

UNIVERSIDAD NACIONAL DE INGENIERIA

FACULTAD DE INGENIERIA ELECTRICA Y ELECTRONICA



**IMPLEMENTACION DE UNA INTERFASE
DE COMUNICACIONES
CAJA REGISTRADORA-COMPUTADOR**

TITULACION POR EXAMEN PROFESIONAL

PARA OPTAR EL TITULO PROFESIONAL DE:

INGENIERO ELECTRONICO

LUIS ALBERTO TAQUIRI ROSALES

PROMOCION 1989-I

LIMA-PERU

1997

A mi esposa Liliana quien
me apoya y alienta
constantemente.

A mis amados padres
Alejandro y Luisa quienes
con esfuerzo, sacrificio e
inteligencia forjaron mi
desarrollo profesional.

SUMARIO

Tareas cotidianas que involucran la compra de un bien o el pago de un servicio nos permiten encontrar un equipo electrónico denominado caja registradora, el desarrollo tecnológico en estos equipos es muy avanzado que es posible almacenar grandes cantidades de información en ellas y establecer un sistema de comunicación de datos con un computador para el envío y recepción de información; el resultado de ello es de gran beneficio para el establecimiento debido a que se dejan de lado los procesos manuales de control de stock y ventas, asimismo, permite en cualquier momento hacer cambios de precios ó crear nuevos artículos, obtener al instante montos de ventas totales ó por medios de pago (efectivo, tarjetas de crédito, etc.), obtener estadísticas de ventas por artículo, obtener listados de mejores clientes, evaluar el rendimiento de vendedores, cajeros, etc.

El presente informe se basa en mi experiencia profesional durante las labores de instalación, programación y puesta en marcha de sistemas con cajas registradoras en red para supermercados, grifos, hoteles, etc. así como en el diseño y desarrollo de software de comunicación de datos para el enlace caja registradora-computador.

**IMPLEMENTACION DE UNA
INTERFASE DE COMUNICACIONES
CAJA REGISTRADORA-COMPUTADOR**

EXTRACTO

TITULO	Implementación de una interfase de comunicaciones caja registradora-computador
AUTOR	Luis Alberto Taquiri Rosales
GRADO QUE OPTA	Ingeniero Electrónico
FACULTAD	Ingeniería Eléctrica y Electrónica
UNIVERSIDAD	Universidad Nacional de Ingeniería
CIUDAD	Lima-Perú
AÑO	1997

El presente informe se divide en cinco capítulos para un mejor estudio:

En el capítulo I se efectúa una presentación de la caja registradora con la finalidad de conocer sus ventajas y aplicaciones en el mercado; se presenta un estudio de sus características, funcionamiento, los modelos existentes, las interfases y su uso como elemento fiscal.

A continuación en el capítulo II se ingresa al estudio de la interfase hardware necesaria para establecer la comunicación de datos entre una caja registradora y un computador. Se analiza al detalle en el computador la salida serial RS-232, se estudia a fondo el UART, sus

registros y su programación. Por otro lado en la caja registradora se estudia la tarjeta de interfase hardware, se indican sus características, las configuraciones permitidas, la descripción de los pines utilizados y el diagrama de conexión con el computador.

En el capítulo III se detalla el protocolo de comunicaciones para el enlace caja registradora-computador, se estudian las secuencias de envío, recepción, procedimientos de espera y control de errores, asimismo, se indican al detalle la estructura de las diversas tramas de datos, el alfabeto de comunicaciones y los caracteres de control.

En el capítulo IV se estudia un modelo de aplicación software que permite plasmar en un lenguaje de programación las secuencias definidas en el protocolo de comunicaciones para establecer la comunicación de datos entre la caja registradora y el computador, se explican los procedimientos que ejecutan el establecimiento de enlace, el envío y recepción de datos y la finalización del enlace.

Finalmente, en el capítulo V se presenta un detalle de los costos y equipos necesarios para llevar a cabo la investigación y desarrollo de la interfase de comunicación de datos caja registradora-computador.

TABLA DE CONTENIDOS

PROLOGO	1
CAPITULO I	
LA CAJA REGISTRADORA	3
1.1 Definición	3
1.2 Características generales	3
1.3 Componentes y funcionamiento	5
1.4 Modelos de cajas registradoras	9
1.5 Dispositivos de interfase opcionales	14
1.6 La caja registradora como elemento fiscal	18
CAPITULO II	
LA INTERFASE HARDWARE	22
2.1 Interfase en el computador	22
2.1.1 Salida serial RS-232	23
2.2 El UART	34
2.2.1 El UART 8250	35
2.2.2 El UART 16550	48
2.3 Tarjeta interfase en la caja registradora	63
2.4 Conexión caja registradora-computador	69
CAPITULO III	
PROTOCOLO DE COMUNICACIONES	73
3.1 Protocolo asíncrono CASIO	73
3.2 Caracteres de control	90
3.3 Alfabeto de comunicaciones	91

3.4	Formatos de comunicación.....	94	
3.4.1	Formato del bloque comando.....	94	
3.4.2	Formato del bloque de datos.....	97	
CAPITULO IV			
IMPLEMENTACION DEL SOFTWARE.....			101
4.1	Menú principal.....	102	
4.2	Definición de parámetros generales.....	102	
4.3	Definición de bloques comando.....	106	
4.4	Procedimiento de transmisión.....	108	
4.5	Procedimiento de recepción.....	117	
4.6	Listado del programa.....	128	
CAPITULO V			
COSTOS.....			134
CONCLUSIONES.....			135
ANEXOS.....			136
BIBLIOGRAFIA.....			159

PROLOGO

El presente informe constituye un documento de referencia para la implementación de sistemas de comunicación de datos entre una caja registradora y un computador, sin embargo, su uso puede extenderse a la implementación de enlaces entre un computador y otro equipo electrónico que cuente con una salida serial para comunicación de datos, con esta finalidad se da preferencia a un estudio detallado de la interfase hardware en el computador que consta de la revisión de la salida serial RS-232, los dispositivos UART mas populares y sus respectivos registros, asimismo, se detallan las características del protocolo de comunicaciones y los formatos de las tramas de datos y finalmente se estudia un modelo de software de comunicación de datos.

Las aplicaciones resultantes de este modelo son diversas y podemos mencionar entre otras: enlace computador y cajas registradoras en supermercados para el control de ventas y existencias, enlace computador y cajas registradoras ubicadas en los surtidores de un grifo para control de ventas y niveles de combustible, enlace computador y cajas registradoras en los diversos ambientes de un hotel (recepción, restaurant, discoteca, etc.) para el control de consumos de los huéspedes.

Actualmente la mayoría de aplicaciones existentes en el mercado vienen del exterior en forma de programas ejecutables no pudiendo efectuarse modificaciones ante cualquier necesidad particular del mercado nacional, por ello el diseñar un modelo propio representa una ventaja significativa debido a la rapidez de adecuación a las necesidades cambiantes de nuestro mercado.

Finalmente, debemos indicar que con dedicación y empeño podremos realizar desarrollos de sistemas que estén a la vanguardia y de esta forma contribuir al desarrollo nacional.

CAPITULO I LA CAJA REGISTRADORA

1.1 Definición

Hoy en día la mayoría de establecimientos comerciales se ven en la imperiosa necesidad de contar con un equipo que les facilite los procesos de emisión rápida de comprobantes de pago (llámese boletas de venta o facturas) así como también tener almacenado en el equipo toda la gama de artículos que expenden con la finalidad que con solo presionar algunas teclas ó leer el código de barras del artículo a través de un scanner se muestre impresa en el comprobante de pago las características del artículo (código, descripción, precio, etc.) y finalmente el equipo lleve un control al detalle de las operaciones realizadas durante el día, esto es, importe total de ventas, totales por medios de pago, totales por cajeros, total de unidades vendidas por artículo, etc, pues bien, el equipo que posee estas funciones se denomina caja registradora y actualmente su uso se encuentra extendido en la totalidad de supermercados y en la mayoría de los minimarkets, restaurantes, grifos, hoteles, etc.

1.2 Características generales

Como características generales de las cajas registradoras podemos mencionar las siguientes:

- a) Poseen dos pantallas donde se muestran la totalidad de operaciones (una para el empleado y otra para el cliente).
- b) Modo de operación con llave para funciones especiales (existen tres tipos de llaves para limitar el acceso a funciones).
- c) Mantiene los datos almacenados alrededor de 30 días aún desconectada de la red eléctrica.
- d) Mantiene las operaciones de registro de ventas aún cuando se registre un corte en el suministro de energía (a diferencia del computador que cuando falla la energía los datos procesados en ese instante generalmente se pierden).
- e) Capacidad de lectura de códigos de barra internacionales.
- f) Ampliación de la capacidad de memoria a través de chips de RAM.
- g) Emisión rápida de comprobantes de pago (boletas de venta y facturas) debido a que poseen impresora de matriz de puntos veloz.
- h) Todas las operaciones de venta se imprimen en dos cintas de papel una de ellas sirve para emitir comprobantes de pago al cliente y la otra denominada cinta auditora o testigo queda en poder del propietario para efectos tributarios.
- i) Bajo costo de los comprobantes de venta debido a que utiliza rollos de papel contómetro.

j) Almacena información permanente de datos de los artículos que expende el establecimiento como:

- Código del artículo
- Descripción del artículo
- Precio del artículo
- Estado de impuesto del artículo, etc.

k) Emisión de reportes de venta periódicos y diarios por diversos detalles:

- Por artículo
- Por cajeros
- Por formas de pago
- Por departamentos, etc.

1.3 Componentes y funcionamiento

La caja registradora en su estructura interna se compone básicamente de una tarjeta principal gobernada por un microprocesador la cual maneja en su configuración standard interfases de entrada y salida como impresora de matriz de puntos, visores, teclado y gaveta.

Impresora.— Se trata de una impresora de matriz de puntos que imprime todas las operaciones de registro y programación del equipo a una velocidad promedio de 2.5 líneas por segundo. La impresora imprime sobre dos rollos de papel denominados papel contómetro las cuales tienen las dimensiones de 44 milímetros de ancho y 60 metros de largo promedio. La cantidad de caracteres por línea en cada rollo es de 21 (veintiuno) y tienen la posibilidad de ser escritas en mayúscula, minúscula y en letra doble (ocupan dos caracteres). El alfabeto es bastante amplio y

Caja Registradora Electrónica

TK-2300

CASIO



**Figura 1.1. Caja registradora electrónica
Marca CASIO. Modelo TK-2300.**

satisface todas las letras y caracteres que uno desee utilizar.

De los dos rollos de papel utilizados, uno de ellos se emplea para emitir los comprobantes de pago al cliente o usuario final y la impresora realiza sobre el un corte automático al finalizar una transacción, mientras que el otro (denominado cinta auditora o testigo) queda en poder del propietario para efectos tributarios. Se debe indicar que los caracteres impresos en el rollo destinado al cliente se repiten en forma idéntica en el rollo testigo no existiendo forma alguna de eludir esta impresión, por ello, la caja registradora representa un elemento totalmente confiable en el aspecto de control tributario.

Visor.— Como parte del equipo existen dos visores en las cuales se muestran todas las operaciones de registro y programación del equipo. El operador del equipo (usualmente el empleado o cajero) tiene la necesidad de ver en pantalla si la operación que registró fue la correcta, por ello el equipo cuenta con un visor alfanumérico de dos líneas; la línea inferior es del tipo numérico de 10 dígitos donde se mostrarán solamente los importes de venta, mientras que la línea superior del visor es totalmente alfanumérica de 12 caracteres en la cual el cajero visualizará el nombre del artículo que esta expendiendo, por otro lado, existe un visor del tipo numérico de 10 dígitos para uso del cliente en la cual el mismo visualizará los montos materia de la transacción. El visor para uso del empleado permanece fijo mientras

que el visor del cliente es del tipo giratorio para ser adecuado a la posición en la cual el comprador tenga una mejor visibilidad.

Teclado.- El medio para ingresar las operaciones de venta y programación en la caja registradora es el teclado. El teclado por su nivel de programación podemos dividirla en dos grupos.

a) Grupo de teclas con programación fija.- Estas teclas tienen una función que no pueden ser variadas de ninguna forma y comprenden:

Teclas numéricas del cero al nueve (0 a 9).

Tecla de SUBTOTAL (tecla que permite obtener los subtotales de venta en una transacción).

Tecla CLEAR (tecla para limpiar buffer de teclado).

b) Grupo de teclas de programación variable.- Estas teclas pueden ser programadas con una gama grande de funciones diseñadas para dar al usuario la máxima libertad para crear la configuración que optimize las necesidades del establecimiento. El número de teclas disponibles varía de acuerdo al modelo, sin embargo, un promedio adecuado es de 60 (sesenta) y el número de funciones disponibles es de 80 (ochenta).

Asimismo existen dos tipos de teclas disponibles en los modelos de cajas registradoras, una de ellas es el teclado tradicional y por otro lado existe el teclado del tipo de membrana plana.

Gaveta.- Viene a ser un cajón metálico con divisiones en su interior y que sirve para guardar dinero ya sea

monedas metálicas o papel moneda. Tienen un juego de cables que van conectadas directamente a la tarjeta de la caja registradora y es a través de ellos que se envía un pulso de corriente que permite abrir la gaveta cada vez que se finaliza una transacción. También poseen una llave que permite abrirla opcionalmente en el momento que uno desee.

1.4 Modelos de cajas registradoras

Existe una gama grande de modelos de cajas registradoras debido a diferentes necesidades de los usuarios finales, sin embargo, podemos agruparlas en tres grupos de acuerdo a la capacidad de almacenamiento de datos y funciones de operación:

a) Modelos pequeños.- En este grupo podemos incluir aquellos equipos que poseen funciones limitadas y dentro de las características principales podemos mencionar:

- Máquina solo numérica.

 - Manejan solo 4 departamentos.

- Tres medios de pago.

 - Posee solo un rollo de impresión.

Su uso se limita a establecimientos que requieren solo un control interno. Estos modelos no pueden emitir comprobantes de pago válidos para la SUNAT debido a que poseen solamente un rollo de papel (la SUNAT reglamenta que las cajas registradoras válidas para emitir comprobantes de pago deben poseer dos rollos de papel).

b) Modelos intermedios.- Los modelos intermedios son los de mayor uso en establecimientos como minimarkets,

190ER

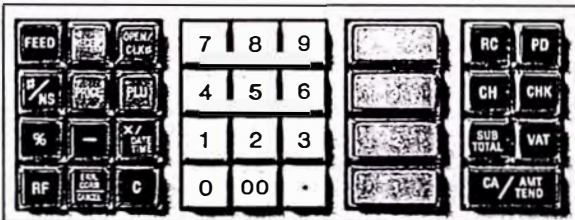


- Pantalla rotativa para el cliente
- Dos formatos de recibo posterior a la finalización de la transacción
- Pantalla de diodos LED grande
- Impresión de número de unidad
- Tres medios de pago (efectivo, cheque y a plazos)

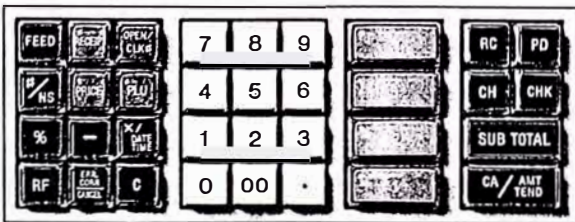


230ER

Europa



Ex-Europa



- Dos formatos de recibo posterior a la finalización de la transacción
- Pantalla de diodos LED grande
- Impresión de número de unidad
- Tres medios de pago (efectivo, cheque y a plazos)
- Función de cancelación



Con gaveta de tamaño pequeño



Con gaveta de tamaño mediano



Figura 1.2. Caja registradora electrónica de capacidad pequeña.

CE-4700



- Asignación de memoria
- Seguimiento de cuenta de cliente
- Operación simultánea de recibo/diario
- Modo REG negativo
- Teclado con disposición de función libre
- Capacidad de usar tarjeta de memoria
- Tablero de comunicación opcional
- Capacidad de conexión a impresora de talón de recibo
- Capacitación de empleado

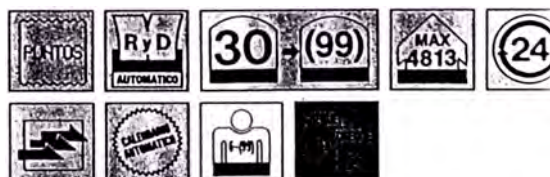
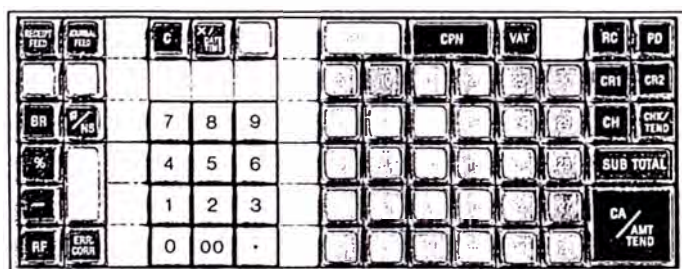


Figura 1.3. Caja registradora electrónica de capacidad intermedia.

restaurants, snacks, farmacias, etc. Poseen un amplio rango de funciones, son aceptadas como válidas por la SUNAT y pueden comunicarse con un computador, podemos mencionar las siguientes características:

- Pantallas alfanuméricas.
- Poseen dos rollos de impresión, por lo tanto son válidas para la SUNAT.
- Pueden almacenar en su memoria interna hasta 10,000 artículos mediante la adición de chips de RAM.

Poseen una gama grande de funciones programables.

- Pueden establecer enlace de comunicación de datos punto a punto con un computador.

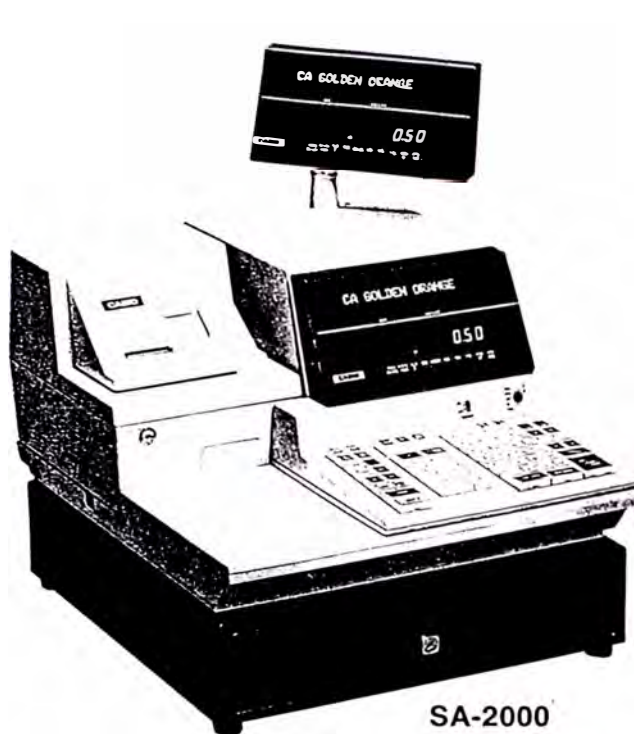
Poseen interfase con balanzas electrónicas e impresora de papeleta.

c) Modelos avanzados.- Las cajas registradoras de mayor nivel tecnológico son ubicadas en este rango. Se encuentran instalados mayormente en supermercados, cadenas de grifos, hoteles, etc. Estas poseen una capacidad de memoria mucho mayor y la posibilidad de enlazarse con diversos dispositivos externos, como características saltantes podemos mencionar:

- Capacidad de almacenar en su memoria interna hasta 65,000 artículos.
- Capacidad de almacenar la cinta auditora en forma electrónica.

Puede formarse una red de hasta 120 cajas registradoras y todas ellas enlazadas a un computador.

SA-2000/SA-2100



SA-2000



SA-2100 (Solamente Europa)

- Sistema de exploración sin caídas
- Tipo integrado o modular
- Sistema de cajero flotante
- Sistema de agrupación múltiple
- Captura de datos de artículo
- Archivo de cliente



Tipo modular



Figura 1.4. Caja registradora electrónica. Modelo avanzado.

Comunicación de datos en tiempo real.

- Interfase con scanner
- Interfase con lectoras de tarjetas magnéticas.

1.5 Dispositivos de interfase opcionales

Los diversos modelos de cajas registradoras tienen la posibilidad de ser integradas con equipos opcionales de trabajo, esto se logra instalando dentro de las cajas registradoras diversas tarjetas de interfase, ampliando la capacidad de memoria y cambiando la EPROM de la tarjeta principal.

Scanner y lectura de código de barras.- El scanner es un equipo electrónico que permite la lectura rápida de códigos de barra internacionales. Este dispositivo puede conectarse a ciertos modelos de cajas registradoras y su uso se encuentra bastante difundido en la mayoría de supermercados y minimarkets porque es en estos lugares donde mayormente se encuentran a la venta artículos de consumo masivo membretados con el respectivo código de barras.

El uso de estos equipos asegura confiabilidad, rapidez y elimina en gran porcentaje los errores de digitación manual de códigos.

En su operación normal el cajero pasa el producto codificado por el área donde el scanner emite el haz de rayos láser y al instante se emite un beep indicando que la lectura fue realizada correctamente, luego de ello el código leído se envía a través del cable interfase hacia la caja en donde se realiza un proceso de búsqueda

interna y si el artículo se encuentra se imprimen los datos de descripción y precio en el comprobante de pago.

Los scanner son diseñados para soportar la totalidad de código de barras existentes en el mundo. En el mercado nacional los códigos de barra mas comunes son el EAN-13, EAN-8, UPC-A, UPC-E. Nuestro país actualmente utiliza la norma del EAN-13 y todos los fabricantes nacionales tienen que adaptarse a esta norma. En el país el ente encargado de administrar y distribuir los códigos de barras es la Asociación Peruana de Código de Barras.

Balanza Electrónica.- La balanza electrónica es un dispositivo que puede conectarse a una caja registradora para efectos de pesaje de mercadería como carnes, embutidos, verduras, etc. Inmediatamente el producto es pesado, la balanza envía una señal codificada a la caja haciendo esta el cálculo del importe total de la venta e imprimiendo luego los datos resultantes en el comprobante de pago final.

Lector de Tarjetas Magnéticas.- Se encuentra bastante difundido en nuestro país y en el exterior el uso de las tarjetas de crédito magnéticas para procesos de compra de productos en autoservicios, restaurants, grifos, etc. Para el proceso de lectura de la tarjeta existe un dispositivo que permite leer la información almacenada en la banda magnética de las tarjetas y puede ser conectado a la caja registradora.

Dentro de la serie de datos almacenados en la banda magnética los mas usados son:

- Código del propietario
- Nombre del propietario
- Fecha de vencimiento de la tarjeta

Al ser leída la tarjeta con este dispositivo, estos datos son enviados a la caja registradora la cual procede a utilizarlas de acuerdo a la programación definida.

Impresora de Cocina.- El uso de este dispositivo se encuentra bastante difundido en restaurantes. Se trata de una impresora de tipo serial que se conecta a la caja registradora y posee un rollo de papel en el cual se pueden imprimir caracteres de colores negro y rojo.

Por el tipo de comunicación serial este equipo puede estar ubicado a una distancia alejada de la caja registradora (usualmente se instala en la cocina de los restaurantes). En la caja registradora se efectúa una programación especial para que al vender ciertos platos del menú la información de descripción del plato y la cantidad se envían a través de la línea de comunicación hacia la impresora de cocina imprimiéndose en el rollo de papel para que el cocinero proceda a preparar el potaje respectivo, con este procedimiento se evita que los pedidos se efectúen a viva voz o a través de intercomunicadores.

Impresoras de Papeleta (Slip Printer).- Como el ancho promedio de los comprobantes de pago emitidos por la caja registradora son de 44mm, se hace necesario tener un dispositivo opcional para aquellos establecimientos que necesiten utilizar formatos de comprobante mas amplios;

este dispositivo se denomina Impresora de Papeleta y su función es la de imprimir en formatos amplios la misma información que se imprime en los tickets de la caja registradora.

Tarjetas de Memoria (RAM Card).- La información almacenada en la caja registradora usualmente es de gran escala, pues se incluyen nombres y precios de los artículos, nombres de cajeros, características generales de la caja, logotipo, teclado, funciones, etc. Si por alguna razón llegaran a perderse estos datos, la reprogramación total del equipo puede demorar alrededor de 5 (cinco) horas si no se cuenta con un respaldo adecuado.

Para almacenar la información completa de la caja registradora existe un dispositivo de almacenamiento denominado tarjeta de memoria, la cual se inserta en una ranura interfaz de la caja registradora y a través de algunos comandos se puede grabar el contenido de la caja en este dispositivo. La capacidad de memoria de este medio varía de 64kbytes a 128kbytes.

Teclado Externo.- Algunos modelos de cajas registradoras requieren que al programarse los caracteres alfabéticos de descripción de artículos, nombres de empleados, etc. se ingresen a través de una tabla numérica de correspondencia, esto a veces resulta demasiado engorroso, para ello el fabricante ofrece la posibilidad de conectar a la caja registradora un teclado pequeño con todos los caracteres alfabéticos permitidos

por el equipo, posibilitando de este modo una programación fluida y segura.

1.6 La caja registradora como elemento fiscal

Actualmente el estado peruano se encuentra abocado en la tarea de incrementar la recaudación de tributos provenientes del pago del impuesto general a las ventas (IGV).

Para ello cuenta con la Superintendencia Nacional de Administración Tributaria (SUNAT) como ente encargado de dar las normas y disposiciones reglamentarias que aseguren un correcto proceso de recaudación.

Es así que dentro de las funciones de la SUNAT una de las tareas es la de controlar en forma rigurosa que todas las empresas, profesionales y/o establecimientos que presten servicios o expendan algún bien deben emitir en forma obligatoria un comprobante de pago.

Comprobantes válidos.- El reglamento de SUNAT reconoce como comprobantes de pago válidos los emitidos -entre otros- por los siguientes medios:

- a) Comprobante emitido por cajas registradoras.
- b) Comprobante emitido por computadoras.
- c) Comprobante emitido en forma manual.

Análisis comparativo.- Podemos hacer un análisis comparativo de las tres alternativas disponibles para darnos cuenta de las ventajas y desventajas que representan el uso de uno u otro medio.

a) Caja Registradora.-

- Medio de impresión veloz, ventaja muy importante en establecimientos con alto movimiento comercial (ej. supermercados).
- Costo bajo de los comprobantes de pago, debido a que la caja registradora puede emitir indistintamente boletas de venta o facturas y el medio de impresión es un rollo de papel contómetro que cuesta alrededor de \$ 0.50 y puede alcanzar a emitir alrededor de 800 comprobantes.
- La caja registradora cuenta con una programación cerrada en cuanto al manejo de totales de venta y acumulados de venta lo cual imposibilita la alteración de estos datos y por ende una imposibilidad de evasión de tributos.

b) Computador.-

- Medio de impresión lento, debido a que tiene que imprimirse sobre formatos de papel de dimensiones 14cm de ancho.
- Costo alto de los comprobantes de pago, debido a que un requisito para utilizar computadoras como emisores de comprobantes es el de usar papeles preimpresos y numeración correlativa preimpresa por imprenta, además de incluir 4 (cuatro) copias.
- Los datos de venta almacenados en la computadora pueden ser alterados fácilmente mediante software.

c) Manual.-

- Medio de impresión lento.

- Costo medianamente alto de comprobantes de pago, debido a que se requieren de talonarios de facturas con 4 (cuatro) copias.
- Los datos de venta pueden ser manipulados a criterio del propietario.

De lo anterior podemos concluir que la caja registradora además de servir de gran ayuda como elemento de control para el establecimiento y ahorro en la emisión de comprobantes de pago protege al estado en la evasión de impuestos.

Validez de la caja registradora como equipo emisor de comprobantes de pago.- La SUNAT autoriza el uso de la caja registradora como equipo válido en la emisión de comprobantes de pago si cumple con las siguientes características:

- Poseer dos rollos de papel contómetro (uno para emitir comprobante para el cliente y el otro como cinta auditora o testigo).
- Imprimir en el comprobante para el cliente los datos de la empresa que presta el servicio o expende el bien (Razón social, dirección, RUC y # de serie de la caja registradora).
- Imprimir en el comprobante el número correlativo autogenerado por el equipo además de la fecha y hora de la transacción.
- Imprimir en el comprobante la descripción del artículo y el precio del mismo.

Para casos de factura se incluirá, además de lo anterior, el número de RUC y razón social del adquiriente y el total de venta desglosado en valor de venta y monto de IGV.

CAPITULO II LA INTERFASE HARDWARE

Cuando se trata de establecer una interfase de comunicación de datos completa entre dos equipos se requieren dos niveles de interfase claramente definidos; hablamos de la interfase hardware y la interfase software. El presente capítulo versará exclusivamente sobre los elementos hardware que intervienen y son necesarios para implementar el sistema de comunicación de datos entre una caja registradora y un computador.

2.1 Interfase en el computador

El avance de las telecomunicaciones e informática es tan amplia que la mayor parte de actividades que realizamos tiene alguna relación con ellas. Detrás de las mismas se encuentra un elemento importante y que es herramienta básica para llevar a cabo estas y otras tareas, se trata del computador; el cual tiene como tarea fundamental llevar a cabo procesos con suma rapidez y exactitud, sin embargo, para enviar los datos procesados a un dispositivo lejano se hace necesario la presencia de una interfase hardware.

De acuerdo a la complejidad de la comunicación, muchas veces será necesario el diseño de tarjetas de interfase especiales a ser instaladas en el computador, sin embargo, en su versión estándar el computador trae un

dispositivo de salida serial para comunicación de datos con el mundo exterior, se trata de la tarjeta RS-232.

2.1.1 Salida serial RS-232

La interfase serial RS-232 viene de la abreviatura resultante de la denominación "Recomendación Standard Número 232, Revisión C", recomendación efectuada por el Departamento de Ingeniería de la Asociación de Industrias Electrónicas (EIA).

El nombre formal del RS-232C es "Interfase entre un equipo terminal de datos y un equipo de comunicación de datos empleando intercambio de datos binarios en serie". Las denominaciones equipo terminal de datos (Data Terminal Equipment, DTE) y equipo de comunicación de datos (Data Communications Equipment, DCE) son términos genéricos para el computador y modem, respectivamente.

A continuación se describe tres aspectos distintos de la conexión DTE/DCE las cuales forman parte de la recomendación de la EIA:

Descripción mecánica de los circuitos de interfase

- Descripción funcional de los circuitos de intercambio.

Característica de las señales eléctricas.

a) Descripción mecánica de los circuitos de interfase

La definición mecánica es inesperadamente escasa en el estándar RS-232. Uno de los errores más comunes sobre este estándar se refiere al conector físico. Lo único que dice el documento es que se usarán dos conectores: el primero hembra para el DCE y el segundo macho para el

DTE. Además, se esbozan algunas indicaciones sobre la posición en que se deben instalar los conectores; sin embargo, no se habla en ningún momento de las características físicas del propio conector. Sí se establece, no obstante, la capacitancia máxima del cable (2,500 picofaradios), luego se definen 25 (veinticinco) "circuitos de intercambio" y se asignan números de pines dentro del conector. Estas asignaciones se presentan en la figura 2.1.

Si se eliminan de la figura los circuitos que están sin asignar o que pertenecen exclusivamente a la transmisión síncrona, se obtiene como resultado la figura 2.2., cada número de pin aparece junto a una abreviatura de su función (en lugar del nombre del circuito) y un comentario sobre su manejo.

Pin	Circuito	Nombre
1	AA	Masa protectora (Malla)
2	BA	Transmisión datos
3	BB	Recepción datos
4	CA	Solicitud de envío
5	CB	Listo para enviar
6	CC	Listo grupo de datos
7	AB	Masa de señal o común
8	CF	Recibida detección de señal de línea
9	--	Reservado para test
10	--	Reservado para test
11	--	Sin asignar
12	SCF	Recibida detección de señal de línea (secundario)
13	SCB	Listo para enviar secundario
14	SBA	Transmisión de datos secundario
15	DB	Temporización de elementos de señal de transmisión
16	SBB	Recepción de datos secundario
17	DD	Temporización de elementos de señal de recepción
18	--	Sin asignar
19	SCA	Solicitud de envío secundario
20	CD	Listo terminal de datos
21	CG	Detector de calidad de señal
22	CE	Indicador de timbre
23	CH/CI	Detector de velocidad de señal de datos
24	DA	Temporización de elemento de señal de transmisión
25	--	Sin asignar

Figura 2.1. Asignaciones de pines en los veinticinco circuitos de intercambio RS-232

Pin	Abreviatura	Nombre	Dirección	Función
1	--	Protective Ground	--	Puesta a tierra de seguridad
2	TD	Transmitted Data	hacia DCE	Salida de datos DTE
3	RD	Received Data	hacia DTE	Entrada de datos DTE
4	RTS	Request to Send	hacia DCE	DTE desea cambiar a modo TX
5	CTS	Clear to Send	hacia DTE	DCE listo para transmitir
6	DSR	Data Set Ready	hacia DTE	DCE listo para comunicar con DTE
7	--	Signal Common	--	Línea común (masa) del circuito
8	DCD	Data Carrier Detect (RLSD)	hacia DTE	Enlace de datos instalándose
20	DTR	Data Terminal Ready	hacia DCE	Pone a trabajar el modem
22	RI	Ring Indicator	hacia DTE	Anuncia una llamada
20	DTR*	Data Terminal Ready	hacia DCE	Pone a trabajar al modem

Figura 2.2. Circuitos RS-232 para E/S asincrónica en computadoras. Se conservan los nombres en inglés con sus siglas.

Funciones de datos y control.- Las funciones RS-232 están divididas estrictamente en dos grupos: funciones de datos y de control. Las primeras, simplemente, corresponden a la transmisión y recepción a través de los pines 2 y 3, respectivamente. Se trata de los dos únicos pines por las que circulan los datos. Todas las funciones restantes son de control, y se llaman así porque transportan el estado u orden que controla el comportamiento del modem.

Convenciones entrada/salida.- Uno de los aspectos menos intuitivos de las figuras 2.1. y 2.2. es que ambas partes de un interfase RS-232 son complementarias desde un punto de vista lógico: la salida de un extremo de la interfase corresponde a la entrada del otro. Recuérdese que los nombres funcionales de los circuitos reflejan el punto de vista del DTE. En estas condiciones es fácil analizar la nomenclatura del polo opuesto: por ejemplo, el nombre "Transmisión de datos" se refiere claramente a una salida. Sin embargo, el pin Transmitted Data (2) es una salida únicamente en el DTE; en el extremo DCE, evidentemente, se trata de una entrada. Para aliviar este problema, la figura 2.2. indica la dirección en que viaja la señal.

El programador y el RS-232.- Las funciones de control de la interfase RS-232 se gestionan, por supuesto, a través del hardware. Se emplean generalmente funciones asíncronas instaladas en un circuito integrado controlador que se conoce con las siglas UART

(Transmisor/Receptor Asíncrono Universal). Este dispositivo gestiona los datos transmitidos y recibidos, y también muchas funciones de control. Cuando se emplea una UART, el programador se ahorra la mayor parte de detalles internos de temporización de E/S asíncrona, formato de datos e interfases RS-232. De hecho, la transmisión o recepción de un byte con una UART se basa simplemente en la lectura o escritura de una posición de RAM o un puerto E/S. Del mismo modo, las entradas y salidas RS-232 pueden ser controladas y manipuladas sin esfuerzo.

b) Descripciones funcionales de los circuitos de intercambio

Se describen a continuación las definiciones formales EIA de las funciones de los circuitos de la figura 2.2.

Masa o tierra protectora (pin 1).— Este pin suele conectarse internamente al chasis del dispositivo, para aprovechar la conexión a tierra del enchufe de la pared.

Común o masa de señal (pin 7).— Es el retorno común de todos los circuitos, y deberá existir en todos los casos. La conexión de este pin a la masa protectora (pin 1) evita la destrucción catastrófica del equipo cuando surge un serio problema eléctrico en algún punto diferente.

Solicitud de envío (RTS, pin 4)

Listo para enviar (CTS, pin 5).— El estándar RS-232 establece que Request to Send (pin 4) "acondiciona" el modem para transmisión. De hecho, su única misión es conmutar un modem semi-duplex entre modos recepción y

transmisión. Mientras el modem semi-duplex está recibiendo, el DTE mantiene inhibida el pin Request to Send; cuando le toca el turno de transmitir, informa al modem de su deseo activando este pin. El DTE no puede empezar a enviar datos al modem inmediatamente, porque este último no realiza la operación de conmutación de forma instantánea. Así, después de activar Request to Send, el DTE comienza a vigilar Clear to Send (pin 5), que se mantiene baja mientras el modem está en modo recepción. Cuando el modem completa la operación de conmutación, lo informa al DTE a través de este pin, indicando que ya puede enviar los datos; este handshaking RTS/CTS se realiza en sentido inverso para pasar desde recepción a transmisión de nuevo.

Las conexiones full-duplex son canales de doble vía, por lo que no necesitan hacer uso de la técnica RTS/CTS. De acuerdo con esto, en los modems full-duplex se deja permanentemente activado el pin 5 (Clear to Send), o bien se conecta a Data Carrier Detect (pin 8).

Listo grupo de datos (DSR, pin 6)..- La señal Data Set Ready se activa únicamente cuando se dan simultáneamente las condiciones siguientes:

1. El modem está "conectado a un canal de comunicaciones"; es decir, descolgado, sin estar en modo test, voz ni marcado.
2. El modem ha realizado "todas las funciones de temporización necesarias en el sistema conmutado para completar la llamada". En modo origen, esta frase

significaría marcar, seguir el proceso de la llamada, y cualquier otra operación que requiera la conexión a través del teléfono.

3. El modem ha comenzado "la transmisión de un tono respuesta discreto..." En modo respuesta, este tono y la señal Data Set Ready se activan dos segundos después de descolgar el teléfono. El modem origen no transmite ni activa su Data Set Ready hasta recibir el tono respuesta del modem remoto.

Por cierto, el término Data Set es el nombre que se da a los modems en las compañías telefónicas (equipo de datos), por su parte, el estándar EIA los llama DCE.

Detección de portadora de datos (DCD, pin 8).- Este pin, cuyo nombre oficial es detección de señal de línea recibida, se activa cuando el modem recibe una portadora remota, y permanece así durante todo el tiempo de la conexión. En modems semi-duplex, por supuesto, Data Carrier Detect se activa únicamente en el modem receptor.

Listo terminal de datos (DTR, pin 20).- Esta señal "prepara" los modems "para ser conectados" a la línea de comunicaciones y "mantiene la conexión establecida por medios externos". Estos fragmentos de frase quieren indicar que Data Terminal Ready activa (pero no produce) la conexión del modem a la línea. En modo origen, Data Terminal Ready deberá activarse para proceder a la autollamada; en modo respuesta, se activará para realizar la autorespuesta.

Una vez que el modem esté conectado a la línea, Data Terminal Ready deberá permanecer activo mientras dure la conexión; si se inhibe se produce la desconexión de la línea de comunicaciones, interrumpiendo bruscamente el enlace. Además, esta señal es también necesaria para transmitir datos por la línea Transmitted Data (pin 2), por lo que, a todos los efectos, se trata de un "control maestro" del modem.

Transmisión de datos (TD, pin 2).- La línea Transmitted Data se encarga de transportar los datos serie desde el DTE al modem. De acuerdo con una tecnología establecida hace mucho tiempo, el transmisor se pasa a MARK durante los períodos de reposo.

Un DTE no puede transmitir datos a menos que hayan sido activados los circuitos siguientes:

1. Request to Send (pin 4)
2. Clear to Send (pin 5)
3. Data Set Ready (pin 6)
4. Data Terminal Ready (pin 20)

Las condiciones 1 y 2 tienen poco sentido en modems full-duplex en los que Request to Send y Clear to Send carecen de utilidad.

Recepción de datos (RD, pin 3).- El funcionamiento de RD no depende de ninguna otra función RS-232. El estándar especifica que Received Data estará asignado a MARK cuando no haya portadora o bien, en modems semi-duplex, durante un "breve" intervalo después de la conmutación de modo transmisión a recepción.

Indicación de timbre (RI, pin 22)..- Este pin se activa cuando aparecen tonos de llamada en la línea. Se supone que Ring Indicator debe estar activado "coincidiendo aproximadamente" con el segmento activo de la señal de timbre, e inhibido entre un timbrado y el siguiente. Por otra parte, esta señal funciona con independencia del estado de Data Terminal Ready (pin 20).

Detector de la velocidad de señal de datos (DSRD, pin 23)..- Cuando existe posibilidad de dos velocidades de transmisión, se representa la mayor de las dos activando DSRD. En la figura 2.2. se observará que esta función puede, por acuerdo entre usuarios, llegar a ser bidireccional. En tal caso, el DTE puede activar DSRD (pin 23) para forzar al modem a utilizar la mayor de las dos velocidades, o alternativamente el modem puede activarla para informar sobre la velocidad que va a utilizar en el enlace.

c) Características de las señales eléctricas

A continuación se hará un estudio de las características eléctricas de la interfase RS-232:

Velocidad y potencia..- La EIA permite que las velocidades se establezcan en un rango entre "0 y un límite superior nominal de 20,000 bits por segundo". En la mayoría de instalaciones, la velocidad queda limitada a 19,200 bps. El estándar limita también la longitud de los cables a 15 metros, a menos que la capacitancia total de los mismos sea menor de 2,500 picofaradios.

La interfase deberá ser capaz de soportar un cortocircuito de duración indefinida entre cualquier pareja de pines sin sufrir daño alguno. En tales casos, la corriente no deberá superar 0.5 amperios. Estas características producen una interfase segura y robusta, la cual, como característica importante, resulta muy tolerante a errores de cableado.

Niveles lógicos.- El estándar RS-232 especifica un nivel lógico bipolar: es decir, los niveles lógicos se representan no solamente por la magnitud de las tensiones, sino también por las polaridades. La tensión máxima permitida en cualquier circuito es ± 15 voltios.

Además, el estándar RS-232 define cuatro niveles lógicos: las entradas tienen diferentes definiciones que las salidas, y las funciones de datos (Transmitted Data - pin 2- y Received Data -pin 3-) son distintas de las funciones de control. En la figura 2.3. se muestran las definiciones de niveles lógicos de las entradas y salidas RS-232. Los niveles binarios de salida son +5 a +15 y -5 a -15; las tensiones entre +5 y -5 quedan indefinidas. Los niveles binarios de entrada van desde +3 a +15 y desde -3 a -15; las tensiones entre +3 y -3 quedan asimismo indefinidas. Esta diferencia de niveles entre entradas y salidas se conoce como margen de ruido. Significa también que la interfase puede tolerar 2 voltios de ruido (pico) o una caída de 2 voltios por ley de Ohm entre DTE y DCE.

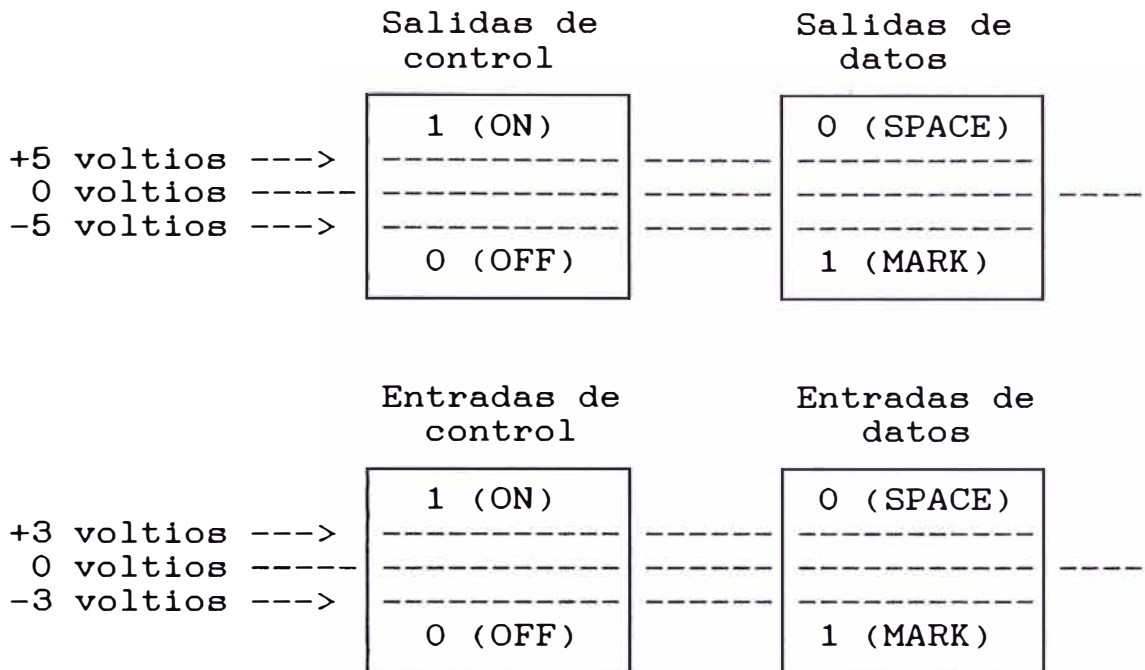


Figura 2.3. Niveles lógicos de las funciones de control RS-232.

Si las definiciones del estándar parecen extrañas para un equipo informático, recordemos que se diseñó en gran medida para acomodarse a la tecnología de teletipos en los que, por tradición centenaria, la línea de comunicaciones bipolar indicaba un estado de reposo (MARK) con una tensión negativa.

Conversión de niveles RS-232.— Las tensiones RS-232 y los niveles lógicos empleados no son los que se usan normalmente en la electrónica de las computadoras; se hace necesaria por lo tanto, una conversión entre niveles. Esta operación la realizan ciertos circuitos integrados especiales que se conocen como controladores y receptores de línea EIA (RS-232). Por razones

electrónicas, los dispositivos suelen ser inversores, lo cual implica que dicha inversión deberá ser compensada en las entradas y salidas del CI controlador de E/S asíncronas (UART).

2.2 El UART

Ahora que tenemos los detalles de la interfase serial RS-232, es el momento de revisar el dispositivo hardware que realmente realiza la tarea en el computador.

La mayoría de microcomputadores usa un dispositivo hardware especial llamado Universal Asynchronous Communications Receiver Transmitter (UART) para implementar una interfase serial RS-232. Se debe señalar que el UART es algunas veces llamado Asynchronous Communications Element (ACE). El IBM PC y las computadoras compatibles utilizan el UART basado en el Chip INS8250 de National Semiconductor's. Aunque algunas interfases seriales con PC aún utilizan el 8250 UART, la mayoría de computadoras fabricadas hoy usan los UARTs 16450 y 16550. Básicamente el UART 16450 es una versión mejorada del 8250 en cuanto a velocidad y el 16550 es una versión mejorada del 16450 que utiliza buffers FIFO. Con los modems de alta velocidad y ambientes de trabajo multitarea el UART 16550 se perfila como el UART de mayor uso en el mundo de las computadoras personales.

Direccionamiento del UART

Todos los UARTs poseen uno o más registros para el control de sus operaciones. Estos registros leen de y escriben hacia utilizando los puertos de entrada/salida. La figura 2.4. muestra las direcciones del registro base (el primer registro) del UART para el COM1 hasta el COM4. Aún cuando algunos dispositivos de comunicación serial soportan puertos mayores a COM4, estos realmente no son estándares y se debe consultar la literatura del dispositivo particular para determinar la dirección del puerto utilizado por el UART como registro base.

Puerto Serial	Dirección de Puerto del Registro Base
COM1	3F8H
COM2	2F8H
COM3	3E8H
COM4	2E8H

Figura 2.4. Direcciones de puerto de los registros base en los puertos seriales.

2.2.1 El UART 8250

Veamos ahora un estudio detallado de los registros del 8250. Como veremos, los registros del 8250 realizan una serie de tareas que simplifican enormemente la programación de software para comunicaciones seriales.

Registro 0 :**Registro del Buffer de Receptor (RBR, Receiver Buffer Register)**

Cuando un caracter es recibido por el UART 8250, este es reconstruido y almacenado en el Receiver Buffer Register (RBR) del UART. El RBR es el primer registro del UART y puede ser direccionado leyendo el puerto del registro base del respectivo puerto serial. Para COM1 el RBR puede ser leído cogiendo el valor del puerto 3F8H. Aunque el RBR siempre cogerá 8 bits, se debe enmascarar los bits no usados cuando se trabaje con caracteres menores a 8 bits.

Bit	7	6	5	4	3	2	1	0
	X	X	X	X	X	X	X	X

- Bit 0 = Bit 0 del caracter entrante
- Bit 1 = Bit 1 del caracter entrante
- Bit 2 = Bit 2 del caracter entrante
- Bit 3 = Bit 3 del caracter entrante
- Bit 4 = Bit 4 del caracter entrante
- Bit 5 = Bit 5 del caracter entrante.
Puede ser indefinido si se usan 5 bits de datos.
- Bit 6 = Bit 6 del caracter entrante.
Puede ser indefinido si se usan 5 o 6 bits de datos.
- Bit 7 = Bit 7 del caracter entrante.
Puede ser indefinido si se usan 5, 6 o 7 bits de datos.

Figura 2.5. El Registro del Buffer de Receptor del UART 8250 (RBR).

Registro 0 :

Registro de Retención (THR, Transmitter Holding Register).

El Transmitter Holding Register (THR) del UART es usado para transmitir un caracter hacia la interfase serial. Comparte el primer registro del UART con el RBR y puede ser direccionada escribiendo el puerto del registro base del puerto serial. Para el COM1, el THR puede ser escrita enviando un valor al puerto **3F8H**.

Bit	7	6	5	4	3	2	1	0									
<table border="1" style="border-collapse: collapse; margin: 0 auto;"> <tr> <td style="padding: 5px 10px;">X</td> <td style="padding: 5px 10px;">X</td> <td style="padding: 5px 10px;">X</td> <td style="padding: 5px 10px;">X</td> <td style="padding: 5px 10px;">X</td> <td style="padding: 5px 10px;">X</td> <td style="padding: 5px 10px;">X</td> <td style="padding: 5px 10px;">X</td> <td style="padding: 5px 10px;">X</td> </tr> </table>									X	X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X	X									

Bit 0 = Bit 0 del caracter saliente
 Bit 1 = Bit 1 del caracter saliente
 Bit 2 = Bit 2 del caracter saliente
 Bit 3 = Bit 3 del caracter saliente
 Bit 4 = Bit 4 del caracter saliente
 Bit 5 = Bit 5 del caracter saliente
 Bit 6 = Bit 6 del caracter saliente
 Bit 7 = Bit 7 del caracter saliente

Figura 2.6. El Registro de Retención del UART 8250 (THR)

Registro 1 :

Activación de Interrupción (IER, Interrupt Enable Register).

El Interrupt Enable Register (IER) del 8250 es usado para habilitar interrupciones. Puede ser direccionado escribiendo o leyendo el registro del puerto base más uno. Para COM1, el IER puede ser escrito o leído enviando o cogiendo el valor del puerto 3F9H.

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	X	X	X	X

- Bit 0 = RxRDY: Cuando este bit es igual a 1, el UART 8250 generará una interrupción cuando un byte esté listo en el Receiver Buffer Register.
- Bit 1 = TBE: Cuando este bit es igual a 1, el UART 8250 generará una interrupción si un byte es movido del Transmitter Holding Register hacia el registro interno Transmitter Shift Register del UART.
- Bit 2 = Cuando este bit es igual a 1, el UART 8250 generará una interrupción cuando un error de paridad, un sobre rango, error de trama o una condición BREAK es detectada durante la llegada de un byte.
- Bit 3 = Cuando este bit es igual a 1, el UART 8250 generará una interrupción si alguno del las líneas de entrada del RS-232 cambia de estado.
- Bit 4 = Siempre 0.
- Bit 5 = Siempre 0.
- Bit 6 = Siempre 0.
- Bit 7 = Siempre 0.

Figura 2.7. El Registro de Activación de Interrupción del UART 8250 (IER)

Registro 2 :**Identificación de Interrupciones (IIR, Interrupt Identification Register).**

El Interrupt Enable Register (IIR) del 8250 es usado para determinar que interrupciones pueden haber ocurrido. Puede ser direccionado leyendo el registro del puerto base más dos. Para COM1, el IIR puede leído cogiendo el valor del puerto 3FAH.

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	0	X	X	X

- Bit 0 = Si este bit está en 1, no existen interrupciones pendientes. Por otro lado, un valor 0 indicará que una interrupción ha ocurrido.
- Bit 2,1 = Si el bit 0 es igual a 0, los bits 2 y 1 indicarán el tipo de interrupción como sigue:
- | Bit 2 | Bit 1 | Tipo de interrupción |
|-------|-------|---------------------------------|
| 0 | 0 | Estado del modem |
| 0 | 1 | Buffer de transmisión vacío. |
| 1 | 0 | Dato recibido. |
| 1 | 1 | Error de serialización o BREAK. |
- Bit 3 = Siempre 0.
- Bit 4 = Siempre 0.
- Bit 5 = Siempre 0.
- Bit 6 = Siempre 0.
- Bit 7 = Siempre 0.

Figura 2.8. El Registro de Identificación de Interrupciones del UART 8250 (IIR).

Registro 3 :**Fomato de Datos (LCR, Line Control Register).**

El Line Control Register (LCR) del 8250 es usado para definir parámetros como número de bits de data, número de bits de stop, tipo de paridad, etc. Puede ser direccionado escribiendo o leyendo el registro del puerto base más tres. Para COM1, el LCR puede ser escrito o leído enviando o cogiendo el valor del puerto **3FBH**.

Bit	7	6	5	4	3	2	1	0
	X	X	X	X	X	X	X	X

Bit 1,0 = Los bits 1 y 0 determinan el número de bits de datos como sigue.

Bit 1	Bit 0	Número de bits de datos
0	0	5
0	1	6
1	0	7
1	1	8

Bit 2 = Si el bit es igual a 0, un bit de stop será usado. Si es igual a 1 se generará 2 bits de stop.

Nota: Si se usan 5 bits de datos y este bit está en 1, el UART usará 1 y 1/2 bits de stop.

Bit 5,4,3=Los bits 5,4 y 3 señalan la paridad a ser usada.

Bit 1	Bit 0	Bit 3	Paridad
0	0	0	NINGUNA
0	0	1	IMPAR
0	1	1	PAR
1	0	1	MARK
1	1	1	SPACE

Bit 6 = BREAK: Si este bit es igual a 1, el UART forzará la línea TD al estado lógico 0. El transmisor permanece incondicionalmente en este estado hasta que se escriba un 0 en este bit.

Bit 7 = Este bit no tiene nada que ver con el formato; se trata de un truco que permite extender el número de registros direccionados con las tres líneas de control. Cuando este bit está activo, los registros 0 y 1 (E/S de interrupciones y transmisor/receptor) se transforman en B-S y B+S de los registros de latch divisor.

Figura 2.9. El Registro de Formato de Datos del UART 8250 (LCR).

Registro 4 :**Control de modem (MCR, Modem Control Register).**

El Modem Control Register (MCR) del 8250 es usado para tareas como definir el estado de la línea RTS, línea DTR, habilitación de las interrupciones UART entre otras. Puede ser direccionado escribiendo o leyendo el registro del puerto base más cuatro. Para COM1, el MCR puede ser escrito o leído enviando o cogiendo el valor del puerto 3FCH.

Bit	7	6	5	4	3	2	1	0
0	0	0	X	X	X	X	X	X

- Bit 0 = Data Terminal Ready (DTR): Cuando se escribe un 1 en este bit se activa la salida Data Terminal Ready.
- Bit 1 = Request To Send (RTS): Cuando se escribe un 1 en este bit se activa la salida Request to Send.
- Bit 2 = Salida de propósitos generales #1 (GPO1): Se trata de las dos primeras salidas definibles por el usuario.
- Bit 3 = Salida de propósitos generales #2 (GPO2): Es la segunda salida definible por el usuario.
- Bit 4 = Test Local Loopback: Asignando este bit a 1, el UART 8250 se encontrará en un estado de loopback local. Cualquier dato enviado por el transmisor será devuelto por el UART como un byte recibido. Bajo operación normal este bit deberá estar en 0.
- Bit 5 = Siempre 0
- Bit 6 = Siempre 0
- Bit 7 = Siempre 0

Figura 2.10. El Registro de Control de Modem del UART 8250 (MCR).

Registro 5 :**Estado de línea (LSR, Line Status Register).**

El Line Status Register (LSR) del 8250 informa de la disponibilidad de algún dato recibido, asimismo la detección de BREAK, errores de receptor y actividad de los registros de transmisión y recepción. Puede ser direccionado leyendo el registro del puerto base más cinco. Para COM1, el LSR puede ser leído cogiendo el valor del puerto 3FDH.

Bit	7	6	5	4	3	2	1	0
	0	X	X	X	X	X	X	X

- Bit 0 = RxDY: Si este bit es igual a 1, un byte está disponible en el buffer de receptor (RBR). El bit queda anulado cuando se lee.
- Bit 1 = Sobreescritura en el receptor: Cuando este bit está activo se indica que un byte del buffer del receptor (RBR) acaba de ser borrado por la entrada de un nuevo byte; el primer byte se pierde. El bit se anula por lectura del registro.
- Bit 2 = Error de paridad: Se activa cuando el bit de paridad del byte recibido no coincide con la paridad debida según el registro de formato de datos. El bit queda anulado cuando se lee.
- Bit 3 = Error de trama: Cuando este bit es 1, indica que tras ensamblar un byte recibido, su bit de STOP era incorrecto (es decir, SPACE en lugar de MARK). El bit queda anulado cuando se lee.
- Bit 4 = Detección de BREAK: Aparece un 1 en este bit cada vez que el receptor detecta una condición SPACE por un período mayor a un caracter.
- Bit 5 = Transmitter Buffer Empty (TBE): El bit TBE se activa para informar que se ha desplazado un byte desde el buffer del transmisor (registro de retención) al registro de desplazamiento. Si se olvida consultar este byte antes de escribir en el registro del buffer del transmisor puede llegar a borrarse un byte que estuviese ya instalado en él. Esta situación, conocida como sobreescritura de transmisor, no es informada por el 8250. Es importante comprender que aunque TBE advierta que el 8250 esté listo para aceptar otro byte de transmisión, esto no implica que se haya completado la transmisión del byte anterior.
- Bit 6 = Transmitter empty (TXE): El bit TXE indica que no existen bytes en el buffer de transmisor ni en su registro de desplazamiento. Así, el UART ha transmitido todos los bytes si este bit está en 1.
- Bit 7 = Siempre 0

Figura 2.11. El Registro de Estado de Línea del UART 8250 (LSR).

Registro 6 :**Estado de modem (MSR, Modem Status Register).**

El Modem Status Register (MSR) del 8250 informa del estado de las líneas CTS y DSR, si ocurre una llamada y si una portadora está presente. Puede ser direccionado leyendo el registro del puerto base más seis. Para COM1, el MSR puede ser leído cogiendo el valor del puerto 3FEH.

Bit	7	6	5	4	3	2	1	0
	X	X	X	X	X	X	X	X

- Bit 0 = Delta CTS: Si este bit está en 1, la línea CTS ha cambiado de estado desde que el MSR fué leído por última vez. Se anula cuando es leído el MSR.
- Bit 1 = Delta DSR: Si este bit está en 1, la línea DSR ha cambiado de estado desde que el MSR fué leído por última vez. Se anula cuando es leído el MSR.
- Bit 2 = Delta Ring Indicator: Si este bit es 1, la línea RI ha cambiado de bajo a alto desde que el MSR fué leído por última vez. Se anula cuando se lee el MSR.
- Bit 3 = Delta Data Carrier Detect: Si este bit es 1, la línea DCD ha cambiado de estado desde que el MSR fué leído por última vez. Se anula cuando se lee el MSR.
- Bit 4 = Clear To Send. (CTS): Este bit indica el estado actual de la línea CTS.
- Bit 5 = Data Set Ready (DSR): Este bit indica el estado actual de la línea DSR.
- Bit 6 = Ring Indicator (RI): Indica el estado actual de la línea RI.
- Bit 7 = Data Carrier Detect (DCD): Indica el estado de la línea DCD.

Figura 2.12. El Registro de Estado de Modem del UART 8250 (MSR).

Registro 7 :**Scratch pad.**

Este registro no tiene función reconocida, además no existe en las versiones iniciales del 8250.

Registro 8 :

Byte menos significativo del latch divisor de velocidad en baudios (DLL, Baud Rate LSB Divisor Latch Register).

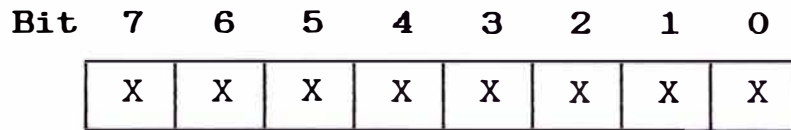
Registro 9 :

Byte mas significativo del latch divisor de velocidad en baudio (DLM, Baud Rate MSB Divisor Latch Register).

Cuando se pone en alto el bit 7 del registro de formato de datos (LCR), los registros 0 y 1 se transforman en los B-S y B+S del latch divisor, respectivamente. (Sus números de registro, 0/8 y 0/9, reflejan simplemente el parche realizado).

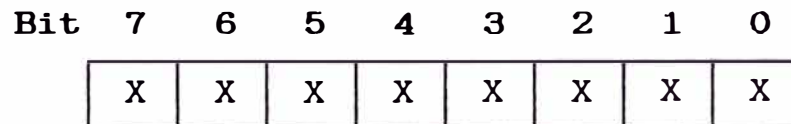
El reloj de referencia del 8250 se divide por un entero de 16 bits contenido en los registros B-S y B+S de latch divisor. La frecuencia resultante se utiliza como reloj maestro para controlar la lógica del transmisor y opcionalmente (por medio de una conexión externa), la del receptor. Este reloj maestro se divide a su vez por 16 para generar el reloj de baudios, que controla la velocidad de transmisión y recepción de datos. El divisor para cualquier velocidad puede calcularse según la fórmula:

$$\text{Divisor} = \frac{\text{Frecuencia del reloj de referencia}}{16 \times \text{velocidad deseada}}$$



Bit 0 = Bit 0 del divisor de velocidad
 Bit 1 = Bit 1 del divisor de velocidad
 Bit 2 = Bit 2 del divisor de velocidad
 Bit 3 = Bit 3 del divisor de velocidad
 Bit 4 = Bit 4 del divisor de velocidad
 Bit 5 = Bit 5 del divisor de velocidad
 Bit 6 = Bit 6 del divisor de velocidad
 Bit 7 = Bit 7 del divisor de velocidad

Figura 2.13. El Registro de Byte Menos Significativo del Divisor de Velocidad en Baudios del UART 8250 (DLL).



Bit 0 = Bit 8 del divisor de velocidad
 Bit 1 = Bit 9 del divisor de velocidad
 Bit 2 = Bit 0 del divisor de velocidad
 Bit 3 = Bit 11 del divisor de velocidad
 Bit 4 = Bit 12 del divisor de velocidad
 Bit 5 = Bit 13 del divisor de velocidad
 Bit 6 = Bit 14 del divisor de velocidad
 Bit 7 = Bit 15 del divisor de velocidad

Figura 2.14. El Registro de Byte Mas Significativo del Divisor de Velocidad en Baudios del UART 8250 (DLM).

2.2.2 El UART 16550

A pesar que el UART 16550 es bastante parecido al UART 8250 y al UART 16450, se debe resaltar que tiene buffers internos del tipo FIFO (first-in, first-out, el primero que entra es el primero que sale) los cuales sirven para almacenar temporalmente los bytes que son recibidos y transmitidos.

Estos buffer son extremadamente importantes cuando el dispositivo serial se utiliza en un ambiente multitarea. Supongamos que un programa de comunicaciones está efectuando una transferencia de archivo como una de las tareas entre varias en el PC. Si el modem es lento y el CPU es rápido, no se perderán caracteres cuando el CPU realiza la transferencia entre programas, Sin embargo, que sucedería si se tuviese una transferencia de datos de alta velocidad o un CPU lento, en ambos casos probablemente el programa de comunicaciones no tendría suficiente tiempo de CPU para grabar los caracteres entrantes desde el UART. Es por ello que los buffers FIFO del UART 16550 son importantes pues ellos pueden almacenar un limitado número de bytes internamente, así el programa de comunicaciones no debería preocuparse por la pérdida de bytes. El buffer de transmisión FIFO del UART 16550 tiene 16 bytes de longitud. El buffer de recepción FIFO del UART 16550 también posee 16 bytes, sin embargo, existe la posibilidad de ajustar el nivel de umbral de recepción (activación de interrupciones cuando se llega a cierto número de bytes recibidos) mediante

software. Para hacer uso de los buffers FIFO del UART 16550 se debe habilitarlos vía comandos software. Es importante resaltar que si el programador decide ignorar los buffers FIFO el UART 16550 resulta totalmente compatible con el UART 8250. A continuación se hará una descripción detallada de los registros del UART 16550 y podremos notar que con pocas excepciones que son necesarias para soportar los buffer FIFO, las funciones de los registros son las mismas que las del UART 8250.

Registro 0 :**Registro del Buffer de Receptor (RBR, Receiver Buffer Register)**

Cuando un caracter es recibido por el UART 16550, este es reconstruido y almacenado en el Receiver Buffer Register (RBR) del UART. El RBR es el primer registro del UART y puede ser direccionado leyendo el puerto del registro base del respectivo puerto serial. Para COM1 el RBR puede ser leído cogiendo el valor del puerto 3F8H. Aunque el RBR siempre cogerá 8 bits, se debe enmascarar los bits no usados cuando se trabaje con caracteres menores a 8 bits.

Bit	7	6	5	4	3	2	1	0
	X	X	X	X	X	X	X	X

- Bit 0 = Bit 0 del caracter entrante
- Bit 1 = Bit 1 del caracter entrante
- Bit 2 = Bit 2 del caracter entrante
- Bit 3 = Bit 3 del caracter entrante
- Bit 4 = Bit 4 del caracter entrante
- Bit 5 = Bit 5 del caracter entrante. Puede ser indefinido si se usan 5 bits de datos.
- Bit 6 = Bit 6 del caracter entrante. Puede ser indefinido si se usan 5 o 6 bits de datos.
- Bit 7 = Bit 7 del caracter entrante. Puede ser indefinido si se usan 5, 6 o 7 bits de datos.

Figura 2.15. El Registro del Buffer de Receptor del UART 16550 (RBR).

Registro 0 :

Registro de Retención (THR, Transmitter Holding Register).

El Transmitter Holding Register (THR) del UART 16550 es usado para transmitir un caracter hacia la interfase serial. Comparte el primer registro del UART con el RBR y puede ser direccionada escribiendo el puerto del registro base del puerto serial. Para el COM1, el THR puede ser escrita enviando un valor al puerto 3F8H.

Bit	7	6	5	4	3	2	1	0
	X	X	X	X	X	X	X	X

Bit 0 = Bit 0 del caracter saliente
 Bit 1 = Bit 1 del caracter saliente
 Bit 2 = Bit 2 del caracter saliente
 Bit 3 = Bit 3 del caracter saliente
 Bit 4 = Bit 4 del caracter saliente
 Bit 5 = Bit 5 del caracter saliente
 Bit 6 = Bit 6 del caracter saliente
 Bit 7 = Bit 7 del caracter saliente

Figura 2.16. El Registro de Retención del UART 16550 (THR).

Registro 1 :

Activación de Interrupción (IER, Interrupt Enable Register).

El Interrupt Enable Register (IER) del 16550 es usado para habilitar interrupciones. Puede ser direccionado escribiendo o leyendo el registro del puerto base más uno. Para COM1, el IER puede ser escrito o leído enviando o cogiendo el valor del puerto 3F9H.

Bit	7	6	5	4	3	2	1	0
	0	0	0	0	X	X	X	X

Bit 0 = RxDY: Cuando este bit es igual a 1, el UART 16550 generará una interrupción cuando un byte esté listo en el Receiver Buffer Register.

Bit 1 = TBE: Cuando este bit es igual a 1, el UART 16550 generará una interrupción si un byte es movido del Transmitter Holding Register hacia el registro interno Transmitter Shift Register del UART.

Bit 2 = Cuando este bit es igual a 1, el UART 16550 generará una interrupción cuando un error de paridad, un sobre rango, error de trama o una condición BREAK es detectada durante la llegada de un byte.

Bit 3 = Cuando este bit es igual a 1, el UART 16550 generará una interrupción si alguno del las líneas de entrada del RS-232 cambia de estado.

Bit 4 = Siempre 0.

Bit 5 = Siempre 0.

Bit 6 = Siempre 0.

Bit 7 = Siempre 0.

Figura 2.17. El Registro de Activación de Interrupción del UART 16550 (IER)

Registro 2 :**Identificación de Interrupciones (IIR, Interrupt Identification Register).**

El Interrupt Enable Register (IIR) del 16550 es usado para determinar que interrupciones pueden haber ocurrido y si los buffers FIFO son habilitados. Puede ser direccionado leyendo el registro del puerto base más dos. Para COM1, el IIR puede leído cogiendo el valor del puerto 3FAH.

Bit	7	6	5	4	3	2	1	0
	X	X	0	0	X	X	X	X

Bit 0 = Si este bit está en 1, no existen interrupciones pendientes. Por otro lado, un valor 0 indicará que una interrupción ha ocurrido.

Bit 3,2,1=Si el bit 0 es igual a 0, los bits 3, 2 y 1 indicarán el tipo de interrupción como sigue:

Bit3	Bit2	Bit1	Tipo de interrupción
0	0	0	Estado del modem
0	0	1	Buffer de TX vacío.
0	1	0	Dato recibido.
0	1	1	BREAK o error de serialización
1	1	0	Dato recibido y han transcurrido por lo menos cuatro bytes de tiempo sin que se haya leído algún byte

Bit 4 = Siempre 0.

Bit 5 = Siempre 0.

Bit 6 = Este bit será puesto en 1 si los FIFOs son habilitados. Si es 0, los FIFOs no están habilitados.

Bit 7 = Este bit será puesto en 1 si los FIFOs son habilitados. Si es 0, los FIFOs no están habilitados.

Figura 2.18. El Registro de Identificación de Interrupciones del UART 16550 (IIR).

Registro 2 :**Control del buffer FIFO (FCR, FIFO Control Register).**

El FIFO Control Register (FCR) del 16550 es usado para controlar los buffer FIFO. Puede ser direccionado escribiendo el registro del puerto base más dos. Para COM1, el FCR puede escrito enviando el valor al puerto 3FAH.

Bit 7 6 5 4 3 2 1 0

X	X	X	X	X	X	X	X
---	---	---	---	---	---	---	---

- Bit 0 = Habilitación FIFO: Si este bit se pone a 1, los buffers del UART 16550 serán habilitados. Si se pone a 0, los buffers del UART 16550 serán deshabilitados y el chip funcionará como el UART 16450.
- Bit 1 = Receiver FIFO Reset: Si este bit se pone a 1 se limpiará el buffer de recepción FIFO.
- Bit 2 = Transmitter FIFO Reset: Si este bit se pone a 1 se limpiará el buffer de transmisión FIFO.
- Bit 3 = Modo de selección MDA: Será 0 para el IBM PC y compatibles.
- Bit 4 = Reservado.
- Bit 5 = Reservado.
- Bit 7,6 = Si son habilitados los FIFOs, los bits 7 y 6 indicarán el umbral de recepción del buffer de recepción FIFO. Esto es, el número de bytes que el UART ubicará en el buffer FIFO antes de que se genere una interrupción de dato recibido. El umbral de recepción puede ser definido como sigue:
- | Bit 7 | Bit 6 | Umbral de recepción |
|-------|-------|---------------------|
| 0 | 0 | 1 byte |
| 0 | 1 | 4 bytes |
| 1 | 0 | 8 bytes |
| 1 | 1 | 14 bytes |

Figura 2.19. El Registro de Control FIFO del UART 16550 (FCR).

Registro 3 :**Fomato de Datos (LCR, Line Control Register).**

El Line Control Register (LCR) del 16550 es usado para definir parámetros como número de bits de data, número de bits de stop, tipo de paridad, etc. Puede ser direccionado escribiendo o leyendo el registro del puerto base más tres. Para COM1, el LCR puede ser escrito o leído enviando o cogiendo el valor del puerto **3FBH**.

Bit 7 6 5 4 3 2 1 0

X	X	X	X	X	X	X	X
---	---	---	---	---	---	---	---

Bit 1,0 = Los bits 1 y 0 determinan el número de bits de datos como sigue.

Bit 1	Bit 0	Número de bits de datos
0	0	5
0	1	6
1	0	7
1	1	8

Bit 2 = Si el bit es igual a 0, un bit de stop será usado. Si es igual a 1 se generará 2 bits de stop.

Nota: Si se usan 5 bits de datos y este bit está en 1, el UART usará 1 y 1/2 bits de stop.

Bit 5,4,3=Los bits 5,4 y 3 señalan la paridad a ser usada.

Bit 1	Bit 0	Bit 3	Paridad
0	0	0	NINGUNA
0	0	1	IMPAR
0	1	1	PAR
1	0	1	MARK
1	1	1	SPACE

Bit 6 = BREAK: Si este bit es igual a 1, el UART forzará la línea TD al estado lógico 0. El transmisor permanece incondicionalmente en este estado hasta que se escriba un 0 en este bit.

Bit 7 = Este bit no tiene nada que ver con el formato; se trata de un truco que permite extender el número de registros direccionados con las tres líneas de control. Cuando este bit está activo, los registros 0 y 1 (E/S de interrupciones y transmisor/receptor) se transforman en B-S y B+S de los registros de latch divisor.

Figura 2.20. El Registro de Formato de Datos del UART 16550 (LCR).

Registro 4**Control de modem (MCR, Modem Control Register).**

El Modem Control Register (MCR) del 16550 es usado para tareas como definir el estado de la línea RTS, línea DTR, habilitación de las interrupciones UART entre otras. Puede ser direccionado escribiendo o leyendo el registro del puerto base más cuatro. Para COM1, el MCR puede ser escrito o leído enviando o cogiendo el valor del puerto 3FCH.

Bit	7	6	5	4	3	2	1	0
	0	0	0	X	X	X	X	X

- Bit 0 = Data Terminal Ready (DTR): Cuando se escribe un 1 en este bit se activa la salida Data Terminal Ready.
- Bit 1 = Request To Send (RTS): Cuando se escribe un 1 en este bit se activa la salida Request to Send.
- Bit 2 = Salida de propósitos generales #1 (GPO1): Se trata de las dos primeras salidas definibles por el usuario.
- Bit 3 = Salida de propósitos generales #2 (GPO2): Es la segunda salida definible por el usuario.
- Bit 4 = Test Local Loopback: Asignando este bit a 1, el UART 16550 se encontrará en un estado de loopback local. Cualquier dato enviado por el transmisor será devuelto por el UART como un byte recibido. Bajo operación normal este bit deberá estar en 0.
- Bit 5 = Siempre 0
- Bit 6 = Siempre 0
- Bit 7 = Siempre 0

Figura 2.21. El Registro de Control de Modem del UART 16550 (MCR).

Registro 5 :**Estado de línea (LSR, Line Status Register).**

El Line Status Register (LSR) del 16550 informa de la disponibilidad de algún dato recibido, asimismo la detección de BREAK, errores de receptor y actividad de los registros de transmisión y recepción. Puede ser direccionado leyendo el registro del puerto base más cinco. Para COM1, el LSR puede ser leído cogiendo el valor del puerto **3FDH**.

Bit 7 6 5 4 3 2 1 0

X	X	X	X	X	X	X	X
---	---	---	---	---	---	---	---

- Bit 0 = RxDY: Si está en 1, un byte está disponible en el RBR. El bit queda anulado cuando el RBR y el FIFO quedan vacíos.
- Bit 1 = Sobreescritura de recepción: Si está en 1 indica que un error de sobreescritura ha ocurrido, esto es, un nuevo byte ha sido recibido y el buffer FIFO está lleno. El bit se anula cuando se lee.
- Bit 2 = Error de paridad: Se activa cuando el bit de paridad del byte recibido no coincide con la paridad debida según el formato de datos. El bit se anula cuando se lee.
- Bit 3 = Error de trama: Cuando este bit es 1, indica que tras ensamblar un byte recibido, su bit de STOP era incorrecto (es decir, SPACE en lugar de MARK). El bit queda anulado cuando se lee.
- Bit 4 = BREAK: Aparece un 1 en este bit cada vez que el receptor detecta una condición SPACE por un período mayor a un carácter.
- Bit 5 = Transmitter Buffer Empty (TBE): El bit TBE se activa para informar que se ha desplazado un byte desde el buffer del transmisor (registro de retención) al registro de desplazamiento. Si se olvida consultar este byte antes de escribir en el registro del buffer del transmisor puede llegar a borrarse un byte que estuviese ya instalado en él. Esta situación, conocida como sobreescritura de transmisor, no es informada por el 16550. Es importante comprender que aunque TBE advierta que el 16550 esté listo para aceptar otro byte de transmisión, esto no implica que se haya completado la transmisión del byte anterior.
- Bit 6 = El bit Transmitter Empty (TXE) indica que no existen bytes en el buffer de transmisor ni en su registro de desplazamiento, ni en el buffer de transmisión FIFO. Así, el UART ha transmitido todos los bytes si este bit está en 1.
- Bit 7 = Error de recepción FIFO: Si está en 1, existe al menos un error de paridad, error de trama, o BREAK en el buffer de recepción FIFO. Se anula cuando se lee.

Figura 2.22. El Registro de Estado de Línea del UART 16550 (LSR).

Registro 6 :**Estado de modem (MSR, Modem Status Register).**

El Modem Status Register (MSR) del 16550 informa del estado de las líneas CTS y DSR, si ocurre una llamada y si una portadora está presente. Puede ser direccionado leyendo el registro del puerto base más seis. Para COM1, el MSR puede ser leído cogiendo el valor del puerto **3FEH**.

Bit	7	6	5	4	3	2	1	0
	X	X	X	X	X	X	X	X

- Bit 0 = Delta CTS: Si este bit está en 1, la línea CTS ha cambiado de estado desde que el MSR fué leído por última vez. Se anula cuando es leído el MSR.
- Bit 1 = Delta DSR: Si este bit está en 1, la línea DSR ha cambiado de estado desde que el MSR fué leído por última vez. Se anula cuando es leído el MSR.
- Bit 2 = Delta Ring Indicator: Si este bit es 1, la línea RI ha cambiado de bajo a alto desde que el MSR fué leído por última vez. Se anula cuando se lee el MSR.
- Bit 3 = Delta Data Carrier Detect: Si este bit es 1, la línea DCD ha cambiado de estado desde que el MSR fué leído por última vez. Se anula cuando se lee el MSR.
- Bit 4 = Clear To Send (CTS): Este bit indica el estado actual de la línea CTS.
- Bit 5 = Data Set Ready (DSR): Este bit indica el estado actual de la línea DSR.
- Bit 6 = Ring Indicator (RI): Indica el estado actual de la línea RI.
- Bit 7 = Data Carrier Detect (DCD): Indica el estado de la línea DCD.

Figura 2.23. El Registro de Estado de Modem del UART 16550 (MSR).

Registro 7 :**Scratch pad.**

Este registro no tiene función reconocida, además no existe en las versiones iniciales del 16550.

Registro 8 :

Byte menos significativo del latch divisor de velocidad en baudios (DLL, Baud Rate LSB Divisor Latch Register).

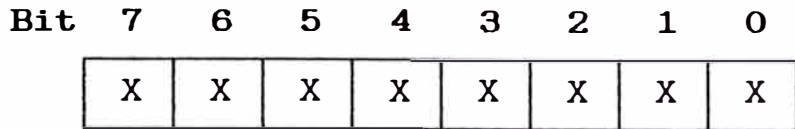
Registro 9 :

Byte mas significativo del latch divisor de velocidad en baudio (DLM, Baud Rate MSB Divisor Latch Register).

Cuando se pone en alto el bit 7 del registro de formato de datos (LCR), los registros 0 y 1 se transforman en los B-S y B+S del latch divisor, respectivamente. (Sus números de registro, 0/8 y 0/9, reflejan simplemente el parche realizado).

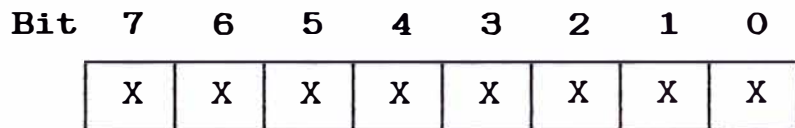
El reloj de referencia del 16550 se divide por un entero de 16 bits contenido en los registros B-S y B+S de latch divisor. La frecuencia resultante se utiliza como reloj maestro para controlar la lógica del transmisor y opcionalmente (por medio de una conexión externa), la del receptor. Este reloj maestro se divide a su vez por 16 para generar el reloj de baudios, que controla la velocidad de transmisión y recepción de datos. El divisor para cualquier velocidad puede calcularse según la fórmula:

$$\text{Divisor} = \frac{\text{Frecuencia del reloj de referencia}}{16 \times \text{velocidad deseada}}$$



Bit 0 = Bit 0 del divisor de velocidad
 Bit 1 = Bit 1 del divisor de velocidad
 Bit 2 = Bit 2 del divisor de velocidad
 Bit 3 = Bit 3 del divisor de velocidad
 Bit 4 = Bit 4 del divisor de velocidad
 Bit 5 = Bit 5 del divisor de velocidad
 Bit 6 = Bit 6 del divisor de velocidad
 Bit 7 = Bit 7 del divisor de velocidad

Figura 2.24. El Registro de Byte Menos Significativo del Divisor de Velocidad en Baudios del UART 16550 (DLL).



Bit 0 = Bit 8 del divisor de velocidad
 Bit 1 = Bit 9 del divisor de velocidad
 Bit 2 = Bit 0 del divisor de velocidad
 Bit 3 = Bit 11 del divisor de velocidad
 Bit 4 = Bit 12 del divisor de velocidad
 Bit 5 = Bit 13 del divisor de velocidad
 Bit 6 = Bit 14 del divisor de velocidad
 Bit 7 = Bit 15 del divisor de velocidad

Figura 2.25. El Registro de Byte Mas Significativo del Divisor de Velocidad en Baudios del UART 16550 (DLM).

2.3 Tarjeta interfase en la caja registradora

Nos corresponde ahora revisar la interfase hardware correspondiente a la caja registradora.

Existen diversos modelos de cajas registradoras, ello implica, que existirán diversos modelos de tarjetas interfase de acuerdo a las necesidades específicas del usuario final, tomaremos como base de estudio el modelo de caja registradora marca CASIO modelo TK-2300, en este modelo se puede instalar una tarjeta interfase denominada I/O-PB-11, esta tarjeta tiene como características las funciones siguientes:

- a) Sistema de comunicación On-line, que consiste en la comunicación de datos punto a punto con un computador.
- b) Sistema de comunicación In-line, que consiste en la comunicación de datos entre cajas registradoras.
- c) Salida para Impresora Centronics, que permite conectar la caja registradora a una impresora del tipo Centronics para la emisión de una gama de reportes.
- d) Sistema de Balanza, que permite la conexión a una balanza electrónica.

Sistema de comunicación On-line.- El sistema de comunicación On-line consiste en la comunicación de datos entre una caja registradora y un computador personal. La velocidad de comunicación puede ser programada en la caja registradora y son posibles los valores de 1200, 2400, 5800, 9600 ó 19200 bps. Se puede lograr esta comunicación de dos formas, la primera es la conexión en forma directa caja registradora-PC vía la interfase RS-232C en el

computador y la otra es utilizando modems en ambos extremos a través de una línea privada o pública.

a) Comunicación On-line directa.- En esta modalidad se hace uso de la interfase RS-232C existente en el micro-computador. Para el enlace respectivo se utiliza un cable mallado de 9 hilos (este es un cable comercial) de los cuales sólo se utilizan 8. Este modelo de conexión será el utilizado más adelante para llevar a cabo la implementación de la interfase de comunicación caja registradora-computador. El diagrama es el mostrado en la figura 2.26.

b) Comunicación On-line utilizando modem.- Bajo esta modalidad se hace uso de dos modems, uno se ubica al lado del computador y otro modem al lado de la caja registradora. La configuración se muestra en la figura 2.27.

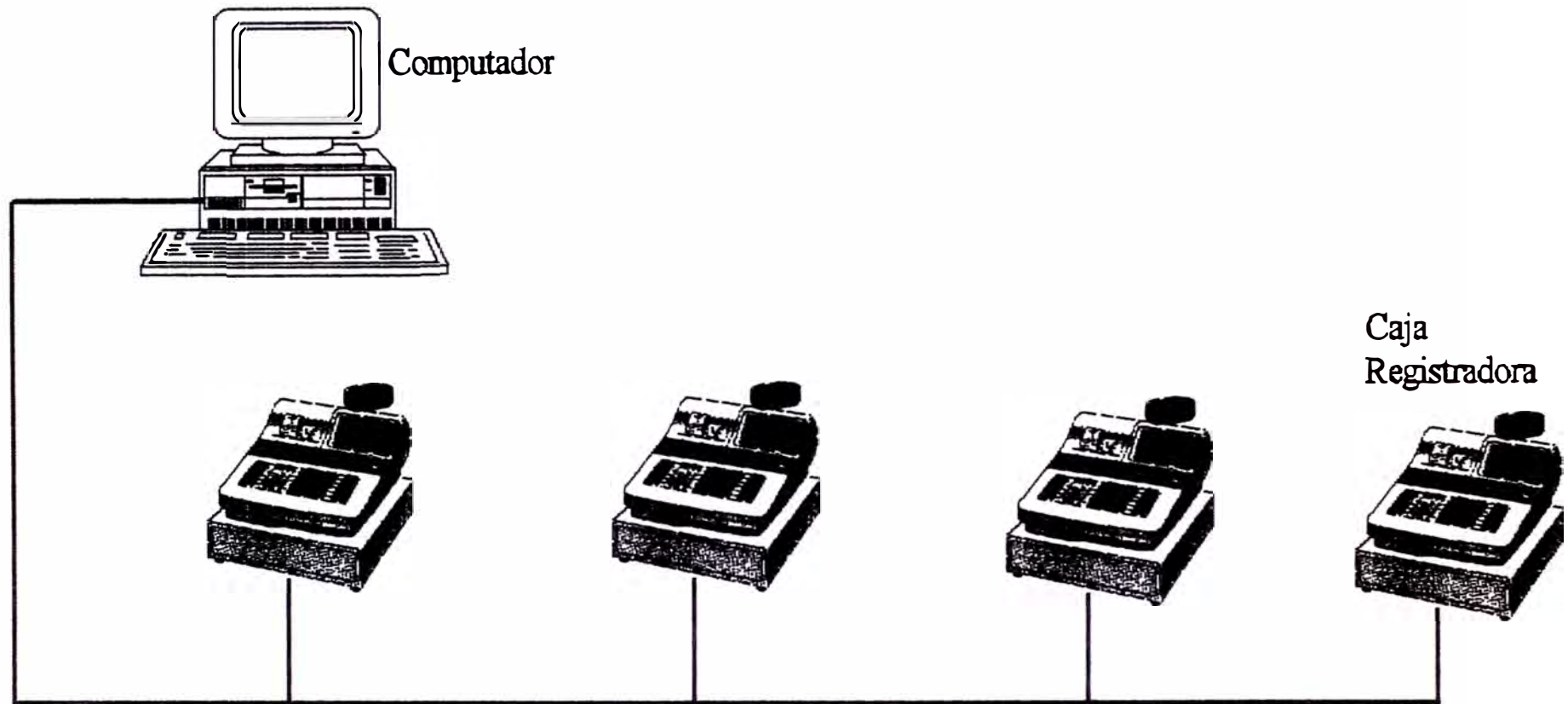


Figura 2.26. Sistema de comunicación On-line directa

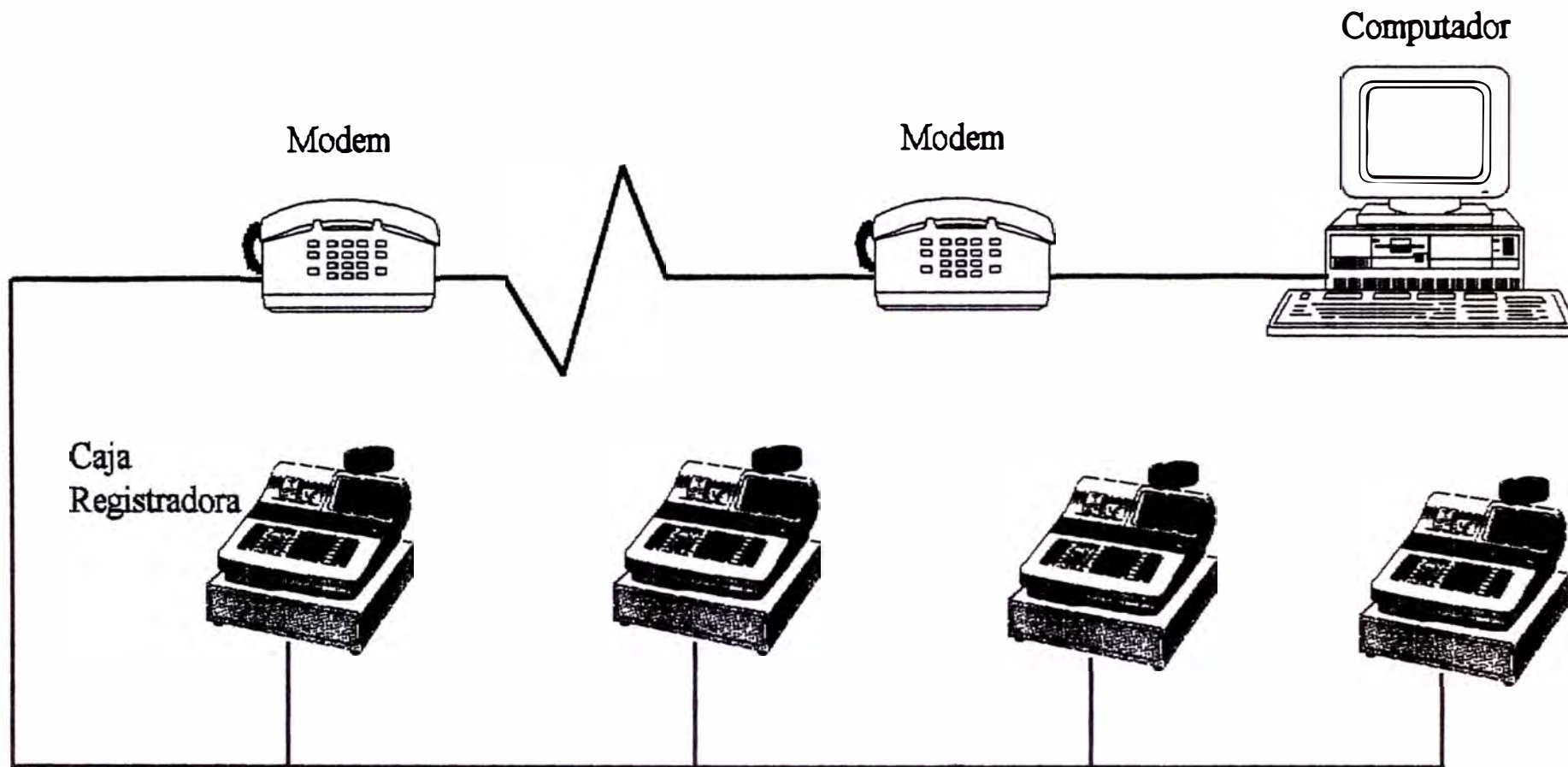


Figura 2.27. Sistema de comunicación On-line con modem

Sistema de comunicación In-line.- El sistema de comunicación In-line consiste en la interconexión de más de una caja registradora. Para ello se tiene el concepto de caja registradora maestra y caja registradora satélite. La caja registradora maestra viene a ser aquella que centraliza todas las operaciones de programación de datos y el envío y recepción de datos con el resto de cajas. Por ejemplo, si se desea programar una lista de artículos en la línea completa de cajas, esta se efectuará primero en la caja registradora maestra y luego mediante algunos comandos se transmitirá al resto de cajas registradoras. Las cajas registradoras satélites, como resulta obvio, vienen a ser el resto de cajas conectadas a la maestra y que tienen como función el efectuar las transacciones comerciales.

El sistema de comunicación In-line se utiliza ampliamente en supermercados, tiendas de varios ambientes y niveles, grifos, etc. El número máximo de satélites que pueden ser conectados es variable de acuerdo al modelo de caja registradora. La velocidad de transmisión puede ser programada a valores desde 4800 bps hasta 1.25 M bps. La distancia máxima entre la maestra y la última satélite es de 1 (uno) kilómetro. La comunicación se realiza a través de un cable de 2 (dos) hilos. La figura 2.28. muestra la configuración posible.

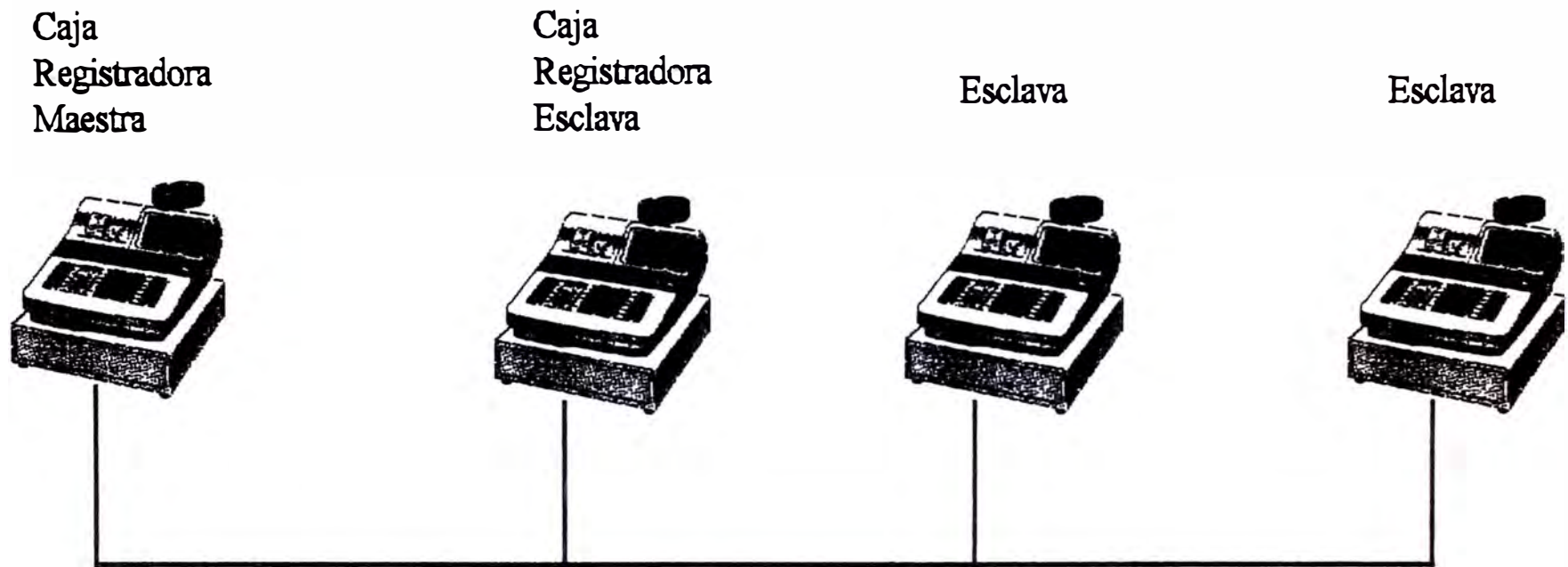


Figura 2.28. Sistema de comunicación In-line

2.4 Conexión caja registradora-computador

Se ha visto anteriormente las características de los dos elementos hardware que intervienen en la comunicación caja registradora-computador, esto es, la interfase serial RS-232 en el computador y la tarjeta de interfase I/O-PB-11 en la caja registradora.

Corresponde ahora el estudio de la conexión entre ambos equipos. El sistema de comunicaciones a utilizar es el tipo On-line que consiste en un enlace directo caja registradora-computador.

Para información adicional remitirse al apéndice A.

Conectores

Se utilizan dos conectores.

El conector por el lado de la caja registradora es uno que cumple la norma DIN 13P. Se trata de un conector macho de trece pines que hace juego con una entrada del tipo hembra presente en la tarjeta de interfase I/O-PB-11 de la caja registradora.

En el computador se utiliza el bastante conocido DB-25, conector hembra de 25 pines que hace juego con la salida serial del computador.

Cable

Para el enlace se necesitan 8 líneas, es por ello que se hará uso de un cable comercial de 9 hilos mallado. Es necesario la presencia del cable mallado para evitar interferencias de comunicación. Se debe tener muy presente que el tendido del cable debe ubicarse a una

distancia mínima de 15 centímetros respecto a los cables de energía eléctrica del establecimiento.

Longitud máxima de cable

La longitud teórica máxima (recomendada por la norma 232) del cable de conexión caja registradora-computador es 15 metros.

Diagrama de cableado

El diagrama de cableado es mostrado en la figura 2.29.

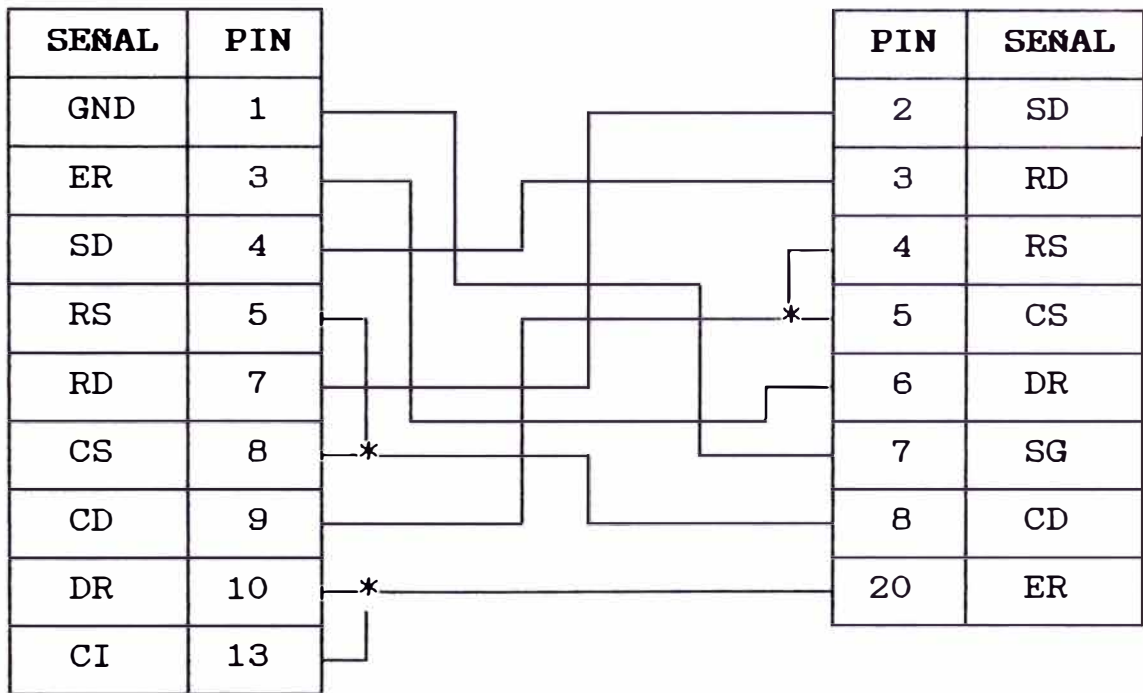


Figura 2.29. Diagrama de cableado caja registradora-PC.

Descripción de las líneas utilizadas

Las líneas utilizadas en el diagrama de cableado se describen a continuación:

Línea SD. Pin 4.

TXD: Transmit Data.- Transmite datos desde el terminal (PC) al data set (modem).

Línea RD. Pin 7.

RXD: Receive Data.- Recibe datos desde el data set hacia el terminal.

Línea RS. Pin 5.

RTS: Request To Send.- Señal RTS desde el terminal hacia el data set. La señal se va al nivel alto cuando un bloque de datos es transmitido mientras la señal DR está en ON. Cuando la transmisión es finalizada, la línea pasa al estado OFF.

Línea CS. Pin 8.

CTS: Clear To Send.- Señal solicitando transmisión, es enviado desde el data set hacia el terminal. Cuando la señal está en ON, el terminal está disponible para enviar información.

Línea DR. Pin 10.

DSR: Data Set Ready.- Señal informando al terminal que el data set está listo. Cuando la señal está en ON, el terminal está disponible para enviar o recibir datos.

Línea CD. Pin 9.

CD: Carrier Detect.- Señal informando al terminal que el data set ha recibido una portadora.

Línea ER. Pin 3.

DTR: Data Terminal Ready.- Señal informando al data set que el terminal está listo para enviar o recibir datos.

Línea SG. Pin 1.

SG: Signal Ground.- Señal de masa.

Línea CI. Pin 13.

CI: Calling Indicator.- Indicación de llamada

Línea FG.

FG: Frame Ground.-

CAPITULO III PROTOCOLO DE COMUNICACIONES

El protocolo utilizado para establecer la comunicación de datos entre la caja registradora y el computador se basa en el protocolo IBM BSC 3780 y se denomina Protocolo Asíncrono CASIO.

3.1 Protocolo asíncrono CASIO

El Protocolo Asíncrono CASIO es usado para establecer una comunicación del tipo ON-LINE entre la caja registradora y el computador.

El proceso de enlace computador-caja registradora consta de dos partes fundamentales, en primer lugar el computador decide que tarea se ha de llevar a cabo (transmisión o recepción de datos), este proceso se efectúa mediante el envío por parte del computador de un bloque de datos denominado "bloque comando", en este bloque se indica el tipo de comunicación a efectuar, es decir, si se trata de un proceso de lectura de los archivos de ventas existentes en la caja registradora, transmisión de los archivos de programación de la caja registradora o recepción de los archivos de programas existentes en la caja registradora, asimismo, se define el tipo de reporte de ventas a recibir o el número del archivo de programación a ser enviado o recibido, también se define si luego del proceso de lectura de ventas se

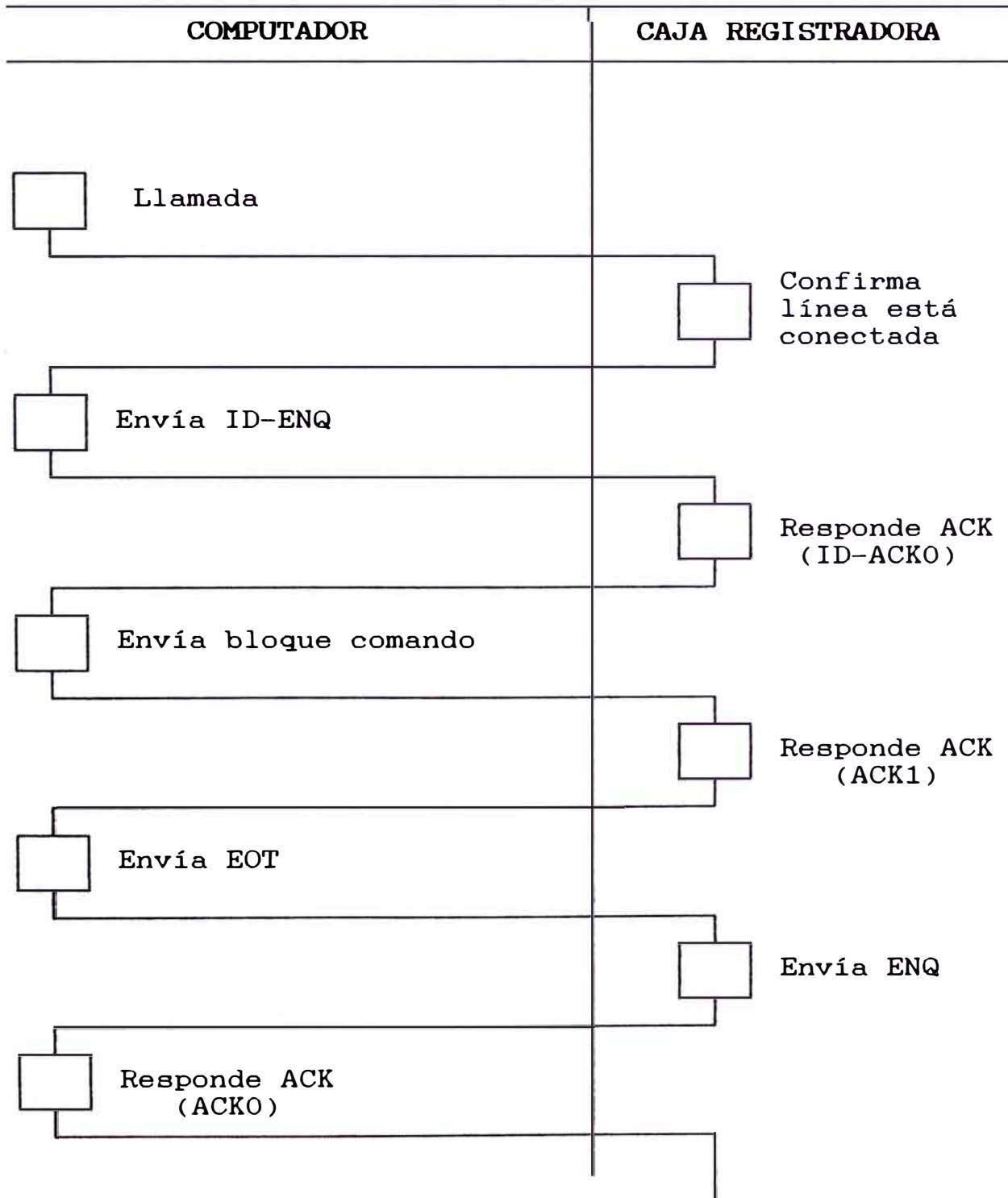
borran o permanecen los datos existentes en la caja registradora, una vez definida las características del proceso a efectuar se ingresa de llano al proceso de envío o recepción de datos el cual está regido por un protocolo de comunicación del tipo asíncrono, este protocolo de comunicaciones define los procedimientos a seguir, tanto por parte del computador, como por la caja registradora para ejecutar un correcto proceso de envío y/o recepción de datos, el protocolo define, asimismo, los pasos a seguir en caso de existencia de errores en uno u otro lado del enlace.

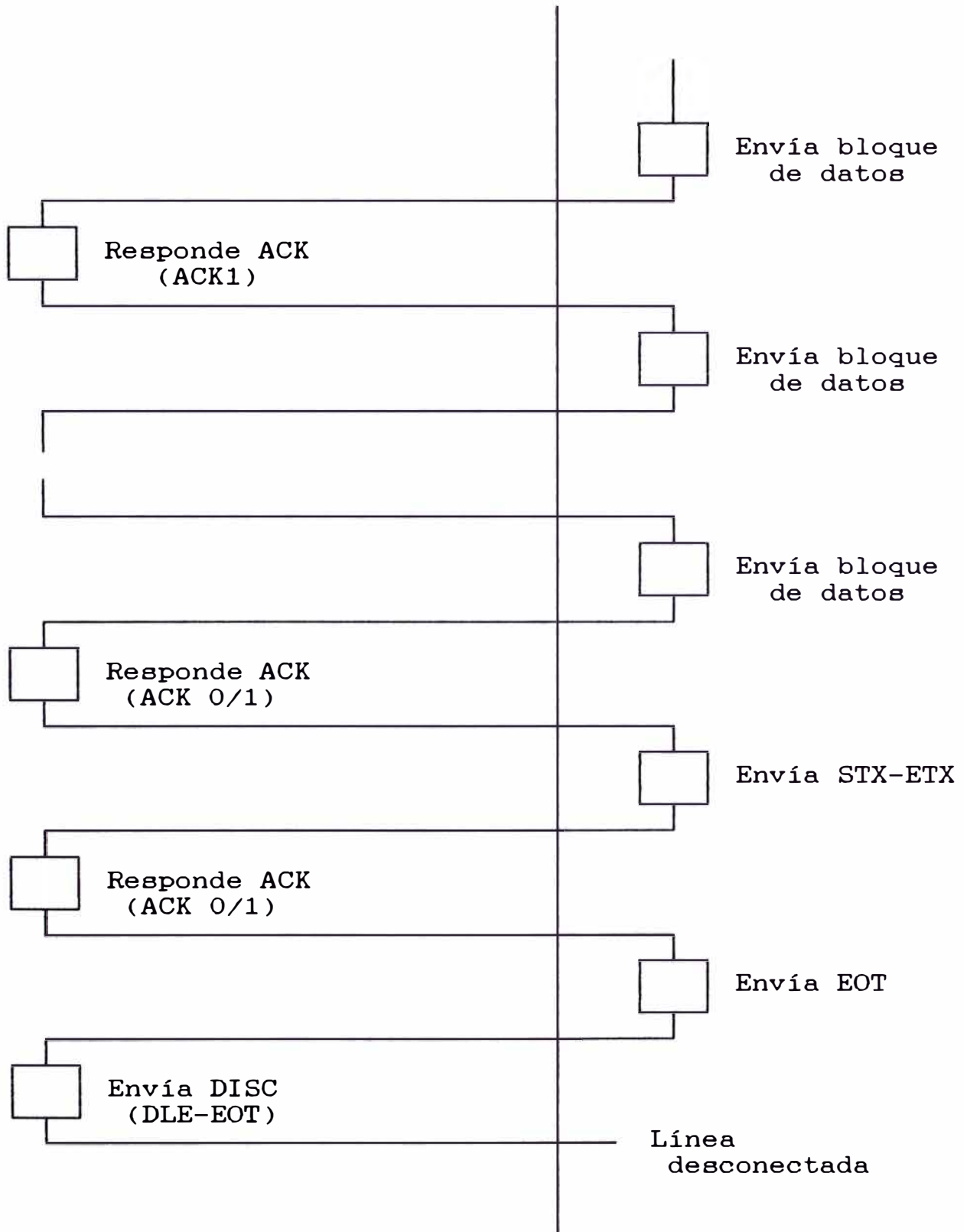
Se detalla a continuación los diagramas de flujo para los siguientes procesos:

- a) Transmisión de datos
- b) Recepción de datos
- c) Respuesta negativa (NAK)
- d) Repetición de solicitud (ENQ)
- e) Verificación de secuencia ACK0/1
- f) Respuesta WACK (espera)
- g) Respuesta TTD (retardo)

a) Recepción de datos

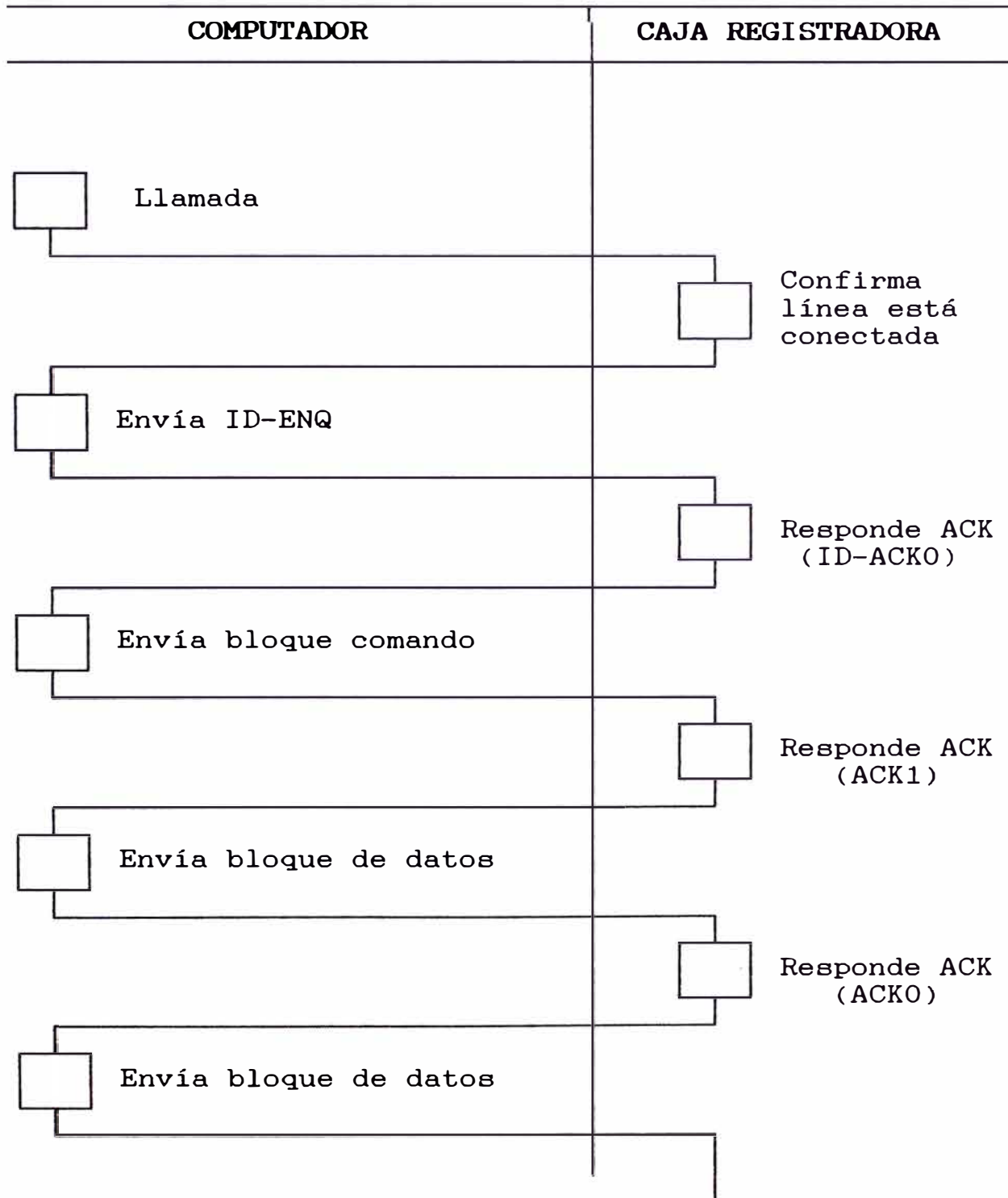
El siguiente es el diagrama que muestra la secuencia de sucesos para llevar a cabo la recepción de datos enviada por la caja registradora.

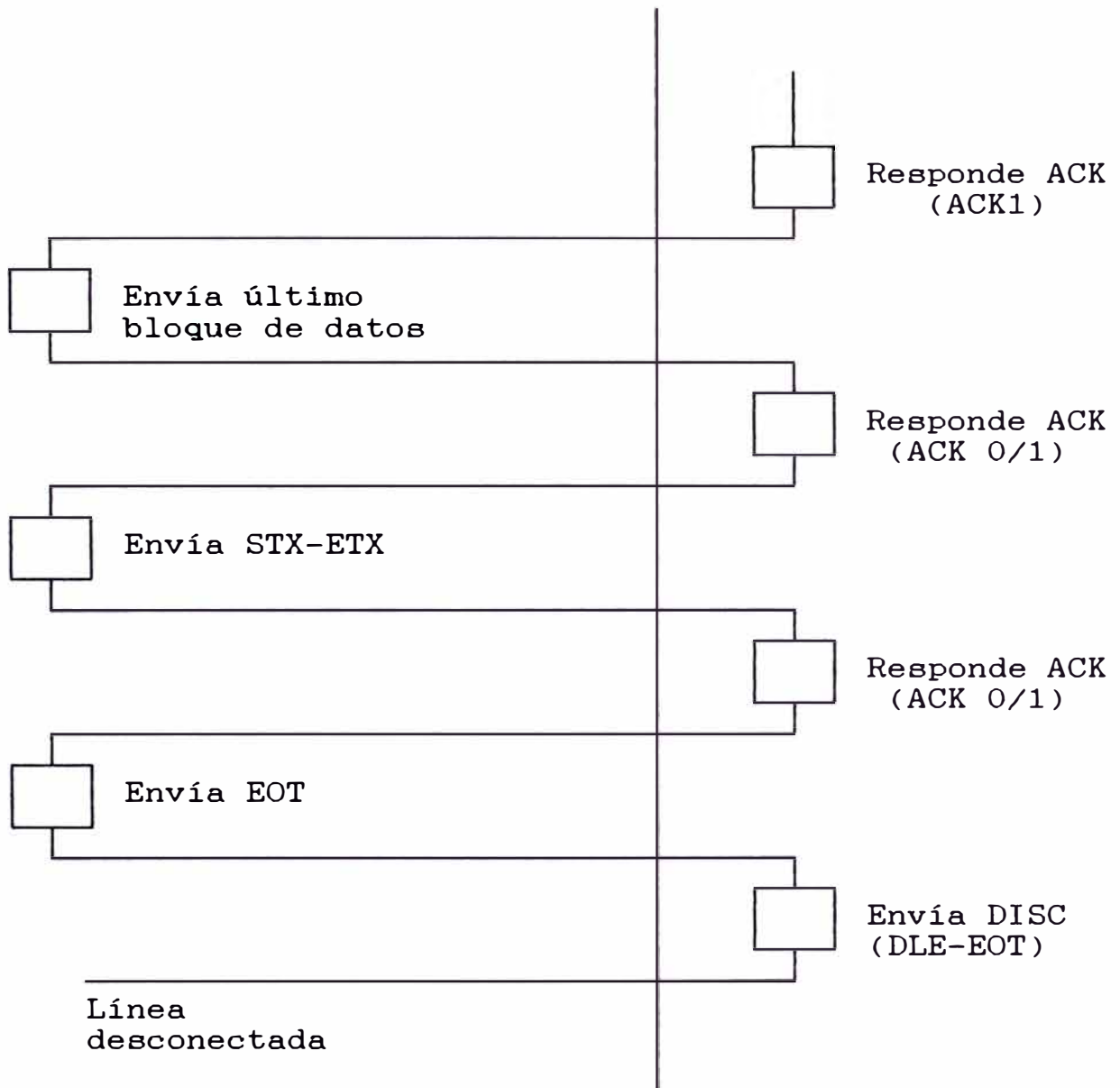




b) Transmisión de datos Computador-Caja Registradora

El siguiente es el diagrama que muestra la secuencia de sucesos para llevar a cabo la transmisión de datos desde el computador hacia la registradora.



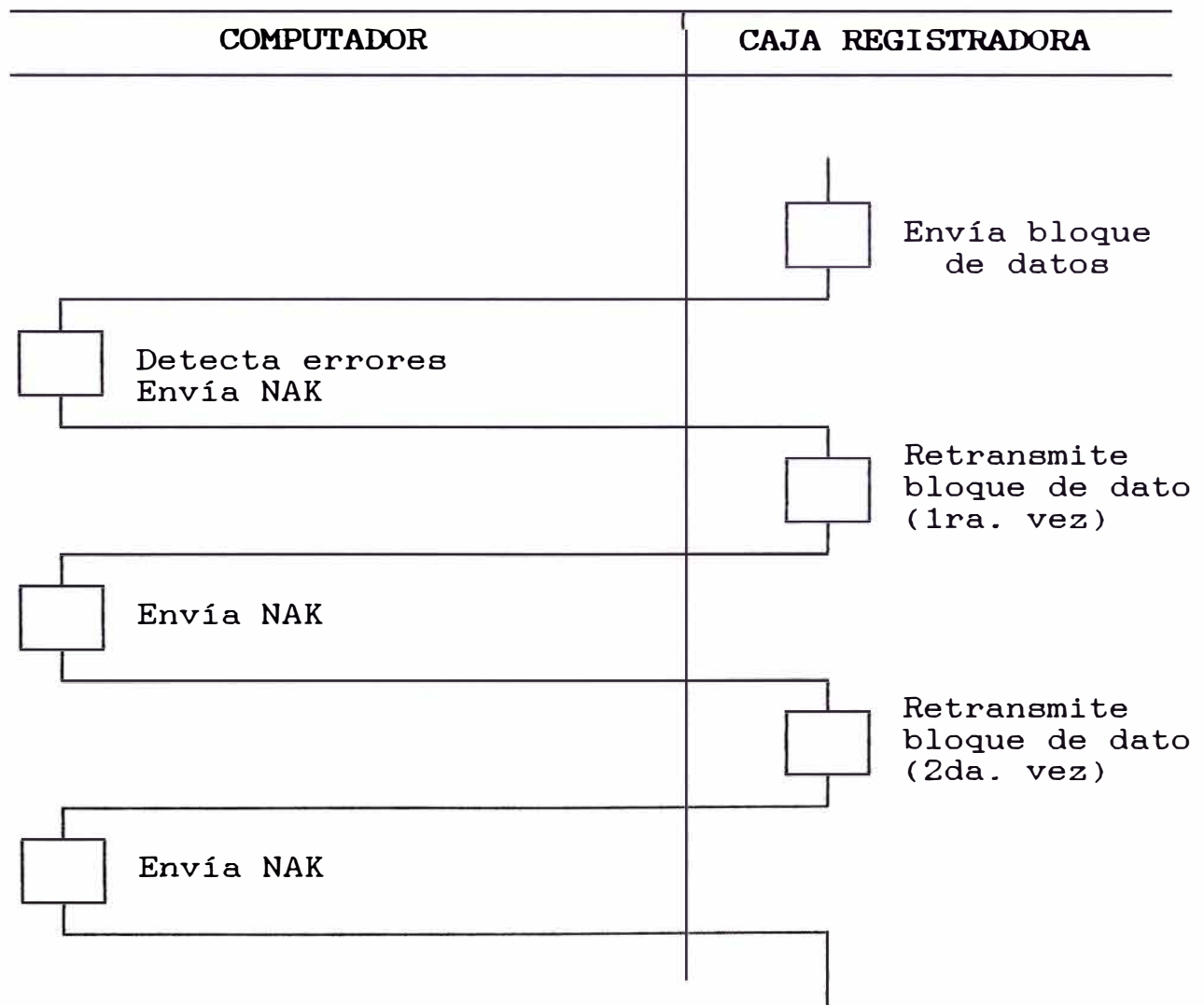


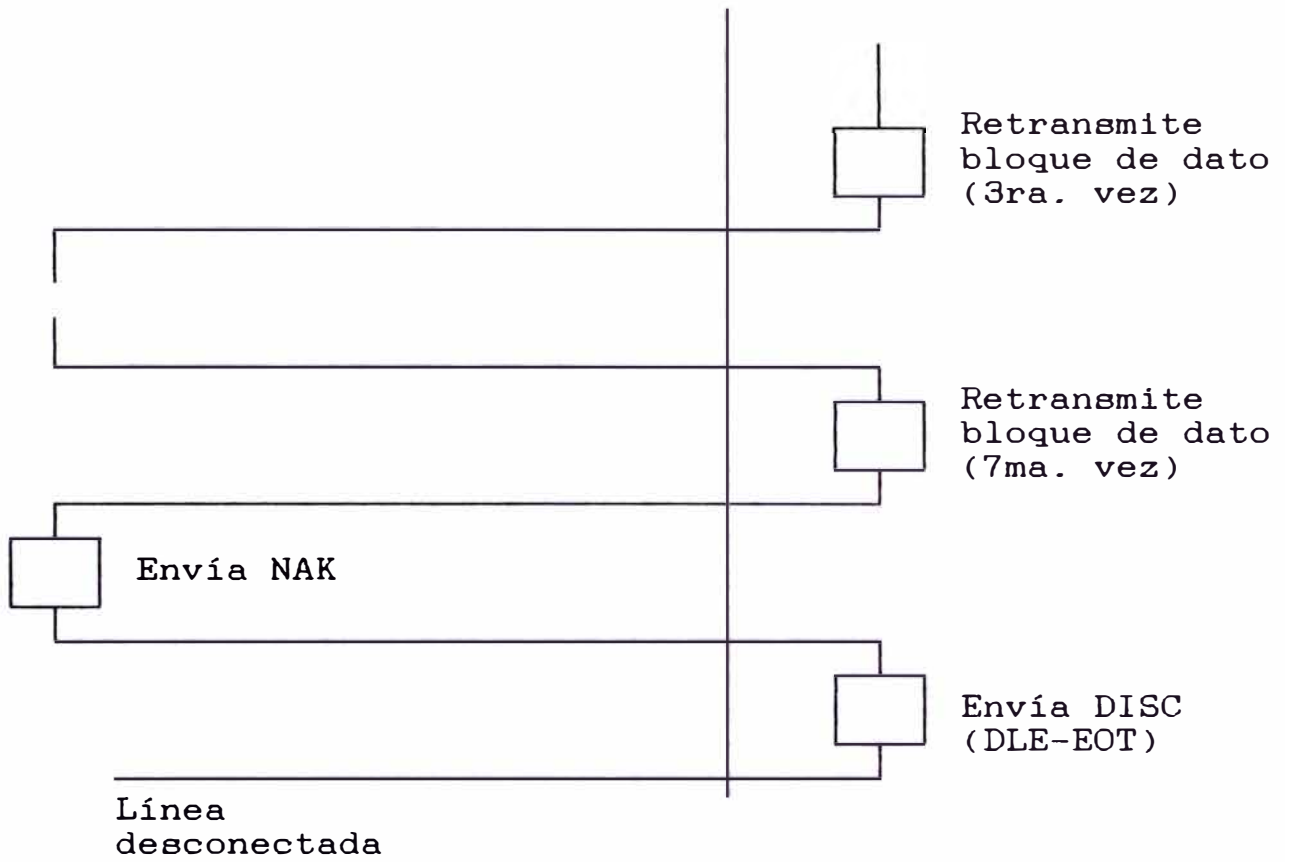
c) Respuesta negativa (NAK)

El caracter de control NAK puede ser transmitido desde el computador hacia la caja registradora en dos casos:

- Cuando la computadora no puede contestar a una solicitud ENQ de la caja.
- Cuando un bloque de datos no ha sido recibido completamente.

El mismo bloque de datos puede ser transmitido hasta siete veces por la caja registradora.

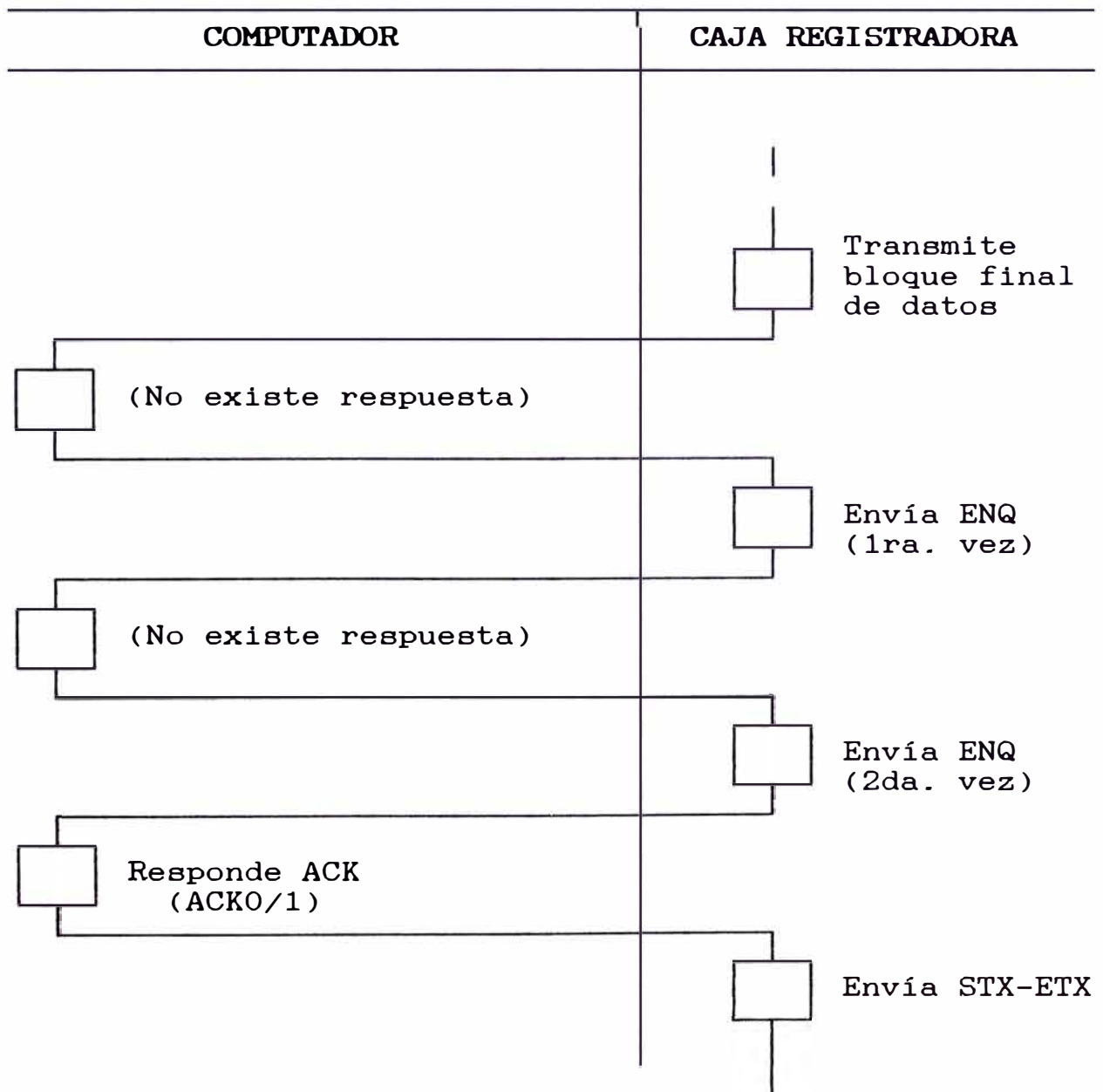


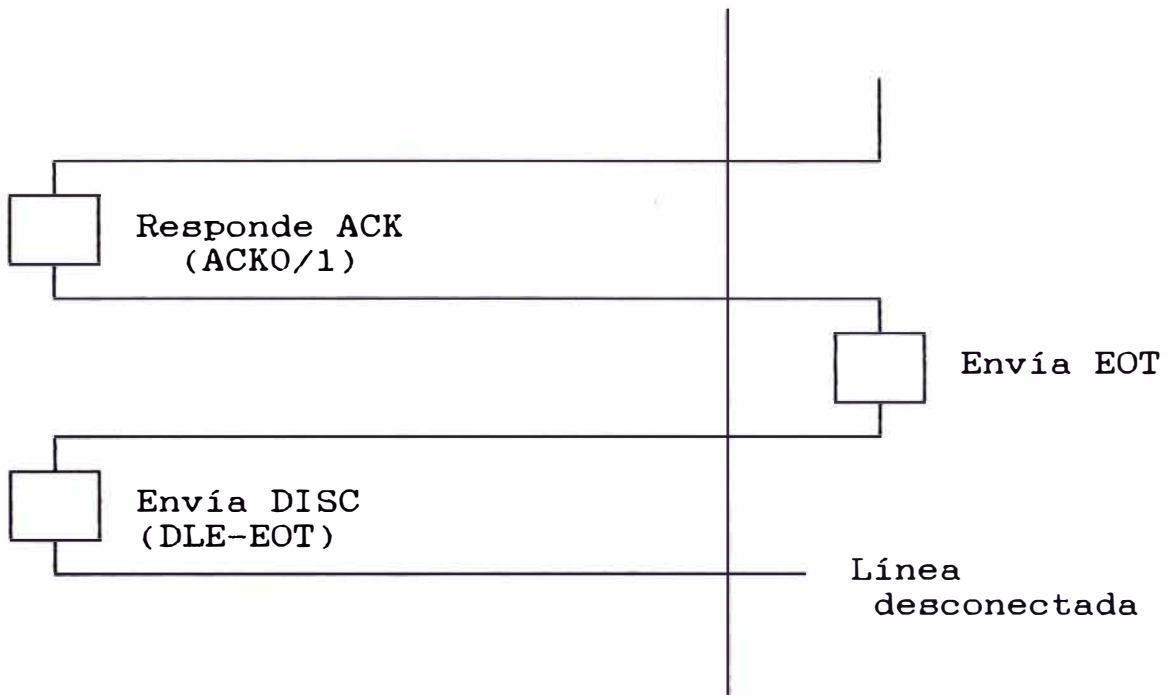


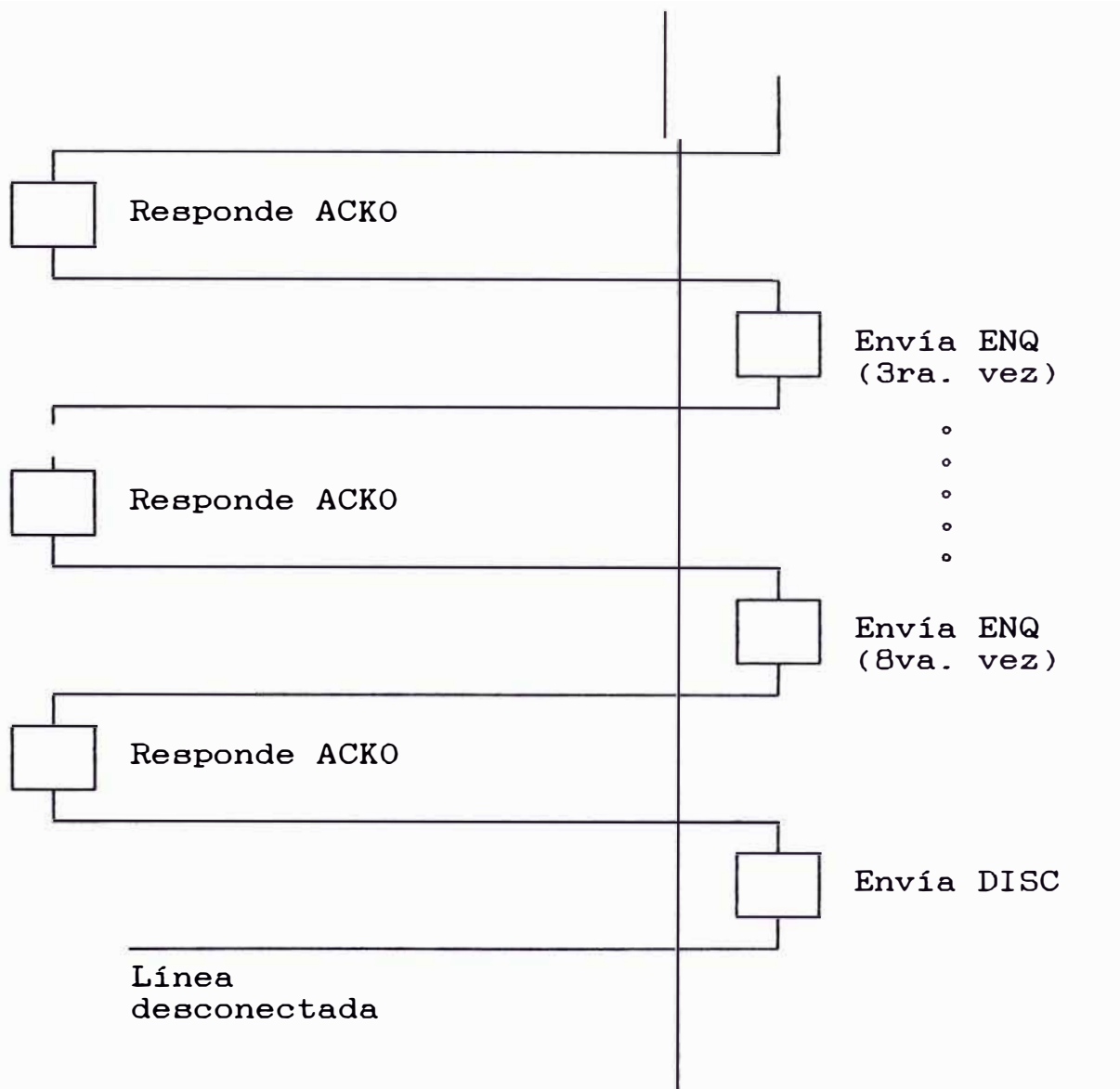
d) Repetición de solicitud (ENQ)

La caja registradora puede repetir el envío de solicitud (ENQ) hasta ocho veces en los casos siguientes:

- La caja registradora no recibe un ACK0/1 luego de enviar un ENQ.
- La caja registradora no recibe un ACK0/1 luego de enviar el final de un bloque de datos.



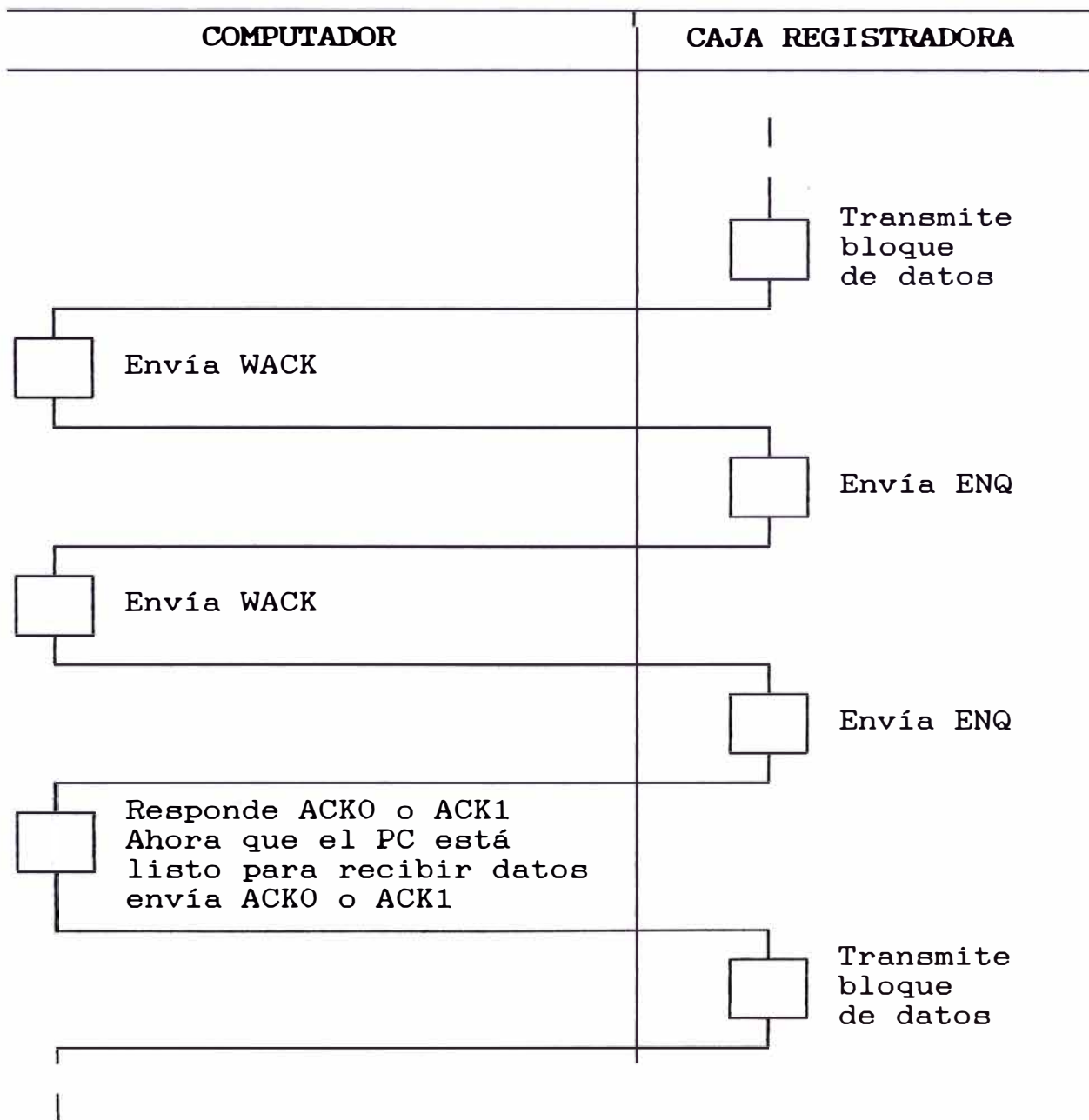




En el diagrama anterior, la caja registradora recibe un ACK0 cuando estaba esperando un ACK1, por lo tanto, envía un ENQ con el fin de solicitar una respuesta ACK1. Como ningún ACK1 es recibido luego de ocho intentos, la caja registradora envía al computador un código DISC dando por terminada la comunicación.

f) Respuesta WACK (espera)

Si se da el caso que mientras la caja registradora está enviando información al computador, este por alguna razón se ve congestionado por otras tareas y está temporalmente imposibilitado de recibir datos, el computador puede enviar un WACK hacia la caja registradora indicando que ingrese al modo espera.



La respuesta WACK es un reconocimiento positivo a una solicitud ENQ recibida. La caja registradora, habiendo recibido un WACK, continúa enviando códigos ENQ para revisar si el computador se encuentra listo para recibir el siguiente bloque.

No existe límite al número de intentos que puede efectuar la caja registradora. Si el computador desea terminar la comunicación con la caja registradora puede enviar un código DISC.

Es importante notar que la caja registradora no puede enviar un código WACK.

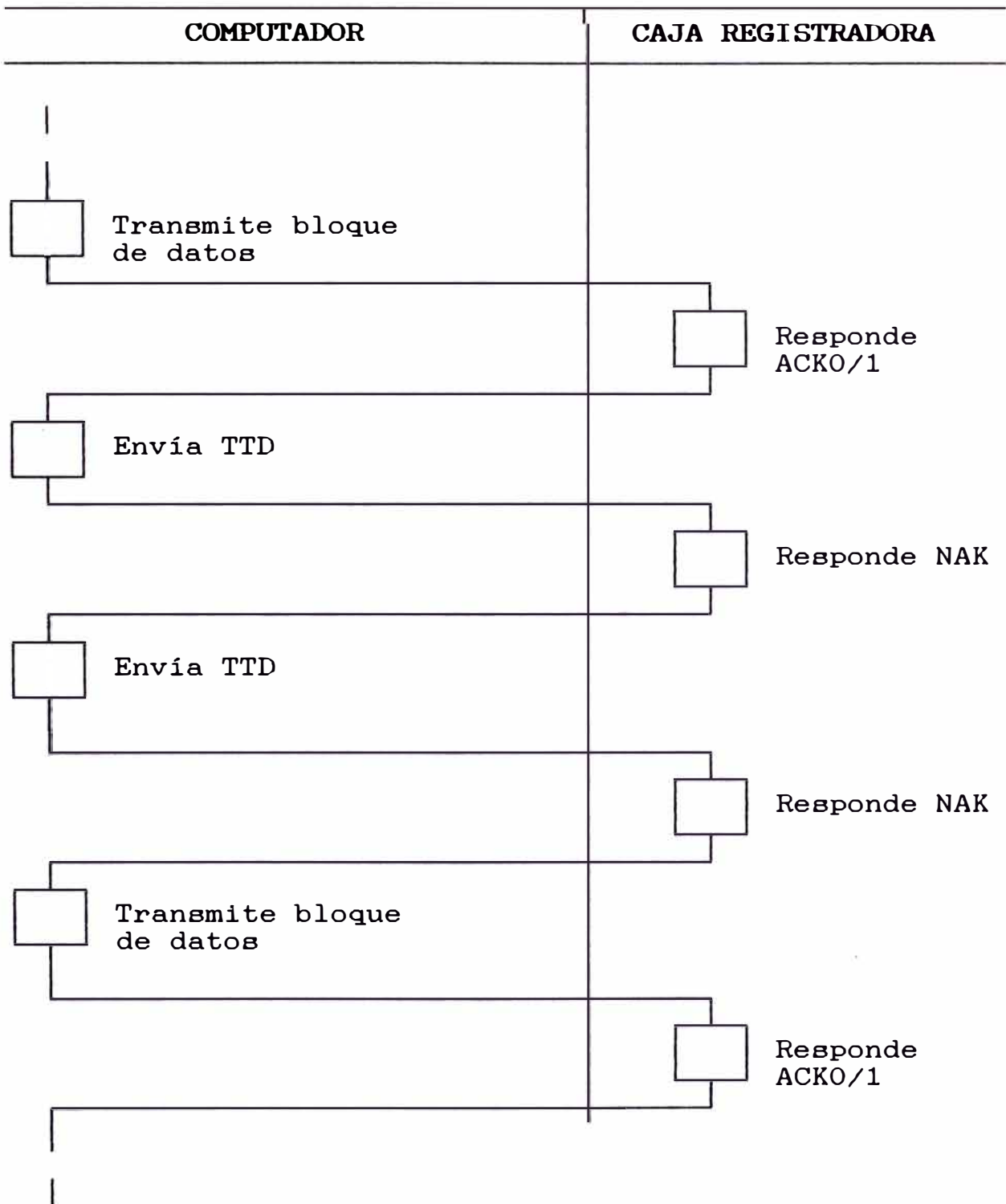
g) Respuesta TTD (retardo)

El caracter TTD es enviado por el computador hacia la caja registradora cuando el primero no puede enviar el siguiente dato dentro de los dos segundos siguientes a la recepción de un ACK. Este código previene la ocurrencia de un time-out.

Cuando la caja recibe un TTD, este responde un NAK al computador. Como en el caso de un código WACK, no existe límite al número de veces que el código TTD puede ser enviado. Así el computador puede enviar códigos TTD hasta que se encuentre listo a enviar el siguiente dato.

Cuando la caja registradora recibe un TTD desde el computador, se envía una respuesta NAK.

La caja registradora no puede enviar un código TTD.



Time-Out

Un resultado Time-Out se da en los siguientes casos:

Luego de enviar el código ID-ENQ, la caja registradora espera 1.5 segundos antes de repetir el código ENQ. Si ninguna respuesta es recibida del computador, la caja envía el código ENQ al computador hasta quince veces.

Si luego de enviar los códigos STX, DATA, ETB ningún reconocimiento es recibido dentro de los tres segundos. Se reintenta hasta siete veces.

Si han transcurrido mas de dos segundos luego que un código WACK o TTD ha sido enviado.

Si la caja registradora no recibe el código ID-ENQ dentro de veinte segundos después de establecida la línea de comunicaciones.

3.2 Caracteres de control

Se ha visto en los diagramas del protocolo de comunicaciones caja registradora-computador que para señalar los procesos se hace uso de ciertos códigos especiales tales como ENQ, ACK0, ACK1, STX, ETB, NAK, WACK, etc., este conjunto de caracteres recibe el nombre de caracteres de control.

La siguiente es una lista de los caracteres de control utilizados para establecer el protocolo de comunicaciones caja registradora-computador y se trata de un conjunto de caracteres EBCDIC.

CARACTER DE CONTROL	FUNCION	EBCDIC (Hex)
STX	Inicio de Texto	02
ETB	Fin de bloque de Texto	26
EOT	Fin de Transmisión	37
ENQ	Solicitud	2D
ACK0	Reconocimiento positivo Par	1070
ACK1	Reconocimiento positivo Impar	1061
NAK	Reconocimiento negativo	3D
WACK	Espera antes de transmitir	1068
TTD	Retardo temporal	022D
DLE	Extensión en control de TX	10
DISC	Desconexión de línea	1037

3.3 Alfabeto de comunicaciones

El conjunto de caracteres permitido en la caja registradora cubre los números decimales, los caracteres mayúscula y minúscula de la "A" a la "Z" y muchos caracteres adicionales.

A continuación se muestra la tabla indicada y asociada a ella un método de conversión de texto a hexadecimal que será utilizada para la codificación de los textos a ser enviados o para traducir los textos recibidos.

ALFABETO EN CAJA REGISTRADORA Y TABLA DE CONVERSION HEXADECIMAL

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		0	G	X	i	y	è	£								
1	/	1	H	Y	j	z	à	¥								
2	'	2	I	Z	k	+	ò	@								
3	.	3	J		l	-	ù	*								
4	&	4	K	?	m	%	ø									
5	#	5	L	l	n	o	ó									
6		6	M	.	o	Ä	ä	:								
7		7	N	,	p	ö										
8		8	P	a	q	ü										
9		9	Q	b	r	á										
A		A	R	c	s	ó										
B		B	S	d	t	ú	Å									
C		C	T	e	u	í										
D		D	U	f	v	é	(
E		E	V	g	w	ß)				°					
F		F	W	h	x	Ñ	\$				~					Db

Db es el símbolo que indica caracter doble

Traducción de texto

Cuando se envían datos desde el computador hacia la caja registradora, es necesario traducir el texto a un formato especial en el cual cada caracter original se traduce a dos bytes. El proceso de traducción es el siguiente:

Paso 1 : Se codifica el texto de acuerdo a la tabla de conversión hexadecimal anterior.

Paso 2 : Cada caracter resultante (número o caracter) del paso 1 se codifica de acuerdo a la tabla de conversión binaria siguiente:

TABLA DE CONVERSION BINARIA

Número	Traducción		Caracter	Traducción
0	F0		A	C1
1	F1		B	C2
2	F2		C	C3
3	F3		D	C4
4	F4		E	C5
5	F5		F	C6
6	F6			
7	F7			
8	F8			
9	F9			

Ejemplo 1 :

- Traducir la palabra "UNI"

Paso 1 : Se traduce de acuerdo a la tabla de conversión hexadecimal.

UNI = hex 2D2722

Paso 2 : Se traduce el resultado del paso uno de acuerdo a la tabla de conversión binaria.

2 D 2 7 2 2
2D2722 = F2C4F2F7F2F2

Por lo tanto :

UNI = F2C4F2F7F2F2

U N I = F2 C4 F2 F7 F2 F2

3 bytes 6 bytes

Ejemplo 2 :

- Traducir la palabra "TESIS"

Paso 1 : Se traduce de acuerdo a la tabla de conversión hexadecimal.

TESIS = hex 2C1E2B222B

Paso 2 : Se traduce el resultado del paso uno de acuerdo a la tabla de conversión binaria.

2 C 1 E 2 B 2 2 2 B
2C1E2B222B = F2C3F1C5F2C2F2F2F2C2

Por lo tanto :

TESIS = F2C3F1C5F2C2F2F2F2C2

5 bytes 10 bytes

3.4 Formatos de comunicación

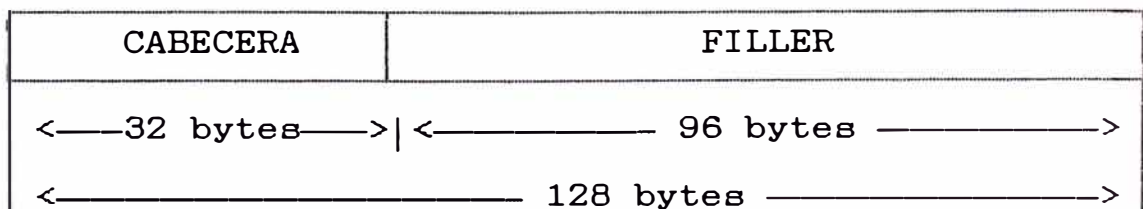
A continuación se describen los formatos de las tramas de comunicación utilizadas para el envío y/o recepción de datos entre la caja registradora y el computador. Se define en primer lugar el formato del bloque comando y a continuación el formato de la trama de datos.

3.4.1 Formato del bloque comando

Como se comentó en la parte inicial del presente capítulo, el computador define que tipo de proceso ha de llevar a cabo con la caja registradora, es decir, si se reciben archivos de ventas o programación, si se envían archivos de datos o programación a la caja registradora.

Para indicar estas características el computador envía una trama denominada "bloque comando" cuyo formato se define a continuación.

La longitud del bloque se fija en 128 bytes, de los cuales los primeros 32 bytes se denominan la cabecera del bloque y en ella se define las características de la comunicación; los 96 bytes restantes se completan con caracteres "0".



Formato de cabecera del bloque comando

La estructura del bloque comando es el siguiente :

	1	2	2	4	6	4	1	1	3	3	3	2
Modelo Caja	00	00	0000	000000	0000	Tipo Enlace	0	000	Código Comando	000	00	

Modelo de Caja.

1 : CE-4700

2 : TK-2300

Tipo Enlace.

5 : Lectura/Borrado de Ventas

7 : Recepción de archivo de datos Caja-PC

8 : Transmisión de archivo de datos PC-Caja

Código Comando.

El código comando varía de acuerdo al tipo de enlace:

a) Lectura/Borrado de Ventas.

Se genera un código de tres dígitos de la forma siguiente :

D3	D2	D1
----	----	----

D3 :

0 : Solo lectura de archivo de ventas

4 : Borrado de archivo de ventas

D2,D1 :

11 : Venta por Totales Fijos

12 : Venta por Funciones Libres

14 : Venta por Artículos

15 : Venta por Departamentos

17 : Venta por Cajero

19 : Venta por Rango Horario

b) Recepción de archivo de datos Caja-PC

Transmisión de archivo de datos PC-Caja

En ambos casos se genera un código de tres dígitos que identifica el número de archivo a enviar o recibir :

D3	D2	D1
----	----	----

D3,D2,D1 :

001 : Totalizadores Fijos

002 : Funciones Libres

003 : Artículos

005 : Departamentos

006 : Grupos

007 : Cajeros

022 : Programa General

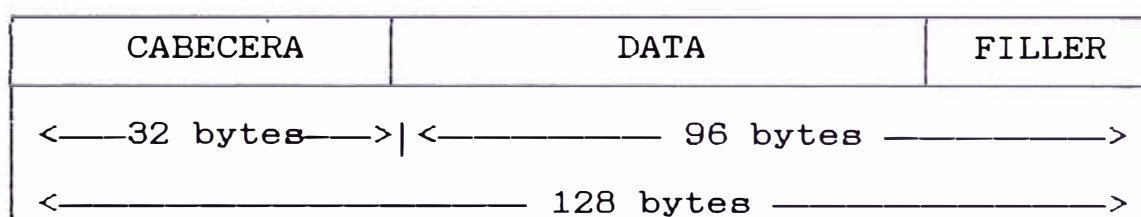
025 : Impuestos

032 : Logotipo

090 : Todo el programa

3.4.2 Formato del bloque de datos

La longitud del bloque de datos está fijado a 128 bytes de los cuales 32 bytes son reservados para la cabecera del bloque de datos y el resto para un registro de datos de longitud variable. Cuando los datos a enviar tienen una longitud menor a los 128 bytes el resto se completa con bytes "0".



Formato de cabecera del bloque de datos

La siguiente cabecera es generada automáticamente cuando la caja registradora envía datos. Por otro lado, el computador debe generar la cabecera mediante un software aplicativo en el mismo formato cada vez que desee enviar datos hacia la caja registradora.

1	2	2	4	6	4	1	1	3	3	3	2
Modelo Caja	Ceros	Numero Nodo	Numero Caja	Fecha	Hora	Codigo Dato	Multi- bloque	Contador Bloque	Numero Comando	Long. Regist	Numero Regist

El siguiente cuadro muestra al detalle la descripción de cada uno de los campos:

Bytes	Nombre del campo	
1	Modelo de CRE	1:CE-4700, 2:TK-2300
2	Ceros	00
2	Numero de Nodo	
4	Número de CRE	
6	Fecha	(Año, Mes, Día)
4	Hora	
1	Tipo Operación	4:Lectura de datos 5:Borrado de datos 7:Programación
1	Multibloque	0:No usa multibloque 1,4,8:Multibloque
3	Contador de bloque	000-999
3	Código de comando	
3	Longitud de registro	
3	Número de registros	

Ejemplo de algunos códigos comando para el proceso de envío y recepción de datos:

Código Comando	Archivo
020	Totalizadores fijos
022	Funciones libres
030	Departamentos
041	Plus
053	Tabla de Impuestos
074	Cajeros

Formato del cuerpo del bloque de datos

El formato del cuerpo de la trama de datos varía de acuerdo al tipo de archivo a enviar o recibir, para ello existe una tabla de estructuras completamente definidas.

En primer lugar tomaremos como ejemplo la estructura del archivo recibido de artículos vendidos y a continuación se describirá la estructura del archivo de artículos a ser enviado a la caja registradora.

a) Archivo de acumulados de venta por artículos.- La caja registradora posee internamente un archivo de acumulado de ventas por artículo en la cual se van almacenando los totales de venta de todas las operaciones realizadas, por ejemplo, si durante el día se efectuaron 20 operaciones de venta en la cual la cantidad total de Coca-Cola vendidas fue de 30 unidades y el importe total fue de 45.00 (asumiendo que una Coca-Cola vale 1.50), entonces se tendrá un archivo de acumulados de la forma siguiente:

Número de Memoria	Unidades	Importe
0324	30.000	45.00

Donde el número de memoria viene a ser de cuatro dígitos y representa la ubicación del producto en el archivo de artículos. La estructura del formato de datos del archivo de ventas por artículo es la siguiente:

← 128 bytes →									
← 32 →			4	10	10	10	10	10	42
CABECERA	054	01	Número Memoria	Unidades	Importe	Descuento	Total Bruto	Total Neto	Filler

Donde se tiene:

054 -- Longitud de registro

01 -- Número de registros en el bloque

b) Archivo de programación de artículos.- La caja registradora posee internamente un archivo de artículos donde se almacenan los productos que pueden ser vendidos a través de este equipo. Dentro de este archivo se almacenan características del artículo tales como: Número de memoria, código del artículo, descripción, precio, impuesto, etc.

Cuando se hace necesario crear un nuevo artículo en caja o variar alguna característica del mismo se hace necesario enviar una trama con las características completas del artículo. La estructura de esta trama es la que se define a continuación:

← 32 →			4	24	12	2	2	6	6	
CABECERA	092	01	Número Memoria	Nombre Artículo	Atributo	Enlace Depto	Enlace Grupo	Empaque	Precio Unit	
8	6	2	6	2	2	4	5	1	4	
Enlace Botella	Código Artículo	Dígito Chequeo	Máxima Venta	No Usado	Tipo Recibo	0000	Stock Mínimo	0	Filler	

Donde se tiene:

092 -- Longitud de registro

01 -- Número de registros en el bloque

CAPITULO IV IMPLEMENTACION DEL SOFTWARE

Al tener completa la información referente a la interfase hardware y al protocolo de comunicaciones utilizado en el enlace caja registradora-computador, la implementación del software puede ser desarrollada haciendo uso de los diversos lenguajes de programación existentes en el mercado (Lenguaje C, Turbo Pascal, etc.).

En el presente informe se muestra una aplicación desarrollada en el lenguaje de programación BASIC (versión GW-BASIC) el cual se encuentra disponible en la mayoría de sistemas operativos existentes en el mercado y su sintaxis es bastante sencilla de entender. Se hará una descripción de las partes principales del programa tales como: parámetros, bloques de comando, transmisión y recepción de datos.

Para describir los diversos procesos que involucran la transmisión y recepción de datos se toman como base extractos del programa principal, sin embargo, para efectuar una explicación mas clara en algunos casos se hace uso de líneas de programación resumidas.

4.1 Menú principal

El Menú Principal es el siguiente:

```

** COMUNICACION CAJA REGISTRADORA-PC **
1. RECEPCION DE DATOS (CRE a PC)
2. TRANSMISION DE DATOS (PC a CRE)
3. DEFINIR BLOQUE COMANDO
4. SALIR

```

Recepción de Datos (CRE a PC).— Esta opción permite la recepción de datos en el computador de los datos enviados por la caja registradora electrónica (CRE).

Transmisión de Datos (PC a CRE).— Esta opción permite el envío de datos desde el computador hacia la caja registradora.

Definir Bloque Comando.— Esta opción permite definir los archivos de control para establecer el envío o recepción de datos.

Salir.— Finaliza el uso del software.

4.2 **Definición de parámetros generales**

En el software a ser diseñado se tiene como punto de partida la definición de parámetros tales como códigos de control, puerto de comunicación a utilizar, velocidad de transmisión, etc.

Códigos de control

Los códigos de control descritos en el párrafo 3.2 del capítulo anterior son asignados a las siguientes variables para ser fácilmente entendibles:

```

230  ^*** CODIGOS DE CONTROL ***
250  ZSTX$ = CHR$(&H2)
260  ZETB$ = CHR$(&H26)
270  ZETX$ = CHR$(&H3)
280  ZACKO$ = CHR$(&H70)
290  ZACK1$ = CHR$(&H61)
300  ZNACK$ = CHR$(&H3D)
310  ZENQ$ = CHR$(&H2D)
320  ZDLE$ = CHR$(&H10)
330  ZEOT$ = CHR$(&H37)
340  ZAKO$ = ZDLE$ + ZACKO$
350  ZAK1$ = ZDLE$ + ZACK1$
360  ZDLEEOT$ = ZDLE$ + ZEOT$
370  ZSTXETX$ = ZSTX$ + ZETX$ + ZETX$
380  ZIDENQ$ = STRING$(8, 240) + ZENQ$

```

Configuración del puerto serial

El puerto a utilizar es el puerto serial 1 (COM1) y la velocidad definida es de 9600 bps. Las siguientes líneas definen estas características:

```

630  SPEED$ = "9600"
1030 OPEN "COM1:"+SPEED$+",N,8,1,RS,CSO,DS" FOR
      RANDOM AS #1

```

Procedimiento para cálculo del dígito de verificación CRC y conversión de binario a ASCII

En los procesos de envío y recepción de datos existen tiempos límites de espera que no deben ser sobrepasados porque el enlace sufriría un corte abrupto y todo el proceso debe iniciarse nuevamente.

Durante los procesos de envío y recepción de datos se hace necesario ejecutar ciertos cálculos en forma repetitiva, estos procesos pueden programarse en el mismo lenguaje de programación que el resto del programa fuente, sin embargo, el problema presentado es que ocasionan un retardo considerable de tiempo; debido a ello se considera conveniente hacer uso del lenguaje de programación mas cercano al hardware: el lenguaje assembler.

En el modelo presentado se hace uso de un procedimiento en lenguaje assembler que es invocado desde el programa escrito en lenguaje BASIC que permite el cálculo del dígito de comprobación de redundancia cíclica CRC y además convierte una trama de datos en formato binario -que es utilizado en los archivos almacenados en la caja registradora- al formato ASCII convencional.

A continuación se muestra el código fuente escrito en lenguaje BASIC y su correspondiente código fuente escrito en lenguaje assembler.

Código fuente en lenguaje BASIC

```

500 ~***** LENGUAJE ASSEMBLER PARA CALCULO DEL CRC *****
505 ~***** Y CONVERSION BINARIO A ASCII *****
510 ZL1 = 0: ZL2 = 0: ZI1$ = ""
520 DATA 55,8B,EC,50,53,51,52,56,57,06,1E,8B,76,0A,8B,74
530 DATA 01,8B,7E,0C,8B,7D,01,33,C0,B9,81,00,8A,04,32,E0
540 DATA E8,2A,00,88,05,46,47,E2,F3,B0,30,3A,24,74,02,B0
550 DATA 31,8B,76,06,8B,74,01,88,04,8B,76,08,8B,74,01,88
560 DATA 24,1F,07,5F,5E,5A,59,5B,58,5D,CA,08,00,55,8B,EC
570 DATA 53,56,33,DE,3C,60,75,02,B0,FF,3C,F0,72,02,2C,10
580 DATA 2C,DO,24,1F,8A,D8,8B,76,02,83,C6,4B,2E,8A,00,5E
590 DATA 5B,5D,C3,7D,4A,4B,4C,4D,4E,4F,50,51,52,00,30,31
600 DATA 32,33,34,35,36,37,38,39,00,00,00,00,2D,00,00:
610 RESTORE 520: DEF SEG = VARSEG(ZZBCC(1)):
FOR ZL1 = 0 TO &H8F: READ ZI1$:
POKE (VARPTR(ZZBCC(1)) + ZL1), VAL("&H" + ZI1$):
NEXT ZL1

```

Código fuente en lenguaje assembler

```

INTERFAZ    SEGMENT 'CODE'
PUBLIC      ZBCC

ZBCC        PUSH    BP
            MOV BP,SP
            PUSH   AX
            PUSH   BX
            PUSH   CX
            PUSH   DX
            PUSH   SI
            PUSH   DI
            PUSH   ES
            PUSH   DS
            MOV SI,[BP+0A]
            MOV SI,[SI+01]
            MOV DI,[BP+0C]
            MOV DI,[DI+01]
            XOR AX,AX
            MOV CX,0081
L011C:     MOV AL,[SI]
            XOR AH,AL
            CALL  R014D
            MOV [DI],AL
            INCSI
            INC DI
            LOOP  L011C
            MOV AL,30
            CMP AH,[SI]
            JZ   L0131
            MOV AL,31
L0131:     MOV SI,[BP+06]
            MOV SI,[SI+01]
            MOV [SI],AL
            MOV SI,[BP+08]
            MOV SI,[SI+01]
            MOV [SI],AH
            POP DS
            POP ES
            POP DI
            POP SI
            POP DX
            POP CX

```

```

        POP BX
        POP AX
        POP BP
        RETF    0008

R014D   PUSH    BP
        MOV BP,SP
        PUSH    BX
        PUSH    SI
        XOR BX,BX
        CMP AL,60
        JNZ L015A
L015A:  MOV AL,FF
        CMP AL,F0
        JB  L0160
L0160:  SUB AL,10
        SUB AL,DO
        AND AL,1F
        MOV BL,AL
        MOV SI,[BP+02]
        ADD SI,+4B
        CS:
        MOV AL,[BX+SI]
        POP SI
        POP BX
        POP BP
        RET

ZBCC    ENDP
INTERFAZ ENDS

        END

```

4.3 Definición de bloques comando

Un bloque de comando es un conjunto de caracteres que indican a la caja registradora que acciones ha de realizar y que archivos ha de procesar.

Antes de analizar en detalle un bloque de comando, debemos mencionar que existen tres tipos de procesos de comunicación con la caja registradora:

a) Lectura de Reportes X/Z

Durante el proceso de venta en la caja registradora se va acumulando en registros de datos los totales de venta por rubros, estos registros incluyen variables como:

- Detalle de Artículos vendidos
- Totales por Medios de Pago.
- Totales por Cajero.

- Totales por Rango Horario.
- Totales por Departamento, etc.

Esta información es de suma importancia para el análisis que pueda establecer el propietario del establecimiento y a través de esta opción pueden ser leídos por el computador y almacenados en forma de archivos tipo texto para ser procesados por algún software aplicativo.

b) Envío de Programación

La caja registradora guarda en memoria RAM un conjunto de archivos para los procesos de venta. Estos archivos incluyen:

- Archivo de Datos de Artículos.
- Archivo de Datos de Departamentos.
- Archivo de Datos de Cajeros.
- Archivo de Datos de Programación de Teclado, etc.

Cuando se requiere crear o modificar en caja un artículo, un medio de pago, un cajero, etc., es necesario enviar una actualización del archivo respectivo, para ello, en primer lugar se crea un archivo con las modificaciones respectivas y luego efectúa el envío.

c) Lectura de Programación

Los datos mencionados en el punto (b) son posibles de ser leídos por el computador, esto se hace con la finalidad de analizar que datos se encuentran activos en la caja registradora.

Estructura de un bloque comando

La estructura de un bloque comando es el siguiente:

estudio del proceso de transmisión este será dividido en cuatro partes:

- Establecimiento del enlace.
- Envío del archivo comando.
- Envío de datos.
- Fin de enlace.

Transmisión - Establecimiento del enlace

El establecimiento del enlace consiste en revisar y fijar condiciones físicas y lógicas de los estados presentes en los puertos de entrada/salida, solamente verificadas ciertas condiciones puede continuarse con el procesos de transmisión de datos.

Para establecer el enlace se hace uso del siguiente bloque de programa:

```

5000 `*** TX - ESTABLECIMIENTO DEL ENLACE ***
5005 `*** Se activa RTS ON, se verifica CTS ON ***
5010 ZM1% = INP(&H3FC) `RTS OFF ---> RTS ON
5020 ZM2% = ZM1% OR &H2
5030 OUT &H3FC, ZM2%
5040 ZM1% = INP(&H3FE)
5050 IF (ZM1% AND &H10) = 0 THEN 5040 ELSE 5060
5055 `*** Se envía caracter de control ID-ENQ ***
5060 ZBF$ = ZIDENQ$
5070 PRINT #1, ZBF$
5075 `*** Se regresa RTS OFF ***
5080 ZM1% = INP(&H3FC) `RTS ON --->RTS OFF
5090 ZM2% = ZM1% AND &HFD
5100 OUT &H3FC, ZM2%
```

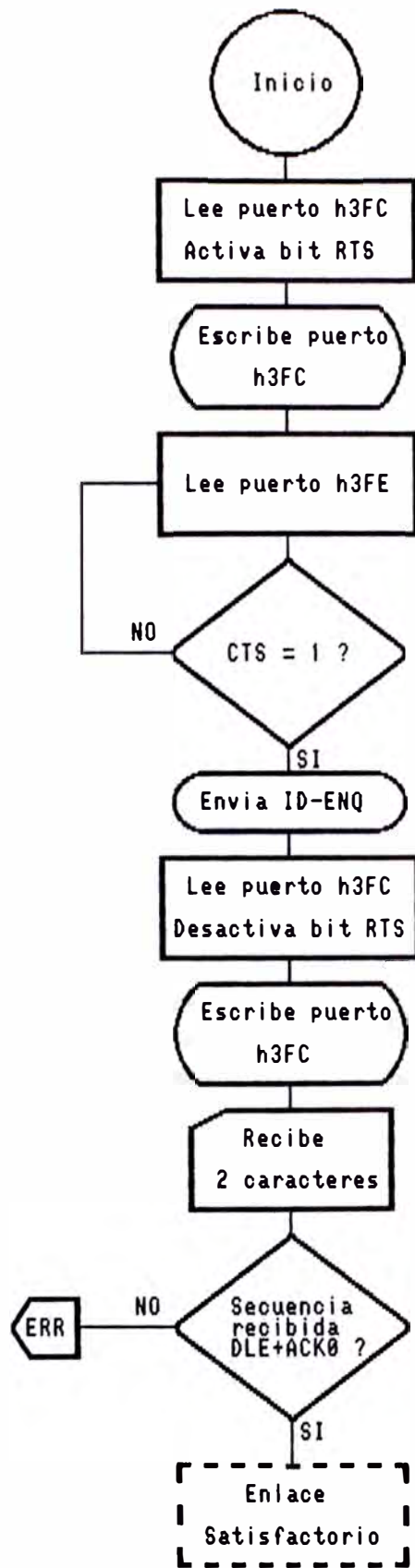


Figura 4.1. Establecimiento de enlace para transmisión

```
5105 `** Enlace OK si se recibe secuencia DLE+ACKO
```

```
5110 Z$ = INPUT$(1, #1) `lee un caracter
```

```
5210 IF Z$ <> ZDLE$ THEN "no sigue"
```

```
5230 Z$ = INPUT$(1, #1)
```

```
5240 IF Z$ = ZACKO$ THEN "sigue" ELSE "no sigue"
```

En primer lugar se lee el valor almacenado en el puerto **H3FC** denominado Modem Control Register (MCR) y se activa el bit Request To Send (RTS), luego se envía el nuevo valor al mismo puerto H3FC.

Inmediatamente se lee el puerto **H3FE** denominado Modem Status Register (MSR) y se comprueba el valor del bit Clear To Send (CTS).

Si el valor del bit Clear To Send es uno el proceso de establecimiento de enlace continúa.

Se envía el código de control ID-ENQ.

Se reciben dos caracteres, si la secuencia recibida es **DLE+ACKO** entonces el establecimiento del enlace es satisfactorio.

Transmisión - Envío del archivo comando

El siguiente paso en el procedimiento de transmisión es enviar hacia la caja registradora una secuencia de 128 caracteres denominado archivo comando. Este archivo define el tipo de archivo de datos que será transmitido y el modelo de caja registradora destino, las siguientes líneas de programa ejecutan tal proceso:

```
5300 `*** TX - ENVIO DE ARCHIVO COMANDO ***
```

```
5305 `*** Lectura de archivo comando ***
```

```
5310 PRINT "INGRESE NOMBRE DEL ARCHIVO COMANDO !"
```

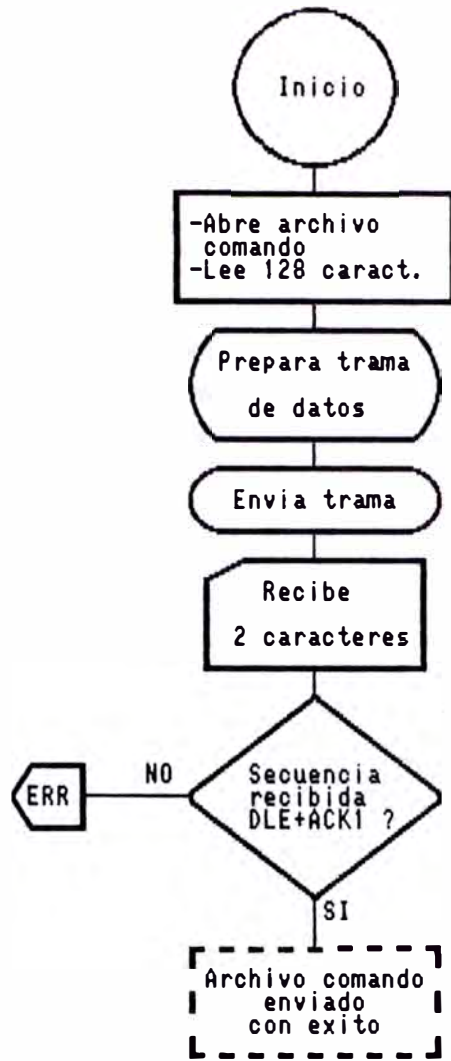


Figura 4.2. Envío de archivo comando para transmisión.


```

5320 LOCATE 7, 20: INPUT "* ", CNAM$
5330 OPEN CNAM$ FOR RANDOM AS #3
5340   FIELD #3, 128 AS ZA$
5350   GET #3
5355   *** Se prepara DATA a enviar ***
5360 ZEND$ = ZETB$
5370 Z$ = ZA$ + ZEND$ + "0":
      DEF SEG =VARSEG(ZZBCC(1)):
      CALL absolute(VARPTR(ZBCC(ZASC$,Z$,ZBCC$,ZR$)))
5380 Z$ = ZSTX$ + ZA$ + ZEND$ + ZBCC$
5385   *** Se transmite DATA ***
5390 ZBF$ = Z$
5400 PRINT #1, ZBF$
5410   ** Transmisión correcta si se recibe
      'secuencia DLE+ACK1 **
5420 Z$ = INPUT$(1, #1) 'lee un caracter
5430 IF Z$ <> ZDLE$ THEN "no sigue"
5440 Z$ = INPUT$(1, #1)
5450 IF Z$ = ZACK1$ THEN "sigue" ELSE "no sigue"

```

En primer lugar se pide al usuario ingrese el nombre del archivo comando.

Luego, se lee el contenido y se procede a armar la trama de datos a enviar, para ello se hace uso del procedimiento **CALL absolute** para generar el dígito de control de errores conforme al método de comprobación de redundancia cíclica CRC.

Se envía la trama generada y se procede a leer dos caracteres del puerto serial.

Si la secuencia recibida es DLE+ACK1, el envío del archivo comando finaliza correctamente.

Transmisión - Envío de datos

Una vez ejecutados los procesos de establecimiento de enlace y envío de archivo comando el computador se encuentra listo para enviar y la caja registradora se encuentra lista para recibir.

El envío de datos consiste en la lectura de un archivo que contiene información para la caja registradora y luego se transmiten bloques de 128 caracteres de longitud, las siguientes líneas de programa ejecutan tal proceso:

```

2191 `*** TX - ENVIO DE ARCHIVO DE DATOS ***
2192 `** Petición de nombre de archivo de datos **
2192 PRINT "INGRESE NOMBRE DE ARCHIVO DE DATOS A
          ENVIAR:
2193 INPUT "* ", ZNAME$
2194   OPEN ZNAME$ FOR RANDOM AS #2
2195   FIELD #2, 128 AS ZC$
2196 ZODEV = 0: ZSTEP = 0: ZBLC = 0
2197 `** Lazo de Transmisión de datos **
2198 `** ZSTEP = 0 Enviando datos
          ZSTEP = 1 Ultimo dato
          ZSTEP - 2 Desconexión          **
2200 ON ZSTEP GOTO 2420, 2500
2230 GET #2: ZA$ = ZC$ `** Lee archivo de datos **
2240 ZBLC=ZBLC+1: PRINT "Enviando Bloque: "; ZBLC;
2270 ZEND$ = ZETB$

```

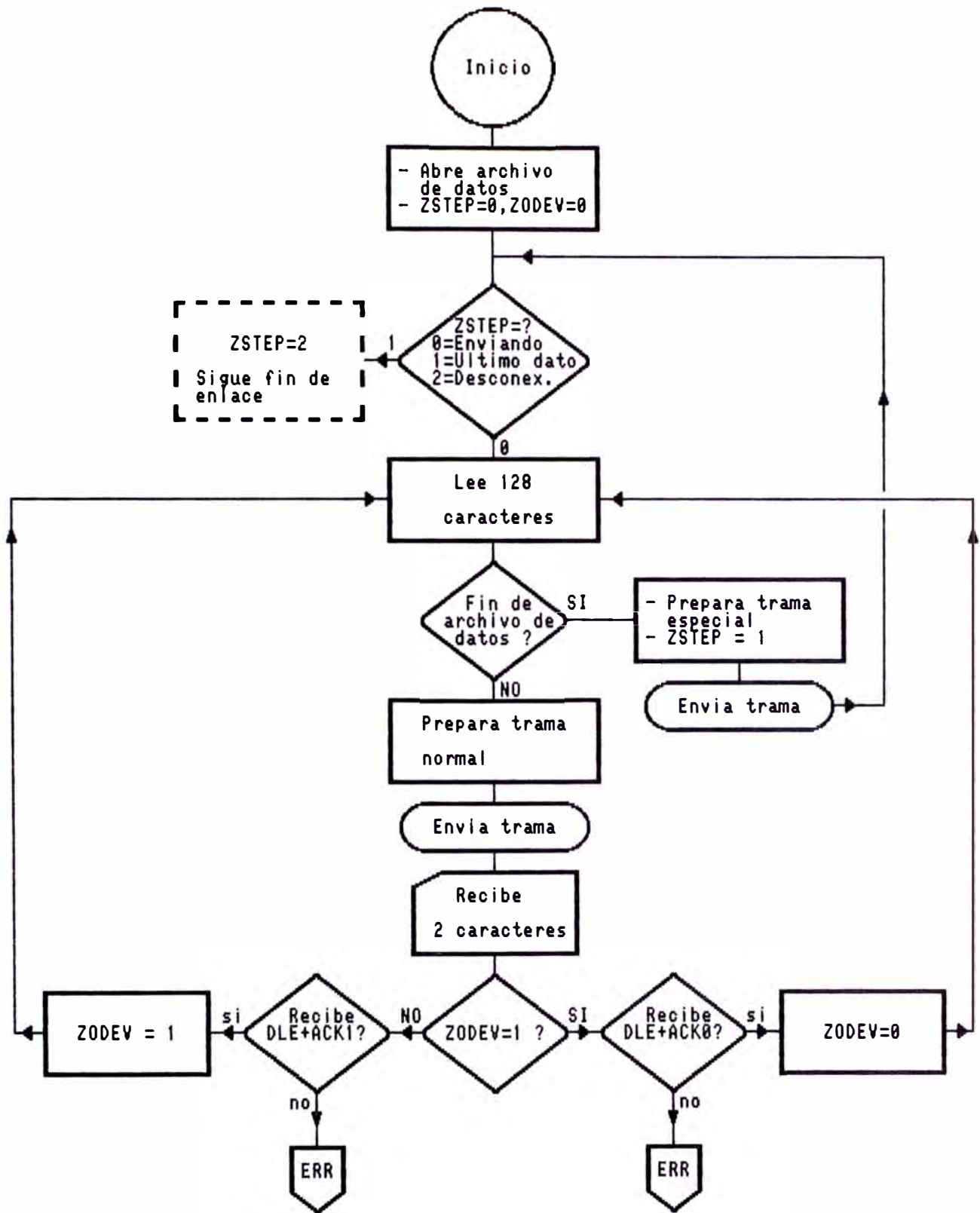


Figura 4.3. Diagrama de flujo para el proceso de envío de datos.

```

2280 Z$ = ZA$ + ZEND$ + "0":
      DEF SEG = VARSEG(ZZBCC(1)):
      CALL absolute(VARPTR(ZBCC(ZASC$,Z$,ZBCC$,ZR$)))
2290 Z$ = ZSTX$ + ZA$ + ZEND$ + ZBCC$
2300 IF EOF(2) THEN ZSTEP = 1: GOTO 2430
2310   ZX$ = Z$
2320   ZBF$ = ZX$: PRINT #1, ZBF$
2330   Z$ - INPUT$ (1, #1)
2340 IF Z$ <> ZDLE$ THEN "no sigue"
2350 Z$ = INPUT$(1, #1)
2360 IF ZODEV = 1 THEN 2410
2370 IF Z$ = ZACK1$ THEN ZODEV = 1: GOTO 2200
2410 IF Z$ - ZACKO$ THEN ZODEV = 0: GOTO 2200
2430 Z$ = ZSTXETX$: ZSTEP = 2: GOTO 2310
2500 "continua secuencia final"

```

El proceso consiste en leer el archivo de datos e ir transmitiendo bloques de 128 caracteres.

Al enviar la primera línea la caja registradora debe responder con la secuencia **DLE+ACKO**.

Al enviar la segunda línea la caja registradora debe responder con la secuencia **DLE+ACK1**.

Esta alternancia de respuestas (ACKO/ACK1) por parte de la caja registradora se repiten para las demás líneas enviadas.

Cuando se envía la última línea de datos se activa una variable ZSTEP=1 para enviar un bloque especial de datos, luego de ello se activa la variable ZSTEP=2 para indicar que el enlace va a finalizar.

Transmisión - Fin de enlace

El proceso de fin de enlace tiene la misión de indicar a la caja registradora que no existen mas datos a enviar por lo tanto se finalizará la comunicación de datos.

Consiste en enviar el caracter de control ZDLEEOT\$:

2150 ZBF\$ = ZDLEEOT\$

2152 `*** Se transmite caracter fin de enlace ***

2156 PRINT #1, ZBF\$

4.5 Procedimiento de recepción

Para llevar a cabo la recepción de los datos enviados por la caja registradora se siguen una serie de pasos, por ello, con la finalidad de efectuar un mejor estudio del proceso de recepción este será dividido en cinco partes:

Establecimiento del enlace.

- Envío de archivo comando.

Solicitud de envío por parte de caja registradora.

Recepción de datos.

Fin de enlace.

Recepción - Establecimiento del enlace

El establecimiento del enlace consiste en revisar y fijar condiciones físicas y lógicas de los estados presentes en los puertos de entrada/salida, solamente verificadas ciertas condiciones puede continuarse con el proceso de recepción de datos.

Para establecer el enlace se hace uso del siguiente bloque de programa:

```

6000 `*** RX - ESTABLECIMIENTO DEL ENLACE ***
6005 `*** Se activa RTS ON, se verifica CTS ON ***
6010 ZM1% = INP(&H3FC) `RTS OFF ---> RTS ON
6020 ZM2% = ZM1% OR &H2
6030 OUT &H3FC, ZM2%
6040 ZM1% = INP(&H3FE)
6050 IF (ZM1% AND &H10) = 0 THEN 6040 ELSE 6060
6055 `*** Se envía caracter de control ID-ENQ ***
6060 ZBF$ = ZIDENQ$
6070 PRINT #1, ZBF$
6075 `*** Se regresa RTS OFF ***
6080 ZM1% = INP(&H3FC) `RTS ON --->RTS OFF
6090 ZM2% = ZM1% AND &HFD
6100 OUT &H3FC, ZM2%
6105 `** Enlace OK si se recibe secuencia DLE+ACKO
6110 Z$ = INPUT$(1, #1) `lee un caracter
6210 IF Z$ <> ZDLE$ THEN "no sigue"
6230 Z$ = INPUT$(1, #1)
6240 IF Z$ = ZACKO$ THEN "sigue" ELSE "no sigue"

```

En primer lugar se lee el valor almacenado en el puerto **H3FC** denominado Modem Control Register (MCR) y se activa el bit Request To Send (RTS), luego se envía el nuevo valor al mismo puerto H3FC.

Inmediatamente se lee el puerto **H3FE** denominado Modem Status Register (MSR) y se comprueba el valor del bit Clear To Send (CTS).

Si el valor del bit Clear To Send es uno el proceso de establecimiento de enlace continúa.

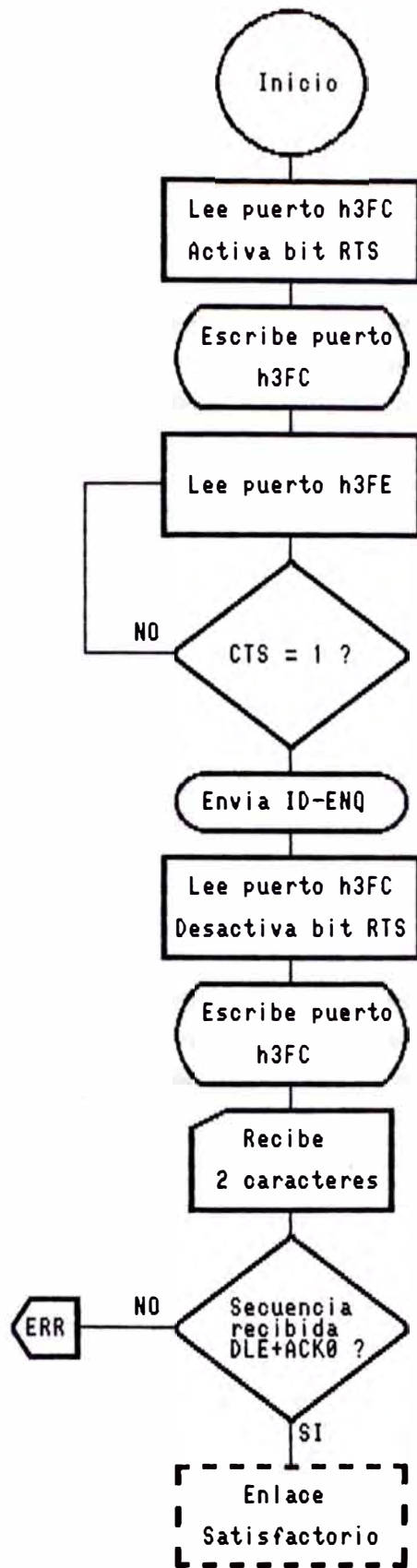


Figura 4.4. Establecimiento de enlace para recepción.

Se envía el código de control ID-ENQ.

Se reciben dos caracteres, si la secuencia recibida es **DLE+ACKO** entonces el establecimiento del enlace es satisfactorio.

Recepción - Envío de archivo comando

El siguiente paso en el procedimiento de recepción es enviar hacia la caja registradora una secuencia de 128 caracteres denominado archivo comando. Este archivo define si se recibirá un archivo de datos ó un archivo de programación, también define el modelo de caja registradora con el cual se establecerá comunicación, las siguientes líneas de programa ejecutan tal proceso:

```

6300 `*** RX - ENVIO DE ARCHIVO COMANDO ***
6305 `*** Lectura de archivo comando ***
6310 PRINT "INGRESE NOMBRE DEL ARCHIVO COMANDO !"
6320 LOCATE 7, 20: INPUT "* ", CNAM$
6330 OPEN CNAM$ FOR RANDOM AS #2
6340   FIELD #2, 128 AS ZC$
6350 GET #2: ZA$ = ZC$
6355 `*** Se prepara DATA a enviar ***
6360 ZEND$ = ZETX$
6370 Z$ = ZA$ + ZEND$ + "0":
      DEF SEG =VARSEG(ZZBCC(1)):
      CALL absolute(VARPTR(ZBCC(ZASC$,Z$,ZBCC$,ZR$)))
6380 Z$ = ZSTX$ + ZA$ + ZEND$ + ZBCC$
6385 `*** Se transmite DATA ***
6390 ZBF$ = Z$
6400 PRINT #1, ZBF$

```

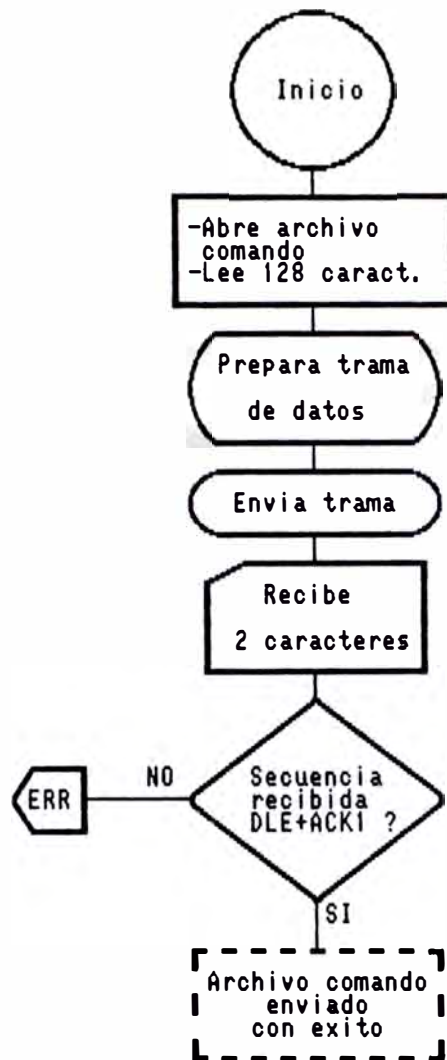



Figura 4.5. Envío de archivo comando para recepción.

```

6410 `** Transmisión correcta si recibe DLE+ACK1 *
6420 Z$ = INPUT$(1, #1) `lee un caracter
6430 IF Z$ <> ZDLE$ THEN "no sigue"
6440 Z$ = INPUT$(1, #1)
6450 IF Z$ = ZACK1$ THEN "sigue" ELSE "no sigue"

```

En primer lugar se pide al usuario ingrese el nombre del archivo comando.

Luego, se lee el contenido y se procede a armar la trama de datos a enviar, para ello se hace uso del procedimiento **CALL absolute** para generar el dígito de control de errores conforme al método CRC.

Se envía la trama generada y se procede a leer dos caracteres del puerto serial, si la secuencia recibida es **DLE+ACK1**, el envío del archivo comando finaliza correctamente.

Recepción Solicitud de envío por parte de caja registradora

Las líneas de programa que indican solicitud de envío por parte de la caja registradora son las siguientes:

```

6500 ZBF$ = ZEOT$
6510 `*** Se transmite EOT ***
6520 PRINT #1, ZBF$
6530 `** La caja debe enviar caracter ENQ **
6540 Z$ = INPUT$(1, #1) `lee un caracter
6550 IF Z$ = ZENQ$ THEN 6560 ELSE "no sigue"
6555 `*** Se transmite ACK0 ***
6560 ZBF$ = ZAKO$:
6570 PRINT #1, ZBF$

```

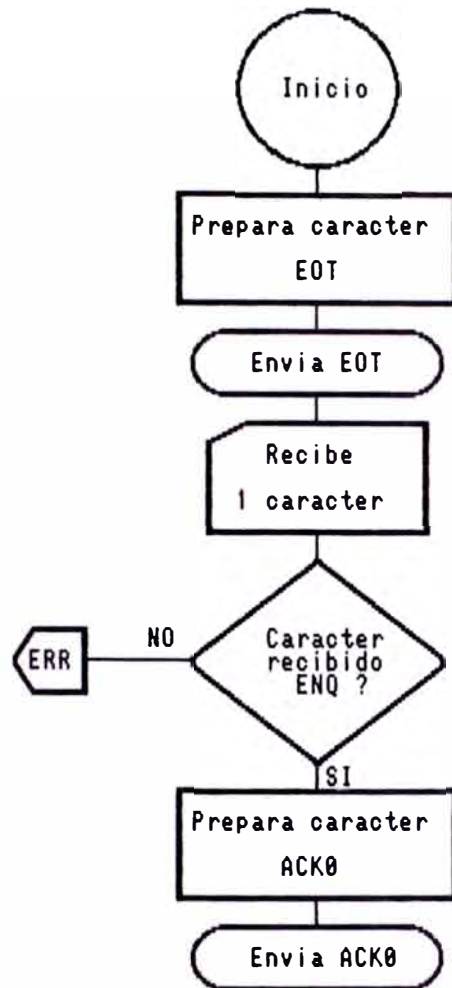


Figura 4.6. Solicitud de envío por parte de la caja registradora.

En primer lugar el computador envía el caracter EOT, tras lo cual la caja registradora debe enviar el caracter ENQ.

Luego de ello el computador debe enviar ACK0 en señal de conformidad.

El envío de ACK0 se considera como parte del proceso de solicitud de envío, sin embargo, será repetido en el proceso de recepción de datos solo la primera vez, esto con la finalidad de facilitar la explicación del programa.

Recepción - Recepción de datos

Una vez ejecutados los procesos de establecimiento de enlace, envío de archivo comando y solicitud de envío por parte de la caja registradora, el computador se encuentra listo para recibir y la caja registradora se encuentra lista para enviar datos.

La recepción de datos consiste en leer a través del puerto serial bloques de 130 caracteres y luego almacenarlos en un archivo en disco, las siguientes líneas de programa ejecutan tal proceso:

```

1080 ZIN$ = "$$TMP$$$": OPEN ZIN$ FOR RANDOM AS #2
1090 FIELD #2, 128 AS ZA$
1100 GOSUB 1610
1600 `*** SUBROUTINA DE RECEPCION ***
1610 ZRS$="E": ZBLC=1: ZODEV=0: ZSTEP=0: ZZH$=""
1670 `***** Envía ACK0 *****
1680 ZBF$ = ZAKO$: PRINT #1, ZBF$: ZX$ = ZAKO$
1690 Z$ = INPUT$(1, #1) `lee un caracter

```

```

1700 IF Z$ = ZSTX$ THEN 1770
1730 IF Z$ = ZEOT$ THEN 1750 `FIN
1740 `***** Secuencia final *****
1750     ZBF$ = ZDLEEOT$: PRINT #1, ZBF$
1760 IF ZSTEP = 1 THEN "finaliza bien"
           ELSE "finaliza con error"
1770 Z$ = INPUT$(1, #1): ZMODE$ = Z$
1800 Z$ = INPUT$(129, #1): Z$ = ZMODE$ + Z$
1810 DEF SEG = VARSEG(ZZBCC(1)):
           CALL absolute(VARPTR(ZZBCC(ZASC$, Z$, ZBCC$, ZR$))):
           DEF SEG : ZH$ = MID$(Z$, 22, 3)
1820 IF ZR$ = "0" THEN 1850 ELSE 1840
1830 `***** Envía NACK *****
1840 ZBF$ = ZNACK$: PRINT #1, ZBF$: ZX$ = ZNACK$:
           GOTO 1690
1850 IF ZZH$ <> ZH$ THEN 1870 ELSE 1930
           `BLOQUE DUPLICADO
1860 `***** Escribe bloque recibido en disco *****
1880 IF ZODEV = 1 THEN LSET ZA$ = ZASC$:
           ZODEV = 0: GOTO 1910
1900 IF ZODEV = 0 THEN LSET ZA$ = ZASC$:
           ZODEV = 1: GOTO 1910
1910 PUT #2, ZBLC: LOCATE 1, 1: PRINT "Bloque:";
           ZBLC; : PRINT ZASC$`***** PRUEBA
1920 ZBLC = ZBLC + 1
1930 IF ZODEV = 0 THEN 1680

```

```

1940 ^***** Envía ACK1 *****
1950 ZBF$ = ZAK1$: GOSUB 2510: ZX$ = ZAK1$:
      GOTO 1690
1960 Z$ = INPUT$(1, #1)
1970 IF Z$ = ZEOT$ THEN 1760 ELSE 1690
1980 Z$ = INPUT$(1, #1)
1990 IF Z$ = ZETX$ THEN ZSTEP = 1: GOTO 2000
      ELSE 1950
2000 IF ZODEV = 0 THEN ZODEV = 1: GOTO 1930
      ELSE ZODEV = 0: GOTO 1930

```

Luego de recibir de parte de la caja registradora el caracter ENQ, el computador envía ACK0.

Se inicia entonces el proceso de recepción de datos en el computador.

Primeramente se lee un caracter y se verifica que sea el caracter de control STX.

Luego procedemos a leer el siguiente caracter y efectuamos la pregunta si el caracter leído es ETX ú otro.

De ser ETX indica que la caja registradora a finalizado su envío.

Caso contrario se trata de un bloque de datos y se procede a leer 129 caracteres adicionales.

Luego se envía como conformidad el caracter de control ACK1/ACK0 en forma alternada.

Recepción - Fin de enlace

El proceso de fin de enlace tiene la misión de indicar que finalizará la comunicación de datos.

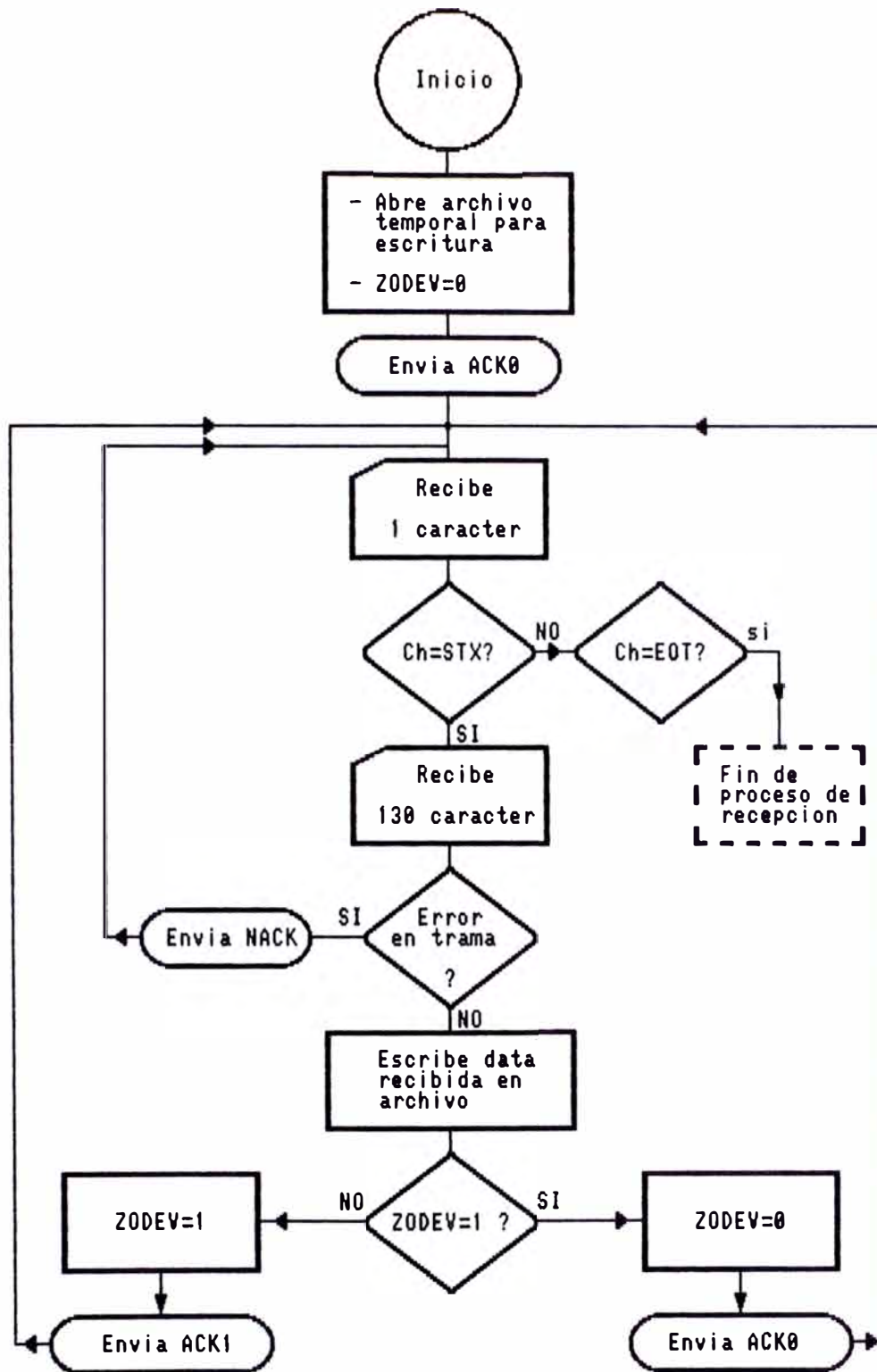


Figura 4.7. Diagrama de flujo para el proceso de recepción de datos.

Para ello en primer lugar la caja registradora envía el caracter de control EOT\$ tras lo cual el computador debe responder enviando el caracter ZDLEEOT\$:

```

8000 Z$ = INPUT$(1, #1):
8010 IF Z$ = ZEOT$ THEN 8020
      ELSE "continua recibiendo bloque de datos"
8020 `*** Se transmite caracter fin de enlace ***
8030 ZBF$ = ZDLEOT$
8040 PRINT #1, ZBF$

```

4.6 Listado del programa

El siguiente es el listado completo del programa e incluye la totalidad de procedimientos descritos:

```

100 `*****
110 `* SOFTWARE DE COMUNICACIONES CAJA REGISTRADORA-PC *
120 `*           DISEÑO POR LUIS TAQUIRI ROSALES      *
130 `*   UNIVERSIDAD NACIONAL DE INGENIERIA - FIEE   *
200 `*****
210
220 CLEAR : DEFINT Z: DIM ZZBCC(&H9F)
225
230 `***** CODIGOS DE CONTROL *****
250 ZSTX$ = CHR$(&H2)
260 ZETB$ = CHR$(&H26)
270 ZETX$ = CHR$(&H3)
280 ZACK0$ = CHR$(&H70)
290 ZACK1$ = CHR$(&H61)
300 ZNACK$ = CHR$(&H3D)
310 ZENQ$ = CHR$(&H2D)
320 ZDLE$ = CHR$(&H10)
330 ZEOT$ = CHR$(&H37)
340 ZAK0$ = ZDLE$ + ZACK0$
350 ZAK1$ = ZDLE$ + ZACK1$
360 ZDLEEOT$ = ZDLE$ + ZEOT$
370 ZSTXETX$ = ZSTX$ + ZETX$ + ZETX$
380 ZIDENQ$ = STRING$(8, 240) + ZENQ$
390 ZASC$ = STRING$(130, 0) ^ DUMMY
400 ZBCC$ = CHR$(48) ^ "0" ^ DUMMY
410 ZR$ = CHR$(48) ^ "0" ^ DUMMY
420 CLOSE
425
430 `***** VALORES PARA RETARDO *****
490 ZT1.5S=176: ZT3S=352: ZT24S=2820: ZT0.1S=12: ZTAJ=16
500 `***** LENGUAJE ASSEMBLER PARA CALCULO DEL CRC *****
510 ZL1 = 0: ZL2 = 0: ZI1$ = ""
520 DATA 55,8B,EC,50,53,51,52,56,57,06,1E,8B,76,0A,8B,74
530 DATA 01,8B,7E,0C,8B,7D,01,33,C0,B9,81,00,8A,04,32,E0
540 DATA E8,2A,00,88,05,46,47,E2,F3,B0,30,3A,24,74,02,B0
550 DATA 31,8B,76,06,8B,74,01,88,04,8B,76,08,8B,74,01,88
560 DATA 24,1F,07,5F,5E,5A,59,5B,58,5D,CA,08,00,55,8B,EC

```



```

570 DATA 53,56,33,DB,3C,60,75,02,B0,FF,3C,F0,72,02,2C,10
580 DATA 2C,DO,24,1F,8A,D8,8B,76,02,83,C6,4B,2E,8A,00,5E
590 DATA 5B,5D,C3,7D,4A,4B,4C,4D,4E,4F,50,51,52,00,30,31
600 DATA 32,33,34,35,36,37,38,39,00,00,00,00,00,2D,00,00:
610 RESTORE 520: DEF SEG = VARSEG(ZZBCC(1)):
    FOR ZL1 = 0 TO &H8F: READ ZI1$:
    POKE (VARPTR(ZZBCC(1)) + ZL1), VAL("&H" + ZI1$):
    NEXT ZL1
615
620 ^***** MENU PRINCIPAL *****
630 SPEED$ = "9600"      VELOCIDAD DEL PUERTO SERIAL
640 CLS
650 LOCATE 5, 16: PRINT "** COMUNICACION CAJA REG. - PC **":
    ZIN$ = "XXX"
660 LOCATE 7, 14: PRINT "1. RECEPCION DE DATOS (CRE a PC)"
670 LOCATE 9, 14: PRINT "2. TRANSMISION DE DATOS (PC a CRE)"
700 LOCATE 11, 14: PRINT "3. DEFINIR BLOQUE COMANDO"
710 LOCATE 13, 14: PRINT "4. SALIR "
720 LOCATE 15, 21: PRINT "POR FAVOR SELECCIONE UNA OPCION "
770 LOCATE 17, 25: INPUT "Y PRESIONE <ENTER> . . . ", ZN
790 ON ERROR GOTO 2720: ON ZN GOTO 970,1240,2920,930: GOTO 640
930 ^***** TERMINAR EL PROGRAMA *****
940 CLS : LOCATE 10,25 : PRINT "Proceso Terminado. Gracias..."
950 SYSTEM
960 ^***** MODULO DE RECEPCION *****
970 CLS : LOCATE 5, 10:
    PRINT "INGRESE NOMBRE DEL ARCHIVO COMANDO !"
980 LOCATE 7, 20: INPUT "* ", CNAM$
990 OPEN CNAM$ FOR RANDOM AS #2
1000 LOCATE 13, 14: PRINT "ESTA SEGURO ? Y/N "
    ENT$ = INPUT$(1): PRINT ENT$:
1010 IF (ENT$="Y" OR ENT$="y") THEN 1020 ELSE CLOSE : GOTO 640
1020 FIELD #2, 128 AS ZC$
1030 OPEN "COM1:"+SPEED$+",N,8,1,RS,CS0,DS" FOR RANDOM AS #1
1040 RCV = 1: ZCOM = 1
1050 GOSUB 2030
1060 IF ZRS$ = "E" THEN CLOSE : BEEP: LOCATE 23, 15:
    PRINT "ERROR END": GOTO 650
1070 CLOSE #2
1080 ZIN$ = "$$TMP$$$": OPEN ZIN$ FOR RANDOM AS #2 ^TEMPORAL
1090     FIELD #2, 128 AS ZA$
1100 GOSUB 1610
1110 LOCATE 23, 50: IF ZRS$ = "N" THEN PRINT "NORMAL END":
    GOTO 1130
1120 IF ZRS$ = "E" THEN PRINT "ERROR END": CLOSE :
    GOTO 650 ELSE CLOSE : GOTO 1080
1130 GET #2, 1
1140 IF LEFT$(ZA$, 1) <> CHR$(&HF4) THEN 1170 ^AUTO PROGRAM
1160 ^* ZNAME=X/Z/P(1)+MAQUINA NO.(4)+CLASE(2)+FECHA,0-9(1) *
1170 ZNAME$ = MID$(ZASC$, 20, 1)
1180 IF ZNAME$ = "4" THEN ZNAME$ = "X": GOTO 1200
1190 IF ZNAME$ = "5" THEN ZNAME$ = "Z" ELSE ZNAME$ = "P"
1200 ZNAME$ = ZNAME$ + MID$(ZASC$, 6, 4) + MID$(ZASC$, 25, 3)
1210 ON ERROR GOTO 2700: CLOSE : KILL ZNAME$:
    NAME ZIN$ AS ZNAME$
1220 LOCATE 23, 15:
    PRINT "NOMBRE DEL ARCHIVO RECIBIDO ES " + ZNAME$:
    GOTO 650
1230 ^***** MODULO DE TRANSMISION *****
1240 CLS:LOCATE 5,10:
    PRINT "INGRESE NOMBRE DEL ARCHIVO COMANDO !"
1250 LOCATE 7, 20: INPUT "* ", CNAM$
1260     OPEN CNAM$ FOR RANDOM AS #3
1270 LOCATE 9, 10:
    PRINT "INGRESE NOMBRE DEL ARCHIVO DE DATOS A ENVIAR: "
1280 LOCATE 11, 20: INPUT "* ", ZNAME$
1290     OPEN ZNAME$ FOR RANDOM AS #2
1300 LOCATE 13, 14:

```

```

PRINT "ESTA SEGURO ? Y/N ", . Z$ = INPUT$(1):
PRINT Z$;
1310 IF (Z$ = "Y" OR Z$ = "y") THEN 1320 ELSE CLOSE : GOTO 640
1320     FIELD #2, 128 AS ZC$
1330     FIELD #3, 128 AS ZA$
1340 OPEN "COM1:"+SPEED$+" ,N,8,1,RS,CSO,DS" FOR RANDOM AS #1
1350 ZCOM = 0: RCV = 0
1360 GOSUB 2070: CLOSE : CLS
1370 LOCATE 23, 15: PRINT "ENVIADO DATOS "+ZNAME$+" ES "; :
    IF ZRS$ = "N" THEN PRINT "NORMAL END"
        ELSE PRINT "ERROR END"

1380 GOTO 650
1390 -
1440 ***** SUBROUTINA (RECEPCION & ENVIO) *****
1450 -
1460 -
1470 * SUBROUTINA DE RECEPCION DE N CARACTERES CON TIMEOUT *
1480 * 1CH LEIDO-TIMEOUT 1.5 S --> Z$=DATA,I=0 OK,I=1 TIMEOUT
1490 ZN = 1: I = ZT1.5S: GOTO 1580
1500 * 1CH LEIDO-TIMEOUT 3 S ----> Z$=DATA,I=0 OK,I=1 TIMEOUT
1510 ZN = 1: I = ZT3S: GOTO 1580
1520 * 1CH LEIDO-TIMEOUT 24 S --> Z$=DATA,I=0 OK,I=1 TIMEOUT
1530 ZN = 1: I = ZT24S: GOTO 1580
1540 * 1CH LEIDO-TIMEOUT 0.1 S --> Z$=DATA,I=0 OK,I=1 TIMEOUT
1550 ZN = 1: I = ZT0.1S: GOTO 1580
1560 * 129CH LEIDO-TIMEOUT 24 S -> Z$=DATA,I=0 OK,I=1 TIMEOUT
1570 ZN = 129: I = ZT24S: GOTO 1580
1580 IF LOC(1) >= ZN THEN Z$ = INPUT$(ZN, #1): I = 0: RETURN
1590 I = I - 1: FOR ZZ = 1 TO 2: NEXT ZZ:
    IF I = 0 THEN I = 1: RETURN ELSE 1580
1600 ***** SUBROUTINA DE RECEPCION *****
1610 ZRS$="E": ZBLC=1: ZODEV=0: ZSTEP=0: ZZH$="": ZTIME=0
1620 ZN1 = 0
1630 GOSUB 1510: ON I GOTO 1660: ZN1 = ZN1 + 1`1CH-3S
1640 IF Z$ = ZENQ$ THEN 1650 ELSE 1630
1650 IF ZN1 < 8 THEN 1680 ELSE 1620
1660 ZTIME = ZTIME + 1: IF ZTIME > 8 THEN ZRS$ = "T":
    RETURN ELSE 1620
1670 ***** ENVIA ACKO *****
1680 ZBF$ = ZAKO$: GOSUB 2510: ZX$ = ZAKO$`** PRINT ZAKO$
1690 GOSUB 1530: ON I GOTO 1750`1CH-24S
1700 IF Z$ = ZSTX$ THEN 1770`1CH-24S
1710 IF Z$ = ZENQ$ THEN ZBF$=ZX$:GOSUB 2510:GOTO 1690
1720 IF Z$ = ZDLE$ THEN 1960`1CH-24S
1730 IF Z$ = ZEOT$ THEN 1750 ELSE 1690
1740 ***** SECUENCIA FINAL *****
1750 ZBF$ = ZDLEEOT$: GOSUB 2510`** PRINT ZDLEEOT$
1760 IF ZSTEP = 1 THEN ZRS$ = "N": RETURN ELSE ZRS$="E": RETURN
1770 GOSUB 1530: ON I GOTO 1750: ZMODE$ = Z$`1CH-24S
1780 IF Z$ = ZENQ$ THEN 1690
1790 IF Z$ = ZETX$ THEN GOTO 1980`1CH-24S
1800 GOSUB 1570: ON I GOTO 1750: Z$ = ZMODE$ + Z$`130CH-24S
1810 DEF SEG = VARSEG(ZZBCC(1)):
    CALL absolute(VARPTR(ZZBCC(ZASC$, Z$, ZBCC$, ZR$))):
    DEF SEG : ZH$ = MID$(Z$, 22, 3)
1820 IF ZR$ = "0" THEN 1850 ELSE 1840
1830 ***** ENVIA NACK *****
1840 ZBF$=ZNACK$:GOSUB 2510:ZX$=ZNACK$:GOTO 1690`** ? ZNACK$
1850 IF ZZH$ <> ZH$ THEN 1870 ELSE 1930 `BLOQUE DUPLICADO
1860 ***** ESCRIBE EL BLOQUE EN DISCO *****
1870 IF ZODEV = 1 AND ZMODE$ <> CHR$(&HF4)
    THEN IF MID$(Z$, 20, 1) <> CHR$(&HF7)
        THEN LSET ZA$ = ZASC$: ZODEV = 0: GOTO 1910
1880 IF ZODEV = 1 THEN LSET ZA$ = Z$: ZODEV = 0: GOTO 1910
1890 IF ZODEV = 0 AND ZMODE$ <> CHR$(&HF4)
    THEN IF MID$(Z$, 20, 1) <> CHR$(&HF7)
        THEN LSET ZA$ = ZASC$: ZODEV = 1: GOTO 1910
1900 IF ZODEV = 0 THEN LSET ZA$ = Z$: ZODEV = 1: GOTO 1910

```

```

1910 PUT #2, ZBLC: LOCATE 1, 1: PRINT "Bloque:"; ZBLC;
      PRINT ZASC$`***** PRUEBA
1920 ZBLC = ZBLC + 1
1930 IF ZODEV = 0 THEN 1680
1940 `***** ENVIA ACK1 `*****
1950     ZBF$ = ZAK1$: GOSUB 2510: ZX$ = ZAK1$: GOTO
1690`***** PRINT ZAK1$
1960 GOSUB 1530: ON I GOTO 1750`1CH-24S
1970 IF Z$ = ZEOT$ THEN 1760 ELSE 1690
1980 GOSUB 1530: ON I GOTO 1750`1CH-24S
1990 IF Z$ = ZETX$ THEN ZSTEP = 1: GOTO 2000 ELSE 1950
2000 IF ZODEV = 0 THEN ZODEV = 1: GOTO 1930 ELSE ZODEV = 0:
      GOTO 1930

2010
2020
2030 -
2040 -
2050 -
2060 `***** SUBROUTINA DE ENVIO *****
2070 ZERR = 0: ZRS$ = "E": ZBLC = 0: ZODEV = 0: ZSTEP = 0
2080 ZBF$ = ZIDENQ$: GOSUB 2510`***** PRINT ZIDENQ$
2090 GOSUB 1490: ON I GOTO 2120`1CH-1.5S
2100 IF Z$ = ZDLE$ THEN GOSUB 1490: ON I GOTO 2120: GOTO
2170`1CH-1.5S
2110 IF Z$ = ZNACK$ THEN 2150 ELSE 2090
2120 ZERR = ZERR + 1: IF ZERR > 14 THEN 2150 ELSE 2080
2130 GOSUB 1550: ON I GOTO 2120: GOTO 2130`1CH-0.1S
2140 `***** SECUENCIA FINAL *****
2150     ZBF$ = ZDLEEOT$: GOSUB 2510`***** PRINT ZDLEEOT$
2160 IF ZSTEP = 2 THEN ZRS$ = "N":
      RETURN ELSE ZRS$ = "E": RETURN
2170 IF Z$ = ZACKO$ THEN 2200
2180 IF Z$ = ZEOT$ THEN 2160 ELSE 2130
2190 `***** ZSTEP=0 ENVIANDO DATOS , ZSTEP=1 ULTIMO DATO ,
      ZSTEP=2 DESCONEXION *****
2200 ON ZSTEP GOTO 2420, 2150
2210 ON ZCOM GOTO 2230
2220 IF ZBLC = 0 THEN GET #3:GOTO 2240 ELSE ZCOM = 1:ZBLC = 0
2230 GET #2: ZA$ = ZC$
2240 ZBLC = ZBLC + 1: LOCATE 1, 1: PRINT "Bloque: "; ZBLC;
2250 ON ZCOM GOTO 2270
2260 ZEND$ = ZETX$: GOTO 2280
2270 IF RCV = 1 THEN ZEND$ = ZETX$ ELSE ZEND$ = ZETB$
2280 Z$ = ZA$ + ZEND$ + "0":
      DEF SEG = VARSEG(ZBCC(1)):
      CALL absolute(VARPTR(ZBCC(ZASC$, Z$, ZBCC$, ZRS$)))
2290 Z$ = ZSTX$ + ZA$ + ZEND$ + ZBCC$ `CONSTRUCCION DE DATA
2300 IF EOF(2) THEN ZSTEP = 1:
      GOTO 2420`SIGUIENTE BLOQUE ES STX+ETE+BCC
2310 ZERR = 0: ZX$ = Z$
2320 ZBF$ = ZX$: GOSUB 2510
2330 GOSUB 1510: ON I GOTO 2390`1CH-3S ***** PRINT ZX$
2340 IF Z$ = ZNACK$ THEN 2320
2350 IF Z$ = ZDLE$ THEN GOSUB 1510:
      ON I GOTO 2390 ELSE 2440`1CH-3S
2360 IF ZODEV = 1 THEN 2410
2370 IF Z$ = ZACK1$ THEN ZODEV = 1: GOTO 2200
2380 IF Z$ = ZEOT$ THEN 2160 ELSE 2440
2390 ZERR = ZERR + 1: IF ZERR > 65 THEN 2150
2400     ZBF$ = ZENQ$: GOSUB 2510:
      GOTO 2330`***** PRINT ZENQ$
2410 IF Z$ = ZACKO$ THEN ZODEV = 0: GOTO 2200 ELSE 2380
2420 ON RCV GOTO 2450
2430 Z$ = ZSTXETX$: ZSTEP = 2: GOTO 2310
2440 GOSUB 1550: ON I GOTO 2390: GOTO 2440`1CH-0.1S
2450 ZBF$ = ZEOT$: GOSUB 2510: ZRS$ = "N": RETURN
2460
2470

```

```

2480 -
2490 -
2500 ***** SUBROUTINA DE CONTROL *****
2510 ZM1%=INP(&H3FE):ZM2%=ZM1% AND &H80
2520 IF ZM2%<>=0 THEN 3000 IF CD=1 THEN WAIT ELSE NEXT
2530 ZM1% = INP(&H3FC) RTS OFF ---->RTS ON CON CTS CHECK
2540 ZM2% = ZM1% OR &H2
2550 OUT &H3FC, ZM2%
2560 ZM1% = INP(&H3FE)
2570 IF (ZM1% AND &H10) = 0 THEN 2560 ELSE 2580
2580 PRINT #1, ZBF$: : GOSUB 2660
2590 ZM1% = INP(&H3FC) RTS ON ---->RTS OFF
2600 ZM2% = ZM1% AND &HFD
2610 OUT &H3FC, ZM2%: RETURN
2620 -
2630 -
2640 -
2650 ***** SUBROUTINA DE RETARDO *****
2660 ZL2 = LEN(ZBF$) * ZTAJ`8.7S/1CH <--- AJUSTE TIEMPO *****
2670 FOR ZL1 = 0 TO ZL2: NEXT ZL1: RETURN
2680 -
2690 -
2700 ** CUANDO SUCEDE UN ERROR (RUTINA DE RECEPCION) **
2710 RESUME NEXT
2720 ** CUANDO SUCEDE UN ERROR PARA RUTINAS NORMALES **
2730 ZERR = ERR: ZERL = ERL: IF ERR = 57 THEN RESUME NEXT
2740 IF ERR = 53 THEN RESUME NEXT ELSE LOCATE 1, 1:
PRINT "line: "; ERL; " code: "; ERR
2750 CLOSE : STOP: END
2760 -
2770 -
2780 -
2880 -
2890 -
2900 -
2910 ***** DEFINICION DE BLOQUE COMANDO *****
2920 EBC = 240: X$ = " ": CLS : LOCATE 5, 20:
PRINT "Definir el Bloque Comando"
2950 MC$ = "2" CASIO MODELO TK-2300
3000 X$ = " ": GOSUB 3370: GOSUB 3380: GOSUB 3390
3010 GOSUB 3350
3020 IF (ENT = 0) OR (ENT > 4) THEN BEEP: GOTO 3000
3030 IF ENT = 2 THEN ENT = 4
3040 MD$ = STR$(ENT + 4)
3050 IF VAL(MD$) = 5 THEN X$ = "*" ELSE X$ = " "
3060 GOSUB 3370
3070 IF VAL(MD$) = 8 THEN X$ = "*" ELSE X$ = " "
3080 GOSUB 3380
3090 IF VAL(MD$) = 7 THEN X$ = "*" ELSE X$ = " "
3100 GOSUB 3390
3110 LOCATE 21, 13: PRINT " "
3120 LOCATE 14, 17: INPUT "CODIGO COMANDO ", ENT$
3130 IF VAL(ENT$) > 999 THEN BEEP: LOCATE 14, 27:
PRINT " ": GOTO 3120
3140 KCD$ = ENT$
3150 LOCATE 14, 17: PRINT "CODIGO COMANDO "; KCD$
3160 LOCATE 16, 17:
INPUT "NOMBRE DEL ARCHIVO A GRABAR ", CNAM$
3170 LOCATE 21, 17: PRINT "SEGURO ? (Y/N) "; :
ENT$ = INPUT$(1): PRINT ENT$
3180 IF (ENT$ = "Y" OR ENT$ = "y") THEN 3190 ELSE 640
3190 MC$ = CHR$(EBC + VAL(MC$)):
MD$ = STRING$(18, 240) + CHR$(EBC +
VAL(RIGHT$(MD$, 1))) + STRING$(4, 240)
3200 KCD$ = RIGHT$(KCD$, 3)
3210 KCD$ = CHR$(EBC + VAL(LEFT$(KCD$, 1))) +
CHR$(EBC + VAL(MID$(KCD$, 2, 1))) + CHR$(EBC +
VAL(RIGHT$(KCD$, 1))) + STRING$(7, 240)

```



```
3220 CMD$ = MC$ + MD$ + KCD$ + STRING$(96, 240)
3230 LOCATE 21, 17: PRINT "COMANDO ["; CNAM$; "]" GRABADO"
3240 CMIN$ = "$$$TMP$$": OPEN CMIN$ FOR RANDOM AS #2
3250 FIELD #2, 128 AS CM$
3260 LSET CM$ = CMD$
3270 PUT #2
3280 ON ERROR GOTO 2720: CLOSE : KILL CNAM$:
    NAME CMIN$ AS CNAM$
3290 GOTO 640
3300
3310
3320 .                SUBROUTINA DE SELECCION *****
3350 LOCATE 21, 13: INPUT "INGRESE LA OPCION DESEADA ", ENT
3360 LOCATE 21, 13: PRINT "                                ": RETURN
3370 LOCATE 10, 13:
    PRINT X$; "1. LECTURA DE REPORTE X/Z ": RETURN
3380 LOCATE 11, 13:
    PRINT X$; "2. ENVIO DE PROGRAMACION": RETURN
3390 LOCATE 12, 13:
    PRINT X$; "3. LECTURA DE PROGRAMACION": RETURN
```

CAPITULO V COSTOS

Para llevar a cabo este proyecto se necesita un período de investigación y desarrollo de aproximadamente dos meses y además contar con los equipos necesarios para efectuar las pruebas del caso:

DESCRIPCION	COSTO US\$
Caja registradora marca CASIO modelo TK-2300	2,200
Tarjeta de Interfase marca CASIO modelo IO-PB-11	900
Computador compatible 486	800
Impresora	300
Cable de comunicación 3 metros	3
Conector DIN 13	15
Conector DB-25	2
Investigación y desarrollo durante dos meses	4,000
TOTAL	8,220

Teniendo en cuenta que una interfase de comunicación de datos ya desarrollada tiene un costo en el mercado de US\$ 1,000 se requiere efectuar la venta de nueve interfases para lograr el retorno de la inversión.

CONCLUSIONES

Para la implementación de un sistema de comunicación de datos entre un computador y algún otro equipo electrónico se requiere conocer en el computador las características del puerto serial RS-232, el UART sus registros y programación y por otro lado el protocolo de comunicaciones aceptado por el equipo electrónico.

El desarrollo propio de un sistema de comunicación de datos caja registradora-computador representa una gran ventaja contra una aplicación extranjera -que mayormente es inmodificable- debido a su rápida adecuación a futuros cambios en la política fiscal y la facilidad de adaptarse a situaciones particulares del mercado nacional.

Como elemento fiscal la caja registradora representa una gran ventaja debido a que su programación interna es cerrada y no pueden ser alterados los datos de totales de venta.

Un establecimiento comercial obtiene grandes ventajas en automatización y disminución de errores cuando posee una caja registradora enlazada a un computador y deja de lado los procesos de venta y control manual.

ANEXOS

A. DIAGRAMAS DE CONEXION

B. PLANOS DEL CIRCUITO DE LA TARJETA INTERFASE

C. DESCRIPCION DEL CIRCUITO DE LA TARJETA INTERFASE

ANEXO A

DIAGRAMAS DE CONEXION

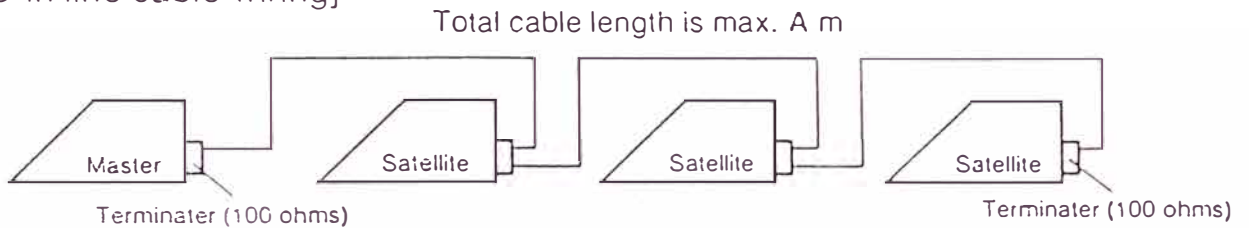
2. INTERFACE

2-1. C-In line (ECR to ECR)

1) C-In line connection diagram

The maximum C-In line cable length and the maximum connection unit will be varied by the in-line communication speed as shown below list.

[C-In line cable wiring]



C-In line communication speed (Baud rate)	Maximum connection unit	Maximum cable length A (m)
1.25 M bps	8	86
625 K bps	12	200
312.5 K bps	20	430
156.25 K bps	32	890

2) C-In line cable, Connector, and Terminator

C-In line cable

Code No.	Parts name	Specification
1904 4023	In-line cable	IPEV-SLA0.5x1P

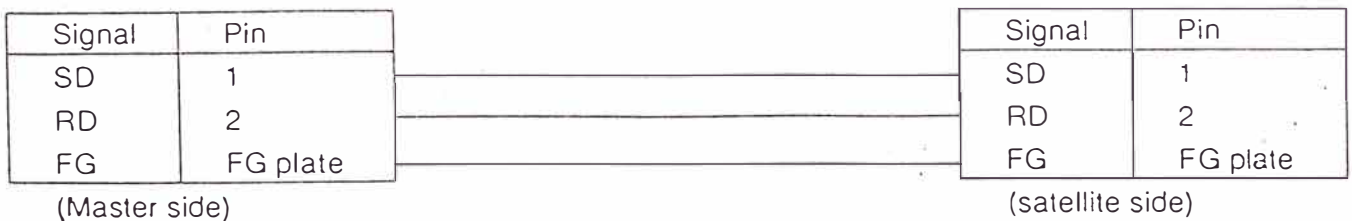
C-In line connector

Code No.	Parts name	Specification
3500 5820	In-line connector kit	XLP-KIT-1

Terminator

Code No.	Parts name	Specification
2600 2516	Carbon film resistor	R-25-100-J

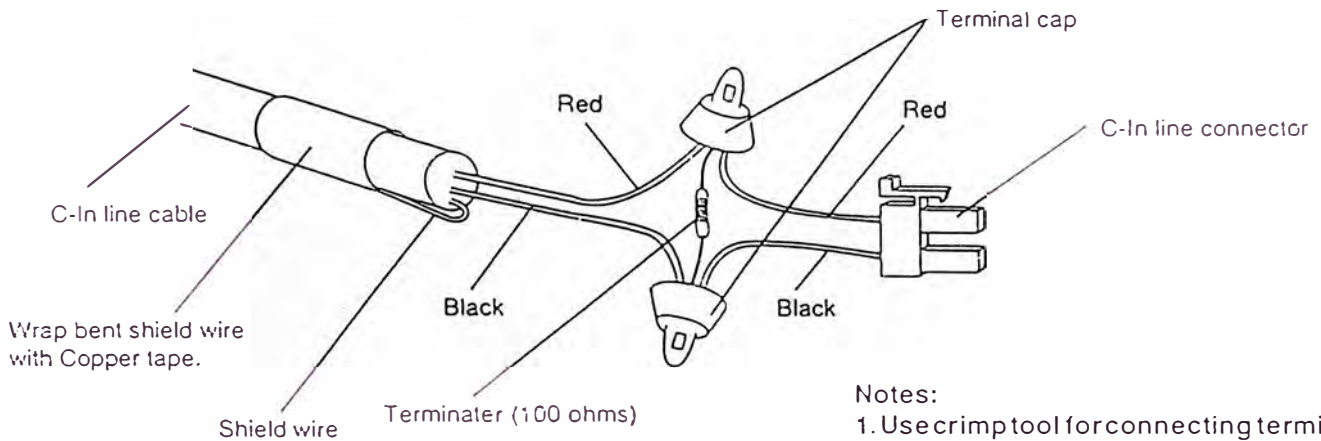
3) Cable connection diagram



4) To fix C-In line connector to ECR.

Connect the C-In line cable and C-In line connector as shown below:

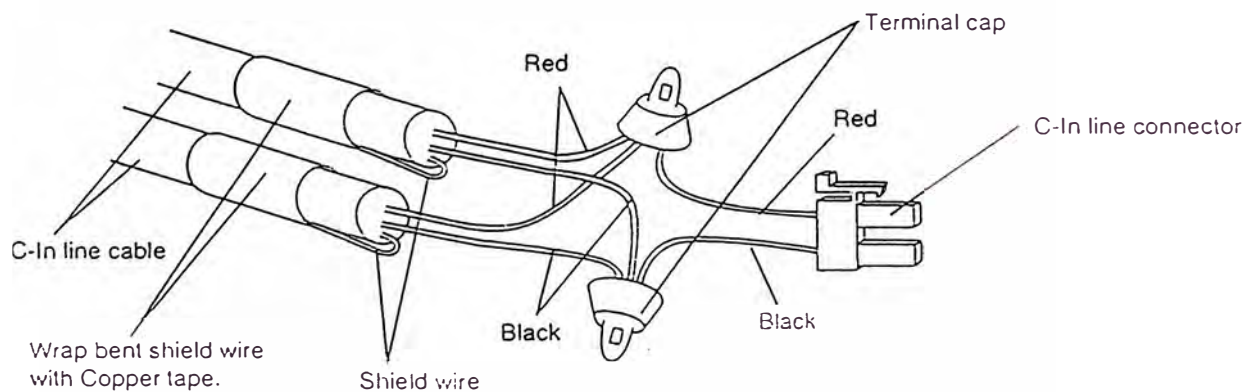
[Connector on the end of C-In line]



Notes:

1. Use crimp tool for connecting terminal caps.
2. If crimp tool is not available, connect the wires with solder instead of terminal cap. After the soldering, insulate the wires with vinyl tape.

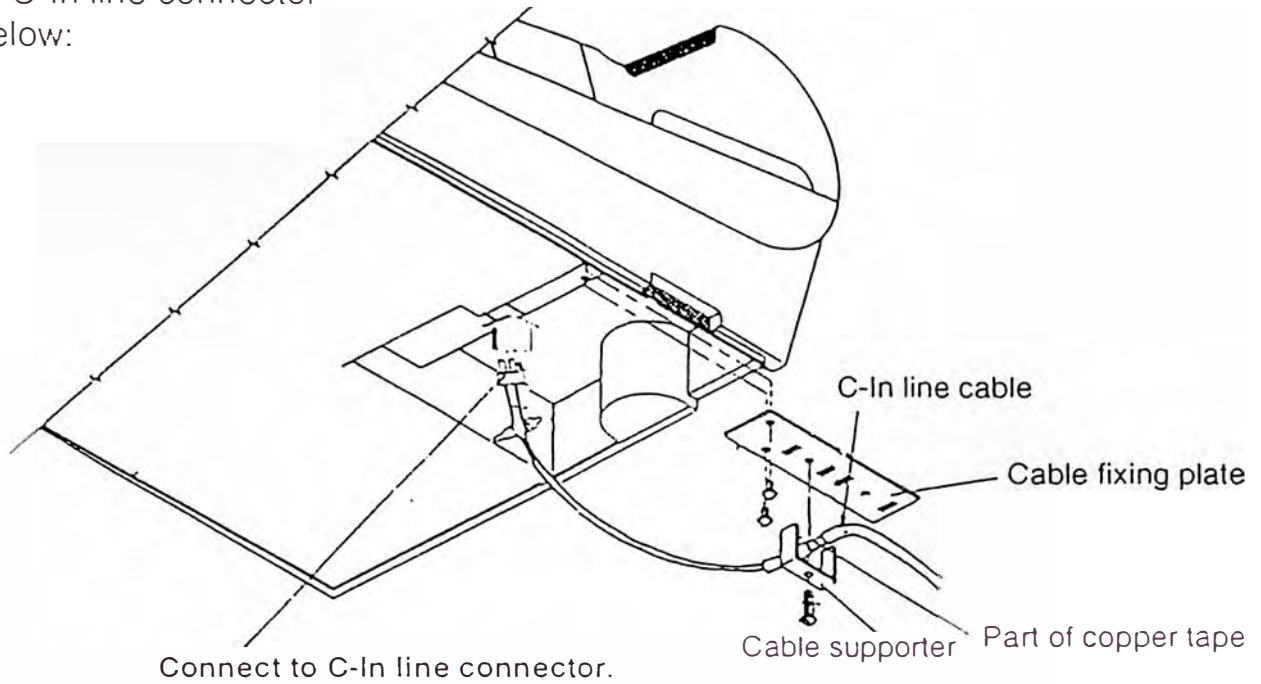
[Connector between ECRs]



Notes:

1. Use crimp tool for connecting terminal caps.
2. If crimp tool is not available, connect the wires with solder instead of terminal cap. After the soldering, insulate the wires with vinyl tape.

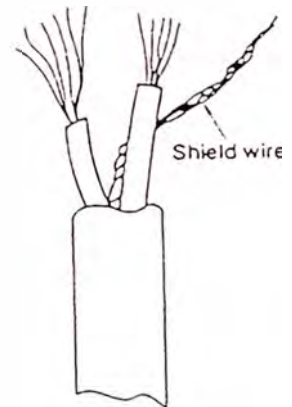
Connect the C-In line connector as shown below:



6) C-In line cable specification

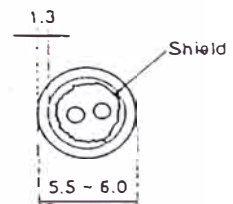
1. Structure

Conductor	Material:	Tin plated mild copper twist
	Cross section area:	0.5mm ²
	Construction:	20 pieces/0.18 mm
	O.D.	0.95 mm
Insulation	Material:	Polyethylene
	Thickness	0.3 mm
Sheath	Material:	PVC
	Thickness	1.3 mm
Finished O.D.:		6.0 -6.5 mm
Approx. weight:		55 kg/km



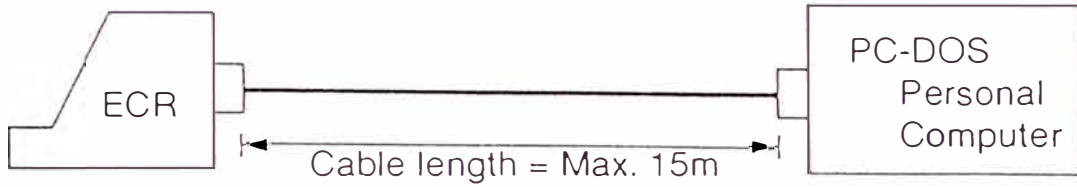
2. Properties (20°C)

Maximum conductor resistance	38.7 Ω/km
Minimum insulation resistance	10 kMΩ·km
Electric strength	AC350 V/1 minute
Electrostatic capacity (Standard)	75 nF/km

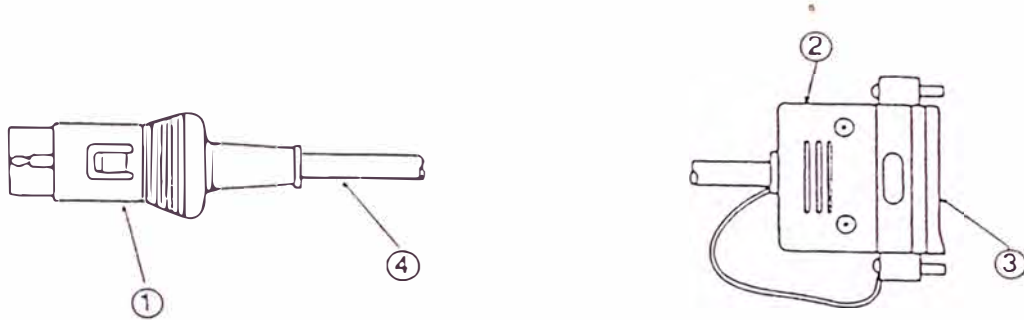


2-2. On line

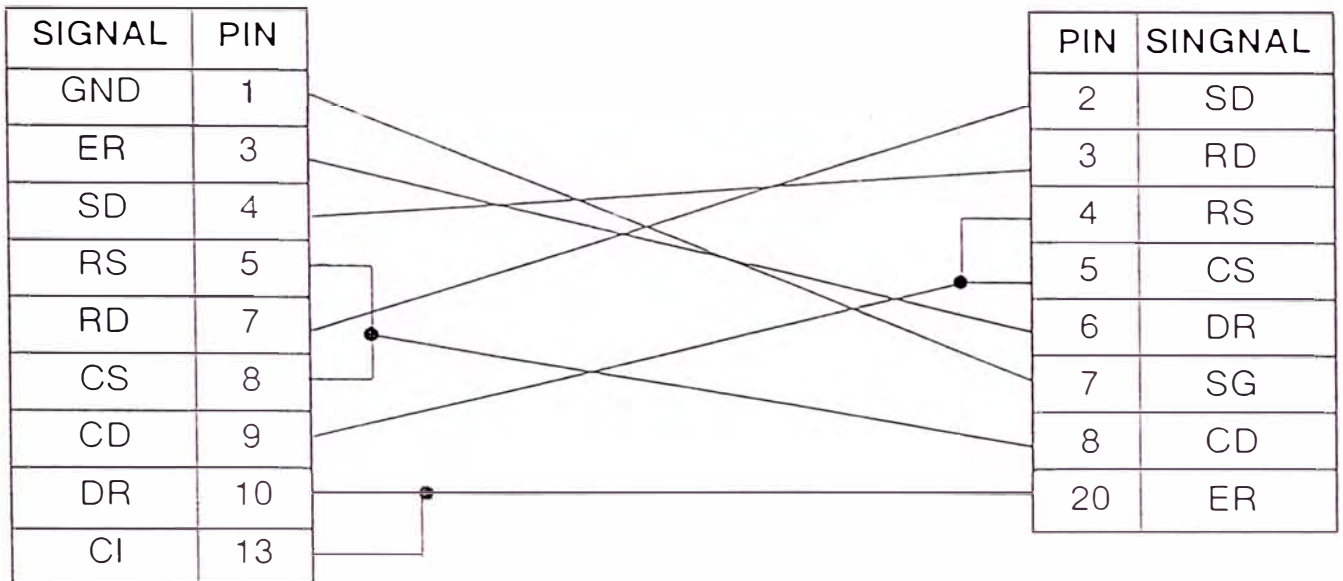
(1) Direct connection to P/C



[Cable]



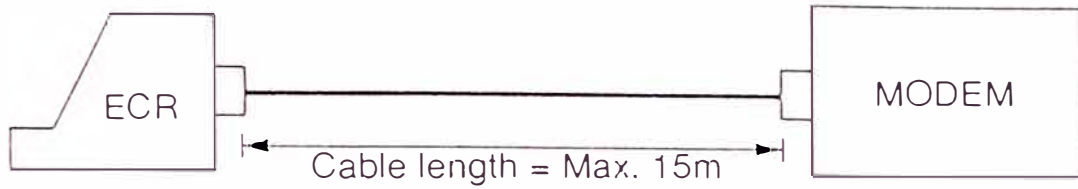
1) Wiring diagram (A) A SYNC.



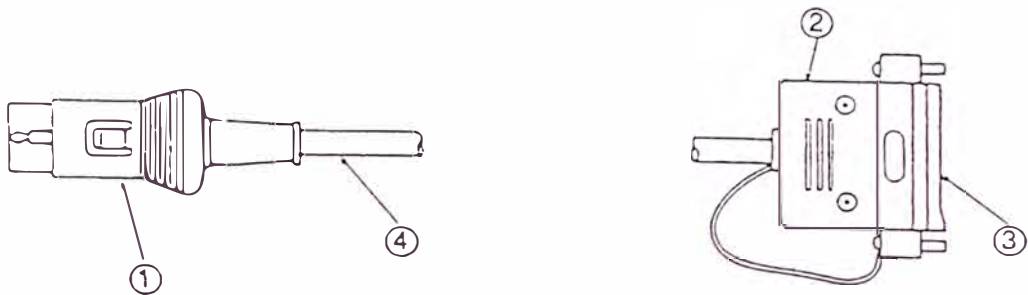
2) Parts list

No.	Code Number	Part	Specification
1	3612 0762	DIN 13P PLUG	TCP9361-71-111
2	3513 1019	Junction shell	DB-C2-J9
3	3510 2043	Shell connector	DB-25S
4		Cable	PLGW-3456-01A

(2) MODEM connection



[Cable]



1) Wiring

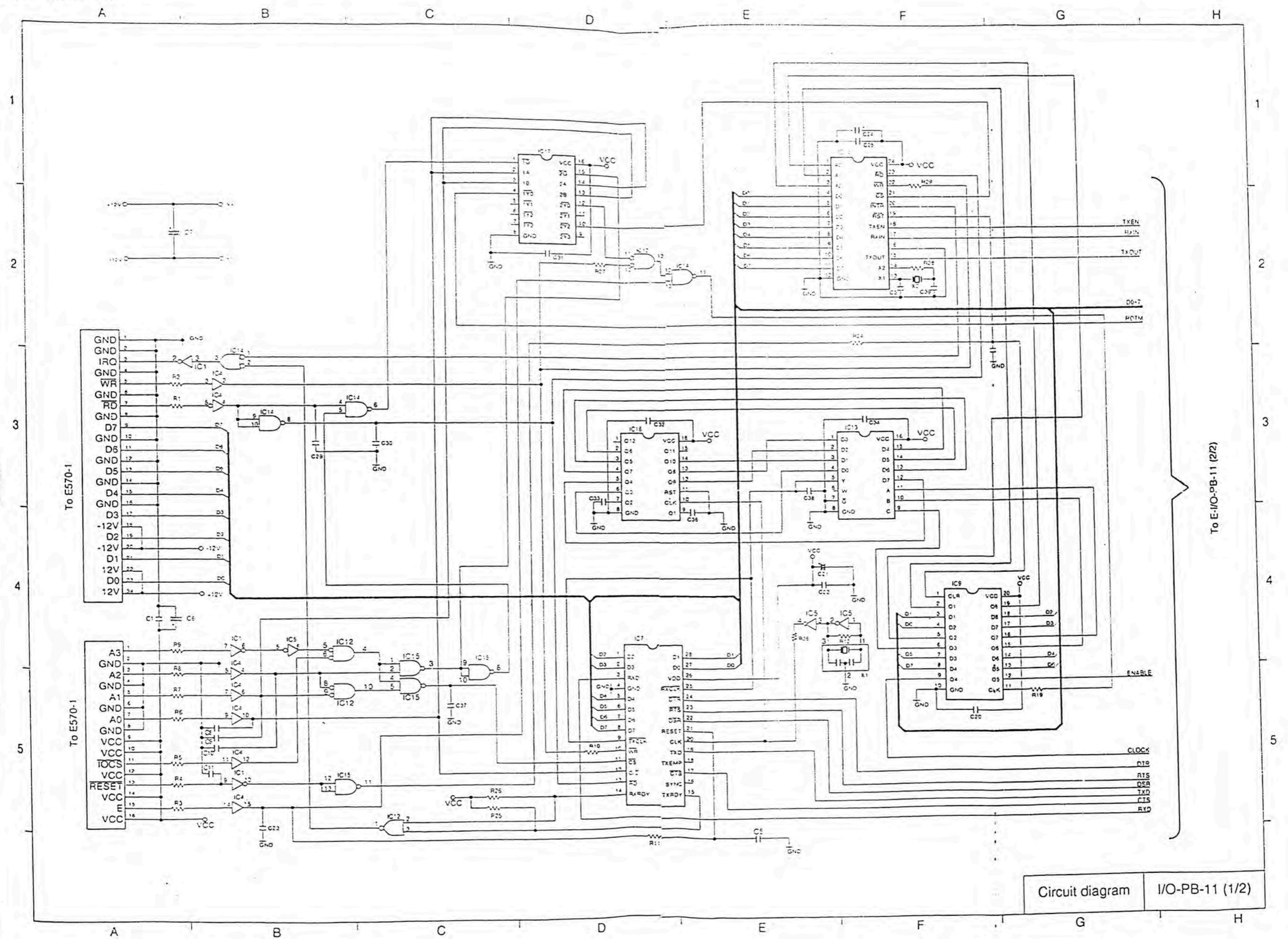
SIGNAL	PIN	PIN	SIGNAL
GND	1	2	SD
ER	3	3	RD
SD	4	4	RS
RS	5	5	CS
NC	6	6	DR
RD	7	7	SG
CS	8	8	CD
CD	9	15	-
DR	10	17	-
NC	11	20	ER
NC	12	22	CI
CI	13	24	-

2) Parts list

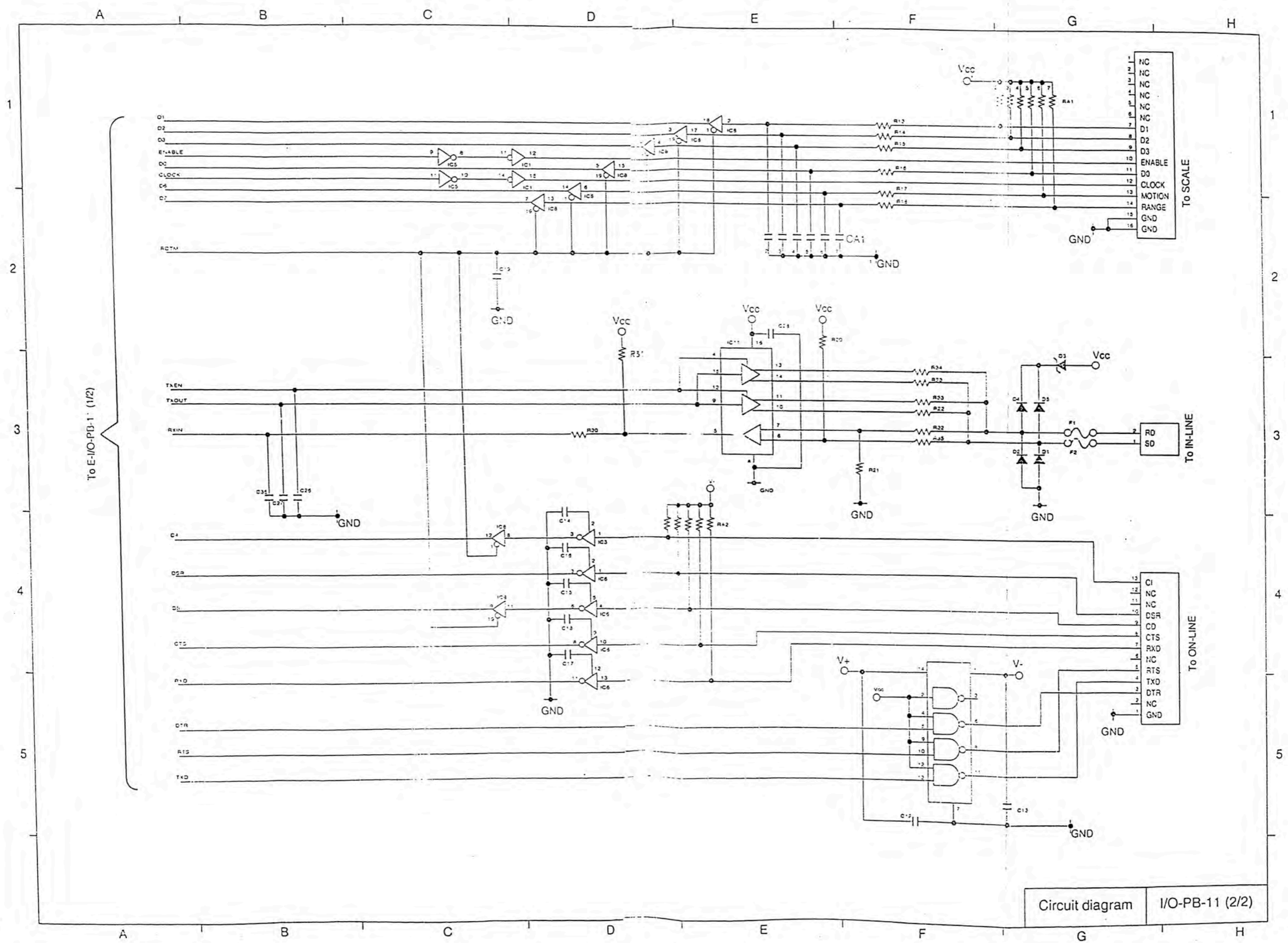
No.	Code Number	Part	Specification
1	3612 0762	DIN 13P PLUG	TCP9361-71-111
2	3513 1019	Junction shell	DB-C2-J9
3	3510 2043	Shell connector	DB-25S
4		Cable	PLGW-3456-01A

ANEXO B

PLANOS DEL CIRCUITO DE LA TARJETA INTERFASE



Circuit diagram	I/O-PB-11 (1/2)
-----------------	-----------------



To E-I/O-PB-11 (1/2)

To SCALE

To IN-LINE

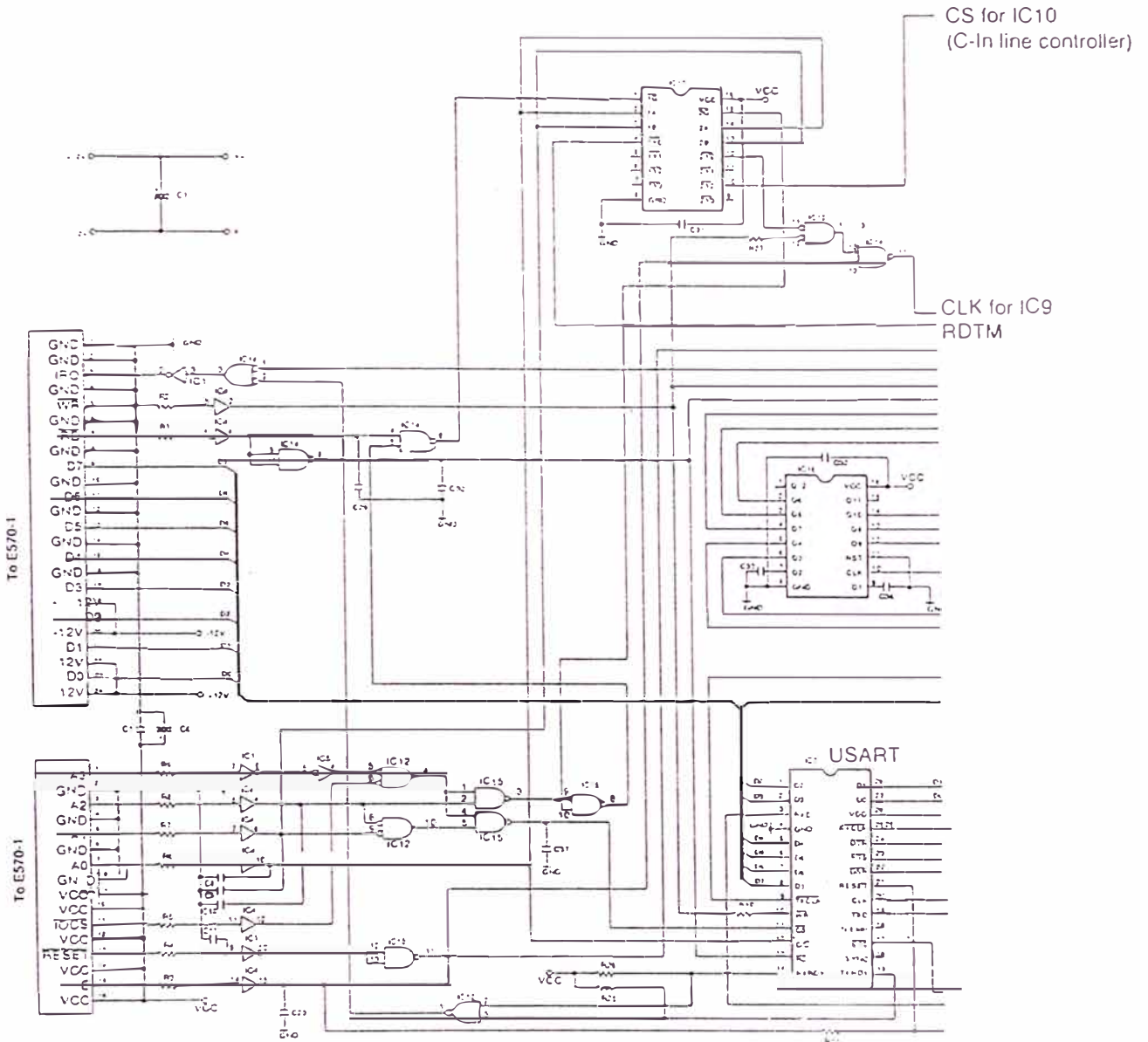
To ON-LINE

ANEXO C

DESCRIPCION DEL CIRCUITO DE LA TARJETA INTERFASE

3. CIRCUIT EXPLANATIONS

3-1. Address selection

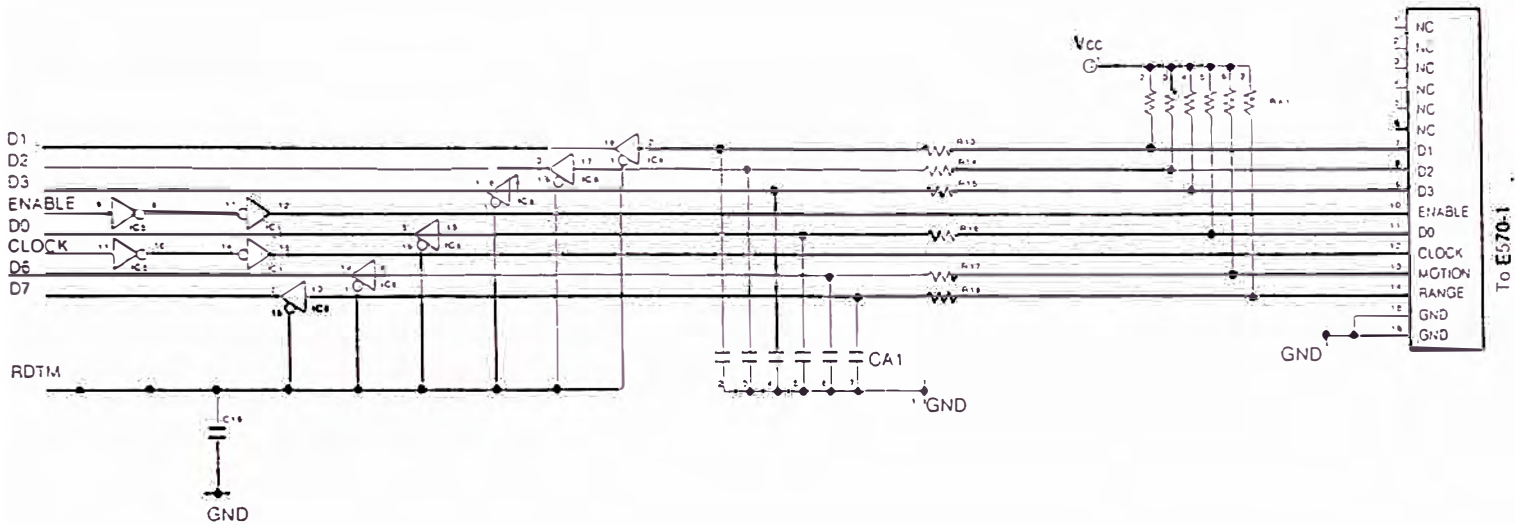


According to the address of main CPU, the decoder selects the following optional devices

IC No.	Pin No.	CPU address	Descriptions
IC17	4	$\overline{1Y0}$	RDTM gate enable of IC8 (For scale data read)
	10	$\overline{2Y2}$	Chip selection of C-In line controller
IC14	11	\overline{CLK}	Clock signal of IC9 (For C-In line controller data)
IC15	6	\overline{CS}	Chip selection of USART (Data)
			Chip selection of USART (Control status)

3-2. Scale buffer circuit

When a measured object is put on the scale, the scale starts weighing. During weighing, MO signal is "H". When the weighing is finished with indication of the weight of measures object, the MO signal becomes "L". Then EN signal is output to the scale and the scale output the weight data (D0~D3) to the scale buffer.



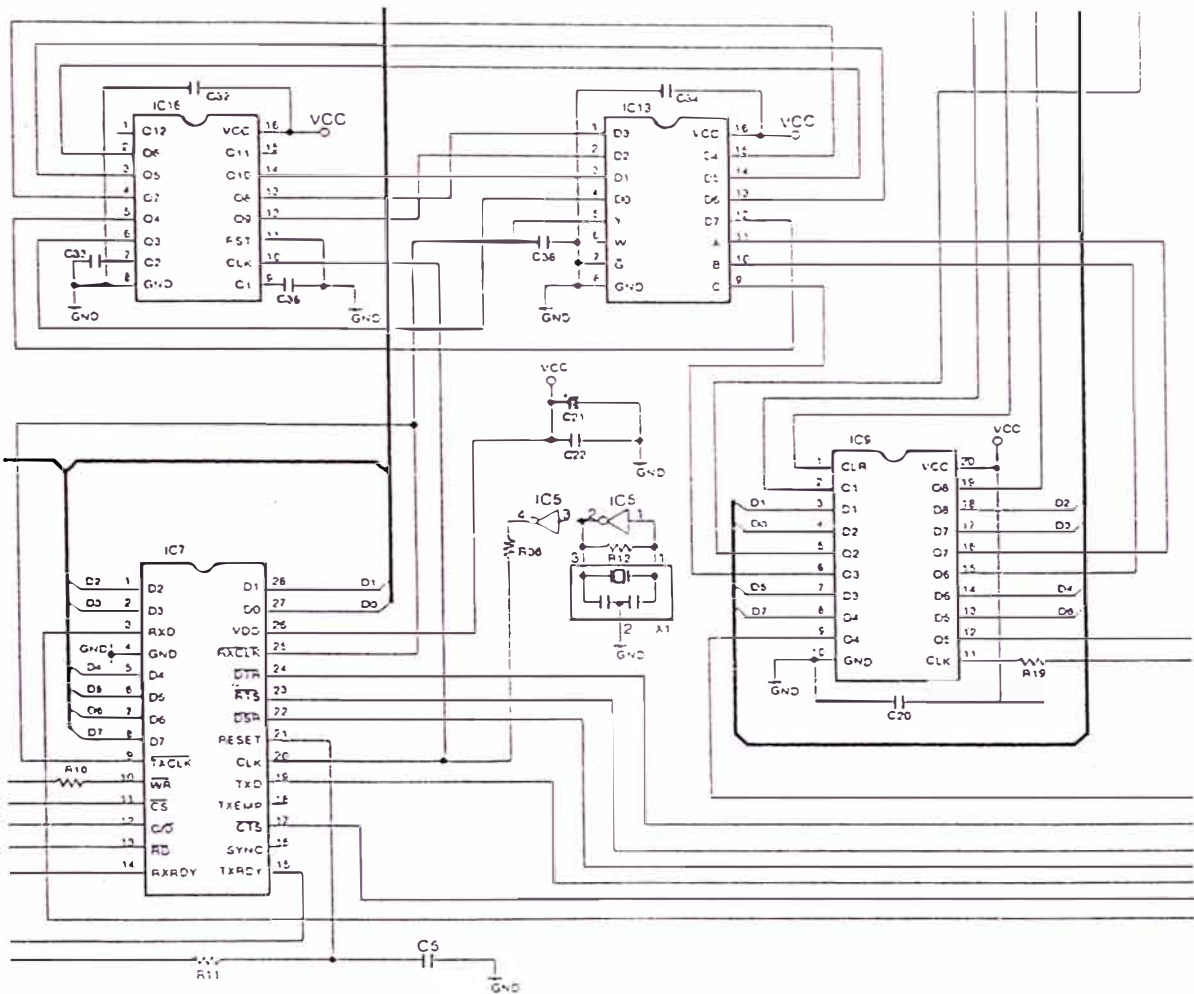
Pin description of scale connector

Pin No.	Signal Name	Descriptions
1	NC	Non connection
2	NC	Non connection
3	NC	Non connection
4	NC	Non connection
5	NC	Non connection
6	NC	Non connection
7	D1	Data
8	D2	Data
9	D3	Data
10	ENABLE	Enable signal
11	D0	Data
12	CLOCK	Clock signal
13	MOTION	Motion signal
14	RANGE	Out of range signal
15	GND	GND
16	GND	GND

3-3. On line circuit (RS-232C)

1. Baud rate selection for On line (RS-232C)

Basic clock (307.2 KHz) is divided into seven clocks by binary counter IC16 and they are supplied to D0~D6 terminals of multiplexer IC13. The IC13 outputs one clock to USART according to the condition of select signals A, B, and C.



Output of IC16

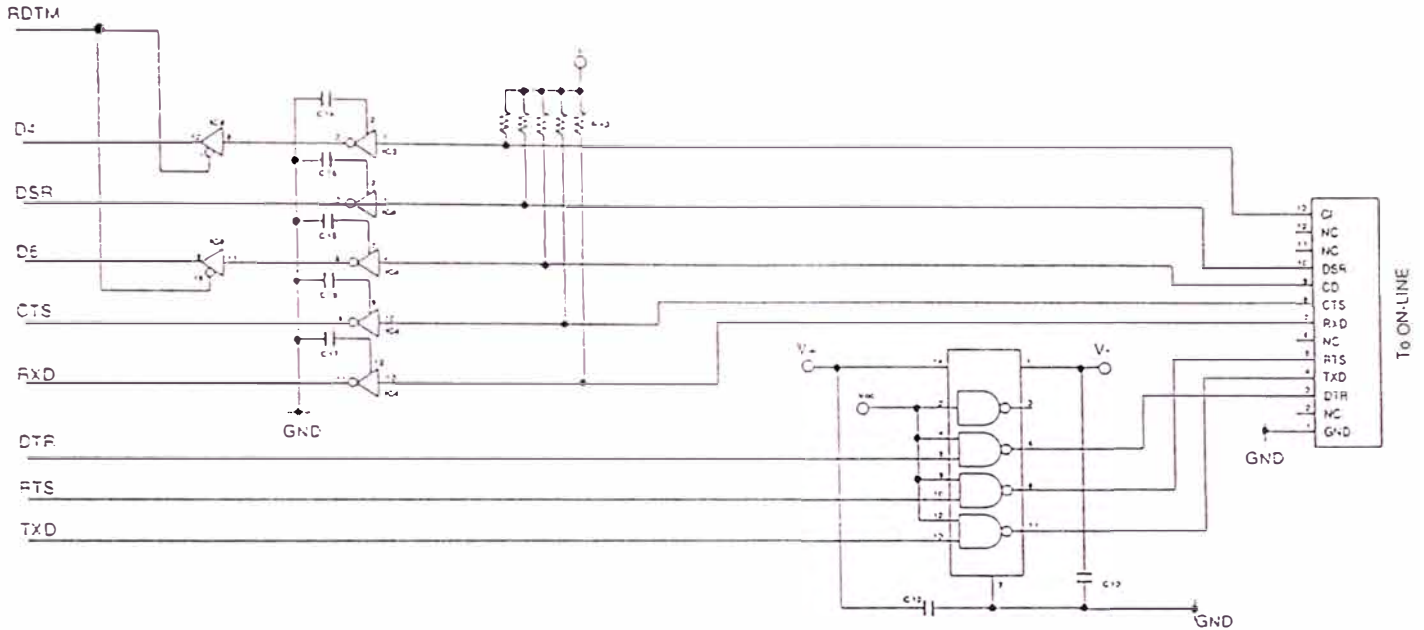
Q3	38.4 K
Q4	19.2 K
Q5	9,600
Q6	4,800
Q7	2,400
Q8	1,200
Q9	600
Q10	300

Truth table of IC13

Input			Output
A	B	C	Y
L	L	L	D0
H	L	L	D1
L	H	L	D2
H	H	L	D3
L	L	H	D4
H	L	H	D5
L	H	H	D6
H	H	H	D7

2. On line circuit (RS-232C)

The on line port (RS-232C) is controlled by the USART μ PD71051 and supported only asynchronous communication.



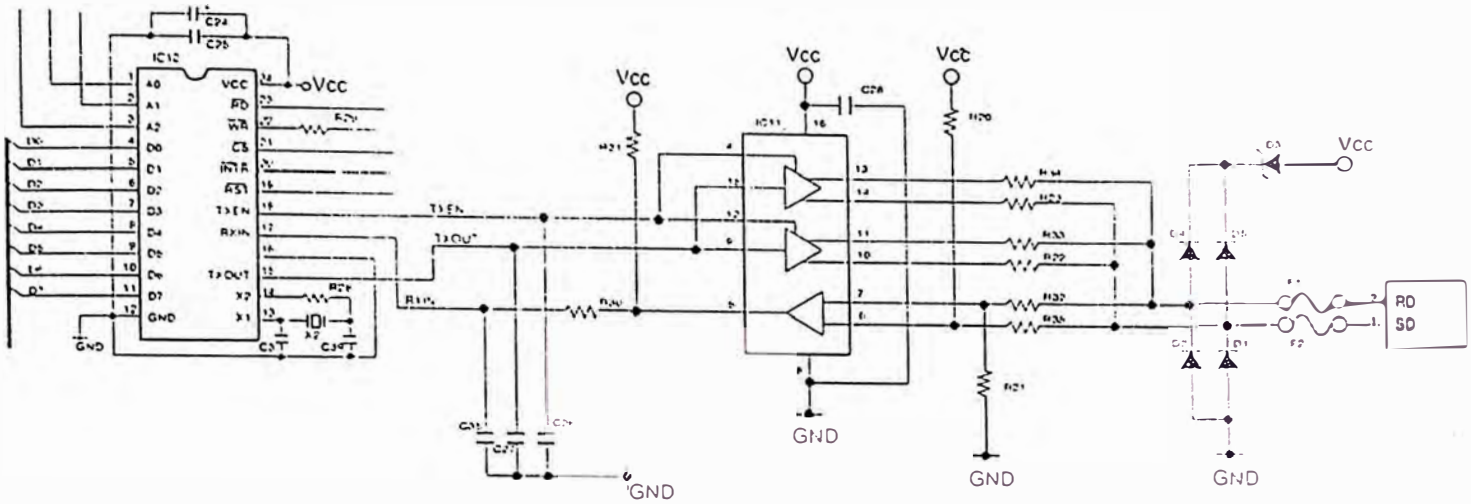
Pin description of UPD71051C

Pin No.	Signal	Description
1,2,5-8, 27,28	D0-D7	Data bus
3	RxDATA	Receive data input
9	TxCLK	Transmitter clock input
10	WR	Write signal
11	CS	Chip select signal Input
12	C/D	Control or data change
13	RD	Read signal
14	RxRDY	Receive ready output
15	TxRDY	Transmitter ready output
16	SYNC/BRK	Synchronization/Break (Not used)
17	CTS	Clear to send signal input
18	TxEMP	Transmitter empty
19	TxDATA	Transmit data output
20	CLK	Clock input
21	RESET	Reset input
22	DSR	Data set ready signal input
23	RTS	Request to send signal output
24	DTR	Data terminal ready signal output
25	RxCLK	Receiver clock input
26	VDD	VCC(+5V) terminal

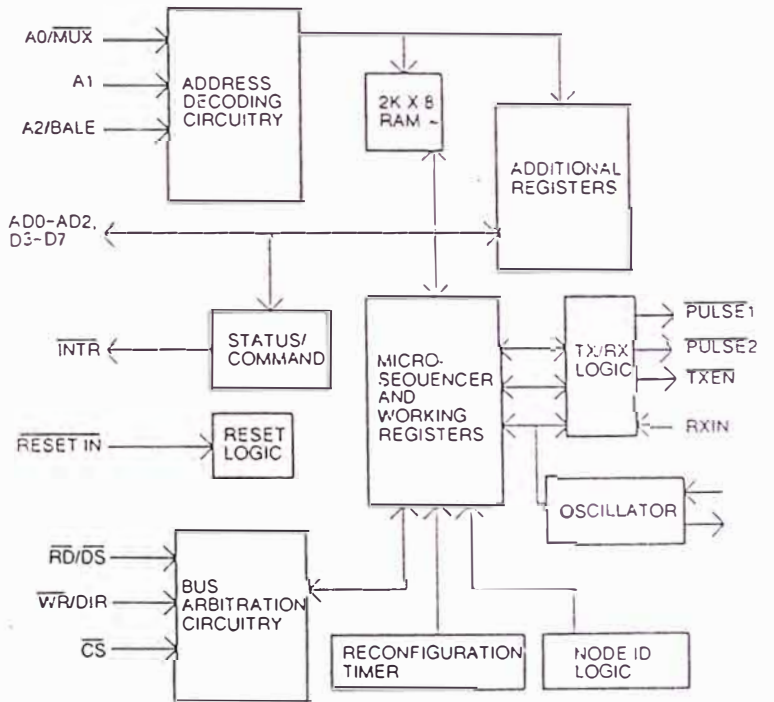
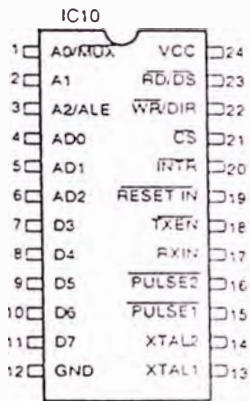
Pin description of on line connector

Pin No.	Signal	Description
1	SG	Signal ground
2	NC	Non connection
3	ER	Equipment ready
4	SD	Send data
5	RTS	Request to send
6	NC	Non connection
7	RD	Receive data
8	CTS	Clear to send
9	CD	Carrier detect
10	DR	Data ready
11	NC	No connection
12	NC	No connection
13	CI	Calling indicator

3-4. C-In line circuit



1. C-In line controller (COM20020B)



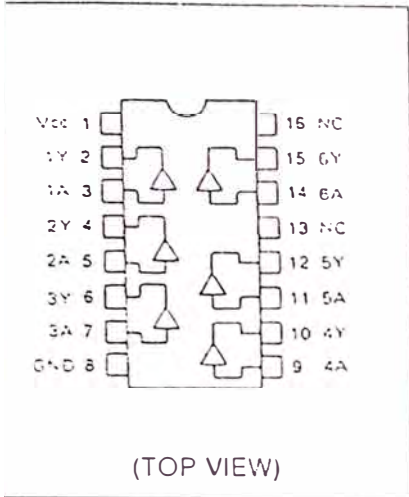
2. COM20020B pin description

Pin No.	Name	In/Out	Status of OFF	Status of ON No Token	Status of ON Token	Description
1 2 3	A0/MUX A1 A2/ALE	In In In	L L L	L L L	Pulse L L	Input. On a non-multiplexed bus, these signals are directly connected to the low bits of the host address bus. On a multiplexed address/data bus, A0/MUX is tied low, A1 is left open, and A2 is tied to the address latch enable signal of the host. A1 is connected to an internal pull-up resistor.
4 5 6 7 8 9 10 11	AD0 AD1 AD2 D3 D4 D5 D6 D7	In/Out In/Out In/Out In/Out In/Out In/Out In/Out In/Out	L L L L L L L L	Pulse Pulse Pulse Pulse Pulse Pulse Pulse Pulse	Pulse Pulse Pulse Pulse Pulse Pulse Pulse Pulse	Input/Output. On a non-multiplexed bus, these signals are used as the data lines for the device. On a multiplexed address/data bus, AD0-AD2 act as the address lines (latched by ALE) and as the low data lines for the device. D3-D7 are always used for data only. These signals are connected to internal pull-up resistors.
12	GND	Power	GND	GND	GND	Signal ground.
13 14	XTAL1 XTAL2	In In	L L	Pulse Pulse	Pulse Pulse	An external crystal should be connected to these pins. If an external TTL clock is used instead, it must be connected to XTAL1 with a 390 ohms pull-up resistor, and XTAL2 should be left floating.
15 16	<u>PULSE1</u> <u>PULSE2</u>	Out Out	L L	H L	Pulse L	Output. In normal mode, these active low signals carry the transmit data information, encoded in pulse format, from the COM20020 to the media driver circuitry. When the device is in backplane mode, the PULSE1 signal driver is programmable (push/pull or open-drain), while the PULSE2 signal provides a clock with frequency of crystal/4. PULSE1 is connected to a weak internal pull-up resistor in backplane mode.
17	RXIN	In	L	H	Pulse	Receive input. This signal carries the receive data information from the line receiver circuitry to the COM20020.
18	TXEN	Out	L	L	Pulse	Transmit Enable output. This signal used in backplane mode to enable the line drivers for transmission. The polarity of the signal is programmable by grounding the PULSE2 pin. This option is valid only in backplane mode.
19	RESET IN	In	L	H	H	Input. This active low signal issued by the microcontroller executes a hardware reset. It is used to activate the internal reset circuitry within the COM20020.

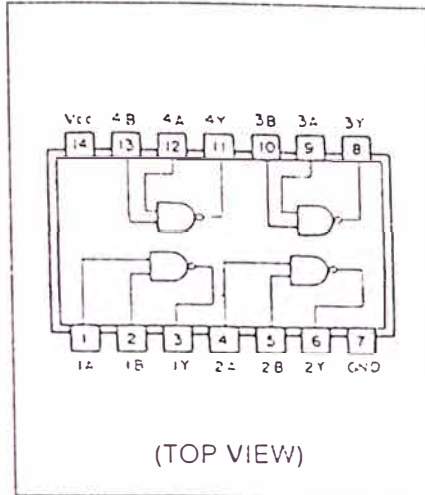
Pin No.	Name	In/Out	Status of OFF	Status of ON No token	Status of ON Token	Description
20	INTR	Out	L	H	Pulse	Interrupt output. This active low signal is generated by the COM20020 when an enabled interrupt condition occurs. INTR returns to its inactive state when the interrupt status condition or the corresponding interrupt mask bit is reset.
21	CS	In	L	H	Pulse	Chip select input. This active low signal issued by the microcontroller selects the COM20020 for an access.
22	WR/DIR	In	L	Pulse	Pulse	Input. On a 68XX-like bus, this signal is issued by the microcontroller as the Read/Write signal to determine the direction of data transfer. In this case, a logic "1" selects a read operation, while a logic "0" selects a write operation. In this case, data is actually strobed by the DS signal. On an 80XX-like bus, this active low signal is issued by the microcontroller to indicate a write operation. In this case, a logic "0" on this pin, when the COM20020 is accessed, enables data from the data bus to be written to the device.
23	RD/DS	In	L	L	Pulse -	Read/ Data strobe signal input. On a 68XX-like bus, this active low signal is issued by the microcontroller as the data strobe signal to strobe the data onto the bus. On a 80XX-like bus, this active low signal is issued by the microcontroller to indicate a read operation. In this case, a logic "0" on this pin, when the COM20020 is accessed, enables data from the device to the data bus to be read by the microcontroller.
24	VCC	Power	GND	+5V	+5V	Power supply +5V.

IC DATA

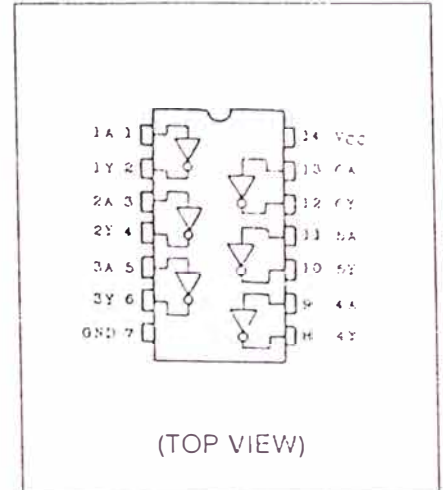
TC74HC4050AP



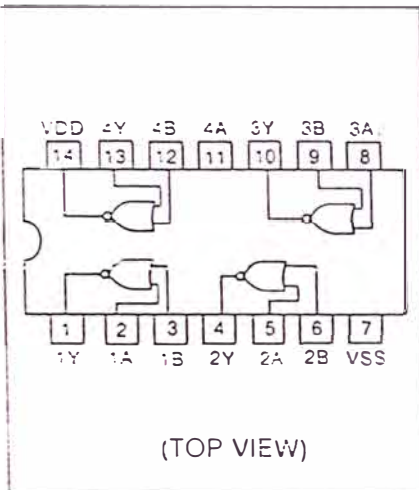
TC74HC00AP
QUAD 2-INPUT NAND GATE



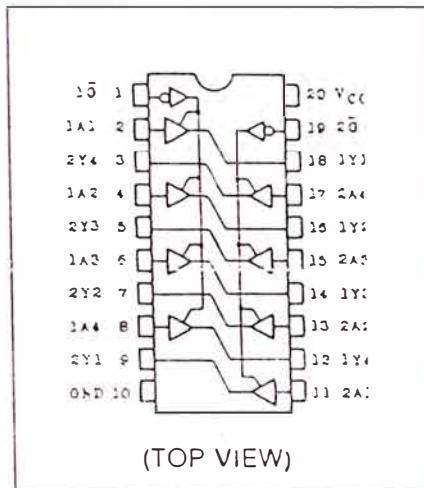
TC74HCU04AP
HEX INVERTER



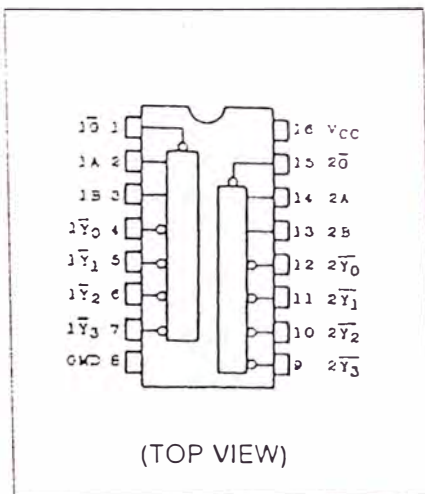
TC74HC02AP



TC74HC244AP



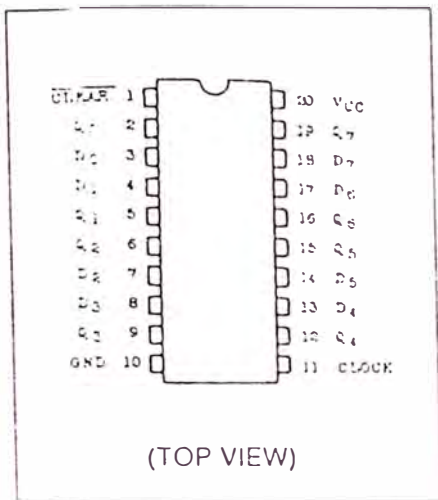
TC74HC139AP



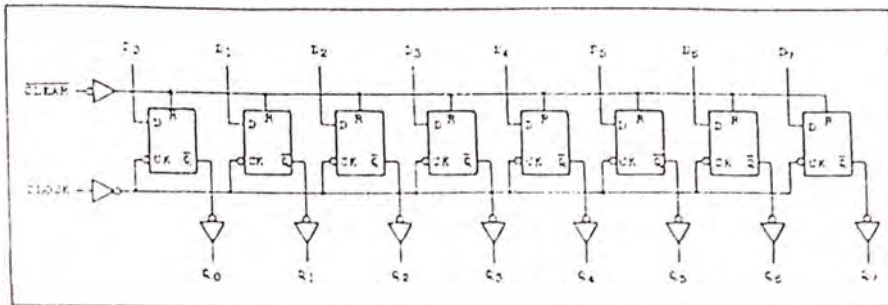
ENABLE	INPUTS		OUTPUTS				SELECTED OUTPUT
	SELECT		\bar{Y}_0	\bar{Y}_1	\bar{Y}_2	\bar{Y}_3	
\bar{G}	B	A	\bar{Y}_0	\bar{Y}_1	\bar{Y}_2	\bar{Y}_3	SELECTED OUTPUT
H	X	X	H	H	H	H	NONE
L	L	L	L	H	H	H	\bar{Y}_0
L	L	H	H	L	H	H	\bar{Y}_1
L	H	L	H	H	L	H	\bar{Y}_2
L	H	H	H	H	H	L	\bar{Y}_3

X: Don't care

TC74HC273AP
OCTAL D-TYPE FLIP FLOP WITH CLEAR

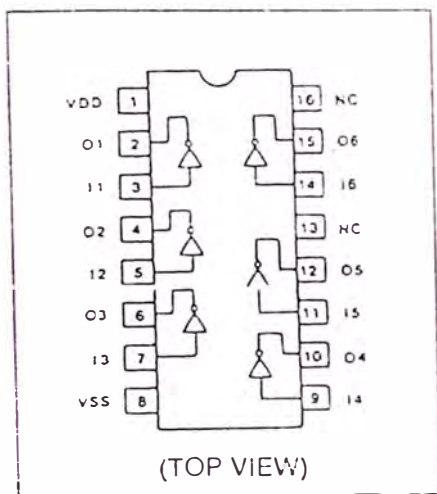


LOGIC DIAGRAM

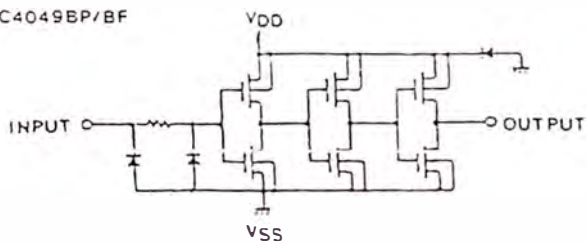


INPUTS		OUTPUT		FUNCTION
CLEAR	D	CLOCK	Q	
L	X	X	L	Clear
H	L		L	-
H	H		H	-
H	X		Qn	No change

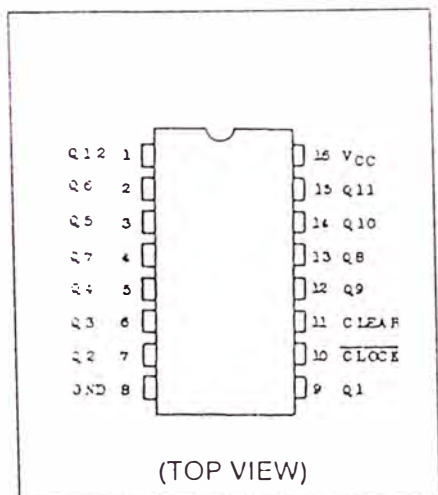
TC74HC4049AP



1/6 TC4049BP/BF



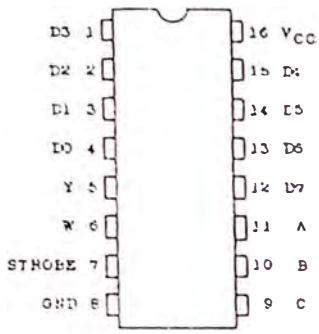
TC74HC4040AP



CLOCK	CLEAR	OUTPUT STATE
X	H	ALL CUPUTS = "L"
	L	NO CHANGE
	L	ADVANCE TO NEXT STATE

X : DONT CARE

TC74HC151AP

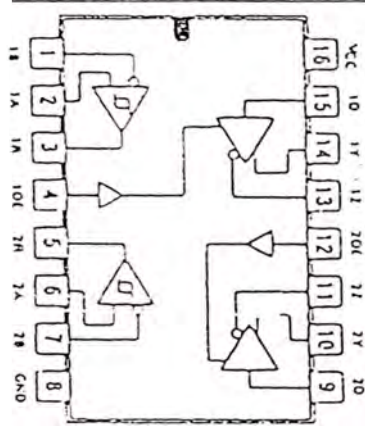


(TOP VIEW)

INPUTS				OUTPUTS	
SELECT			STROBE	W	Y
	B	A			
X	X	X	H	L	H
L	L	L	L	D0	D0
L	L	H	L	D1	D1
L	H	L	L	D2	D2
L	H	H	L	D3	D3
H	L	L	L	D4	D4
H	L	H	L	D5	D5
H	H	L	L	D6	D6
H	H	H	L	D7	D7

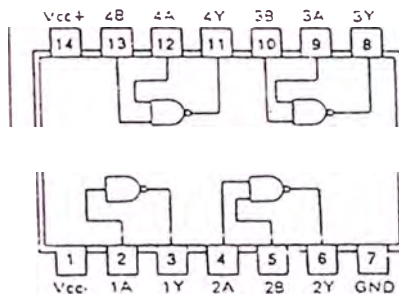
X: Don't care

SN751178N



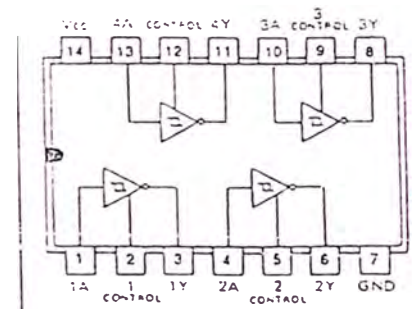
(TOP VIEW)

SN75188N



(TOP VIEW)

SN75189AN



(TOP VIEW)

7. PARTS LIST

Item	Ccode No.	Parts Name	Specification	Version
IC7	2002 1330	I/O-PB-11 ass'y	E311890-1	
IC10	2006 0364	LSI	UPD71051C	
D3	2310 9409	Zenner diode	COM20020BIP	
X1	2520 1818	Ceramic oscillator	RD5.1FB3	
X2	2590 1197	Ceramic oscillator	CST4.91MGW	
RA2	2720 4376	Module resistor	CSA20.00MXZ01C	
RA1	2730 C023	Module resistor	MS1035F	
CA1	2845 0182	Module capacitor	MS1036F	
CN5	3500 1121	PCB connector	CNB6X101K	
CN3	3540 3918	Connector	IMSA-9603S-16C	
CN2	3540 4279	Connector	HiF3FC-16PA-DSA	
CN4	3540 4731	PCB connector	B02P-XL	
CN1	3612 0327	Connector	IMSA-9603C-24C	
F1,F2	3630 2344	Time lag fuse	TCS5044-01-401	
F1,F2	3640 2331	Fuse holder	TSC-0.5A	
IC17	2101 0162	MOS IC	UF-0033	
IC1	2101 0243	MOS IC	TC74HC139AP	
IC13	2101 0774	C-MOS IC	TC74HC4049AP	
IC9	2101 0782	C-MOS IC	TC74HC151AP	
IC16	2101 0791	C-MOS IC	TC74HC273AP	
IC8	2101 1037	MOS IC	TC74HC4040AP	
IC14	2105 1008	C-MOS IC	TC74HC244AP	
IC12	2105 1035	C-MOS IC	TC74HC00AP	
IC5	2105 1260	C-MOS IC	TC74HC02AP	
IC4	2105 1463	MOS IC	TC74HCU04AP	
IC2	2111 2275	Bipolar IC	TC74HC4050AP	
IC11	2111 6238	Bipolar IC	SN75188N	
IC3,IC6	2113 0217	Bipolar IC	SN751176N	
D1,2,4,5	2315 2619	Diode	SN75109AN	
R25,26	2614 0013	Carbon film resistor	1SS142-T-77-T	
R20,21	2614 0043	Carbon film resistor	R-25-10K-J-T24-T	
R1~11,19,24, 27~30,35	2614 0145	Carbon film resistor	R-25-15K-J-T24-T	
R13~18,31, 32,35	2614 0234	Carbon film resistor	R-25-100-J-T24-T	
R12	2614 0242	Carbon film resistor	R-25-1K-J-T24-T	
R22,23,33,34	2614 0706	Carbon film resistor	R-25-1M-J-T24-T	
C6,21,24	2802 9931	Electrolytic capacitor	R-25-10-J-T24-T	
C7	2807 1523	Electrolytic capacitor	16RE3-100-T2-T	
C2,5,10	2818 0365	Ceramic capacitor	35RE3-100-T2-T	
C4,14~19,26, 27,30,35,37	2818 0446	Ceramic capacitor	RT-HE50TKYB100-T	
C3,36	2818 3203	Ceramic capacitor	RT-HE40TKYB100-T	
C1,22,25	2820 3093	TF capacitor	RT-HE50TKCH330-T	
C12,13,20,28, 31,32,34	2825 0364	TF capacitor	ECO-V1H-104-J22-T	
	4303 0976	PCB E-I/O-PB-11	ECO-B1H103JF3-T	
		Others	E211627-1	
	6231 9707	FG wire sub ass'y A	E310996A-15	
	6245 5330	FFC joiner G-PB-9	E411511B-7	
	6245 5340	FFC joiner H-PB-9	E411511B-8	

	Item	Code No.	Parts Name	Specification	Version
		3540 3924	IN-LINE connector kit	XLP-KIT-2	
		5111 1648	Screw	3X8 ZMC-3	
		5112 0124	Screw with washer	3X8 ZMC-3.....	
		5550 1459	Locking spacer	SPLS-10	
		6221 4285	Earth metal sub ass'y	E411552-1	
		6246 0740	Name sheet E/O11	E411873-18	
		2800 2516	Carbon film resistor	R-25-100-J	

BIBLIOGRAFIA

1. CAMPBELL, Joe. Comunicaciones Serie. Madrid Ed. Anaya Multimedia. 1991
2. CAMPBELL, Joe. The RS-232 Solution. 2da. ed. EE.UU. Ed. Sybex Inc. 1989.
3. CASIO COMPUTER CO, LTD. I/O-PB-11 Interface Board. Dealer's Manual. Tokyo 1995
4. CASIO COMPUTER CO, LTD. Electronic Cash Register TK-2300. Service Manual. Tokyo 1995
5. CASIO COMPUTER CO, LTD. Caja Registradora Electrónica TK-2300. Manual de Programación. Tokyo 1995
6. CASIO COMPUTER CO, LTD. Caja Registradora Electrónica TK-2300. Manual de Instrucciones. Tokyo 1995
7. GOODWIN, Mark. Serial Communications in C and C++. EE.UU. Ed. MIS:Press. 1992
8. HOLMES, Michael and FLANDERS, Bob. C++ Communications Utilities. EE.UU. Ed. Ziff-Daves Press. 1993
9. RODRIGUEZ ROSELLO, Miguel Angel. 8088-8086/8087 Programacion Ensamblador en entorno MS DOS. Madrid Ed. Anaya Multimedia. 1988