutina mesero que distingue si la operación es mesa o mesero.
17h	R23	Registro que almacena el número de mesero dado de alta.
18h	R24	Registro para manejo de desplegado de mensaje.
19h	R25	Registro de número de mesa dada de alta.
1Ah	R26	Registro para uso de rutina tablas.
1Bh	R27	Contador para rutina de mesa.
1Ch	R30	Inicio de área de <i>stack pointer</i> .
. . .		Área de <i>stack</i> .
47h	R71	Fin de área de <i>stack pointer</i> .
A8h	R168 (ADM)	Registro donde inicia la memoria de mensajes.
. . .		Área de memoria de mensajes.
F7	R247	Registro donde termina la memoria de mensaje.
FD	R253	$N^{\circ}1 N^{\circ}2 (B_7 B_6 B_5 B_4 = N^{\circ}1 / B_3 B_2 B_1 B_0 N^{\circ}2)$ Clave de mesero.
FE	R254	$N^{\circ}3 N^{\circ}4 (B_7 B_6 B_5 B_4 = N^{\circ}3 / B_3 B_2 B_1 B_0 N^{\circ}4)$ Clave de mesero.
FF	R255	Registro de número de mesa

Tabla 6.1.5 Funciones de los registros de la memoria RAM.

R5	X <sub>0</sub>	1er. caracter teclado (bandera que se enciende cuando se inicia el formado de un mensaje).
	X <sub>1</sub>	ALTA de TCI - bandera que se enciende cuando el Módulo Central polea por primera vez a la TCI.
	X <sub>2</sub>	ERROR EN RECEPCION - bandera que se enciende cuando los datos llegaron mal (con errores) a la TCI.
	X <sub>3</sub>	SI HAY INFORMACION PARA TRANSMITIR - bandera que se enciende cuando ya se formó un mensaje completo para transmitir.
	X <sub>4</sub>	ERROR EN TRANSMISION - bandera que se enciende cuando el Módulo Central manda el carácter que informa que le llegaron mal los datos.
	X <sub>5</sub>	NO EMPLEADO
	X <sub>6</sub>	NO EMPLEADO
	X <sub>7</sub>	Bandera para rutina de cuenta manejo mesa/comensal.
R10	X <sub>0</sub>	Teclado
	X <sub>1</sub>	1er. caracter rutinas.
	X <sub>2</sub>	NO EMPLEADO
	X <sub>3</sub>	Memoria de mensajes llena.
	X <sub>4</sub>	NO EMPLEADO
	X <sub>5</sub>	NO EMPLEADO
	X <sub>6</sub>	NO EMPLEADO
	X <sub>7</sub>	NO EMPLEADO
R18	X <sub>0</sub>	Bandera que indica si la operación realizada es alta o baja. ALTA = 0      BAJA = 1
	X <sub>1</sub>	bandera rutina manejo de teclado (bandera 1er caracter)
	X <sub>2</sub>	Rastreo de memoria 1= rastreando memoria 0= no rastreando memoria.
	X <sub>3</sub>	bandera que se enciende para dar permiso para transmitir
	X <sub>4</sub>	bandera que permite cambiar tipo de cuenta, lo enciende alta de platillo y no se pueden cambiar las cuentas hasta que se da de baja la mesa.
	X <sub>5</sub>	1= inicia rutina de encendido de luz 0= termina rutina de encendido de luz
	X <sub>6</sub>	bandera que se enciende cuando se va a mandar un mensaje - bip
	X <sub>7</sub>	1= formando tablas con <i>SHIFT</i> . 0= formando tablas sin <i>SHIFT</i> .

Tabla. 6.1.6 Banderas de control del programa.

### 6.1.3 DESARROLLO DE LA PROGRAMACION DE LA TCI.

La programación de la TCI se divide en las siguientes rutinas:

- Declaración de registros.
- Inicialización de registros.
- Inicialización de la pantalla.
- Lectura de número de TCI en *dip switch*.
- Alta de sistema.
- Manejo de teclado.
- Manejo de mensajes-bip.
- Asignación de tareas.
- Tarea de mesero.
- Tarea mesa.
- Mensaje error.
- Limpiar un caracter a la izquierda.
- Tarea *SHIFT*: Mandar el total de cuenta.  
Tablas 9-16.
- Tarea tablas (grupos de platillos).
- Letreros tablas (grupos de platillos).
- Borrado de área letreros y mensajes.
- Desplegado de caracteres en en pantalla.
- Tiempo de espera para inicializar la pantalla.
- Envío de comando a la pantalla.
- Comunicación.
- Transmisión de dirección y datos a TCI's.
- Recepción de datos de TCI's.

A continuación se describirán las rutinas del programa del microcontrolador de la TCI.

#### DECLARACION DE REGISTROS.

En esta parte del programa se le asigna un identificador a los registros que se emplearán en el programa.

#### INICIALIZACION DE REGISTROS.

En esta rutina se realiza la configuración necesaria para que el  $\mu\text{C}$  realice las funciones requeridas para el correcto funcionamiento de la TCI. Las características principales de la inicialización son :

- Formato de comunicación:
  - 1 bit de arranque.
  - 2 bits de parada.
  - 1 bit de paridad par.
  - 8 bits de datos.
  - Velocidad de 300 bauds.
  - Se habilita la interrupción de recepción de la *UART (universal asynchronous receiver transmitter)*.
  - Se deshabilita la interrupción de la transmisión de la *UART*.
- Puerto C se define como entrada para lectura de número de TCI.
- Puerto A se define como salida para manejo de datos de la pantalla.
- Puerto B se define como salida para control de pantalla y encendido de renglones de teclado.
- Puerto D se define como entrada digital para lectura de columnas de teclado.
- Limpia todas las banderas de control del programa
- Inicializa con el valor #FFh los registros de números de mesero y mesa

#### INICIALIZACION DE PNTALLA.

Esta rutina da el tiempo necesario a la pantalla para que alcance los 4.5 V, voltaje necesario para que funcione adecuadamente, además se inicializa la pantalla con las siguientes características:

- Palabras de 8 bits.
- El desplazamiento del cursor es hacia la derecha.
- Aparecerá el cursor parpadeando.

### LECTURA DE NUMERO DE TCI EN DIP SWITCH.

Esta rutina realiza la lectura de 4 bits en el *dip switch* donde se puede configurar el número de TCI en esta rutina se lee cada *pin* al que se conecta el *dip switch* colocando el bit en la posición adecuada para formar el número.

### ALTA DE SISTEMA.

En esta rutina la TCI espera que el Módulo Central le envíe información por primera vez preguntando a la TCI si tiene información para enviar a la PC con esto la TCI espera que un mesero realice su alta para iniciar la captura de información, el diagrama de flujo de los procesos mencionados anteriormente se muestran en la figura 6.1.6.

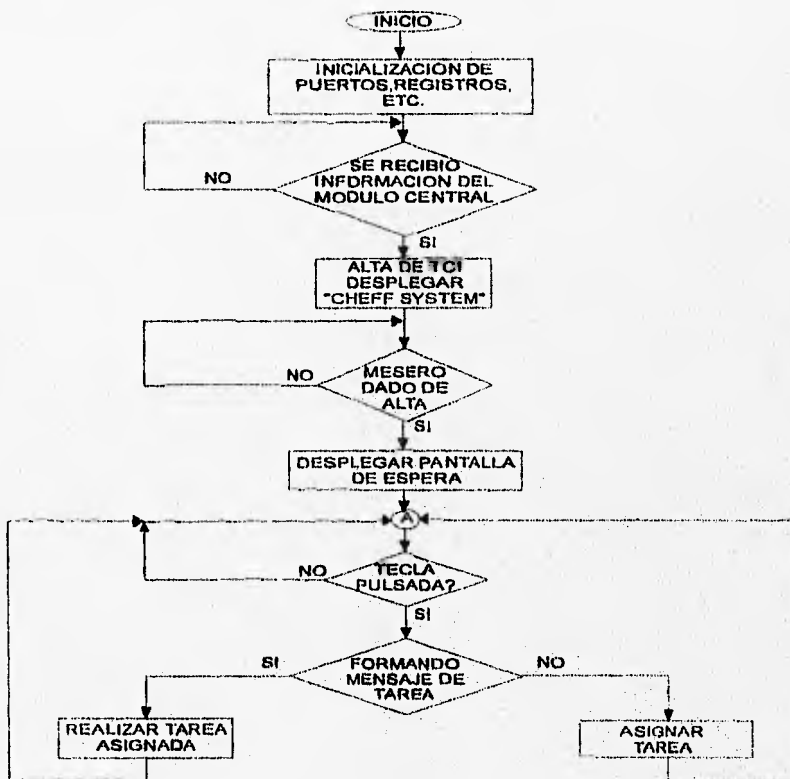


Fig. 6.1.6 Diagrama de flujo de la rutina principal de la TCI.

MANEJO DE TECLADO.

En esta parte del programa se lleva a cabo la lectura del teclado esto se logra encendiendo el pin asociado a cada renglón, al encenderse el pin se realiza una lectura de las columnas para ver si alguna columna está encendida lo que equivaldría a una tecla oprimida, si en el renglón revisado no hay una tecla pulsada se revisa el siguiente, realizando un puleo de todo el teclado hasta encontrar una tecla pulsada, cuando se detecta una tecla pulsada se revisa 7 veces el teclado para detectar si la tecla sigue pulsada y evitar señales de rebote que equivaldrían a una lectura errónea. Al detectar una tecla que está siendo oprimida se revisa una bandera que indica si hay un mensaje en formación, en cuyo caso el programa va a la rutina abierta anteriormente para seguir formando el mensaje, si no se estaba formando un mensaje el programa va a una rutina que asigna una tarea de acuerdo a la tecla pulsada. En esta rutina también se lleva a cabo el encendido del circuito de sonido mientras la tecla permanece oprimida así como el encendido de la iluminación de la TCI, el diagrama de flujo de esta rutina se muestra en la figura 6.1.7.

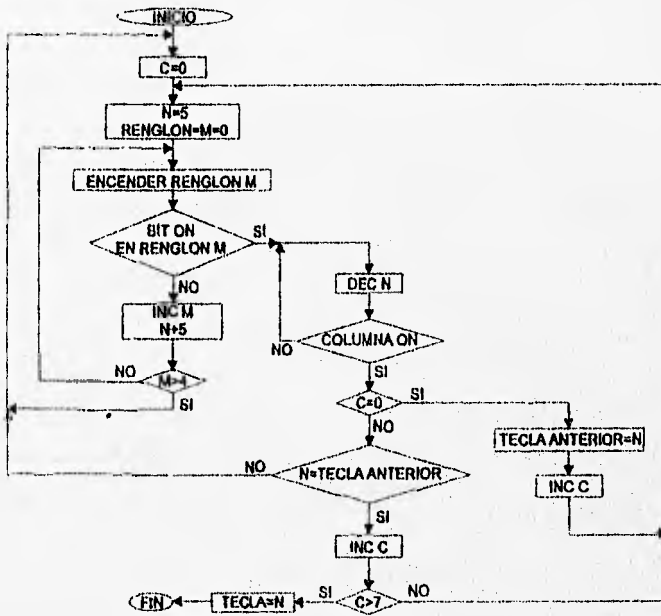


Fig. 6.1.7 Rutina de lectura de teclado.

#### MANEJO DE MENSAJES BIP.

Esta rutina se encarga del manejo de los mensajes especiales enviados por el Módulo Central a la TCI (por ejemplo "platillo listo en cocina") los cuales son desplegados para su visualización, encendiéndose los circuitos de iluminación de la TCI y enviando una serie de pulsos al circuito de sonido para llamar la atención del mesero.

#### ASIGNACION DE TAREAS.

Esta parte del programa lleva a cabo la asignación de la tarea que debe efectuarse al oprimir una tecla. Cuando la tecla es la primera en la secuencia de formación de un mensaje se le asigna la tarea relacionada a la tecla, en caso de que se esté formando un mensaje el programa continúa en la tarea que se había asignado con anterioridad. El diagrama de flujo de esta rutina se muestra en la figura 6.1.8.

#### TAREA MESERO.

Esta rutina sirve para formar el mensaje y realizar la comunicación del mismo en las tareas de alta y baja de mesero la secuencia de esta rutina es la siguiente:

- Desplegado de la palabra mesero/mesa.
- Espera de una a dos teclas que sólo pueden ser números o "clear".
- Espera de tecla ALTA/BAJA de confirmación de mensaje.
- Transmisión de información y espera de respuesta para su manejo en *display*.
- Manejo de la clave de mesero que viene codificada en los bits 15 y 16 del mensaje.

El diagrama de flujo de esta rutina se muestra en la figura 6.1.9.

#### TAREA MESA.

Esta rutina junto con la de mesero realiza la comunicación de la tarea mesa, la parte de formado del mensaje la realiza la tarea mesero, y al regresar de la comunicación esta rutina se encarga de las siguientes funciones:

- Si es baja de mesa, borra del área correspondiente del *display* el número de mesa dada de baja
- Si es alta de mesa, coloca el número de mesa dada de alta en el área correspondiente del *display*.

El diagrama de flujo de esta rutina se muestra en la figura 6.1.10.

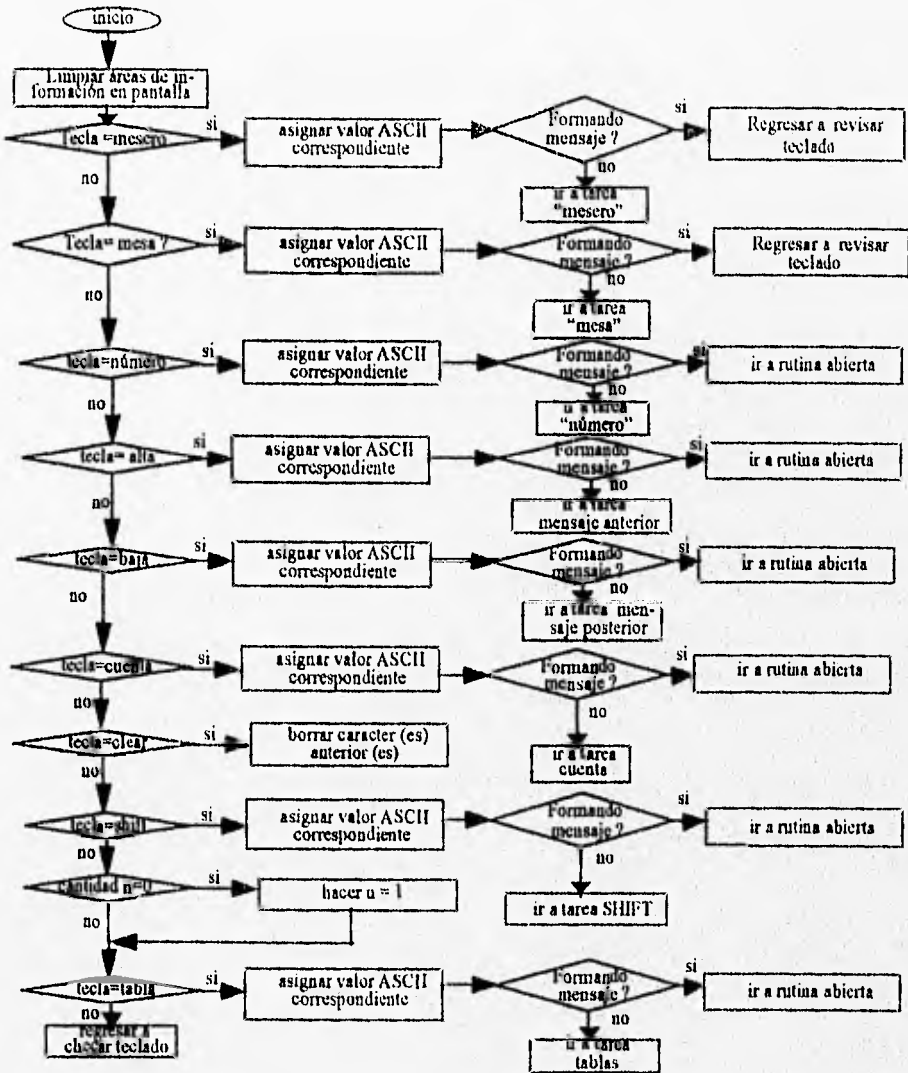


Fig. 6.1.8 Rutina de asignación de tareas.

**MENSAJE ERROR.**

Esta rutina lleva a cabo el despliegue de un mensaje de error cuando se quiere realizar una operación no permitida en la TCI.



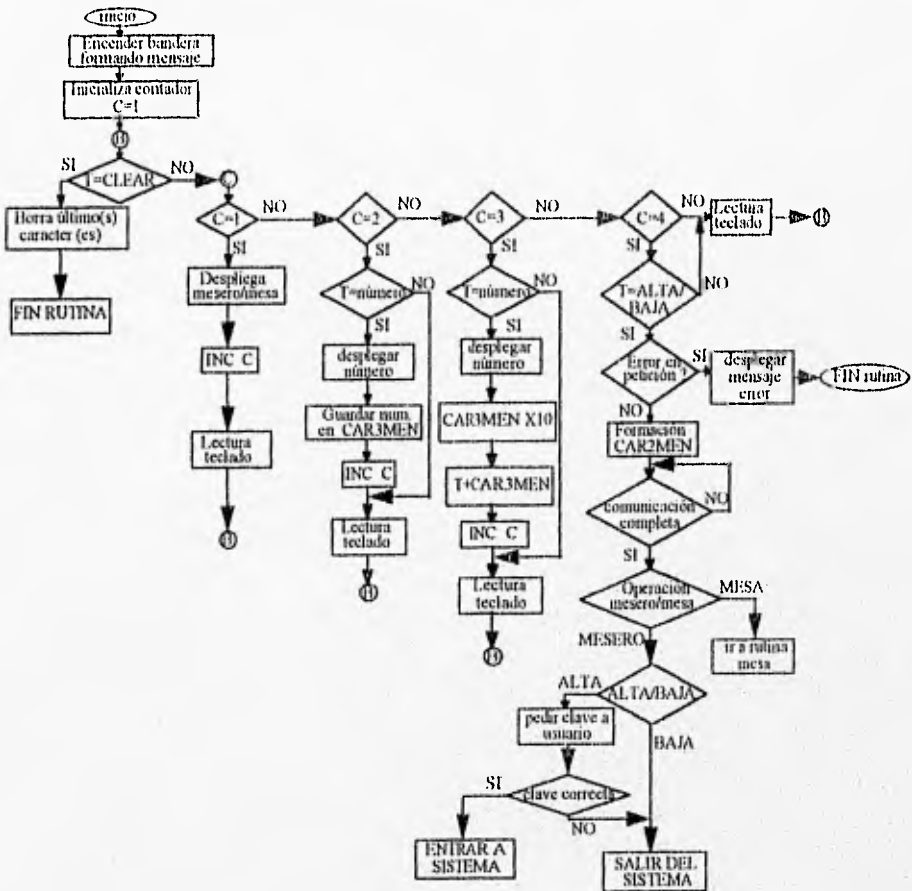


Fig. 6.1.9 Diagrama de flujo de la rutina de tarea mesero.

**LIMPIAR UN CARACTER A LA IZQUIERDA.**

Esta rutina se encarga de borrar el carácter situado a la izquierda de la posición del cursor verificando si la posición del cursor es la primera columna antes de borrar el carácter para que no se pierda la dirección en la memoria de la pantalla. El diagrama de flujo de esta rutina se muestra en la figura 6.1.11.

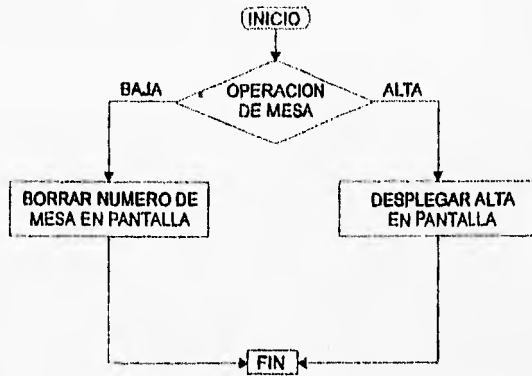


Fig. 6.1.10 Diagrama de flujo de la tarea mesa

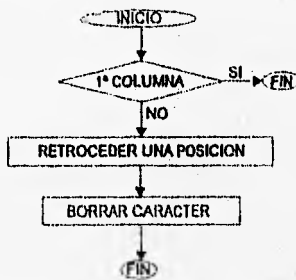


Fig. 6.1.11 Rutina de borrado de un caracter.

#### TAREA SHIFT.

Esta rutina tiene por objeto asignar una doble función a algunas teclas de la TCI pulsando la tecla "SHIFT" seguida de una tecla que tiene doble función, se tienen las siguientes posibilidades de tecla alterna:

- MANDAR EL TOTAL DE CUENTA. Esta rutina sirve para que la cuenta total de alguna mesa sea enviada a la TCI sin realizar la impresión de la cuenta, la secuencia de teclas para realizar esto es: "SHIFT" + "3" + "ALTA", la figura 6.1.12 muestra el diagrama de flujo de esta rutina.
- TABLAS 9-16. Para el manejo de alta de platillos se emplearán las teclas denominadas tablas las cuales se configurarán con los nombres de los grupos de platillos de cada restaurante (por ejemplo carnes, bebidas, sopas, etc.), cada platillo tendrá un número dentro del grupo de tal forma que cada platillo tenga asociado un grupo (tabla n) y un identificador del grupo (X1X2) (por ejemplo para dar de alta una sopa de habas se enviarían la información: tabla n = sopas e identificador X1X2=01, si este platillo fuera el primero en el grupo de sopas).

Con objeto de tener un número suficiente de teclas tablas para incluir los diferentes grupos de alimentos que se manejan en los restaurantes se empleará la tecla *SHIFT* para ampliar las tablas del 1-8 y contar con 16 tablas (grupos de platillos); la rutina sirve para formar el carácter 2 del mensaje que se transmite la TCI (CARZMEN) el orden de las teclas a pulsar es: #1 + #2 + *SHIFT* + TABLA n + #3 + #4.

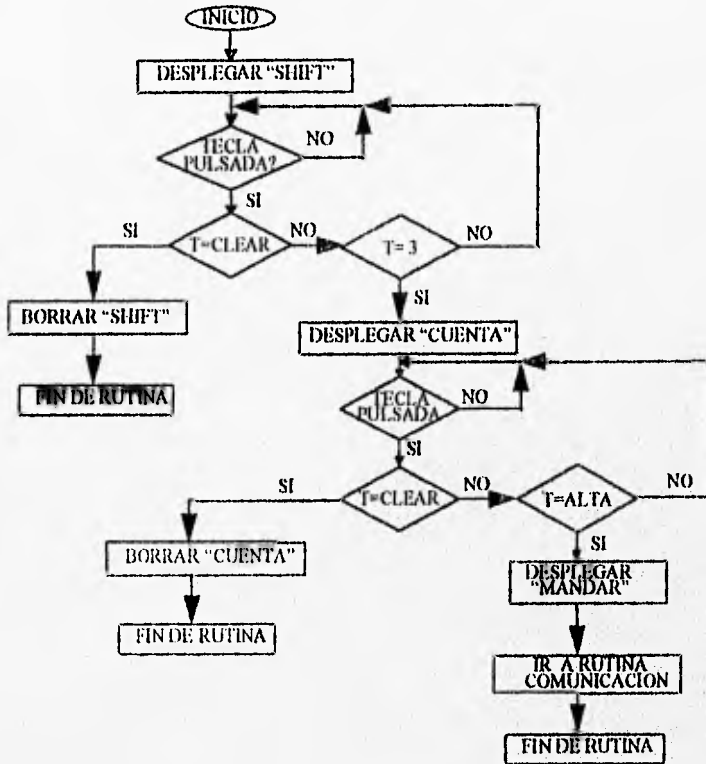


Fig. 6.1.12 Rutina de total de cuenta.

**TAREAS TABLAS.**

Esta rutina lleva a cabo la petición de alta y baja de platillos al Módulo Central, así como el manejo del desplegado de la información de acuerdo al Byte 16 del mensaje en el que viene codificado el número de altas, bajas y la operación realizada (Baja, Alta o Error).

La secuencia de teclas para realizar esta operación es la siguiente:

- 1 o 2 números que indican el número de comensal, este número se puede omitir cuando se va a pedir un platillo para el comensal "1" en este caso por default se considerará el número "1".
- Alguna de las tablas T1 a T8 o una combinación *SHIFT* + Tn para las tablas T9 a T16.
- 1 o 2 números que sirven para identificar que platillo del grupo (tabla) se está solicitando.
- La tecla ALTA o BAJA de acuerdo a la operación que se quiera realizar.

El diagrama de flujo de esta rutina se muestra en la figura 6.1.13.

#### **LETREROS TABLAS.**

En esta área del programa se encuentran los letreros asociados a cada tecla de tabla (Estos letreros son los grupos de alimentos que se configurarán de acuerdo a cada restaurante).

#### **BORRADO DE AREA DE LETRERO Y MENSAJES.**

Esta rutina se encarga del borrado del área de letreros ( $A_1 - A_8$ ) y de mensajes ( $A_{11} - A_{12}$ ) del *display*, esto se logra posicionando el cursor en los caracteres A8 o A32 y realizando un borrado de caracteres hacia la izquierda del cursor hasta llegar a la primera columna, en la figura 6.1.14 se muestra el diagrama de flujo de esta rutina.

#### **DESPLEGADO DE LETRAS EN PANTALLA.**

En esta parte del programa se encuentran las instrucciones para desplegar caracteres en pantalla (letras y números).

#### **TIEMPO DE ESPERA PARA INICIALIZAR LA PANTALLA.**

Esta rutina sirve para proporcionarle a la pantalla los comandos necesarios en el tiempo que requiere para inicializar la pantalla, el diagrama de flujo de esta rutina se muestra en la figura 6.1.15.

#### **ENVIO DE DATOS A PANTALLA.**

Esta rutina lleva a cabo el envío de información a la pantalla ya sean datos para ser desplegados o comandos de control para la pantalla. Con esta rutina se proporcionan las señales RS, R/W y E requeridas por la pantalla para que la información enviada a las líneas de datos sea reconocida como un carácter a desplegar.

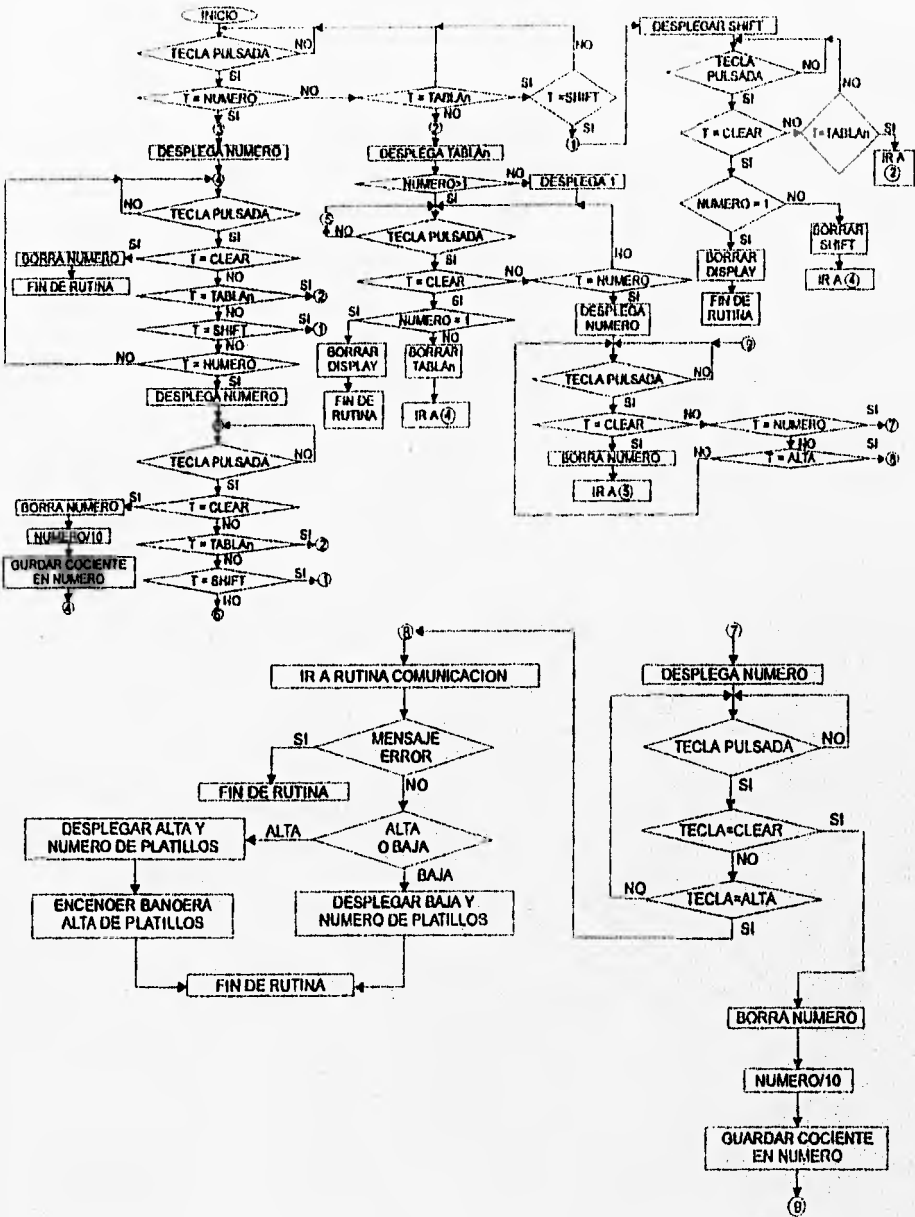


Fig. 6.1.13 Diagrama de flujo de la rutina de alta o baja de platillos.

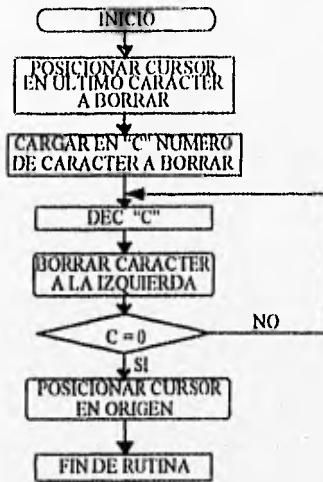


Fig. 6.1.14 Rutina de borrado de área de letreros y mensajes.

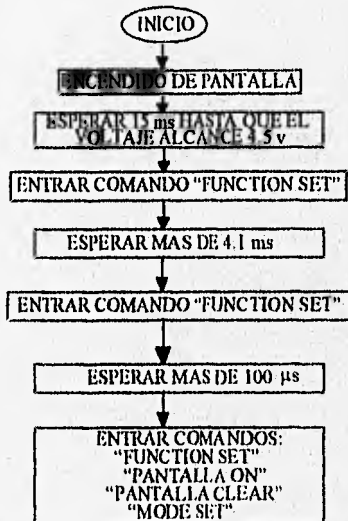


Fig. 6.1.15 Rutina de tiempo de espera para inicializar la pantalla.

COMUNICACION.

Esta rutina se encarga de realizar la transmisión de la información hacia el Módulo Central, verificando si en la transmisión hubo algún error para realizar la retransmisión de los datos. Al concluir la transmisión espera la respuesta del Módulo Central para almacenar y desplegar el mensaje de respuesta, el diagrama de flujo de esta rutina se muestra en la figura 6.1.16.

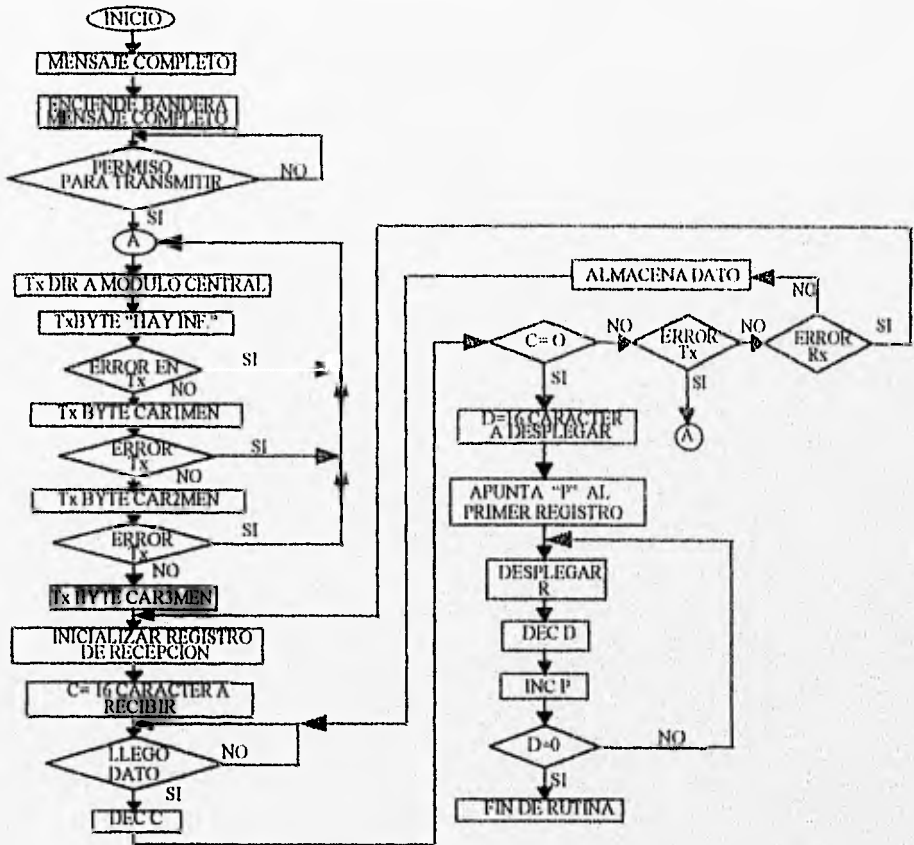


Fig. 6.1.16 Diagrama de flujo de la rutina de comunicación.

TRANSMISION DE DIRECCION Y DATOS.

Esta rutina realiza la transmisión de la dirección al Módulo Central que lo pone en estado de recepción de datos, así como la transmisión de datos.

## RECEPCION DE DATOS EN SCI (Serial Communication Interface).

En esta rutina se lleva a cabo la atención a la interrupción generada por la recepción de datos provenientes del Módulo Central tanto de caracteres de información como de caracteres de control para la comunicación, el diagrama de flujo de esta rutina se muestra en la figura 6.1.17.

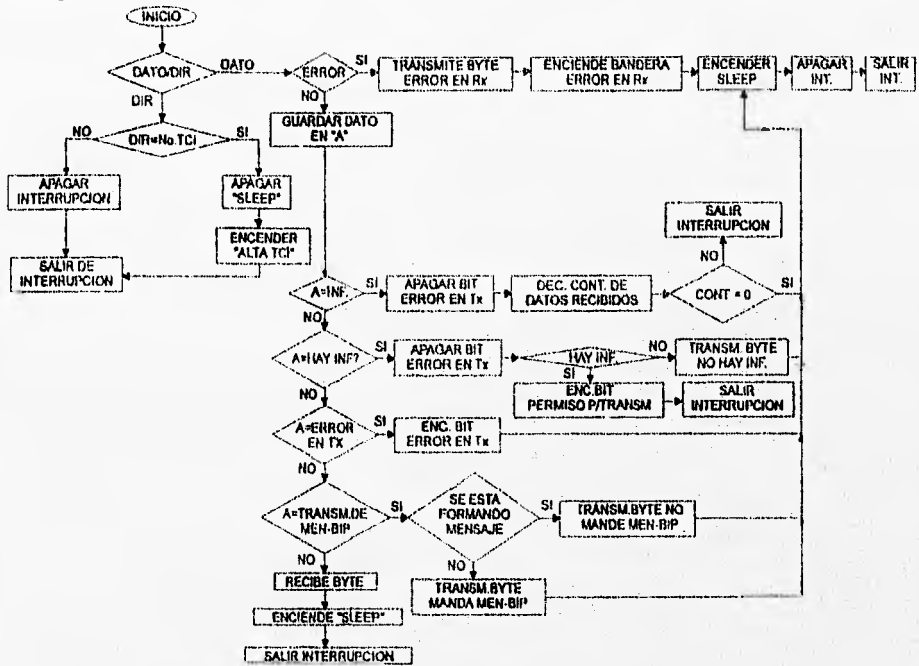


Fig. 6.1.17 Diagrama de flujo de la rutina de recepción de datos.

## 6.2 PROGRAMACION DEL MICROCONTROLADOR DE LA INTERFAZ DE LA PC.

La interfaz de la PC (IPC), es el medio por el cual la información capturada por el Módulo de Comunicaciones de la Interfaz (MCI), es enviada hacia la PC, el control del flujo de la información entre las TCI's y la PC es realizada por el microcontrolador de la IPC el cual debe realizar las siguientes funciones:

- Esperar la inicialización enviada por el programa de la PC.
- Solicitar a la PC el número de TCI's presentes en el sistema.
- Solicitar información a las TCI's para la PC.
- Transmitir información enviada por la PC hacia alguna TCI.
- Informar a las TCI's que el programa de la PC no está residente.



Para realizar estas funciones se tomarán la siguientes consideraciones:

- Se emplean los comandos mostrados en la tabla 6.1.3 para controlar la comunicación con las TCI's.
- Se emplean los comandos para controlar la comunicación con la PC, mostrados en la tabla 6.2.1.

CLAVE	
160d	Se envía de PC a IPC e indica que el programa de la PC está residente.
161d	Se envía de PC a IPC e indica que el programa de la PC va a salir de funcionamiento.
162d	Se envía de PC a IPC y sirve para resetear el microcontrolador de la IPC.
163d	Se envía de la IPC a la PC y sirve para solicitar número de TCI's presentes en el sistema.

Tabla 6.2.1 Comandos adicionales.

- Los registros de la memoria RAM se configurarán como se muestra en las tablas 6.2.2 y 6.2.3.

DIRECCION	No.DE REGISTRO	INFORMACION
00h	A	Registro de uso general.
01h	B	Registro de uso general.
02h	R2(PRUEBDIS)	Contador de rutina de recepción.
03h	R3(DATOSAL)	Dato a transmitir en SCI.
04h	R4(BANDERA1)	Banderas de control.
05h	R5(DATO)	Registro de dato recibido en SCI.
06h	R6(CONTB)	Contador de rutina de recepción.
08h	R8(MAFPOL)	Número de TCI a polear.
09h	R9(ADM)	Registro de información procedente de TCI.
0Ah	R10	Registro de información procedente de TCI.
0Bh	R11	Registro de información procedente de TCI.
0Ch	R12(BYTECHEC)	Byte de verificación de error.
0Dh	R13(NMMC)	Número de TCI en mensaje colectivo.
1Dh	R29(CONT1)	Contadores para salir de rutina de espera de información de TCI's.
1Eh	R30(CONT2)	
1Fh	R31(CONT3)	
20h	R32(ADMB)	Área de mensajes especiales.
4Fh	R79	Fin de área de mensajes especiales.
50h	R80(MEMEN)	Registro que apunta a inicio de último mensaje especial que llegó.
51h	R81(BANMBIT)	Banderas de control de mensaje especial de transmisión.
52h	R82(NUMAFPOL)	Número de TCI's presentes en el sistema.
53h	R83(AMMC1)	Área de memoria de mensajes colectivos.
54h	R84(AMMC2)	
62h	R98(AMMC16)	Fin de área de memoria de mensajes colectivos.
63h	R99(BANCOL1)	Bandera que indica si un TCI recibió mensaje colectivo, TCIs 1 a 8.
64h	R100(BANCOL2)	Bandera que indica si un TCI recibió mensaje colectivo, TCIs 9 a 16.
6Ah	R108	Área de Stack pointer.
9Fh	R159(NUMDAT)	Número de datos a recibir de PC.
A0h	R160	Área de mensajes de memoria.
Ff'h	R255	Fin de área de mensajes de memoria.

Tabla 6.2.2 Funciones de los registros de la memoria RAM.

R4	X0	Error en recepción en datos de SCI.
	X1	Indica que llegó caracter A7.
	X2	Error en datos enviados a TCI.
	X3	Indica que no hay información en TCI.
	X4	Avisa que se puede mandar mensaje especial.
	X5	Indica que el programa de la PC está residente.
	X6	Indica que PC salió del sistema.
	X7	Indica que PC desea resetear al MC.
R81	X0	Indica que hay información para transmitir a TCI's en el área de mensaje 1.
	X1	Indica que hay información para transmitir a TCI's en el área de mensaje 2.
	X2	Indica que hay información para transmitir a TCI's en el área de mensaje 3.
	X3	Indica que hay información para transmitir a TCI's en el área de mensajes colectivos.
	X4	Indica que realiza envío de mensajes colectivos.
	X5	NO EMPLEADO
	X6	NO EMPLEADO
	X7	NO EMPLEADO

Tabla. 6.2.3 Banderas de control del programa.

#### 6.2.1 DESARROLLO DE LA PROGRAMACION DE LA IPC.

El programa de la IPC consta de las siguientes rutinas:

- Inicialización de registros.
- Inicialización de TCI's.
- Comunicación TCI - PC - TCI.
- Transmisión de mensajes especiales a TCI's.
- Solicitud a PC de número de TCI's presentes en el sistema.
- Recepción de información de PC.
- Transmisión de datos y dirección a TCI's.
- Recepción de información de TCI's.

Estas rutinas serán descritas a continuación.

#### INICIALIZACIÓN DE REGISTROS.

En esta rutina se configura al microcontrolador para realizar las funciones requeridas en la TCI. La configuración es la siguiente:

- 1 *bit* de arranque.
- 2 *bits* de parada.
- 1 *bit* de paridad par.
- 8 *bits* de datos.
- Velocidad de 300 bauds.
- Deshabilita interrupción de Tx y Rx.
- Puerto A configurado como entrada para leer datos de *bus* de PC.
- Dos pines del puerto D se configuran como salida y uno como entrada para manejar la transmisión de información a la PC.
- Se activa la interrupción INT1 para detectar cuando la PC quiere enviar información a la IPC.
- *Timer1* configurado como salida para controlar la recepción de información de la PC.
- Limpia todas las banderas de control del programa.

#### INICIALIZACIÓN DE TCI's.

Esta rutina se encarga de inicializar a todas las TCI's presentes en el sistema cuando comienza a operar el programa de la PC; el diagrama de flujo de esta rutina se muestra en la figura 6.2.1.

#### COMUNICACIÓN TCI - PC - TCI.

Esta es la rutina principal del programa y se encarga de solicitar información a cada una de las TCI's, en caso de que una TCI tenga información la recibe y la transmite a la PC para su procesamiento, luego se queda esperando la respuesta para ser enviada a la TCI y seguir solicitando información a las TCI's restantes. En esta rutina también se verifica si llegó un mensaje especial para alguna TCI en cuyo caso lo transmite, la figura 6.2.2 muestra el diagrama de flujo de esta rutina.

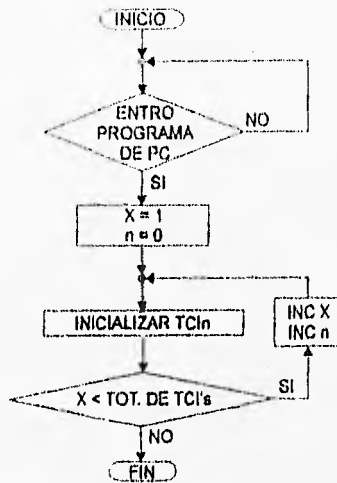


Fig. 6.2.1 Rutina de inicialización de las TCI's.

#### TRANSMISIÓN DE MENSAJES ESPECIALES A TCI's.

En esta rutina se realiza el control de la transmisión de mensajes que son enviados de la PC a las TCI's sin que estos sean solicitados por las TCI's, estos mensajes sirven para avisar de alguna situación especial a los meseros ("Platillo listo en cocina", "Atender mesa 2", etc.), debido a que estos mensajes no son esperados por el usuario de la TCI se verifica si en la TCI no se está realizando una operación para poderlo enviar, si la TCI está ocupada se intentará después, se dispone de espacio para un total de tres mensajes, la figura 6.2.3 muestra el diagrama de flujo de esta rutina.

#### SOLICITUD A PC DE NÚMERO DE TCI's PRESENTES EN EL SISTEMA.

Esta rutina se encarga de solicitar a la PC el número de TCI's que forman parte del sistema para saber a que TCI's se les debe solicitar información.

#### RECEPCIÓN DE INFORMACIÓN DE PC.

Esta rutina se encarga de la recepción de mensajes especiales enviados por la PC para alguna TCI y de caracteres que le informan a la IPC si el programa de la PC está residente o si la PC desea resetear el microcontrolador de la IPC, el diagrama de flujo de esta rutina se muestra en la figura 6.2.4.

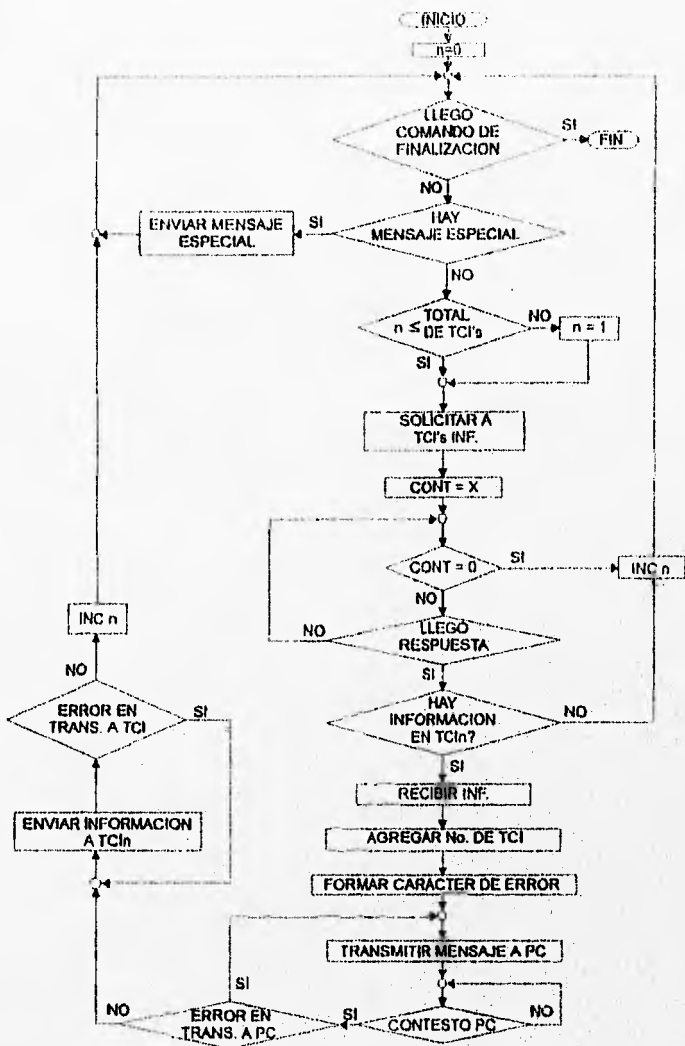


Fig. 6.2.2 Rutina de comunicación TCI - PC -TCI.

**TRANSMISIÓN DE DATOS Y DIRECCIÓN A TCI'S.**

Esta rutina tiene la misma función que la empleada en el programa de la TCI.

RECEPCIÓN DE INFORMACIÓN DE TCI'S.

Esta rutina se encarga de recibir información proveniente de las TCI's tanto información para la PC como comandos de control para la comunicación con las TCI's (comando de error, comando que indica que no hay información en TCI y comando que indica que se puede enviar mensaje especial a TCI. El diagrama de flujo de esta rutina es similar al de la figura 6.1.22.

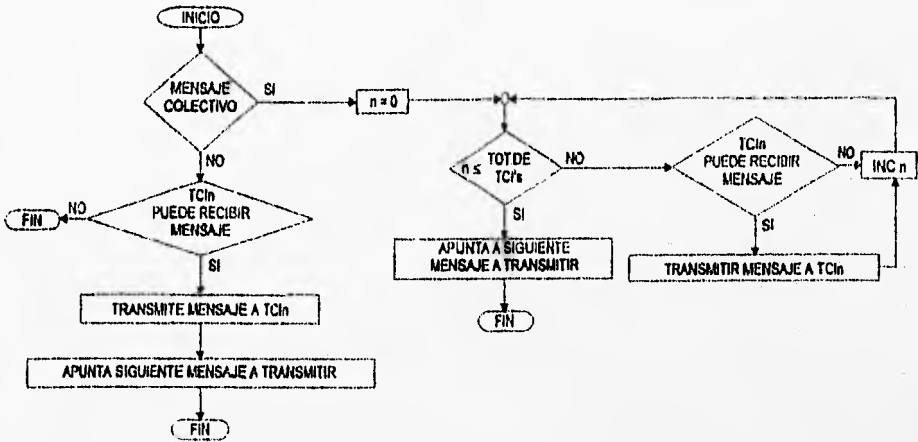


Fig. 6.2.3 Rutina de transmisión de mensajes especiales a TCI's.

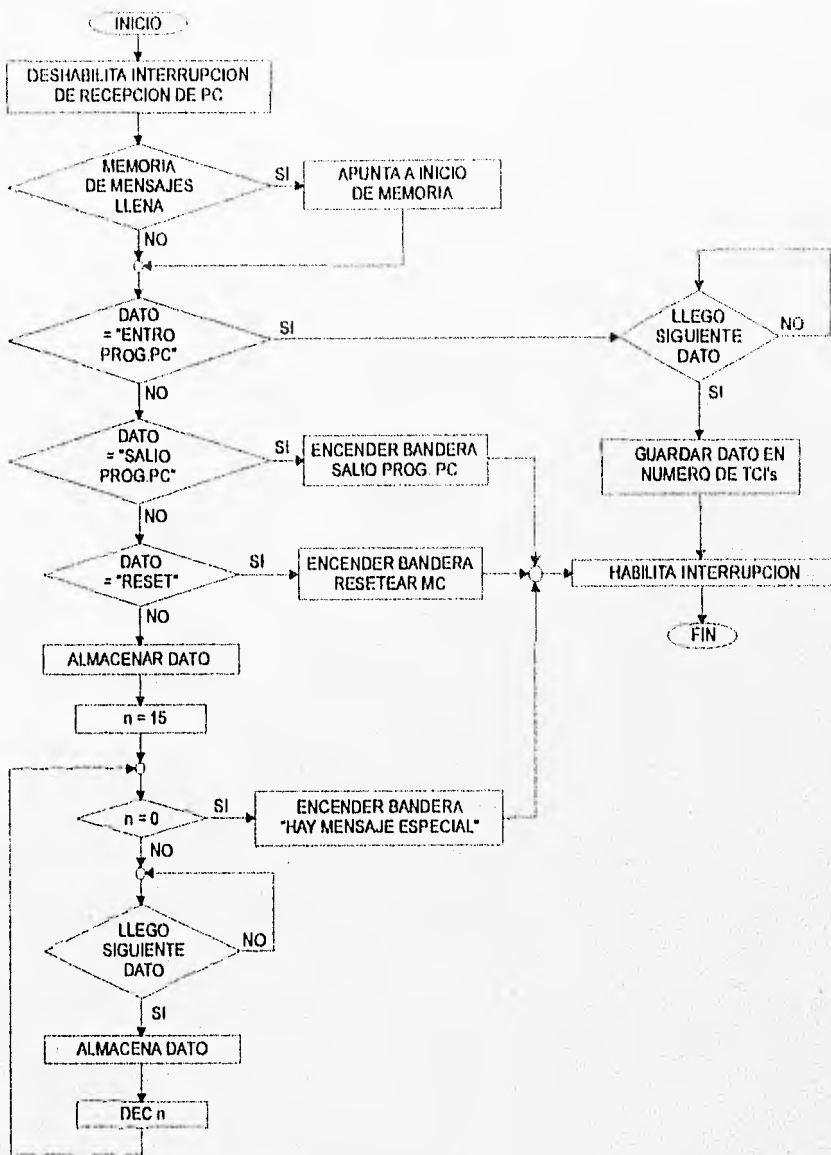


Fig. 6.2.4 Rutina de recepción de información de PC.

### 6.3 PROGRAMA DE LA PC.

La finalidad de este programa es recibir, almacenar y procesar la información procedente de las TCI's. El programa de la PC consta de una rutina principal que se encarga de configurar y establecer los parámetros necesarios para la comunicación con las TCI's a través del *bus-slot* de la PC, y una rutina denominada *handler*, invocada cuando la IPC se comunica con la PC a través del *bus-slot*.

#### -Rutina principal.

Esta rutina es la encargada de realizar el despliegue de la información enviada por las TCI's en el monitor de la PC para su visualización, además de controlar el envío de los mensajes especiales mencionados en el punto 6.1.1 a las TCI's.

El diagrama de flujo del proceso anterior se muestra en la fig. 6.3.1.

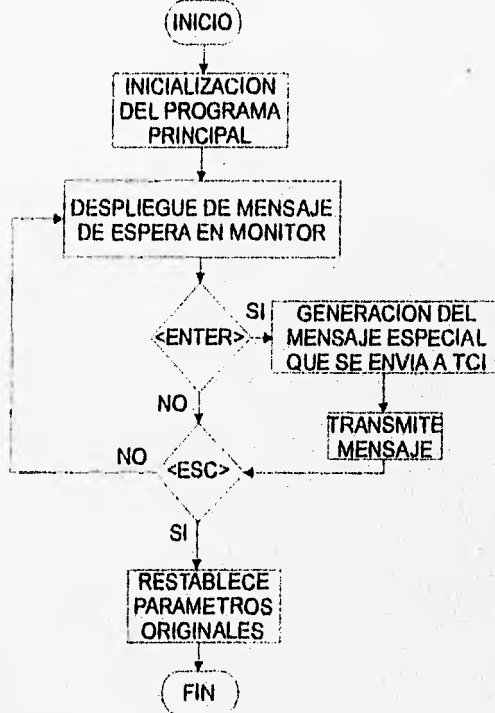


Fig. 6.3.1 Rutina principal.



**-Rutina handler.**

Como ya se mencionó contiene las rutinas que realizan las operaciones solicitadas por una TCI, estas rutinas son las siguientes:

- Alta/baja de mesero o mesa.
- Alta de platillos.
- Petición de cuenta única.

La comunicación enviada de la IPC a la PC tiene el formato mostrado en la fig. 6.3.2.

# DE TCI	CANTIDAD/ # DE PERSONA	COMANDO	# DE MESERO, MESA, PLATILLO	CORRECCION DE ERROR
----------	---------------------------	---------	--------------------------------	------------------------

Fig. 6.3.2 Formato de la información de IPC a PC.

Los elementos del *frame* enviado de IPC a PC son los siguientes: el primer byte es el número de TCI que envía la información, los tres comandos siguientes tienen la información descrita en el punto 6.1.2 y el último se emplea para detectar si la información enviada tiene errores en cuyo caso se pide una retransmisión de la misma. El formato de información enviada de PC a IPC es la mostrada en la fig. 6.1.5.

El *frame* de datos enviado por la IPC es almacenado en un archivo, posteriormente con los primeros cuatro bytes del *frame* de datos se forma el byte de detección de error el cual es comparado con el byte de detección de error enviado por la IPC, de existir algún error, el programa de la función *handler* terminará en ese momento y mandará un mensaje de retransmisión. Si no existe error en la lectura del *frame* de datos, el *handler* realizará la operación solicitada por la TCI especificada en el tercer byte del *frame* de datos, estas operaciones se explican a continuación.

La subrutina que efectúa la operación de alta de mesero, transmite a la TCI el mensaje de confirmación mencionado en el punto 6.1.2, donde se indica el número de mesero que se da de alta, y la clave con la que el mesero puede operar la TCI, el mensaje transmitido es también desplegado en el monitor de la PC. Una vez que la *frame* de caracteres ha sido transmitida la función *handler* finaliza.

La subrutina que realiza la operación de alta de mesa, al igual que la rutina anterior transmite a la TCI un mensaje que indica el número de mesa que se da de alta, el mensaje transmitido es también desplegado en la pantalla de la PC, en este momento se crea un archivo para llevar el registro de precios de los platillos consumidos en la mesa dada de alta. Después de lo anterior la función *handler* finaliza.

Las subrutinas para efectuar la operación de baja de mesero o de mesa, transmiten a la TCI el mensaje que indica el número del mesero o mesa que se da de baja, este mensaje también es desplegado en el monitor de la PC. Después la función *handler* finaliza.

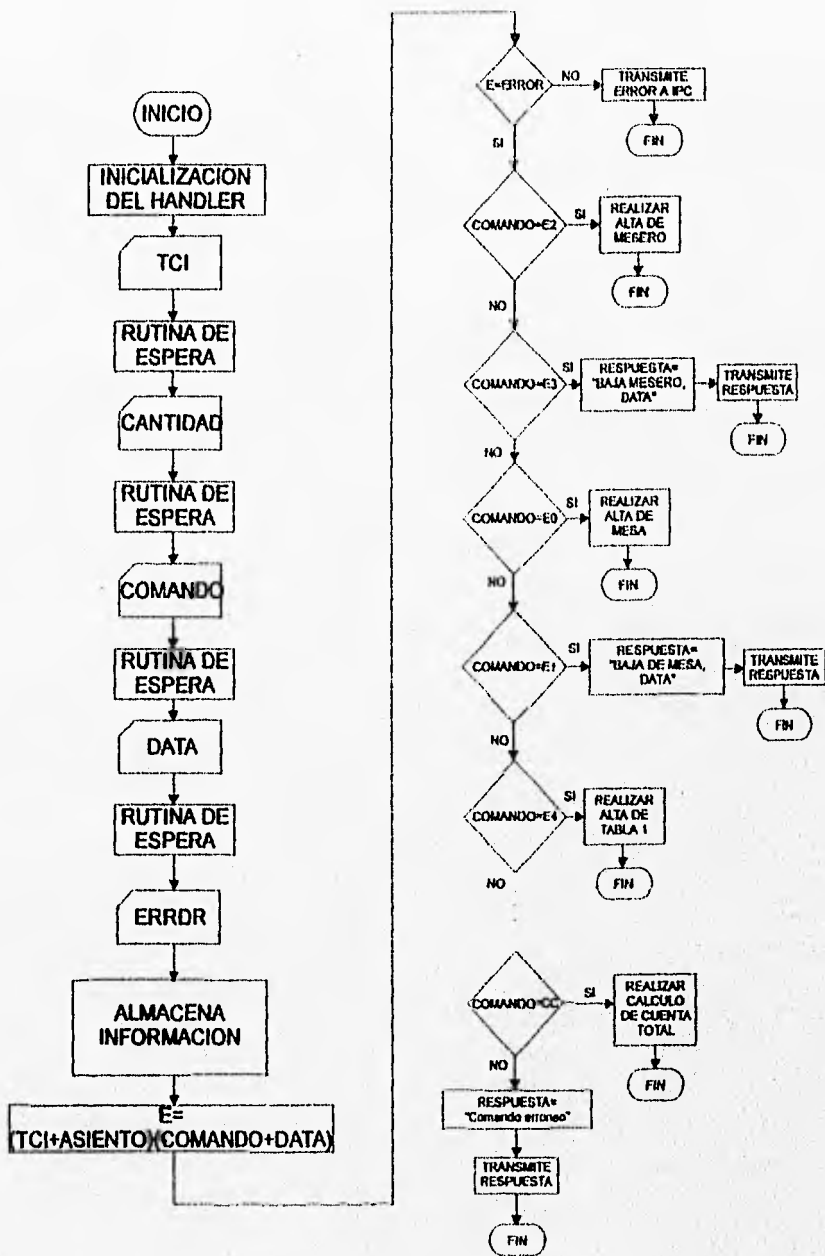
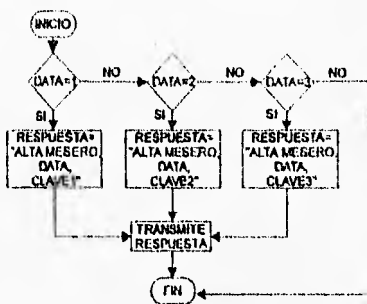
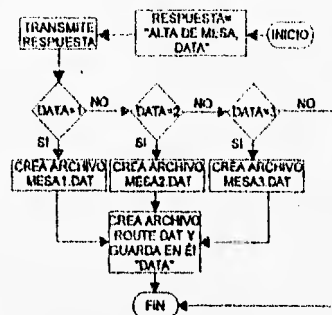


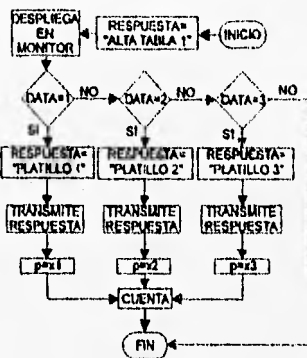
Fig. 6.3.3 Rutina del handler.



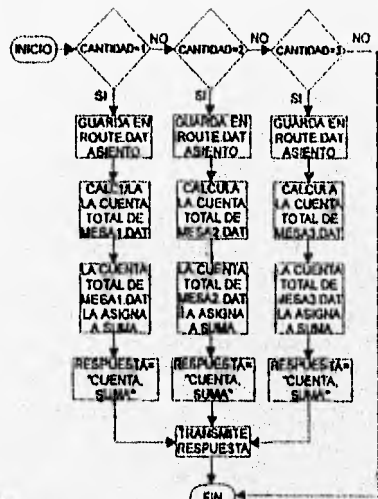
(a)



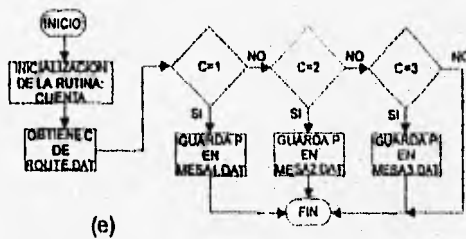
(b)



(c)



(d)



(e)

Fig. 6.3.4 Subrutinas del handler: (a) Alta de mesero, (b) Alta de mesa, (c) Alta de tabla n, (d) Petición de cuenta y (e) función cuenta para almacenar precios.

La subrutina para realizar la operación de alta de platillos, transmite a la TCI el mensaje del platillo que se da de alta, despliega dicho mensaje en el monitor de la PC y guarda en el archivo de la mesa dada de alta anteriormente, el precio del platillo que se dio de alta. Después de esto la función *handler* finaliza.

La subrutina de petición de cuenta, busca el archivo que lleva el registro de los precios de los platillos consumidos en la mesa que la TCI especifica, efectúa la suma total de los precios contenidos en dicho archivo, y el resultado lo transmite a la TCI en un mensaje que indica la cuenta total de dicha mesa, este mensaje es también desplegado en el monitor de la PC. Después de lo anterior la función *handler* finaliza.

La fig. 6.3.3 muestra el diagrama de flujo de la función *handler* con las subrutinas mencionadas anteriormente.

La fig. 6.3.4 muestra los diagramas de flujos de las subrutinas empleadas por la rutina *handler*.

## RESULTADOS Y CONCLUSIONES.

Con el objeto de determinar el alcance máximo del sistema presentado, se realizaron pruebas utilizando dos terminales de captura inalámbricas (TCI's), un módulo de comunicaciones de la interfaz (MCI) y una interfaz de la PC (IPC) la cual residía en una PC (procesador 486, 66 MHz).

Las pruebas que se realizaron fueron las siguientes:

-Prueba de distancia máxima de transmisión en exteriores del Módulo Central hacia una TCI, lográndose un alcance de 110 m.

-Prueba de distancia máxima de transmisión en exteriores de una TCI hacia el Módulo Central, lográndose un alcance máximo de 110 m.

Para poder comparar las ventajas del sistema expuesto con algún otro sistema equivalente existente en el mercado, se revisó un sistema comercial denominado Terminales Workstation III de la empresa MICROTECH, que también realiza la captura y procesamiento de información en el ramo restaurantero. Este sistema tiene la misma función del sistema descrito en este trabajo de tesis, cuenta con terminales inalámbricas de captura que realizan una comunicación vía Radio Frecuencias, el sistema puede transmitir las ordenes a una PC la cual realiza básicamente las mismas funciones de la PC del presente trabajo de tesis. Estas terminales de captura, son demasiado voluminosas para ser portadas cómodamente por el mesero, ya que son equipos que no fueron creados específicamente para esta aplicación, funcionan bajo una plataforma MS-DOS y tiene la estructura de una IBM-PC, la forma de introducir información a estas terminales es a través de varias "ventanas" de menús. En el caso de las terminales inalámbricas del presente trabajo, la captura de la información se efectúa a través de un teclado que directamente ejecuta la función deseada.

A continuación se presenta la comparación del precio de venta cotizado en 1996, de las terminales de captura desarrolladas por la empresa "MICROTECH" y las TCI's presentadas en este trabajo de tesis.

Descripción	Precio unitario
Terminales Workstation III, (Terminales inalámbricas con arquitectura IBM-PC que emplean una plataforma MS-DOS).	\$ 28,312.00
Terminales de captura inalámbrica presentadas en este trabajo de tesis	\$ 5,452.00

El sistema expuesto en este trabajo de tesis, se diseñó analizando las necesidades y la forma de operar de los restaurantes con el fin de que el sistema no desplazara el factor humano sino que lo eficientizara.

El objetivo del sistema es lograr la comunicación inalámbrica entre una PC y una o más terminales portátiles para capturar y procesar sin errores, la información manejada en los restaurantes.

La terminal de captura inalámbrica (TCI), del sistema que se expone en esta tesis tiene las ventajas de tener un tamaño compacto, lo que la hace más fácil de manipular, además de que la captura de información se efectúa directamente a través de comandos.

Con el empleo de radio frecuencias como medio de transmisión, se tiene la ventaja de que al realizar la captura de las peticiones del cliente, estas llegan inmediatamente al área de atención (cocina, bar, caja, etc.).

Debido a que el sistema trabaja con una computadora personal (PC), la programación de la misma puede extenderse tanto como lo requieran las necesidades del restaurante. De esta forma en la PC puede residir un programa para manejar funciones tales como:

- Envío de información a impresoras localizadas en cocinas, bares, islas de meseros, caja, etc., con el fin de presentar una forma impresa de la información capturada.
- Manejo de una PC operando en diferentes puntos del restaurante u hotel (bar, cocina, fuente de sodas, oficina, etc.), para visualizar la información capturada y enviar información a las TCI's.
- Realizar el control y la actualización de la existencia en almacén de acuerdo a las comandas (plátillos) pedidas.
- Generación de los reportes de ventas.

Además de la aplicación descrita en este trabajo de tesis, el sistema podría utilizarse en alguna otra área donde se requiera agilizar la captura y procesamiento de información. Tal es el caso de los talleres mecánicos de automóviles, donde el personal de servicio operaría las TCI's para efectuar las cotizaciones de refacciones y servicios necesarios para los automóviles que se revisan en ese preciso momento.

Para lograr que el sistema presentado en este trabajo tenga otra aplicación, se debe de efectuar el análisis correspondiente de las necesidades de la aplicación en cuestión, con el objeto de saber las modificaciones que se deben efectuar al sistema presentado.

APENDICE A

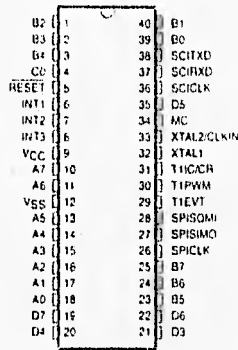
HOJAS DE ESPECIFICACIONES  
TECNICAS

## 2.2 TMS370Cx2x Pinouts and Pin Descriptions

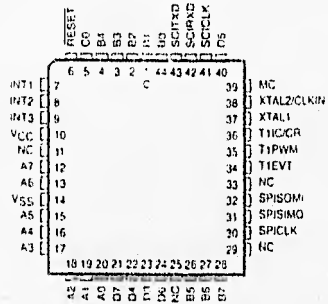
**2**

The pinouts and pin descriptions for the TMS370Cx2x devices are shown in Figure 2-2 and Table 2-2, respectively.

Figure 2-2. Pinouts for TMS370Cx2x



- A. 40-Pin PDIP (N)
- B. 40-Pin CDIP (JD)
- C. 40-Pin SDIP (N2)
- D. 40-Pin CSDIP (JC)



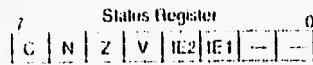
- E. 44-Pin PLCC (FN)
- F. 44-Pin CLCC (FZ)



Table 2-2. TMS370Cx2x Pin Descriptions

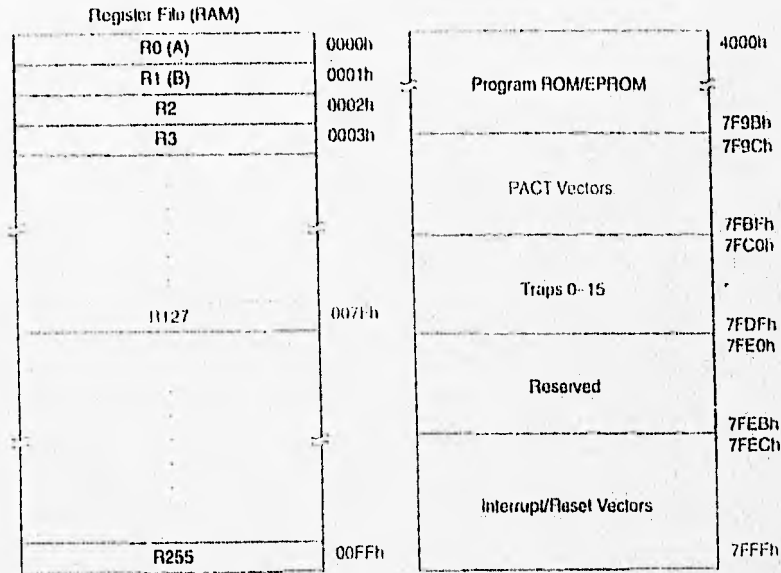
Name	Pin		I/O	Description
	No. DIP / PLCC			
	40	44		
A0	18	20	I/O	Port A is a general-purpose bidirectional I/O port
A1	17	19	I/O	
A2	16	18	I/O	
A3	15	17	I/O	
A4	14	16	I/O	
A5	13	15	I/O	
A6	11	13	I/O	
A7	10	12	I/O	
B0	30	34	I/O	Port B is a general-purpose bidirectional I/O port
B1	40	1	I/O	
B2	1	2	I/O	
B3	2	3	I/O	
B4	3	4	I/O	
B5	23	26	I/O	
B6	24	27	I/O	
B7	25	28	I/O	
C0	4	5	I/O	Port C is a general-purpose bidirectional I/O port
D3	21	23	I/O	Port D is a general-purpose bidirectional I/O port; D3 is also configurable as a CLKOUT
D4	20	22	I/O	
D5	35	10	I/O	
D6	22	24	I/O	
D7	19	21	I/O	
SCITXD	38	33	I/O	SCI transmit data output pin/general-purpose bidirectional pin
SCIRXD	37	32	I/O	SCI receive data input pin/general-purpose bidirectional pin
SCICLK	36	31	I/O	SCI bidirectional serial clock pin/general-purpose bidirectional pin
INT1	6	7	I	External nonmaskable or maskable interrupt/general-purpose input pin
INT2	7	8	I/O	External maskable interrupt input/general-purpose bidirectional pin
INT3	8	9	I/O	External maskable interrupt input/general-purpose bidirectional pin
T1IC/CR	31	36	I/O	Timer 1 input capture/counter reset input pin/general-purpose bidirectional pin
T1PWM	30	35	I/O	Timer 1 PWM output pin/general-purpose bidirectional pin
T1EVT	29	34	I/O	Timer 1 external event input pin/general-purpose bidirectional pin
SPISOMI	28	32	I/O	SPI slave output pin, master input pin/general-purpose bidirectional pin
SPISIMO	27	31	I/O	SPI slave input pin, master output pin/general-purpose bidirectional pin
SPICLK	26	30	I/O	SPI bidirectional serial clock pin/general-purpose bidirectional pin
RESET	5	6	I/O	System reset bidirectional pin; as an input, it initializes the microcontroller; as an open-drain output, it indicates an internal failure was detected by the watchdog or oscillator fault circuit
MC	34	39	I	Mode control input pin; enables the EEPROM write-protection override (WPO) mode
XTAL1	32	37	O	Internal oscillator output for crystal
XTAL2/CLKIN	33	38	I	Internal oscillator crystal input/external clock source input
NC	-	11		No connections
	-	25		
	-	29		
	-	33		
VCC	9	10		Positive supply voltage
VSS	12	14		Ground reference

Figure 3-1. Programmer's Model



Legend:

- C = Carry
- N = Negative
- Z = Zero
- V = Overflow
- IE2 = Level 2 Interrupts Enable
- IE1 = Level 1 Interrupts Enable



### 3.3 Memory Map

Figure 3-4 shows the memory map of the TMS370 family members. The partitioning of memory and physical location of memory (that is, on- or off-chip) depends on the device used and the memory mode of operation. The memory modes of operation are discussed in Section 3.4.

3

Each device that has memory expansion can be programmed to use up to sixteen address bits. This allows access of up to 56K bytes of memory. In addition, memory expansion features allow up to 112K bytes of external memory. (The expansion features are described further in subsection 3.4.2.) †

Figure 3-4. TMS370 Memory Map

0000h	256-Byte RAM (Register File/Stack)
00FFh	
0100h	
	RAM Expansion (On-Chip)†
1000h	Peripheral File
10BFh	
10C0h	Peripheral File Expansion
10FFh	Data EEPROM Expansion (On-Chip)
1100h	
1EFFh	256-Byte Data EEPROM
1F00h	
1FFFh	32K-Byte Program Memory Start or Microprocessor Mode Memory Expansion†
2000h	
4000h	6K-Byte Program Memory Start or Microprocessor Mode Memory Expansion
6000h	
6000h	8K-Byte Program Memory Start or Microprocessor Mode Memory Expansion
7000h	
7000h	4K-Byte Program Memory Start or Microprocessor Mode Memory Expansion
7000h	
7000h	2K-Byte Program Memory Start
7000h	
7F00h	Interrupt and Reset Vectors; Trap Vectors
7FFFh	
8000h	Memory Expansion/External Memory
FFFFh	

† In devices with more than 256 bytes of RAM, only the first 256-byte block can be used as registers/stack.

† In devices that have 32K bytes of program memory, it begins at 2000h and ends at 40FFh.

### 3.3.2 Peripheral File

The peripheral file (PF) is a set of memory-mapped registers that provide access to all internal peripheral modules, system-wide control functions, and EEPROM/EPROM programming control.

The PF includes 256 addresses in the memory map from 1000h-10FFh. The PF is divided into sixteen frames of 16 bytes each. Each peripheral frame is allocated its own set of control registers. In addition, some frames are dedicated to specific functions.

The instruction set includes some instructions which access the peripheral file directly. These instructions designate the register by the number of the file register relative to 1000h, preceded by P0 for a hexadecimal designator or P for a decimal designator. For example, the system configuration control register 0 is located at address 1010h; its peripheral file hexadecimal designator is P010, and its decimal designator is P16.

Table 3-1 shows the address map for the peripheral file.

Table 3-1. Peripheral File Address Map

Frame No.	Address	Description	TMS370C				
			x1x	x2x	x3x	x4x	x5x
0	1000h	Reserved for Factory Test	—	—	—	—	—
1	1010h	System and EEPROM / EPROM Control Registers	Yes	Yes	Yes	Yes	Yes
2	1020h	Digital I/O Port Control Registers	Yes	Yes	Yes	Yes	Yes
3	1030h	SPI Registers	Yes	Yes	NA	NA	Yes
4	1040h	Timer 1 Registers	Yes	Yes	NA	Yes	Yes
		PACT Registers	NA	NA	Yes	NA	NA
5	1050h	SCI Registers	NA	Yes	NA	Yes	Yes
6	1060h	Timer 2 Registers	NA	NA	NA	Yes	Yes
7	1070h	A-to-D Registers	NA	NA	Yes	Yes	Yes
8	1080h	Reserved	NA	NA	NA	NA	NA
9	1090h	Reserved	NA	NA	NA	NA	NA
10	10A0h	Reserved	NA	NA	NA	NA	NA
11	10B0h	Reserved	NA	NA	NA	NA	NA
12	10C0h	External Peripheral Control	NA	NA	NA	NA	Yes†
13	10D0h	External Peripheral Control	NA	NA	NA	NA	Yes†
14	10E0h	External Peripheral Control	NA	NA	NA	NA	Yes†
15	10F0h	External Peripheral Control	NA	NA	NA	NA	Yes†

NA = Not Available

† Does not apply to TMS370Cx58 devices or 64-pin devices.

Programming of the data EEPROM array is controlled by the data EEPROM control register (DEECTL) at memory address 101Ah (P01A) and the corresponding WPRs. See subsections 6.2.1 and 6.2.2 for more details on the WPR and DEECTL registers.

### 3.3.4 Program Memory

The program memory options available in the TMS370 family allow a wide selection of memory types: ROM or EPROM, ranging in size from 2K to 32K bytes. The program memory is arranged as individually addressable bytes in the memory map. Data can be read or code can be executed directly from these locations.

Memory addresses 7F9Ch through 7FBFh and 7FECCh through 7FFFh are reserved for interrupt and reset vectors. Trap vectors, used with TRAP0 through TRAP15 instructions, are at addresses 7FC0h through 7FDFh. Table 3-2 gives the memory map for the reserved vector locations and describes the differences among TMS370 family members.

Table 3-2. Vector Address Map

Address	Description	TMS370C					No. of K bytes
		x1x	x2x	x3x	x4x	x5x	
7F9Ch	PACT INT 1-10	NA	NA	Yes	NA	NA	36
7FC0h	Trap 0-15	Yes	Yes	Yes	Yes	Yes	32
7FE0h	Reserved	Yes	Yes	Yes	Yes	Yes	12
7FECCh	A/D Converter	NA	NA	Yes	Yes	Yes	2
7FEECh	Timer 2	NA	NA	NA	Yes	Yes	2
7FF0h	Serial Communications Interface TX	NA	Yes	NA	Yes	Yes	2
7FF2h	Serial Communications Interface RX	NA	Yes	NA	Yes	Yes	2
7FF4h	Timer 1	Yes	Yes	NA	Yes	Yes	2
7FF6h	Serial Peripheral Interface	Yes	Yes	NA	NA	Yes	2
7FF8h	Interrupt 3	Yes	Yes	Yes	Yes	Yes	2
7FFACh	Interrupt 2	Yes	Yes	Yes	Yes	Yes	2
7FFCCh	Interrupt 1	Yes	Yes	Yes	Yes	Yes	2
7FFEDh	Reset	Yes	Yes	Yes	Yes	Yes	2

NA - Not Available

Figure 4-3. Peripheral File Frame 2: Digital Port Control Registers

Designation	ADDR	PF	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
APORT1	1020h	P020	Reserved							
APORT2	1021h	P021	Port A Control Register 2							
ADATA	1022h	P022	Port A Data							
ADIR	1023h	P023	Port A Direction							
BPORT1	1024h	P024	Reserved							
BPORT2	1025h	P025	Port B Control Register 2							
BDATA	1026h	P026	Port B Data							
BDIR	1027h	P027	Port B Direction							
CPORT1	1028h	P028	Reserved							
CPORT2	1029h	P029	Port C Control Register 2							
CDATA	102Ah	P02A	Port C Data							
CDIR	102Bh	P02B	Port C Direction							
DPORT1	102Ch	P02C	Port D Control Register 1							
DPORT2	102Dh	P02D	Port D Control Register 2							
DDATA	102Eh	P02E	Port D Data							
DDIR	102Fh	P02F	Port D Direction							

Each port has up to four control registers associated with it. They are:

- Port X control register 1 (XPORT1)
- Port X control register 2 (XPORT2)
- Port X data (XDATA)
- Port X direction (XDIR)

The same bit position of each of these registers affects the corresponding bit in the port. For example, bit 0 of registers DPORT1, DPORT2, DDATA, and DDIR control port D, bit 0. This is illustrated in Figure 4-4.

Table 4-6. Port Configuration Registers Set-Up

Port	Pin	Input	Output	Function A ( $\mu$ C Mode)	Function B ( $\mu$ C Mode)	$\mu$ P Mode
		XPORT1 = 0† XPORT2 = 0 XDATA = y XDIR = 0	XPORT1 = 1† XPORT2 = 0 XDATA = q XDIR = 1	XPORT1 = 0† XPORT2 = 1 XDATA = x XDIR = x	XPORT1 = 1† XPORT2 = 1 XDATA = x XDIR = x	
A	0-7	Data IN y	Data OUT q	DATABUS	‡	DATABUS
B	0-7	Data IN y	Data OUT q	LOW ADDR	‡	LOW ADDR
C	0-7	Data IN y	Data OUT q	HI ADDR	‡	HI ADDR
D	0	Data IN y	Data OUT q	CSE2	OCF	OCF
D	1	Data IN y	Data OUT q	CSH3	‡	‡
D	2	Data IN y	Data OUT q	CSH2	‡	‡
D	3	Data IN y	Data OUT q	CLKOUT	CLKOUT	CLKOUT
D	4	Data IN y	Data OUT q	RAW	RAW	RAW
D	5	Data IN y	Data OUT q	CSPF	‡	‡
D	6	Data IN y	Data OUT q	CSH1	EDS	EDS
D	7	Data IN y	Data OUT q	CSE1	WAIT	WAIT

† XPORT1 = 1 and XPORT2 = 0, the function is not valid.

‡ XPORT1 exists for DPOR1 only.

‡ Reserved. Not available.

#### 4.4.1 Function A and Function B Signal Definitions

The following paragraphs define the memory expansion signals of function A and B.

##### CSE1

Chip-Select Eighth 1. This signal selects the first bank of memory. It has the same timing as  $\overline{EDS}$ , but it goes active only during accesses to an eighth of memory (locations 2000h-3FFFh). Setting this pin to a high-level general-purpose output disables the bank.

##### CSE2

Chip-Select Eighth 2. This signal selects a second bank of memory. It has the same timing as  $\overline{EDS}$ , but it goes active only during accesses to an eighth of memory (locations 2000h-3FFFh). Setting this pin to a high-level general-purpose output disables the bank.

##### CSPF

Chip-Select Peripheral File. This signal has the same timing as  $\overline{EDS}$ , but it goes active only during access to external frames of the peripheral file (locations 10C0h-10FFh).

##### CSH1

Chip-Select Half 1. This signal selects the first bank of memory. It has the same timing as  $\overline{EDS}$ , but it goes active only during access to the upper half of memory (locations 8000h-FFFFh). Setting this pin to a high-level general-purpose output disables the bank.

##### CSH2

Chip-Select Half 2. This signal selects a second bank of memory. It has the same timing as  $\overline{EDS}$ , but it goes active only during access to the upper half of memory (locations 8000h-FFFFh). Setting this pin to a high-level general-purpose output disables the bank.





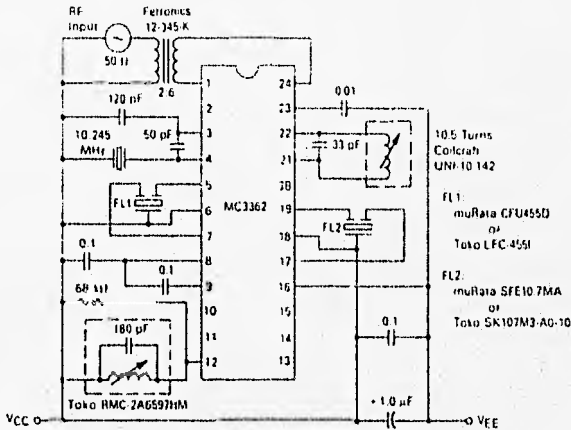
MAXIMUM RATINGS (T<sub>A</sub> = 25°C, unless otherwise noted)

Rating	Pin	Symbol	Value	Unit
Power Supply Voltage	6	V <sub>CC(max)</sub>	8.0	Vdc
Operating Supply Voltage Range (Recommended)	6	V <sub>CC</sub>	2.0 to 7.0	Vdc
Input Voltage (V <sub>CC</sub> = 5.0 Vdc)	1, 24	V <sub>1-24</sub>	1.0	Vrms
Junction Temperature	--	T <sub>J</sub>	150	°C
Operating Ambient Temperature Range	--	T <sub>A</sub>	-40 to +85	°C
Storage Temperature Range	--	T <sub>stg</sub>	-65 to +150	°C

ELECTRICAL CHARACTERISTICS (V<sub>CC</sub> = 5.0 Vdc, I<sub>Q</sub> = 49.7 MHz Deviation = 3.0 kHz, T<sub>A</sub> = 25°C, Test Circuit of Figure 3 unless otherwise noted)

Characteristic	Pin	Min	Typ	Max	Units
Drain Current (Carrier Detect Low — See Figure 5)	6	--	4.5	7.0	mA
Input for -3.0 dB Limiting	--	--	0.7	2.0	μVrms
Recovered Audio (RF signal level = 10 mV)	13	--	350	--	mVrms
Noise Output (RF signal level = 0 mV)	13	--	250	--	mVrms
Carrier Detect Threshold (below V <sub>CC</sub> )	10	--	0.64	--	Vdc
Meter Drive Slope	10	--	100	--	nA/dB
Input for 20 dB IS - NI-N (See Figure 7)	--	--	0.7	--	μVrms
First Mixer 3rd Order Intercept (Input)	--	--	-22	--	dBm
First Mixer Input Resistance (R <sub>p</sub> )	--	--	690	--	Ω
First Mixer Input Capacitance (C <sub>p</sub> )	--	--	7.2	--	pf
First Mixer Conversion Voltage Gain	--	--	18	--	dB
Second Mixer Conversion Voltage Gain	--	--	21	--	dB
Detector Output Resistance	13	--	1.4	--	kΩ

FIGURE 3 -- TEST CIRCUIT



**MC2833**

**LOW POWER FM TRANSMITTER SYSTEM**

MC2833 is a one-chip FM transmitter subsystem designed for cordless telephone and FM communication equipment. It includes a microphone amplifier, voltage controlled oscillator and two auxiliary transistors.

- Wide Range of Operating Supply Voltage (2.8-9.0 V)
- Low Drain Current:  $I_{CC} = 2.9 \text{ mA Typ}$
- Low Number of External Parts Required
- -30 dBm Power Output to 60 MHz Using Direct RF Output
- -10 dBm Power Output Attainable Using On-Chip Transistor Amplifiers

**LOW POWER  
 FM TRANSMITTER  
 SYSTEM**

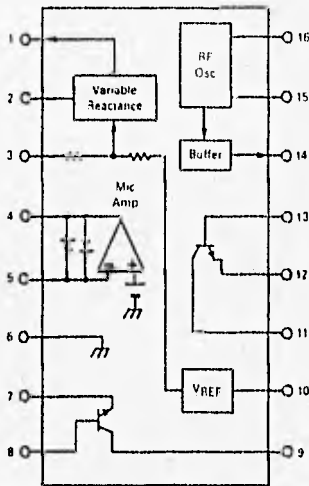


**P SUFFIX**  
 PLASTIC PACKAGE  
 CASE 648

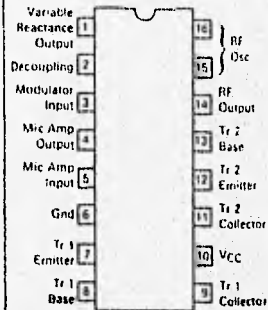


**D SUFFIX**  
 PLASTIC PACKAGE  
 CASE 751B  
 SO 16

**FUNCTIONAL BLOCK DIAGRAM**



**PIN ASSIGNMENTS**



**MAXIMUM RATINGS**

Rating†	Symbol	Value	Unit
Power Supply Voltage <sub>DC</sub>	V <sub>CC</sub>	10 (max)	V
Operating Supply Voltage Range	V <sub>CC</sub>	2.0-9.0	V
Junction Temperature	T <sub>J</sub>	+150	°C
Operating Ambient Temperature	T <sub>A</sub>	30 to -75	°C
Storage Temperature Range	T <sub>STG</sub>	65 to -150	°C

**ELECTRICAL CHARACTERISTICS (V<sub>CC</sub> = 4.0 V, I<sub>A</sub> = 25 C, unless otherwise noted)**

Characteristics	Symbol	Pin	Min	Typ	Max	Unit
Drain Current (No input signal)	I <sub>CC</sub>	10	1.7	2.9	4.3	mA

**FM MODULATOR**

Output RF Voltage (f <sub>o</sub> = 16.6 MHz)	V <sub>out RF</sub>	14	60	90	130	mVrms
Output DC Voltage (No input signal)	V <sub>DC</sub>	14	2.2	2.5	2.8	V
Modulation Sensitivity (f <sub>o</sub> = 16.6 MHz) (V <sub>in</sub> = 0.8 V to 1.2 V)	SEK	34	7.0	10	15	Hz/mVdc
Maximum Deviation (f <sub>o</sub> = 16.6 MHz) (V <sub>in</sub> = 0 V to 2.0 V)	Fdev	34	3.0	6.0	10	kHz

**MIC AMPLIFIER**

Closed Loop Voltage Gain (V <sub>in</sub> = 3.0 mVrms) (f <sub>in</sub> = 1.0 kHz)	A <sub>v</sub>	40	27	30	33	dB
Output DC Voltage (No input signal)	V <sub>out DC</sub>	40	1.1	1.4	1.7	V
Output Swing Voltage (V <sub>in</sub> = 30 mVrms) (f <sub>in</sub> = 1.0 kHz)	V <sub>out P-P</sub>	40	0.8	1.2	1.6	Vp-p
Total Harmonic Distortion (V <sub>in</sub> = 3.0 mVrms) (f <sub>in</sub> = 1.0 kHz)	THD	40	—	0.15	2.0	%

**AUXILIARY TRANSISTOR STATIC CHARACTERISTICS**

Characteristics	Symbol	Min	Typ	Max	Unit
Collector Base Breakdown Voltage (I <sub>C</sub> = 5.0 μA)	V <sub>BRICBQ</sub>	15	45	—	V
Collector Emitter Breakdown Voltage (I <sub>C</sub> = 200 μA)	V <sub>BRICEC</sub>	10	15	—	V
Collector Substrate Breakdown Voltage (I <sub>C</sub> = 50 μA)	V <sub>BRICSO</sub>	—	70	—	V
Emitter Base Breakdown Voltage (I <sub>E</sub> = 50 μA)	V <sub>BRIEBO</sub>	—	6.2	—	V
Collector Base Cutoff Current (V <sub>CB</sub> = 10 V, I <sub>E</sub> = 0)	I <sub>CBO</sub>	—	—	200	nA
DC Current Gain (I <sub>C</sub> = 3.0 mA, V <sub>CE</sub> = 3.0 V)	h <sub>FE</sub>	40	150	—	—

**AUXILIARY TRANSISTOR DYNAMIC CHARACTERISTICS**

Current Gain Bandwidth Product (V <sub>CE</sub> = 3.0 V, I <sub>C</sub> = 3.0 mA)	f <sub>T</sub>	—	500	—	MHz
Collector Base Capacitance (V <sub>CE</sub> = 3.0 V, I <sub>C</sub> = 0)	C <sub>CB</sub>	—	2.0	—	pF
Collector Substrate Capacitance (V <sub>CS</sub> = 3.0 V, I <sub>C</sub> = 0)	C <sub>CS</sub>	—	3.3	—	pF

## Single Chip 300-Baud Modem

The MC145442 and MC145443 silicon-gate CMOS single-chip low-speed modems contain a complete frequency shift keying (FSK) modulator, demodulator, and filter. These devices are with CCITT V.21 (MC145442) and Bell 103 (MC145443) specifications. Both devices provide full-duplex or half-duplex 300-baud data communication over a pair of telephone lines. They also include a carrier detect circuit for the demodulator section and a duplexer circuit for direct operation on a telephone line through a simple transformer.

- MC145442 Compatible with CCITT V.21
- MC145443 Compatible with Bell 103
- Low-Band and High-Band Band-Pass Filters On-Chip
- Simplex, Half-Duplex, and Full-Duplex Operation
- Originate and Answer Mode
- Analog Loopback Configuration for Self Test
- Hybrid Network Function On-Chip
- Carrier Detect Circuit On-Chip
- Adjustable Transmit Level and CD Delay Timing
- On-Chip Crystal Oscillator (3.579 MHz)
- Single - 5 V Power Supply Operation
- Internal Mid-Supply Generator
- Power-Down Mode
- Pin Compatible with MM74HC943
- Capable of Driving -9 dBm into a 600  $\Omega$  Load

**MC145442**  
**MC145443**



P SUFFIX  
 PLASTIC  
 CASE 738

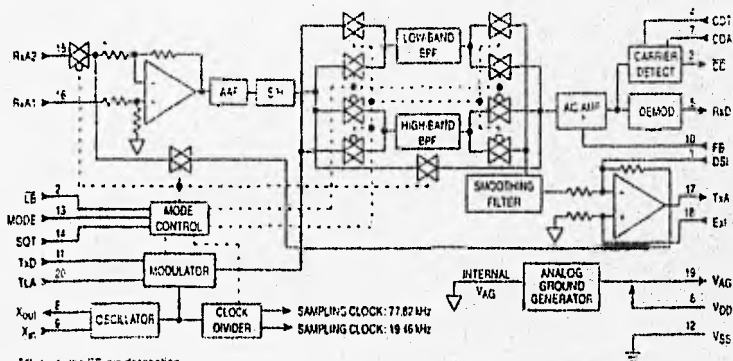


DW SUFFIX  
 SOG  
 CASE 751D

### PIN ASSIGNMENT

DSI	1	20	T1A
TB	2	19	VAG
CD	3	18	E+
CCT	4	17	T1A
RxD	5	16	RxA1
VDD	6	15	RxA2
CDA	7	14	SG1
V <sub>QA</sub>	8	13	MODE
V <sub>PA</sub>	9	12	VSS
FB	10	11	TxD

### BLOCK DIAGRAM



**ABSOLUTE MAXIMUM RATINGS** (Voltages Referenced to V<sub>SS</sub>)

Rating	Symbol	Value	Unit
Supply Voltage	V <sub>DD</sub>	-0.5 to +7.0	V
DC Input Voltage	V <sub>in</sub>	-0.5 to V <sub>DD</sub> + 0.5	V
DC Output Voltage	V <sub>out</sub>	-0.5 to V <sub>DD</sub> + 0.5	V
Clamp Diode Current, per Pin	I <sub>IK, IOK</sub>	±20	mA
DC Output Current, per Pin	I <sub>out</sub>	±28	mA
Power Dissipation	P <sub>D</sub>	500	mW
Operating Temperature Range	T <sub>A</sub>	-40 to +85	°C
Storage Temperature Range	T <sub>stg</sub>	-65 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high impedance circuit. For proper operation it is recommended that V<sub>in</sub> and V<sub>out</sub> be constrained to the range V<sub>SS</sub> ≤ (V<sub>in</sub> or V<sub>out</sub>) ≤ V<sub>DD</sub>.

Unused inputs must always be tied to an appropriate logic voltage level (e.g., either V<sub>SS</sub> or V<sub>DD</sub>).

**RECOMMENDED OPERATING CONDITIONS**

Parameter	Symbol	Min	Max	Unit
Supply Voltage	V <sub>DD</sub>	4.5	5.5	V
DC Input or Output Voltage	V <sub>in</sub> , V <sub>out</sub>	0	V <sub>DD</sub>	V
Input Rise or Fall Time	t <sub>r</sub> , t <sub>f</sub>	—	500	ns
Crystal Frequency	f <sub>crystal</sub>	2.2	5.0	MHz

Changing the crystal frequency from 3.579 MHz will change the output frequencies. The change in output frequency will be proportional to the change in crystal frequency.

**DC ELECTRICAL CHARACTERISTICS** (V<sub>DD</sub> = 5.0 V ± 10%, T<sub>A</sub> = -40°C to +85°C)

Characteristic	Symbol	Min	Typ	Max	Unit
High-Level Input Voltage X <sub>in</sub> , Tx0, Mode: SOT	V <sub>IH</sub>	V <sub>DD</sub> + 0.8 3.15	—	—	V
Low-Level Input Voltage X <sub>in</sub> , Tx0, Mode: SOT	V <sub>IL</sub>	—	—	0.8 1.1	V
High-Level Output Voltage I <sub>OH</sub> = 20 μA I <sub>OH</sub> = 2 mA I <sub>OH</sub> = 20 μA	V <sub>OIH</sub>	V <sub>DD</sub> - 0.1 3.7 —	—	— — V <sub>DD</sub> - 0.05	V
Low-Level Output Voltage I <sub>OL</sub> = 20 μA I <sub>OL</sub> = 2 mA I <sub>OL</sub> = 20 μA	V <sub>OIL</sub>	—	—	0.1 0.4 —	V
Input Current RxA1, RxA2 X <sub>in</sub>	I <sub>in</sub>	—	— 10 —	±1.0 ±12 ±10	μA
Quiescent Supply Current (X <sub>in</sub> or I <sub>crystal</sub> = 3.579 MHz)	I <sub>DD</sub>	—	7	10	mA
Power-Down Supply Current	I <sub>DDP</sub>	—	200	300	μA
Input Capacitance All CMOS Inputs	C <sub>in</sub>	—	10 —	— 10	pF
V <sub>AG</sub> Output Voltage (I <sub>O</sub> = ±10 μA)	V <sub>AG</sub>	2.4	2.5	2.6	V
CDA Output Voltage (I <sub>O</sub> = ±10 μA)	V <sub>CDA</sub>	1.1	1.2	1.3	V
Line Driver Feedback Resistor	R <sub>f</sub>	10	20	30	kΩ

### AC ELECTRICAL CHARACTERISTICS

(V<sub>DD</sub> = 5.0 V ± 10%, T<sub>A</sub> = -40°C to +85°C, Crystal Frequency = ± 0.79 MHz ± 0.1%. See Figure 1)

Characteristic	Min	Typ	Max	Unit
<b>TRANSMITTER</b>				
Power Output on TxA R <sub>L</sub> = 1.2 kΩ RTLA = ∞ R <sub>L</sub> = 1.2 kΩ RTLA = 5.5 kΩ	-13 -10	-12 -3	-11 -8	dBm
Second Harmonic Power R <sub>L</sub> = 1.2 kΩ	---	-5%	---	dBm
<b>RECEIVE FILTER AND HYBRID</b>				
Hybrid input Impedance RxA1, RxA2	40	50	---	kΩ
FB Output Impedance	---	16	---	kΩ
Adjacent Channel Rejection	-48	---	---	dBm
<b>DEMODULATOR</b>				
Receive Carrier Amplitude	-48	---	-12	dBm
Dynamic Range	---	36	---	dB
Bit Jitter (S/N = 30 dB, input = -38 dBm, Bit Rate = 300 baud)	---	100	---	ps
Bit Bias	---	5	---	%
Carrier Detect Threshold (CDA = 1.2 V or CDA grounded through a 0.1 μF capacitor)	On to Off Off to On	---	-46 -47	dBm

### PIN DESCRIPTIONS

#### VDD

Positive Power Supply (Pin 6)  
This pin is normally tied to 5.0 V

#### VSS

Negative Power Supply (Pin 12)  
This pin is normally tied to 0 V

#### VAG

Analog Ground (Pin 19)  
Analog ground is internally biased to (VDD - VSS)/2. This pin must be decoupled by a capacitor from VAG to VSS and a capacitor from VAG to VDD. Analog ground is the common bias line used in the switched capacitor filter, amplifier, and slicer in the demodulation circuitry.

#### TLA

Transmit Level Adjust (Pin 20)

This pin is used to adjust the transmit level. Transmit level adjustment range is typically from -12 dBm to -8 dBm. (See Applications Information.)

#### TxD

Transmit Data (Pin 11)

Binary information is input to the transmit data pin. Data entered for transmission is modulated using FSK techniques. A logic high input level represents a mark and a logic low represents a space (see Table 1).

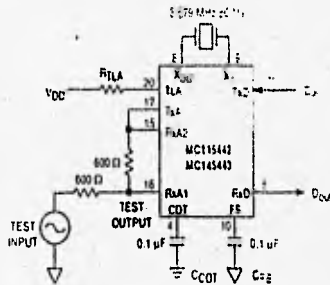


Figure 1. AC Characteristics Evaluation Circuit

Table 1. Bell 103 and CCITT V.21 Frequency Characteristics

Data	Bell 103 (MC145443)			
	Originate Mode		Answer Mode	
	Transmit	Receive	Transmit	Receive
Space	1070 Hz	2025 Hz	2025 Hz	1070 Hz
Mark	1270 Hz	2225 Hz	2225 Hz	1270 Hz
Data	CCITT V.21 (MC145442)			
	Originate Mode		Answer Mode	
	Transmit	Receive	Transmit	Receive
Space	1180 Hz	1850 Hz	1850 Hz	1180 Hz
Mark	980 Hz	1650 Hz	1650 Hz	980 Hz

NOTE: Actual frequencies may be ± 5 Hz assuming 3.579545-MHz crystal is used.

**TxA****Transmit Carrier (Pin 17)**

This is the output of the line driver amplifier. The transmit carrier is the digitally synthesized sine wave output of the modulator derived from a crystal resonator reference. When a 3.570 MHz crystal is used the frequency outputs shown in Table 1 apply. (See *Applications Information*.)

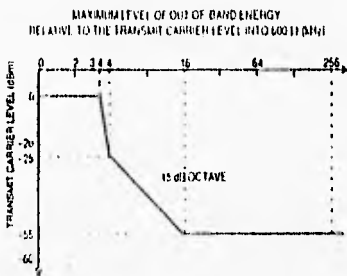


Figure 2 Out-of-Band Energy

**Ext****External Input (Pin 18)**

The external input is the non-inverting input to the line driver. It is provided to combine an auxiliary audio signal or speech signal in the phone line using the line driver. This pin should be connected to V<sub>AG</sub> if not used. The average level must be the same as V<sub>AG</sub> to maintain proper operation. (See *Applications Information*.)

**DSI****Driver Summing Input (Pin 1)**

The driver summing input may be used to connect an external signal, such as a DTMF dialer, to the phone line. A series resistor, R<sub>DST</sub>, is needed to define the voltage gain A<sub>v</sub> (see *Applications Information* and Figure 6). When applying a signal to the DSI pin, the modulator should be squelched by bringing SQT (Pin 14) to a logic high level. The voltage gain, A<sub>v</sub>, is calculated by the formula  $A_v = -R_1/R_{DST}$  (where R<sub>1</sub> = 20 kΩ). For example, a 20 kΩ resistor for R<sub>DST</sub> will provide unity gain ( $A_v = -20\text{ k}\Omega/20\text{ k}\Omega = -1$ ). This pin must be left open if not used.

**RxD****Receive Data (Pin 6)**

The receive data output pin presents the digital binary data resulting from the demodulation of the receive carrier. If no carrier is present, CDT high, the receive data output (RxD) is clamped high.

**RxA2, RxA1****Receive Carrier (Pins 15, 16)**

The receive carrier is the FSK input to the demodulator through the receive band-pass filter. RxA1 is the non-inverting input and RxA2 is the inverting input of the receive hybrid (duplexer) operational amplifier.

**LB****Analog Loopback (Pin 2)**

When a high level is applied to this pin (SQT must be low), the analog loopback test is enabled. The analog loopback test connects the TxA pin to the RxA2 pin and the RxA1 to analog ground. In loopback, the modulator frequencies are switched to the modulation frequencies for the selected mode. (See Tables 1 and 2 and Figures 4c and 4d.)

When LB is connected to analog ground (V<sub>AG</sub>), the modulator generates an echo cancellation tone at 2100 Hz for MC145442 CCITT V.21 and 2225 Hz for MC145443 Bell 103 systems. For normal operation, this pin should be at a logic low level (VSS).

The power-down mode is enabled when both LB and SQT are connected to a logic high level (see Table 2).

Table 2. Functional Table

MODE Pin 13	SQT Pin 14	LB Pin 2	Operating Mode
1	0	0	Originate Mode
0	0	0	Answer Mode
X	0	V <sub>AG</sub> (VDD/2)	Echo Tone
X	0	1	Analog Loopback
X	1	0	Squelch Mode
X	1	V <sub>AG</sub> (VDD/2)	Squelch Mode
X	1	1	Power Down

**MODE****Mode (Pin 13)**

This input selects the pair of transmit and frequencies used during modulation and demodulation. When a logic high level is placed on this input, originate (Bell) or channel 1 (CCITT) is selected. When a low level is placed on this input, answer (Bell) or channel 2 (CCITT) is selected. (See Tables 1 and 2 and Figure 4.)

**CDT****Carrier Detect Timing (Pin 4)**

A capacitor on this pin to VSS sets the amount of time the carrier must be present before CDT goes low (see *Applications Information* for the capacitor values).

**CD****Carrier Detect Output (Pin 3)**

This output is used to indicate when a carrier has been sensed by the carrier detect circuit. This output goes to a logic low level when a valid signal above the maximum threshold level (defined by CDA, Pin 7) is maintained at the input to the hybrid circuit longer than the response (defined by CDT, Pin 4). This pin is held at the logic low level until the signal falls below the minimum threshold level for longer than the turn off time. (See *Applications Information* and Figure 5.)

**CDA****Carrier Detect Adjust (Pin 7)**

An external voltage may be applied to this pin to adjust the carrier detect threshold. The threshold hysteresis is internally fixed at 3 dB (see *Applications Information*).

### 7KLT High Frequency LC Filters Triple Tuned

Frequency Range: 10MHz-150MHz  
Inductance Range: 0.1 $\mu$ H-50 $\mu$ H  
Optional Capacitance Values: 5-100pF



#### Quality Features:

- Triple tuned for very high selectivity. Profile is only 10.5mm high.
  - for LAN, RF transceiver, cordless phone, CATV, TVRO, data transmission, low, high and band pass filter and other applications requiring very high Q. Typically 70-100.
- Coil form has many narrow, closely-spaced horizontal slots and four vertical slots to bring windings down to the terminals. This assures minimum stray capacitance and yields a high Q. It also facilitates duplicating the electrical characteristics more precisely for consistent impedance, center frequency, Q and inductance.
- Coil bobbin is very accessible for prototyping.
- Polycarbonate coil forms are designed with an internally "spined" body matched to the tuning slug providing excellent torque control. By maintaining precision tolerances throughout fabrication, TOKO can guarantee the applications for torque range. Once aligned in the circuit, TOKO coils will maintain characteristics during the PCB assembly process. Locking screws thus can be eliminated.
- Terminals feature rounded ends to facilitate PCB insertion and presoldered to ensure a reliable PCB connection.
- One piece construction makes these coils highly resistant to shock, vibration, humidity and temperature change.
- Expose base to minimal heat during wave soldering. Consult TOKO AMERICA if using etchior than freon cleaning solvents.

### STANDARD DEVICES SELECTION GUIDE

The Part Numbers shown in this table below are standard devices, which are readily available. TOKO will design and manufacture modified and custom devices with specific characteristics to meet your requirements. If you do not find the device for your application in this catalog, please see Modified and Custom Request Form located in the rear of this catalog.

#### Band Pass

TOKO Part Number	Center Frequency (MHz)	Bandwidth (MHz)	Impedance ( $\Omega$ )
335BXS-0050	10	1.5(3dB)	330/330
335BBX-0113	25.5	1.5(3dB)	50/50
335BBS-0034	43	5.5(1dB)	75/75
335BBS-0063	44	12 (1dB)	50/50
335BBV-0223	46.8	0.4(3dB)	50/50
335BBV-0224	49.8	0.4(3dB)	50/50
335BBX-0054	50.75	8 (1dB)	75/75
335BBV-0016	70	20 (1dB)	75/75
335BBS-0055	77.2	6 (3dB)	50/50
335BBS-0058	83.2	6 (3dB)	50/50
335BBS-0057	86.9	6 (3dB)	50/50
335BPV-0038	130	20 (1dB)	75/75

#### Low Pass & High Pass Filters

TOKO Part Number	Cutoff Frequency (MHz)	1st Pole Frequency (MHz)	Stopband Attenuation (dB)	Impedance ( $\Omega$ )
335BHX-0048	58	45	35	75/75 HPF
335BHX-0048	60	49	35	75/75 HPF
335BLV-0112	80	100	40	50/50 LPF
335BLX-0047	80	90	30	75/75 LPF
335BLX-0049	84	95	25	75/75 LPF

continued on next page



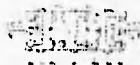
102 TYPE  
7KLT

LC Filters

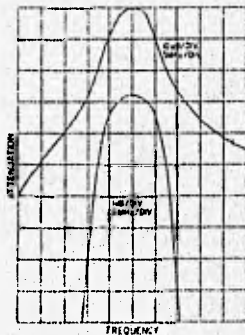
R&TOKO

continued from previous page

335DXS-0055



335BXS-0057



# KAF - Cordless Phone Series SAW Filters

## 46 and 49 MHz - Low Insertion Loss SAW Filters

### Features

- 1) Adjustable trim
- 2) High reliability
- 3) Low insertion loss

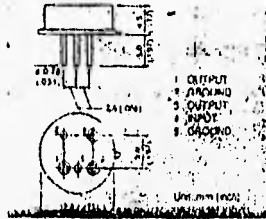
### How To Order

**KAF - 46 NR-ME**

- ①    ②    ③

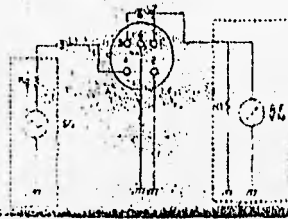
- ① Model (Kyocera Acoustic Filter)
- ② Frequency
- ③ Type number

### Dimensions



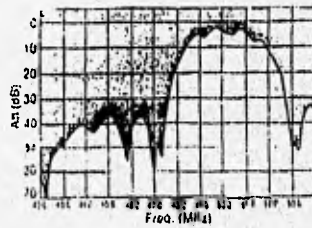
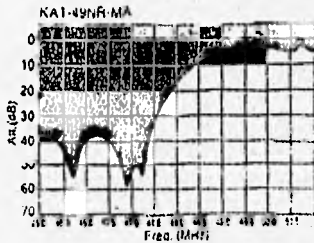
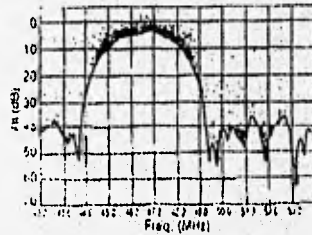
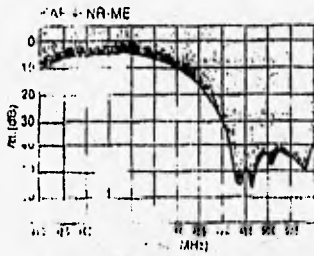
### Test Circuit KAF-46NR-ME/KAF-49NR-MA

	46	49
L1 (H)	0.47	0.39
L2 (H)	0	0.30



Part No.	Insertion Loss (45.61 - 46.97MHz)	Attenuation (49.67 - 49.99MHz)
KAF-46NR-ME	6.0 dB max.	30 dB min.
KAF-49NR-MA	30 dB min.	6.0 dB max.

### Characteristics



APENDICE B

LISTADO DEL PROGRAMA DEL  
MICROCONTROLADOR DEL  
MODULO CENTRAL Y DEL  
PROGRAMA DE LA PC

```

;*****
;*****
;*****
;****
;****          PROGRAMA PARA MICROCONTROLADOR DEL MODULO CENTRAL          ****
;****
;*****
;*****
;*****
;*****
;DECLARACION DE REGISTROS
;*****
INT1      .equ P017 ; REGISTRO DE INTERRUPCION 1
CONPC     .equ P029 ; " DE CONTROL DE PUERTO C
DIRPC     .equ P02B ; " " DIRECCION DE PUERTO C
DATAPC    .equ P02A ; " " DATOS " " "
CONPA     .equ P021 ; " " CONTROL DE PUERTO A
DATAPA    .equ P022 ; " " DATOS " " "
DIRPA     .equ P023 ; " " DIRECCION DE PUERTO A
CONPB     .equ P025 ; " " CONTROL DE PUERTO B
DATAPB    .equ P026 ; " " DATOS " " "
DIRPB     .equ P027 ; " " DIRECCION " " "
DIRPD     .equ P02F ; " " " " " D
DATAPD    .equ P02E ; " " DATOS " " "
CONPD1    .equ P02C ; " 1 " CONTROL " " C
CONPD2    .equ P02D ; " 2 " " " " C

TIEVT     .equ P04D ; TIMER 1 EVT CONTROL Y DATOS
SCICCR    .equ P050 ; " " CONTROL UART (No.DE BITS,PARIDAD,SINC,ETC)
SCICTL    .equ P051 ; " " " " (SLEEP,HABILITACIONTX-RX,ETC)
BAUDMSB   .equ P052 ; " " VELOCIDAD DE UART (BYTE MAS SIGNIFICATIVO)
BAUDLSB   .equ P053 ; " " " " ( " MENOS " )
TXCTL     .equ P054 ; " " CONTROL Y ESTADO TX
RXCTL     .equ P055 ; " " " " " RX
RXBUF     .equ P057 ; " " RECEPCION DATO UART
TXBUF     .equ P059 ; " " TRANSMISION " "
SCIPC1    .equ P05D ; " " CONTROL DE RELOJ
SCIPC2    .equ P05E ; " " DEFINICION DE PINES TXD, RXD COMO I/O
SCIPRI    .equ P05F ; " " CONTROL DE PRIORIDAD UART
B300      .equ 520 ; VELOCIDAD DE TRANSMISION 300 BAUDS
PRUEBDIS  .equ R2 ; CONTADOR DE RUTINA DE RECEPCION
DATOSAL   .equ R3 ; DATO A TRANSMITIR POR PUERTO SERIE
BANDERAL1 .equ R4 ; REGISTRO 1 DE BANDERAS DE CONTROL
DATO      .equ R5 ; REGISTRO DE DATO RECIBIDO EN SCI
CONTB     .equ R6 ; CONTADOR DE RUTINA DE RECEPCION
MAFPOL    .equ R8 ; DIRECCION DE NUMERO DE TCI A POLEAR
CARIMEN   .equ R9 ; AREA (R9,R10 y R11) DE INF. ENVIADA POR TCI (CARMEN)
CAR2MEN   .equ R10
CAR3MEN   .equ R11
BYTECHEC  .equ R12 ; REGISTRO DE BYTE FORMADO PARA CHEQUEO DE ERROR
NMMC      .equ R13 ; REGISTRO DE NUMERO DE TCI EN MENSAJES COLECTIVOS
CONT1     .equ R29 ; CONTADORES PARA MEDIR TIEMPO DE ESPERA DE DATOS DE TCI
CONT2     .equ R30
CONT3     .equ R31
ADMB      .equ R32 ; AREA DE MENSAJES-BIP ENVIADOS POR BUS
MEMMEN    .equ R80 ; REGISTRO QUE APUNTA AL 1er DATO A TX DE ULT. MEN-BIP
BANMBIP   .equ R81 ; BANDERAS DE CONTROL DE MEN-BIP A TX (1,2 o 3)
NUMAPPOL  .equ R82 ; NUMERO DE TCI's A POLEAR (ENVIADO POR PC)
AMMC      .equ R83 ; AREA DE MEMORIA DE MENSAJES COLECTIVOS
BANMCOL1  .equ R99 ; BANDERAS DE MEMORIA COLECTIVA ENVIADA bO=TCIO, etc

```



```

BTJO #020h,BANDERA1,INIPOL ; SI ENTRO PROGRAMA RESETEA TCI'S
BTJO #07h,BANMBIP,BMENBIP ; CHECA SI HAY MENSAJES BIP
BTJO #080h,BANDERA1,INICIO
INC MAFPOL ; APUNTA AL SIGUIENTE TCI
CMP MAFPOL,NUMAFPOL ; CHECA SI NO HA PASADO EL NUMERO DE TCIS
JGE E6POLMAFO ; POLEAR
MOV #00h,MAFPOL ; SI PASO APUNTA AL PRIMER TCI A POLEAR
E6POLMAFO: AND #0F0h,BANDERA1 ; LIMPIA BANDERAS DE CONTROL DE COMUNICACION
CALL TRANSDIR ; TRANSMITE LA DIRECCION A TCI
MOV #0A8h,DATOSAL ; CARGA CLAVE QUE PREGUNTA SI HAY INF. PARA PC
CALL TRANSDAT ; TRANSMITE CLAVE
ES1: BTJZ #040h,TXCTL,ES1
MOV #01h,CONT2
ESP2: MOV #0F7h,CONT1 ; CONTADOR DE ESPERA PARA ENC. TX
ESP1: DJNZ CONT1,ESP1
DJNZ CONT2,ESP2
MOV #00h,SCICTL
MOV #037h,SCICTL
MOV #01h,RXCTL ; ENCIENDE INT RX PTO.SERIE
MOV #0FFh,CONTB ; RUTINA DE ESPERA DE 130 mS PARA SALIR
E2POLMAFO: MOV #0FFh,PRUEBDIS ; DE RUTINA SI NO LLEGAN DATOS
EPOLMAFO: BTJO #0Ah,BANDERA1,CONFDAT ; CHECA SI LLEGO RESP. (HAY INF. O NO)
BTJO #05h,BANDERA1,E6POLMAFO ; SI LLEG DAT MAL A TCI LO VUELVE A TX
; O SI LOS DAT. DE LLEGARON MAL VUELVE A PREGUNTAR
DJNZ PRUEBDIS,EPOLMAFO ; RECARGA DE CONTADOR
DJNZ CONTB,E2POLMAFO ; RECARGA DE CONTADOR
MOV #00h,SCICTL ;*****RESETEAR SCI
MOV #00h,RXCTL
MOV #036h,SCICTL
CONFDAT: BTJO #02h,BANDERA1,EOPOLMAFO ; SI NO HAY RESPUESTA SE VA A POLEAR LA SIGUIENTE TCI
JMP POLMAFO ; SI HAY INF. SE VA A RECIBIRLA
EOPOLMAFO: MOV #03h,CONTB ; SI NO HAY INF. SE VA A POLEAR LA SIG. TCI
MOV #03h,PRUEBDIS ; CARGA REGISTRO QUE CUENTAN EL NUMERO DE
MOV #00h,B ; DATOS A RECIBIR
AND #0F0h,BANDERA1 ; LIMPIA BANDERA DE ERROR EN DATOS RECIBIDOS
MOV #05h,CONT1 ; INICIALIZA CONTADOR DE 0.6 S
E4POLMAFO: MOV #0FFh,CONT2
E5POLMAFO: MOV #0FFh,CONT3
E1POLMAFO: BTJO #01h,BANDERA1,EPOLMAFO ; CHECA SI HAY ERROR EN DATOS RECIBIDOS
DJNZ CONT3,E3POLMAFO ; RECARGA DE REGISTROS DE CONTADOR
DJNZ CONT2,E5POLMAFO
DJNZ CONT1,E4POLMAFO
JMP POLMAFO ; SI SOBREPASA EL CONT. VA A POLEAR OTRA TCI
BMENBIP: JMP MENBIP ; SALTO A MENSAJE BIP (ESCALA -NO ALCANZA SOLO-)
E3POLMAFO: CMP CONTB,PRUEBDIS ; CHECA SI LLEGO UN DATO DE LA TCI
JEQ E1POLMAFO ; SI NO LLEGO SIGUE ESPERANDO
MOV DATO,A ; SI LLEGO DATO GUARDALO
MOV A,*CARIMEN[B] ; EN EL AREA DE RAM DESTINADO PARA ELLO
INC B ; INCREMENTA APUNTADEOR DE RAM
DJNZ PRUEBDIS,E1POLMAFO ; DECREMENTA CONTADOR DE DATOS
BTJO #040h,BANDERA1,NOHAYSIS ; SI PROG. SALIO ENVIAR MENSAJE
MOV CARIMEN,CONTB ; FORMACION DE CARACTER DE CHEQUEO DE ERROR
OR MAFPOL,CONTB ; OP1=(MAFPOL) OR (CARIMEN)
MOV CAR3MEN,BYTECHEC ; OP2=(CAR2MEN) OR (CAR3MEN)
OR CAR2MEN,BYTECHEC ; BYTECHEC=(OP1) AND (OP2)
AND CONTB,BYTECHEC
MOV #00h,CONTB ; LIMPIAR DE REGISTRO AUXILIAR DE FORMACION
MOV #00h,INT1 ; DESHABILITA INT1 PARA LEER DEL BUS SIN INT.
TRANS: MOV #0FFh,DIRPA ; PUERTO DE COMUNICACION CON BUS SALIDA DIG.

```

```

MOV MAPPOL,DATAPA ; EL PRIMER DATO A TX A LA PC ES EL # DE TCI
OR #020h,DATAPD ; SE HACE LA INTERRUPCION A LA PC
TRANS1: BTJO #010h,DATAPD,TRANS1A ; ESPERA A QUE LA PC LEA EL DATO DEL BUS
BTJO #080h,BANDERA1,INICIO
JMP TRANS1
TRANS1A: AND #0D7h,DATAPD ; LIMPIA LINEA DE INT. DE LA PC Y RESETEA EL FF
OR #08h,DATAPD ; PONE AL FLIP-FLOP EN ESTADO DE ESPERA NUEVAMENTE
MOV CAR1MEN,DATAPA ; MANDA EL PRIMER CARACTER DE INF. DE LA TCI
TRANS2: BTJO #010h,DATAPD,TRANS2A ; ESPERA A QUE LA PC LEA EL DATO
BTJO #080h,BANDERA1,INICIO
JMP TRANS2
TRANS2A: AND #0F7h,DATAPD ; RESETEA EL FLIP-FLOP DE LECTURA DEL BUS
OR #08h,DATAPD ; PONE AL FLIP-FLOP EN ESTADO DE ESPERA
MOV CAR2MEN,DATAPA ; MANDA EL SEGUNDO CARACTER DE INF. DE LA TCI
TRANS3: BTJO #010h,DATAPD,TRANS3A ; ESPERA A QUE LA PC LEA EL DATO
BTJO #080h,BANDERA1,INICIO
JMP TRANS3
TRANS3A: AND #0F7h,DATAPD ; RESETEA EL FLIP-FLOP DE LECTURA DEL BUS
OR #08h,DATAPD ; PONE AL FLIP-FLOP EN ESTADO DE ESPERA
MOV CAR3MEN,DATAPA ; MANDA EL TERCER CARACTER DE INF. DE LA TCI
TRANS4: BTJO #010h,DATAPD,TRANS4A ; ESPERA A QUE LA PC LEA EL DATO
BTJO #080h,BANDERA1,INICIO
JMP TRANS4
TRANS4A: AND #0F7h,DATAPD ; RESETEA EL FLIP-FLOP DE LECTURA DEL BUS
OR #08h,DATAPD ; PONE AL FLIP-FLOP EN ESTADO DE ESPERA
MOV BYTECHEC,DATAPA ; MANDA EL CARACTER DE CHEQUE DE ERRORES
TRANS5: BTJO #010h,DATAPD,TRANS5A ; ESPERA A QUE LA PC LEA EL DATO
BTJO #080h,BANDERA1,INICIO
JMP TRANS5
TRANS5A: AND #0F7h,DATAPD ; RESETEA EL FLIP-FLOP DE LECTURA DEL BUS
OR #08h,DATAPD ; PONE AL FLIP-FLOP EN ESTADO DE ESPERA
RECEPPC: MOV #00h,B ; RESETEA REGISTRO APUNT. DE ALMACENAJE DE INF
MOV #00h,DIRPA ; PUERTO A DEFINIDO COMO ENTRADA PARA LEER BUS
LEGODATO: BTJO #040h,INT1,LEGODATO1 ; CHECA SI LLEGO DATO (I/OW*ADDR="1")
BTJO #080h,BANDERA1,INICIO
JMP LEGODATO
LEGODATO1: MOV DATAPA,NUMDAT ; LEE DATO 1 DEL BUS Y GUARDA EN ADM
CMP #0FFh,NUMDAT ; CHECA SI HUBO ERROR EN TRANSMISION A PC
JEQ TRANS ; SI ES ASI VUELVE A TRANSMITIR
CMP #0A3h,NUMDAT
JEQ SALPREF
MOV NUMDAT,CONTB ; CARGA EL NUMERO DE DATOS A ESPERAR
AND #0FBh,T1EVT ; RESETEA FLIP-FLOP QUE INDICA LECTURA DE DATO
OR #04h,T1EVT ; ENCIENDE FLIP-FLOP PARA ESPERAR OTRO DATO
LLEGODATO: BTJO #040h,INT1,LLEGODATA ; CHECA SI LLEGO DATO (I/OW*ADDR="1")
BTJO #080h,BANDERA1,INICIO
JMP LLEGODATO
LLEGODATA: MOV DATAPA,A ; LEE DATO DEL BUS
MOV A,*ADM[B] ; GUARDA DATO EN AREA DE INF. RECIBIDA DE PC
INC B ; APUNTA AL SIGUIENTE REGISTRO DEL AREA DE INF.
AND #0FBh,T1EVT ; RESETEA FLIP-FLOP QUE INDICA LECTURA DE DATO
DJNZ CONTB,ELLEGDAT ; DECREMENTA CONT. DE DATOS RECIBIDOS
OR #04h,T1EVT ; ENCIENDE FLIP-FLOP PARA ESPERAR OTRO DATO
MOV #05h,INT1 ; ENCIENDE INT1 PARA ESPERAR MENSAJE-BIP
CALL TXMAF ; LLAMA RUTINA DE TRANSMISION A TCI
JMP POLMAFO ; SALTA A POLEAR A LA SIGUIENTE TCI
ELLEGDAT: OR #04h,T1EVT ; ENCIENDE FLIP-FLOP PARA ESPERAR OTRO DATO
JMP LLEGODATO ; SALTA A SEGUIR ESPERANDO DATOS DEL BUS
TXMAF: CALL TIME
CALL TRANSDIR ; TRANSMITE DIRECCION A TCI

```

```

MOV MAPOL,DATAPA ; EL PRIMER DATO A TX A LA PC ES EL # DE TCI
OR #020h,DATAPD ; SE HACE LA INTERRUPCION A LA PC
TRANS1: BTJO #010h,DATAPD,TRANS1A ; ESPERA A QUE LA PC LEA EL DATO DEL BUS
BTJO #080h,BANDERA1,INICIO
JMP TRANS1
TRANS1A: AND #0D7h,DATAPD ; LIMPIA LINEA DE INT. DE LA PC Y RESETEA EL FF
OR #08h,DATAPD ; PONE AL FLIP-FLOP EN ESTADO DE ESPERA NUEVAMENTE
MOV CARIMEN,DATAPA ; MANDA EL PRIMER CARACTER DE INF. DE LA TCI
TRANS2: BTJO #010h,DATAPD,TRANS2A ; ESPERA A QUE LA PC LEA EL DATO
BTJO #080h,BANDERA1,INICIO
JMP TRANS2
TRANS2A: AND #0F7h,DATAPD ; RESETEA EL FLIP-FLOP DE LECTURA DEL BUS
OR #08h,DATAPD ; PONE AL FLIP-FLOP EN ESTADO DE ESPERA
MOV CAR2MEN,DATAPA ; MANDA EL SEGUNDO CARACTER DE INF. DE LA TCI
TRANS3: BTJO #010h,DATAPD,TRANS3A ; ESPERA A QUE LA PC LEA EL DATO
BTJO #080h,BANDERA1,INICIO
JMP TRANS3
TRANS3A: AND #0F7h,DATAPD ; RESETEA EL FLIP-FLOP DE LECTURA DEL BUS
OR #08h,DATAPD ; PONE AL FLIP-FLOP EN ESTADO DE ESPERA
MOV CAR3MEN,DATAPA ; MANDA EL TERCER CARACTER DE INF. DE LA TCI
TRANS4: BTJO #010h,DATAPD,TRANS4A ; ESPERA A QUE LA PC LEA EL DATO
BTJO #080h,BANDERA1,INICIO
JMP TRANS4
TRANS4A: AND #0F7h,DATAPD ; RESETEA EL FLIP-FLOP DE LECTURA DEL BUS
OR #08h,DATAPD ; PONE AL FLIP-FLOP EN ESTADO DE ESPERA
MOV BYTECHEC,DATAPA ; MANDA EL CARACTER DE CHEQUE DE ERRORES
TRANS5: BTJO #010h,DATAPD,TRANS5A ; ESPERA A QUE LA PC LEA EL DATO
BTJO #080h,BANDERA1,INICIO
JMP TRANS5
TRANS5A: AND #0F7h,DATAPD ; RESETEA EL FLIP-FLOP DE LECTURA DEL BUS
OR #08h,DATAPD ; PONE AL FLIP-FLOP EN ESTADO DE ESPERA
RECEPPC: MOV #00h,B ; RESETEA REGISTRO APUNT. DE ALMACENAJE DE INF
MOV #00h,DIRPA ; PUERTO A DEFINIDO COMO ENTRADA PARA LEER BUS
LEGODATO: BTJO #040h,INT1,LEGODATO1 ; CHECA SI LLEGO DATO (I/OW*ADDR="1")
BTJO #080h,BANDERA1,INICIO
JMP LEGODATO
LEGODATO1: MOV DATAPA,NUMDAT ; LEE DATO 1 DEL BUS Y GUARDALO EN ADM
CMP #0FFh,NUMDAT ; CHECA SI HUBO ERROR EN TRANSMISION A PC
JEQ TRANS ; SI ES ASI VUELVE A TRANSMITIR
CMP #0A3h,NUMDAT
JEQ SALPREF
MOV NUMDAT,CONTB ; CARGA EL NUMERO DE DATOS A ESPERAR
AND #0FBh,TIEVT ; RESETEA FLIP-FLOP QUE INDICA LECTURA DE DATO
OR #04h,TIEVT ; ENCIENDE FLIP-FLOP PARA ESPERAR OTRO DATO
LLEGODATO: BTJO #040h,INT1,LLEGODATA ; CHECA SI LLEGO DATO (I/OW*ADDR="1")
BTJO #080h,BANDERA1,INICIO
JMP LLEGODATO
LLEGODATA: MOV DATAPA,A ; LEE DATO DEL BUS
MOV A,*ADM[B] ; GUARDA DATO EN AREA DE INF. RECIBIDA DE PC
INC B ; APUNTA AL SIGUIENTE REGISTRO DEL AREA DE INF.
AND #0FBh,TIEVT ; RESETEA FLIP-FLOP QUE INDICA LECTURA DE DATO
DJNZ CONTB,ELLEGDAT ; DECREMENTA CONT. DE DATOS RECIBIDOS
OR #04h,TIEVT ; ENCIENDE FLIP-FLOP PARA ESPERAR OTRO DATO
MOV #05h,INT1 ; ENCIENDE INT1 PARA ESPERAR MENSAJE-BIP
CALL TXMAF ; LLAMA RUTINA DE TRANSMISION A TCI
JMP POLMAFO ; SALTA A POLEAR A LA SIGUIENTE TCI
ELLEGDAT: OR #04h,TIEVT ; ENCIENDE FLIP-FLOP PARA ESPERAR OTRO DATO
JMP LLEGODATO ; SALTA A SEGUIR ESPERANDO DATOS DEL BUS
TXMAF: CALL TIME ; TRANSMITE DIRECCION A TCI
CALL TRANSDIR ; TRANSMITE DIRECCION A TCI

```



```

MOV NUMDAT,CONTB      ; INICIALIZA CONTADOR DE DATOS TRANSMITIDOS 16
MOV #00h,B            ; " APUNTADOR DE AREA DE INF. A TX
TMENSAJ: BTJO #04h,BANDERA1, TXMAF ; SI HAY ERROR EN TX VUELVE A TX TODOS
MOV *ADM[B],A        ; TRANSMITE DATO AL QUE APUNTA B
MOV A, TXBUF
ETXMENSAJ: BTJZ #080h, TXCTL, ETXMENSAJ ; CHECA SI YA SE ENVIO TODO EL BYTE
INC B                ; APUNTA AL SIGUIENTE DATO A ENVIAR
DJNZ CONTB, TMENSAJ ; DECREMENTA EL CONT. DE DATOS TX. ENV. EL SIG.
RTS                  ; SI TERMINO DE ENVIARLOS SAL DE ROUTINA
SALPREF: AND #0FBh, TIEVT
OR #04h, TIEVT
MOV #05h, INT1
JMP POLMAFO
;*****
; AREA DE LETREROS
;*****
NOHAYSIS: CALL TIME
CALL TRANSDIR        ; TRANSMITE DIRECCION A TCI
BTJO #04h, BANDERA1, NOHAYSIS
MOV #04Eh, DATOSAL   ; TRANSMITE "N"
CALL TRANSDAT
BTJO #04h, BANDERA1, NOHAYSIS
MOV #04Fh, DATOSAL   ; TRANSMITE "O"
CALL TRANSDAT
BTJO #04h, BANDERA1, NOHAYSIS
MOV #020h, DATOSAL   ; TRANSMITE " "
CALL TRANSDAT
BTJO #04h, BANDERA1, NOHAYSIS
MOV #048h, DATOSAL   ; TRANSMITE "H"
CALL TRANSDAT
BTJO #04h, BANDERA1, NOHAYSIS
MOV #041h, DATOSAL   ; TRANSMITE "A"
CALL TRANSDAT
BTJO #04h, BANDERA1, NOHAYSIS
MOV #059h, DATOSAL   ; TRANSMITE "Y"
CALL TRANSDAT
BTJO #04h, BANDERA1, NOHAYSIS
MOV #020h, DATOSAL   ; TRANSMITE " "
CALL TRANSDAT
BTJO #04h, BANDERA1, NOHAYSIS
MOV #053h, DATOSAL   ; TRANSMITE "S"
CALL TRANSDAT
BTJO #04h, BANDERA1, NOHAYSIS
MOV #049h, DATOSAL   ; TRANSMITE "I"
CALL TRANSDAT
BTJO #04h, BANDERA1, NOHAYSIS
MOV #053h, DATOSAL   ; TRANSMITE "S"
CALL TRANSDAT
BTJO #04h, BANDERA1, NOHAYSIS
MOV #054h, DATOSAL   ; TRANSMITE "T"
CALL TRANSDAT
BTJO #04h, BANDERA1, NOHAYSIS
MOV #045h, DATOSAL   ; TRANSMITE "E"
CALL TRANSDAT
BTJO #04h, BANDERA1, NOHAYSIS
MOV #04Dh, DATOSAL   ; TRANSMITE "M"
CALL TRANSDAT
BTJO #04h, BANDERA1, NOHAYSIS
MOV #041h, DATOSAL   ; TRANSMITE "A"
CALL TRANSDAT

```

```

BTJO #04h,BANDERA1,NOHAYSIS
MOV #020h,DATOSAL ; TRANSMITE " "
CALL TRANSDAT
BTJO #04h,BANDERA1,NOHAYSIS
MOV #020h,DATOSAL ; TRANSMITE " "
CALL TRANSDAT
BTJO #04h,BANDERA1,NOHAYSIS
JMP POLMAFO
;*****
; RUTINA DE TRANSMISION DE MENSAJES-BIP A TCI'S
;*****
MENBIP: AND #0E7h,BANDERA1
MOV MEMMEN,B ; CARGA POSICION DE # TCI A TX
DEC B
MOV *ADMB[B],A ; LEER NUMERO DE TCI
MOV A,MAFPOL ; CARGAR NUMERO DE TCI EN TCI A POLEAR
CMP #0FFh,MAFPOL ; CHECA SI EL MENSAJE ES PARA TODOS LAS TCI'S
JEQ E3MENBIP ; SI ES ASI MANDARLOS A TODOS
CALL E2MENBIP ; SI ES PARA UNO EN ESPECIFICO MANDARLO A ESE
; BTJO #08h,BANMBIP,HAYMCOI
JMP POLMAF
E2MENBIP: CALL TRANSDIR ; TRANSMITE DIRECCION DE TCI A POLEAR
MOV #0A4h,DATOSAL ; TRANSMITE CARACT. PARA PREGUNTAR A TCI SI SE
CALL TRANSDAT ; TRANSMITE EL MENSAJE-BIP
ES2: BTJZ #040h,TXCTL,ES2
MOV #01h,CONT2
ESP3: MOV #0F7h,CONT1
ESP4: DJNZ CONT1,ESP4
DJNZ CONT2,ESP3
MOV #00h,SCICTL
MOV #037h,SCICTL
MOV #01h,RXCTL
MOV #0FFh,CONTB
E5MENBIP: MOV #0FFh,PRUEBDIS
EMENBIP: BTJO #010h,BANDERA1,E4MENBIP ; ESPERA A QUE LA TCI PERMITA LA TX
BTJO #08h,BANDERA1,EMEN7BIP
DJNZ PRUEBDIS,EMENBIP
DJNZ CONTB,E5MENBIP
JMP EMEN7BIP
E4MENBIP: CALL TIME
; LOCAL: JMP LOCAL
CALL TRANSDIR ; TRANSMITE DIRECCION DE TCI
MOV MEMMEN,B ; APUNTADOR DE DATO A TRANSMITIR
SUB #010h,B
MOV #0Fh,CONTB ; INICIALIZA NUMERO DE DATOS A TRANSMITIR
E0MENBIP: BTJO #04h,BANDERA1,E2MENBIP ; CHECA SI NO HUBO ERROR EN LA TX DE INF.
MOV *ADMB[B],A ; SACA DATO+APUNTADOR DE MEMORIA
MOV A,TXBUF ; MUEVE DATO A REGISTRO DE TRANSMISION
E1MENBIP: BTJZ #080h,TXCTL,E1MENBIP ; CHECA SI YA TRANSMITIO TODOS LOS BITS
INC B ; APUNTA AL SIGUIENTE DATO
DJNZ CONTB,E0MENBIP ; DECREMENTA EL CONT. DE DATOS TX
MOV #020h,DATOSAL ; COMO ULTIMO CARACT. TX UN BLANCO " "
CALL TRANSDAT
BTJO #010h,BANMBIP,EMEN8BIP
CALL CHECBAND
JMP E7MENBIP
CHECBAND: CMP #010h,MEMMEN ; APAGA BANDERA QUE INDICA QUE HAY
JNE E6MENBIP ; "MENSAJE 1"
AND #0Feh,BANMBIP
RTS

```

```

E6MENBIP:  CMP #020h, MEMMEN      ; APAGA BANDERA QUE INDICA QUE HAY
           JNE E8MENBIP          ; "MENSAJE 2"
           AND #0FDh, BANMBIP
           RTS
E8MENBIP:  CMP #030h, MEMMEN      ; APAGA BANDERA QUE INDICA QUE HAY
           JNE E8MENBIP          ; "MENSAJE 3"
           AND #0FBh, BANMBIP
           RTS
EMENBBIP:  RTS
E7MENBIP:  SUB #010h, MEMMEN      ; APUNTA A SIGUIENTE MENSAJE A TRANSMITIR
EMEN7BIP:  AND #0E7h, BANDERA1    ; LIMPIA BANDERA DE CONTROL DE TX MEN-BIP
           RTS                  ; REGRESA DE SUBRRUTINA
E3MENBIP:  MOV #00h, NMHC
           OR #010h, BANMBIP
E3MENB:    MOV NMHC, MAFPOL        ; MANDA MENSAJE COLECTIVO A TODOS LAS TCIS
           CALL E2MENBIP
           INC NMHC
           CMP NUMAFPOL, NMHC
           JL  E3MENB
           AND #0EFh, BANMBIP
           CALL CHECBAND
           CALL E7MENBIP
           JMP POLMAFO

```

\*\*\*\*\*

;RUTINA DE TIEMPO DE ESPERA

\*\*\*\*\*

```

TIEMPO:    MOV #020h, CONT1
TIE1:      MOV #0FFh, CONT2
TIE2:      MOV #0FFh, CONT3
TIE3:      DJNZ CONT3, TIE3
           DJNZ CONT2, TIE2
           DJNZ CONT1, TIE1
           RTS
TIME:      MOV #020h, CONT1
TIM1:      MOV #0FFh, CONT2
TIM2:      DJNZ CONT2, TIM2
           DJNZ CONT1, TIM1
           RTS

```

\*\*\*\*\*

;LECTURA DE NUMERO DE TCI's A POLEAR DE BUS PC

\*\*\*\*\*

```

; LEERNMAF:  MOV #00h, INT1        ; DESHABILITA LA INTERRUPCION DE I/OW
;           MOV #0FFh, DIRPA      ; DEFINE PUERTO A COMO SALIDA DIGITAL
;           MOV #0A4h, DATAPA     ; TX CHARACTER "MANDAME # DE TCI's A POLEAR"(0A4h)
;           OR #020h, DATAPD      ; ENCIENDE INTERRUPCION DE BUS DE PC
;LNMAF1:    BTJZ #010h, DATAPD, LNMAF1 ; ESPERA A QUE BUS LEA DATO
;           AND #0D7h, DATAPD    ; RESETEA FLIP-FLOP I/OR
;           OR #08h, DATAPD      ; ENCIENDE INT. PARA ESPERAR OTRO DATO
;           MOV #00h, DIRPA      ; DEFINE PUERTO COMO ENTRADA DIGITAL
;LNMAF2:    BTJZ #040h, INT1, LNMAF2 ; ESPERA QUE BUS PC PONGA UN DATO AL uc
;           MOV DATAPA, NUMAFPOL ; GUARDA EL DATO EN EL REGISTRO DE # TCI
;           AND #0FBh, TIEVT     ; RESETEA FLIP-FLOP DE IO/W
;           OR #04h, TIEVT       ; ENCIENDE FLIP-FLOP PARA ESPERAR OTRO DATO
;           MOV #05h, INT1       ; ENCIENDE INT. PARA ESPERAR MEN-BIP

```

\*\*\*\*\*

;RUTINA DE ATENCION A LA INTERRUPCION 1 (LLEGADA DE MENSAJES-BIP DEL BUS

\*\*\*\*\*

```

RESPINT1:  MOV #00h, INT1        ; APAGAR INTERRUPCION PARA RECIBIR DATOS
           PUSH B                ; GUARDAR REGISTRO EMPLEADOS EN RUTINA EN

```

```

PUSH CONTB ; LA PILA
CMP #030h, MEMMEN ; CHECA SI ESTA LLENA LA MEMORIA
JNE EDATMEN ; SI ES ASI APUNTA AL PRIMER REGISTRO DE LA
MOV #020h, MEMMEN ; DE LA MEMORIA
EDATMEN: RESETEa REGISTRO APUNT. DE ALMACENAJE DE INF
MOV MEMMEN, B ; RESETEa REGISTRO DE # DE DATOS A RECIBIR
MOV #010h, CONTB ; INICIALIZA REGISTRO DE # DE DATOS A RECIBIR
MOV #00h, DIRPA ; PUERTO A DEFINIDO COMO ENTRADA PARA LEER BUS
MOV DATAPA, A ; LEE PRIMER DATO ENVIADO POR PC
CMP #0A0h, A ; CHECA SI EL BYTE ES ENTRO PROGRAMA CHEFFSYS
JNE TOMDAT ; SI NO ES VE APROBAR OTRA OPCION
OR #020h, BANDERA1 ; SI ES ASI ENCIENDE BANDERA ENTRO PROG
AND #0F0h, T1EVT ; RESETEa FLIP-FLOP QUE INDICA LECTURA DE DATO
ESPDA: OR #04h, T1EVT ; ENCIENDE FLIP-FLOP PARA ESPERAR OTRO DATO
BTJZ #040h, INT1, ESPDA ; ESPERA EL SIGUIENTE DATO
MOV DATAPA, NUMAFPOL ; GUARDA EL DATO EN NUMERO DE TCI's A POLEAR
SALRPC: AND #0F0h, T1EVT
OR #04h, T1EVT ; ENCIENDE FLIP-FLOP PARA ESPERAR OTRO DATO
POP CONTB ; REGRESA VALORES EMPLEADOS EN RUTINA
POP B
JMP E2DATMEN ; SALE DE INTERRUPCION
TOMDAT: CMP #0A1h, A ; CHECA SI LA TECLA ES SALIR DE PROGRAMA
JNE TOMDA ; SI NO ES ASI SALE DE RUTINA
OR #040h, BANDERA1 ; SI ES ASI ENCIENDE BANDERA SALIO PROGRAMA
JMP SALRPC ; SAL DE PROGRAMA
TOMDA: CMP #0A2h, A ; RECIBE CARACTER DE RESET
JNE TOMADATO1
OR #080h, BANDERA1
JMP SALRPC
DATMEN: BTJZ #040h, INT1, DATMEN ; CHECA SI LLEGO DATO (I/OW*ADDR="1")
TOMADATO: MOV DATAPA, A ; LEE DATO DEL BUS
TOMADATO1: MOV A, *ADMB(B) ; GUARDA DATO EN AREA DE INF. RECIBIDA DE PC
INC B ; APUNTA AL SIGUIENTE REGISTRO DEL AREA DE INF.
AND #0F0h, T1EVT ; RESETEa FLIP-FLOP QUE INDICA LECTURA DE DATO
DJNZ CONTB, ELLEGDA ; DECREMENTA CONT. DE DATOS RECIBIDOS
OR #04h, T1EVT ; ENCIENDE FLIP-FLOP PARA ESPERAR OTRO DATO
MOV B, MEMMEN
POP CONTB ; REGRESAR VALORES DE PILA A LOS REGISTROS
POP B
CMP #010h, MEMMEN ; ENCIENDE BANDERA QUE INDICA QUE HAY
JNE E1DATMEN ; "MENSAJE 1"
OR #01h, BANMBIP
JMP E2DATMEN
E1DATMEN: CMP #020h, MEMMEN ; ENCIENDE BANDERA QUE INDICA QUE HAY
JNE E3DATMEN ; "MENSAJE 2"
OR #02h, BANMBIP
JMP E2DATMEN
E3DATMEN: CMP #030h, MEMMEN ; ENCIENDE BANDERA QUE INDICA QUE HAY
JNE E2DATMEN ; "MENSAJE 3"
OR #04h, BANMBIP
E2DATMEN: MOV #05h, INT1 ; ENCIENDE INTERRUPCION PARA ESPERAR OTRO
RTI ; REGRESA DE INTERRUPCION
ELLEGDA: OR #04h, T1EVT ; ENCIENDE FLIP-FLOP PARA ESPERAR OTRO DATO
JMP DATMEN ; SALTA A SEGUIR ESPERANDO DATOS DEL BUS
;*****
; RUTINA TRANSMITE DIRECCION Y DATOS A TCI's
;*****
TRANSDIR: OR #08h, SCICTL ; ENCIENDE BIT QUE AVISA QUE SE TRANSMITE DIRECCION
MOV MAFFPOL, TXBUF ; TRANSMITE DIRECCION DE TCI A POLEAR
ETTRDIR1: BTJZ #080h, TXCTL, ETTRDIR1 ; ESPERA HAS'A QUE TX TODOS LOS BITS
RTS

```

```

TRANSDAT:  MOV DATOSAL, TXBUF ; TRANSMITE A TCI EL DATO GUARDADO EN DATOSAL
ETTRDAT:   BTJZ #080h, TXCTL, ETTRDAT
           RTS
;*****
; RUTINA DE RECEPCION DE DATOS ENVIADOS POR TCI's
;*****
RECIBDAT:  PUSH B ; GUARDA CONTENIDO DE REGISTRO "B" EN PILA
           BTJZ #02h, RXCTL, TABDAT ; CHECA SI NO ESTA EN ESTADO DE "SLEEP"
           MOV RXBUF, B ; SI NO ESTA EN SLEEP CHECA SI LA DIRECCION
           CMP NUMTAR, B ; ES LA CORRECTA DEL MODULO CENTRAL
           JNE SALRECIB ; SI NO ES IGUAL SAL DE INTERRUPCION S/CAMBIAR SLEEP
           AND #0F3h, SCICTL ; SI ES LA DIRECCION DE MOD-CENT SAL DE EDO. DE SLEEP
           JMP SALRECIB ; SAL DE INTERRUPCION
TABDAT:    PUSH A ; GUARDA REGISTRO "A" EN PILA
           BTJZ #080h, RXCTL, NOERROR ; CHECA SI HUBO ERROR EN TX
ERRTAB:    CALL TRANSDIR ; SI HAY ERROR MANDA DIRECCION DE TCI
           MOV #0A9h, DATOSAL ; MANDA COMANDO ERROR EN TRANSMISION (0A9h)
           CALL TRANSDAT ; SE TRANSMITE EL COMANDO ERROR EN RECEPCION
           OR #01h, BANDERA1 ; ENCIENDE BANDERA ERROR EN RECEPCION DE DATOS
FINETDAT:  POP A ; REGRESA VALOR A REGISTRO "A" GUARDADO EN PILA
           JMP SALRECIB1 ; SAL DE INTERRUPCION
NOERROR:   MOV RXBUF, A ; SI NO HAY ERROR TRANSFERIR DATO A REGISTRO "A"
           CMP #0A7h, A ; CHECA SI SE CONTESTA QUE VIENE INF. DE MAF O QUE
           JNE ERRTX ; SE ENVIE PREFERENCIAS, SI NO ES PRUEBA OTRA OPCION
           OR #02h, BANDERA1 ; ENCIENDE BANDERA QUE INDICA QUE LLEGO CARACT. A7
           ; CHECAR CUANDO SE HAGA LA RUT. DE ENVIAR PREFEREN
           ; APAGAR TX (MANDAR OTRO CARACT. PARA DIFERENCIARLO)
           ; *****
           POP A
           JMP SALRECIB ; SAL DE INT. SIN ENTRAR A SLEEP
ERRTX:     CMP #0A9h, A ; CHECA SI LLEGARON MAL LOS DATOS ENVIADOS A LA TCI
           JNE NOINF ; SI NO ES ASI PRUEBA OTRA OPCION
           OR #04h, BANDERA1 ; ENCIENDE BANDERA ERROR EN DATOS TX A PC.
           JMP FINETDAT ; SALIR DE INT. Y ENTRAR A SLEEP
NOINF:     CMP #0A6h, A ; CHECA SI SE AVISA QUE NO HAY INF. EN TCI O QUE NO
           JNE ENVMEN ; ENVIE MEN-BIP O PREF. SI NO ES PRUEBA OTRO CARACT.
           OR #08h, BANDERA1 ; ENCIENDE BANDERA DE AVISO DE CARACT. A6
           JMP FINETDAT ; SALIR DE INT. Y ENTRAR A SLEEP
ENVMEN:    CMP #0A4h, A ; CHECA SI ES CARACTER QUE INDICA QUE SI SE PUEDE
           JNE INFORM ; MANDAR MENSAJE-BIP A TCI
           OR #010h, BANDERA1 ; ENCIENDE BANDERA CORRESPONDIENTE A CARACTER A4
           JMP FINETDAT ; SALIR DE INT. Y ENTRAR A SLEEP
INFORM:    AND #0FBh, BANDERA1 ; APAGA BANDERA ERROR EN DATOS TRANSMITIDOS
           MOV A, DATO ; GUARDA INFORMACION EN REGISTRO DATO
           POP A ; REGRESA VALOR DE REGISTRO "A" GUARDADO EN PILA
           DJNZ CONTB, SALRECIB ; DECREMENTA CONTADOR DE DATOS RECIBIDOS
SALRECIB1: MOV #036h, SCICTL ; SI LLEGARON TODOS LOS DATOS ENTRA A SLEEP
           MOV #00h, RXCTL
           POP B
           RTI
SALRECIB:  POP B ; REGRESA VALOR DE "B" DE LA PILA
           AND #03h, RXCTL ; LIMPIA BANDERA RWAKE
           RTI ; REGRESO DE INTERRUPCION
;*****
.sect "VECTORS", 07FF0h
.word INICIO ; DECLARA ARRANQUE DE PROGRAMA EN "INICIO"
.word RECIBDAT ; EN INTERRUPCION DE RX IR A RUTINA "RECIBDAT"
.word INICIO
.word INICIO

```

.word INICIO  
.word INICIO  
.word RESPINT1  
.word INICIO

```

/*****
 *
 * PROGRAMA PARA COMUNICAR A LA PC CON LAS TCI A TRAVES DEL BUS-SLOT
 *
 *****/

```

//APERTURA DE LIBRERIAS

```

#include <stdio.h> //
#include <bios.h> //
#include <dos.h> // Archivos con las funciones de biblioteca
#include <string.h> // empleadas en el programa.
#include <iostream.h> //
#include <conio.h> //

```

//DECLARACION DE CONSTANTES

```

#define NUMTCI 16//Número total de TCI's.
#define PUERTOTCI 640//Número de puerto para comunicación con TCI's.
#define WAITTCI 100//Límite de tiempo de espera para lectura del puerto.
#define WAITPC 300//Límite de tiempo de espera para escribir al puerto.
#define INTR 0X0A//Número de interrupción.

```

```

void interrupt ( *oldhandler)(...)//Apuntador para la función manejadora-
//de interrupción original.
void interrupt handler(...)//Declaración de la función handler como-
//el nuevo manejador de interrupción.

```

//FUNCION QUE ENVIA UN MENSAJE DE ERROR EN LA APERTURA DE ARCHIVOS

```

void Error(void)
{
windows(23,13,55,15);
cprintf(" Archivo imposible de abrir! ");
cprintf(" Oprima <ENTER> ");
getch();
return;
}

```

//FUNCION QUE TRANSMITE UNA CADENA DE CARACTERES A LA TCI

```

int TxTCI(char* st)
//Inicio de la función que transmite una cadena de caracteres a la TCI.
{
int i,j;
for (i=0;i<=15;i++)
{
for (j=0;j<=WAITPC;j++);
outportb(PUERTOTCI,st[i]);//Envío de la cadena de caracteres-
//bdos(0x02,st[i],0); //a la TCI vía el bus-slot de la PC.
}
return(0);
}

```

//FUNCION QUE COPIA UNA CADENA DE CARACTERES EN UN ARREGLO

```

void mystrepy(char* des,char* sou)
//Inicio de función que copia una cadena de caracteres en un arreglo.
{
int i = 0;
while (sou[i] != '\0') des[i] = sou[i++];
}

```

//FUNCION QUE GUARDA EN UN ARCHIVO LOS MENSAJES TRANSMITIDOS

int Archiv(char\* st)

//Inicio de función que guarda en un archivo los mensajes transmitidos

```
{
    FILE* aa;
    char car;
    if((aa=fopen("datos.dat","a"))= =NULL)Error( );
    for (int i=0,i<=15;i++){
        car=st[i];
        putc(car,aa);
    }
    putc(13,aa);
    putc(10,aa);
    fclose(aa);
    return(0);
}
```

//FUNCION QUE BORRA UN ARREGLO DE 16 CARACTERES

int Borrar(char\* st)

//Inicio de función que borra un arreglo de 16 caracteres.

```
{
    for (int j=0;j<16;j++) st[j] = ' ';
    return(0);
}
```

//FUNCION QUE GUARDA EN UN ARCHIVO, EL FRAME DE DATOS

int Archiv(int co)

//Inicio de función que guarda en un archivo, los datos leídos-

//del bus-slot de la PC.

```
{
    FILE* aa;
    if((aa=fopen("datos2.dat","a"))= =NULL)Error( );
    putc(co,aa);
    putc(13,aa);
    putc(10,aa);
    fclose(aa);
    return(0);
}
```

//FUNCION QUE GUARDA EN UN ARCHIVO EL NUMERO DE MESA QUE SE ATIENDE

void Route(int p)

//Inicio de función que guarda en un archivo, el número de-

//la mesa que se atiende.

```
{
    FILE *x;
    if(( x=fopen("Route.dat","w"))= =NULL)Error( );
    putw(p,x);
    fclose(x);
}
```



```
//FUNCION QUE GUARDA EN UN ARCHIVO LOS PRECIOS DE LOS PLATILLOS
```

```
//CONSUMIDOS EN UNA MESA
```

```
void Cuenta(int p)
```

```
//Inicio de función que emplea un archivo, como registro para guardar-
```

```
//en .I, los precios de platillos consumidos en una mesa
```

```
{  
    FILE *x,*aa;  
    int c;  
    x=fopen("Route.dat","r");  
    c=getw(x);//Lectura del archivo que contiene el número de mesa  
    fclose(x);  
    switch(c)  
    //Inicio de la rutina que guarda en el archivo correspondiente al-  
    //número de mesa, el precio del platillo dado de alta.  
    {  
        case 0x1:if((aa=fopen("Mesa1.dat","a"))!=NULL)Error( );  
                putw(p,aa);  
                putc(13,aa);  
                putc(10,aa);  
                putc(10,aa);  
                fclose(aa);  
                break;  
        case 0x2:if((aa=fopen("Mesa2.dat","a"))!=NULL)Error( );  
                putw(p,aa);  
                putc(13,aa);  
                putc(10,aa);  
                putc(10,aa);  
                fclose(aa);  
                break;  
        case 0x3:if((aa=fopen("Mesa3.dat","a"))!=NULL)Error( );  
                putw(p,aa);  
                putc(13,aa);  
                putc(10,aa);  
                putc(10,aa);  
                fclose(aa);  
                break;  
    }  
}
```

```
//FUNCION QUE CALCULA LA CUENTA TOTAL DE LA MESA CORRESPONDIENTE
```

```
int Suma(FILE *x)
```

```
//Inicio de la función que calcula la cuenta total de la mesa-
```

```
//correspondiente.
```

```
{  
    int p,j;  
    p=1;  
    do{  
        j=getw(x);  
        p=p+j;  
        j=getw(x);  
    } while (j!=EOF);  
    return(p);  
}
```

```
//FUNCION QUE TRANSMITE MENSAJES ESPECIALES A LA TCI
```

```
void Mensaje(char *s)
{
  bioskey(0);
  Borrar(s);
  window(1,1,80,25);
  gotoxy(23,13);cprintf("Teclar mensaje: 0000000000000000");
  gotoxy(23,14);cprintf("<ENTER>=Transmitir ");
  gotoxy(40,13);gets(s);
  TxTCI(s);
  textcolor(15+BLINK);
  textbackground(1);
  window(23,13,55,13);cprintf(" ");
  textcolor(11);
  window(23,14,55,14);cprintf(" ");
}
```

```
/*
* PROGRAMA PRINCIPAL:
* Despliega un mensaje en espera de
* que alguna TCI se comuniquen con
* la PC.
*/
```

```
void main(void)
{
  outp(PUERTOTCI,0xA0);//Aviso a TCI's de inicio del programa.
  outp(PUERTOTCI,NUMTCI);//Envio del número total de TCIs.

  oldhandler = getvect(INTR);//Guarda la dirección de la función -
                          //maejadora de interrupción original.
  setvect(INTR, handler);//Direcciona la interrupción hacia la función-
                          //handler.
  disable();//Deshabilitación de interrupciones.
  outp(0X21,0XB8);//Habilitación de irq. del 8259.
  enable();//Habilitación de interrupciones.
}
```

```
char mensaje[16];//Arreglo para mensajes especiales a TCI
```

```
textbackground(0); //
printf(" "); //
clrscr(); //Prepara monitor de PC
textcolor(15+BLINK); //
textbackground(1); //
```

```
//Mensajes desplegados en monitor pc
gotoxy(23,13);cprintf("Esperando comunicación con la TCI");
textcolor(11);
gotoxy(23,14);cprintf("<ESC>=Salir,<ENTER>=Mensaje a TCI");
```

```

do{//Inicio de la rutina do-while.

//Mensajes desplegados en monitor
window(23,14,55,14).printf("Esperando comunicacion con la TCI").
window(23,14,55,14).printf("<ESC>=Salir,<ENTER>=Mensaje a TCI");

//Inicio de rutinas que comparan teclas oprimidas
if(bioskey(1)!=0x1C0D)Mensaje(mensaje);
if(bioskey(1)==0)goto seguir;
if((bioskey(1)!=0x011B)&(bioskey(1)!=0x2F03))bioskey(0).

seguir;}while(bioskey(1)!=0x011B);//Fin de la rutina do-while.

select(INTR, oldhandler);//Retomo de la dirección de la función-
//manejadora de interrupción original.

outp(PUERTOTCI,0xA1);//Aviso a TCI's de finalización del programa.
window(1,1,80,25);
textcolor(7);
printf(" ");
}

/*****
*PROGRAMA DEL HANDLER: *
*Realiza alta/baja de mesero o mesa. *
*alta de platos y petición de cuenta. *
*****/
void interrupt handler(...)
{
disable();//Desabilitación de interrupciones.
int tci = inportb(PUERTOTCI);//Lectura del 1er. byte del frame de 5.
for (int j=0;j<=WAITTCI;j++)//Rutina de espera.
int asiento = inportb(PUERTOTCI);//Lectura del 2do. byte.
for (j=0;j<=WAITTCI;j++)//Rutina de espera.
int comando = inportb(PUERTOTCI);//Lectura del 3er. byte.
for (j=0;j<=WAITTCI;j++)//Rutina de espera.
int data = inportb(PUERTOTCI);//Lectura del 4to. byte del frame de 5.
for (j=0;j<=WAITTCI;j++)//Rutina de espera.
int error = inportb(PUERTOTCI);//Lectura del 5to. y último byte.

Archiv(tci); // \
Archiv(asiento); // \Guarda los datos leídos.
Archiv(comando); // /
Archiv(data); // /

textcolor(14); // \
textbackground(4); // \Prepara monitor de PC.
window(23,14,55,14); // /
cprintf(" "); // /

if (((tci | asiento) & (comando | data)) == error)
//Empleo de un IF que compara los datos leídos para
//detectar algún error.
{
char* FRAME = " //Creación de una variable para una
//cadena de 16 caracteres.

```

```

long p;
FILE *aa;
mystrcpy(FRAME,"Número de TCI ");//
Archivo(FRAME); //
Borrar(FRAME); // \Guarda el mensaje que-
FRAME[0]=(tci/10+48); // /indica el número de TCI
FRAME[1]=(tci%10+48); // / que se comunica
Archivo(FRAME); //

```

Borrar(FRAME); //Borra los caracteres de la variable FRAME

```

switch (comando)
//Inicio de la rutina SWITCH que determina -
//la operación a realizarse
{
//RUTINA DAR ALTA DE MESERO
case 0xE2 : {mystrcpy(FRAME,"Alta Mesero");
FRAME[11] = 0x17;
FRAME[12] = (data/10+48);
FRAME[13] = (data%10+48);
switch (data)
{
case 0x1: FRAME[14]=(12);
FRAME[15]=(34);
break;
case 0x2: FRAME[14]=(45);
FRAME[15]=(67);
break;
case 0x3: FRAME[14]=(89);
FRAME[15]=(10);
break;
}
TxTCI(FRAME);
Archivo(FRAME);
window(23,13,55,15);// 55
cprintf("Alta Mesero: %d Clave: %d%d",data.FRAME[14],FRAME[15]);
cprintf(" Oprima <Enter> ");
getch();
break;
}
//RUTINA PARA DAR BAJA DE MESERO
case 0xE3 :mystrcpy(FRAME,"Baja Mesero ");
FRAME[12] = (data/10+48);
FRAME[13] = (data%10+48);
TxTCI(FRAME);
Archivo(FRAME);
window(23,13,55,15);// 55
cprintf("Baja Mesero: %d ",data);
cprintf(" Oprima <Enter> ");
getch();
break;
}

```

```

//RUTINA PARA DAR ALTA DE MESA
case 0xE0 :{mystncpy(FRAME,"Alta de Mesa ");
FRAME[13] = (data/10+48);
FRAME[14] = (data%10+48);
FRAME[15] = '\0';
TxTCI(FRAME);
Archivo(FRAME);
window(23,13,55,15);// 55
cprintf("Alta de Mesa: %d",data);
cprintf(" Oprima <Enter> ");
getch();
switch(data)
{
case 0x1:if((aa=fopen("Mesa1.dat","w"))= =NULL)Error( );
fclose(aa);
Route(data);
break;
case 0x2:if((aa=fopen("Mesa2.dat","w"))= =NULL)Error( );
fclose(aa);
Route(data);
break;
case 0x3:if((aa=fopen("Mesa3.dat","w"))= =NULL)Error( );
fclose(aa);
Route(data);
break;
}
break;
}

//RUTINA PARA DAR BAJA DE MESA
case 0xE1 :mystncpy(FRAME,"Baja de Mesa ");
FRAME[13] = (data/10+48);
FRAME[14] = (data%10+48);
FRAME[15] = '\0';
TxTCI(FRAME);
Archivo(FRAME);
window(23,13,55,15);// 55
cprintf("Baja de Mesa: %d",data);
cprintf(" Oprima <Enter> ");
getch();
break;

//RUTINA PARA DAR ALTA A 3 TABLAS DE PLATILLOS
case 0xE4 :{mystncpy(FRAME,"Alta de Tabla 1");
Archivo(FRAME);
switch(data)
{
case 0x1:mystncpy(FRAME,"Platillo 1 ");
TxTCI(FRAME);
Archivo(FRAME);
p=100;
Cuenta(p);
break;
case 0x2:mystncpy(FRAME,"Platillo 2 ");
TxTCI(FRAME);
p=110;
Cuenta(p);
break;
}
}

```

```

        case 0x3:mystrcpy(FRAME,"Platillo 3 ");
            TxTCI(FRAME);
            p=120;
            Cuenta(p);
            break;
    }
    window(23,13,55,15);//          55
    cprintf("Alta de Tabla 1: Platillo %d " data);
    cprintf("      Oprima <Enter> ");
    getch();
    break;
}
case 0xE6 :{mystrcpy(FRAME,"Alta de Tabla 2");
    Archivo(FRAME);
    switch(data)
    {
        case 0x1:mystrcpy(FRAME,"Platillo 1 ");
            TxTCI(FRAME);
            Archivo(FRAME);
            p=130;
            Cuenta(p);
            break;
        case 0x2:mystrcpy(FRAME,"Platillo 2 ");
            TxTCI(FRAME);
            p=140;
            Cuenta(p);
            break;
        case 0x3:mystrcpy(FRAME,"Platillo 3 ");
            TxTCI(FRAME);
            p=150;
            Cuenta(p);
            break;
    }
    window(23,13,55,15);//          55
    cprintf("Alta de Tabla 2: Platillo %d " data);
    cprintf("      Oprima <Enter> ");
    getch();
    break;}
case 0xE8 :{mystrcpy(FRAME,"Alta de Tabla 3");
    Archivo(FRAME);
    switch(data)
    {
        case 0x1:mystrcpy(FRAME,"Platillo 1 ");
            TxTCI(FRAME);
            Archivo(FRAME);
            p=160;
            Cuenta(p);
            break;
        case 0x2:mystrcpy(FRAME,"Platillo 2 ");
            TxTCI(FRAME);
            p=170;
            Cuenta(p);
            break;
    }
}

```

```

case 0x3:mystrcpy(FRAME,"Platillo 3
TxTCI(FRAME);
p=180;
Cuenta(p);
break.)
window(23,13,55,15);
cprintf("Alta de Tabla 3: Platillo %d ",data);
cprintf(" Oprima <Enter> ");
getch();
break;
}

```

//RUTINA PARA CALCULAR CUENTA TOTAL DE ALGUNA MESA

```

case 0xCC :{mystrcpy(FRAME,"Cuenta = $");
switch(asiento)
{
case 0x1:Route(asiento);
if((aa=fopen("Mesa1.dat","r"))==NULL)Error( );
p=Suma(aa);
fclose(aa);
printf("Mesa 1: ");
FRAME[10]=(p/10000+48);
FRAME[11]=((p/1000)%10+48);
FRAME[12]=((p/100)%10+48);
FRAME[13]=((p/10)%10+48);
FRAME[14]=(p%10+48);
FRAME[15]='.';
TxTCI(FRAME);
window(23,13,55,15);//
cprintf("Mesa %d: Cuenta = $%d.00",asiento,p);
cprintf(" Oprima <Enter> ");
getch();
break;
case 0x2:Route(asiento);
if((aa=fopen("Mesa2.dat","r"))==NULL)Error( );
p=Suma(aa);
fclose(aa);
printf("Mesa 2: ");
FRAME[10]=(p/10000+48);
FRAME[11]=((p/1000)%10+48);
FRAME[12]=((p/100)%10+48);
FRAME[13]=((p/10)%10+48);
FRAME[14]=(p%10+48);
FRAME[15]='.';
TxTCI(FRAME);
window(23,13,55,15);//
cprintf("Mesa %d: Cuenta = $%d.00",asiento,p);
cprintf(" Oprima <Enter> ");
getch();
break;
}
}

```

```

case 0x3:Route(asiento).
    if((aa=fopen("Mesa3.dat","r"))==NULL)Error( );
    p=Suma(aa);
    fclose(aa);
    printf("Mesa 3: ");
    FRAME[10]=(p/10000+48);
    FRAME[11]=((p/1000)%10+48);
    FRAME[12]=((p/100)%10+48);
    FRAME[13]=((p/10)%10+48);
    FRAME[14]=(p%10+48);
    FRAME[15]='.';
    TxTCI(FRAME);
    window(23,13,55,15);
    cprintf("Mesa %d: Cuenta = %d.00",asiento,p);
    cprintf(" Oprma <Enter> ");
    getch();
    break;
}
break;
}
//RUTINA QUE MANDA MENSAJE EN CASO DE ERROR EN COMANDO TRANSMITIDO
default : {mystrcpy(FRAME,"Comando erroneo");
    TxTCI(FRAME);
    window(23,13,55,15);// 55
    cprintf(" Comando erroneo ");
    cprintf(" Oprima <Enter> ");
    getch();
}
}
//Fin del SWITCH que determina operaci3n a realizarse.
}
//Fin del IF que detecta alg3n error en lectura de datos.

else outp(PUERTOTCI,0xFF);//Aviso de retransmisi3n en caso de error en lectura de datos.

enable();//Habilitaci3n de interrupciones.
int a1 = 0x20;
outp(0x20,a1);//Aviso a PC del EOI (fin de interrupci3n).
textcolor(15+BLINK);
textbackground(1);
window(23,13,55,13);cprintf(" ");
textcolor(11);
window(23,14,55,14);cprintf(" ");

}
//Fin de la funci3n manejadora de interrupci3n handler.

```



## BIBLIOGRAFIA

- Andrew S. Tanenbaum, Structured computer organization, third edition, Prentice Hall, U.S.A., 1990.
- Apuntes de computadoras y programación, Facultad de Ingeniería de la Universidad Nacional Autónoma de México, 1980.
- A. Sedra/K. C. Smith , Dispositivos electrónicos y amplificación de señales, McGraw-Hill, México, 1989.
- Brian W. Kernighan/Dennis M. Ritchie, El lenguaje de programación C, segunda edición, Prentice Hall.
- Data Manual, 8-bit microcontroller family, TMS370 Family, Texas Instrument, 1993.
- David A. Huges/Horace G. Jackson, Analysis and desing of digital integrated circuits, International student editions, U.S.A., 1990.
- Des Watson, High-level lenguajes and their compilers, Addison-Wesley Publishing Company, 1989.
- Douglas V. Hall, Microprocessors and interfacing. Programming and hardware, Editorial McGraw-Hill, U.S.A., 1991.
- Gary M. Miller, Modern electronic communication, third edition, Prentice Hall, New Jersey, U.S.A., 1992.
- Gene H. Miller, Microcomputer engineering, Prentice Hall, U.S.A.
- Herbert Taub, Circuitos digitales y microprocesadores, McGraw-Hill, México, 1990.
- Ira Pohl and Alon Shaw, The nature of computation: and introduction to computer science. Computer science, Press, Inc., U.S.A., 1981.
- M. Morris Mano, Computer system architecture, third edition, Prentice Hall, U.S.A., 1982.
- M. Morris Mano, Diseño digital, 1a. edición, Prentice Hall, México, 1987.
- Robert F. Coughlin/Frederick F. Driscoll, Circuitos integrados lineales y amplificadores operacionales, 2a. edición, Prentice Hall, México, 1987.
- Circuitos electrónicos, Edwing C. Lowenberg, serie Schaum, McGraw-Hill, México 1988.
- Sol Laoatine, Electronics in communication, second edition, .John Wiley & Sons, 1986.
- Wayne Tomasi, Electronic communication systems. Fundamentals through advanced, Printec Hall, U.S.A., New Jersey, 1991.