

ESCUELA POLITECNICA NACIONAL

FACULTAD DE INGENIERIA ELECTRICA

TESIS DE GRADO

LECTOR DE CODIGO DE BARRAS

Tesis previa la obtención del título de  
Ingeniero en Electrónica y Telecomunicaciones

1.989

Mi sincero agradecimiento a mis Padres,  
mi familia y al Area de Sistemas Digitales  
quienes impulsaron y contribuyeron para  
la culminación de mi carrera profesional.

## Certificación

Certifico que el presente trabajo ha sido desarrollado en su totalidad por el Sr. Ramiro Eduardo Morejón Tobar.



---

Ing. Galo F. Valencia P.

## INDICE

CAPITULO I		Pág
1.1	Consideraciones generales sobre código de barras	2
1.1.1	Introducción	2
1.1.2	Técnicas de identificación automática	3
1.1.3	Ventajas de los códigos de barras	4
1.1.4	Aplicaciones	5
1.1.5	Usuarios típicos de códigos de barras	6
1.1.6	Códigos de barras	8
1.1.7	Scanners	11
1.2	Los emisores y detectores ópticos	14
1.3	Descripción general del sistema	19
1.3.1	Módulo de impresión de etiquetas	19
1.3.2	Módulo de comunicación y decodificación	19
1.3.3	Módulo generador y editor de códigos	20
1.3.4	Módulo principal	20
1.3.5	Circuito lector	21
1.3.6	Procesador de información del código impreso en la etiqueta	22
1.4	Software de procesamiento de la información	22

## CAPITULO II

Pág

2.1.	Generación de código de barras	25
2.1.1.	Características del código	25
2.1.2.	Número de caracteres	25
2.1.3.	Tolerancia del sistema	26
2.1.4.	Densidad de información	28
2.1.4.1	Desarrollado del código 3 de 8	37
2.2.	Programa de generación de código de barras	39
2.2.1.	Tabla de equivalencias	40
2.2.2.	Ejemplo de tabla de equivalencias editada utilizando el programa BSGEN.EXE	41
2.2.3.	Parámetros de entrada de BSGEN.EXE	43
2.2.4.	Sintaxis	45
2.3.	Programa de impresión de etiquetas	45
2.3.1.	Modo gráfico de un impresor matricial	45
2.3.2.	Impresión de una etiqueta	47
2.3.3.	Parámetros de entrada de BSIMP.EXE	48
2.4.	Ejemplos de generación de código de barras	51

## CAPITULO III

3.1.	Principio de funcionamiento de un lector óptico	54
3.2.	Descripción del pórtico de comunicaciones con el computador	62

	<b>Pág</b>	
3.3.	Diseño del lector de código de barras	64
3.3.1.	Lápiz lector	64
3.3.2.	Circuito digital	69
3.3.3.	Rutinas del microcontrolador	71
3.4.	Construcción del lector	95
3.4.1.	Diagrama físico	96
3.4.2.	Lista de componentes	97
3.4.3.	Características eléctricas	97
3.4.4.	Diagrama esquemático	98

#### CAPITULO IV

4.1.	Descripción de las rutinas de comunicaciones	100
4.1.1.	Rutina de recepción	100
4.1.2.	Rutina de transmisión	100
4.2.	Software de comunicaciones para un computador PC-XT	102
4.2.1.	Instrucciones utilizadas en comunicaciones	102
4.2.2.	Inicialización del pórtico serial	104
4.2.3.	Rutina de comunicación	105
4.2.4.	Protocolo de comunicaciones	107
4.3.	Identificación de códigos en el computador	109

## CAPITULO V

Pág

5.1	Pruebas de funcionamiento	115
5.1.1	Prueba #1	115
5.1.1.1	Objetivo	115
5.1.1.2	Consideraciones	115
5.1.1.3	Errores tipificados	116
5.1.1.4	Resultados	118
5.1.2	Prueba #2	120
5.1.2.1	Objetivo	120
5.1.2.2	Consideraciones	121
5.2	Análisis de resultados	128
5.3	Análisis económico	130
5.4	Conclusiones y comentarios	134

## APENDICES

A.	MODO GRAFICO DEL IMPRESOR
B.	REGISTROS DE CONTROL DEL INTEL 8031
C.	INSTRUCCIONES PEEK, POKE DEL BASIC
D.	COMUNICACION SERIAL CON QUICK BASIC 4.0
E.	CARACTERISTICAS FISICAS DEL PORTICO SERIAL
F.	DIAGRAMAS DE FLUJO Y LISTADO DE PROGRAMAS
G.	DATOS TECNICOS DE LOS COMPONENTES
H.	MANUAL DE USO

## **CAPITULO I**

- CONSIDERACIONES GENERALES SOBRE CODIGOS DE BARRAS
- LOS EMISORES Y DETECTORES OPTICOS
- DESCRIPCION GENERAL DEL SISTEMA
- DESCRIPCION DEL SOFTWARE PARA EL PROCESAMIENTO  
DE LA INFORMACION



## 1.1. CONSIDERACIONES GENERALES SOBRE CODIGO DE BARRAS

### 1.1.1. INTRODUCCION

El creciente aumento de los volúmenes de información para ser procesados y la rapidez con la que los datos deben manipularse, plantea la necesidad de implantar sistemas de procesamiento de datos. El desarrollo tecnológico ha permitido en este campo realizar progresos acordes a las necesidades de la época, sin embargo, uno de los mayores limitantes para alcanzar una mayor velocidad en la realización de procesos, como la identificación de personal, los sistemas de inventarios, la facturación de productos de consumo masivo, etc; es el ingreso de los datos que requieren un procesamiento en aplicaciones específicas.

Con el propósito de resolver este problema se han desarrollado diversos sistemas de identificación automática, que básicamente tratan de reproducir la versatilidad de los sentidos del hombre, las sensaciones visuales, auditivas, térmicas, táctiles, etc; buscando aquellas que se adapten más fácilmente a los propósitos de la automatización o adquisición de datos y haciéndolas compatibles con equipos de procesamiento, de modo que el ingreso de datos no

represente el mayor costo del tiempo de proceso y un extenuante trabajo para los operadores y usuarios del sistema.

#### 1.1.2. TECNICAS DE IDENTIFICACION AUTOMATICA

Existen diferentes técnicas. Algunas de ellas utilizan representaciones discretas que por su misma naturaleza son fácilmente adaptables a los sistemas digitales de procesamiento de datos. Entre ellas se puede citar:

- |   |                                    |
|---|------------------------------------|
| - Código de barras.                         | - "Bar Codes"                      |
| - Tarjetas inteligentes.                    | - "Smart Cards"                    |
| - Cinta magnética.                          | - "Magnetic Stripe"                |
| - Identificación mediante radio frecuencia. | - "Radio Frequency Identification" |

Otras en las cuales la tarea de adaptación es muy compleja, la información captada debe ser previamente analizada para extraer de ella sus componentes, los mismos que serán comparados con patrones conocidos para establecer ordenes de similitud, antes que pueda ser utilizada. Algunos de estos sistemas son:

- Reconocimiento de caracteres ópticos.
- Máquina para visión.
- Introducción de datos mediante voz.
- "Optical Character Recognition"
- "Machine Vision"
- "Voice Data Entry"

El presente trabajo diseña un sistema completo de procesamiento de datos en el que se utiliza la técnica de código de barras.

#### 1.1.3. VENTAJAS DE LOS CODIGOS DE BARRAS:

- Seguimiento de la información con precisión.
- Bajo costo de impresión y reproducción.
- Capacidad de suministrar datos para bases de datos en línea.
- Rapidez de la recuperación de la información.
- Precisión de los datos.
- Reducción de trabajo de manejo de materiales.
- Minimización en existencia de inventarios.
- Mejora de servicios a los clientes.
- Actualizar información de los inventarios de activos fijos y propiedades.
- Reducir el tiempo en trabajos de proceso.

ancho, de manera similar a las técnicas de modulación por ancho de pulsos, con la particularidad que la codificación se encuentra impresa.

Una secuencia de barras y espacios se conoce como "etiqueta" y a ésta se asocia un único caracter, comando u otro tipo de información, dando origen a una tabla de codificación. Los códigos no necesariamente deben regirse por reglas o algoritmos de generación, no obstante la decodificación y detección de errores se simplifica cuando las barras cumplen con normas pre-establecidas. En una etiqueta el ancho de las barras es el factor que las diferencia. Por tanto el número de dígitos posibles para una representación es igual a la gama de anchos para las barras. Las barras deben tener una representación binaria asociada. De esta manera será posible procesarlas utilizando sistemas digitales.

Las características de un código de barras debe satisfacer necesidades de diversos usuarios en múltiples aplicaciones, por ello distintos fabricantes de sistemas de este tipo desarrollan sus propios códigos para cumplir con especificaciones de seguridad de información, dimensiones de las etiquetas, etc . Los códigos existente se clasifican tomando en cuenta sus características:

#### 1.1.6.1. EN FUNCION DE LOS CARACTERES

- Numéricos, son aquellos en que el set de caracteres abarca única y exclusivamente los dígitos del 0 - 9, con las secuencias de inicio y fin. Como ejemplo, el código 2 de 5 utilizado por la Hewlett Packard.
  
- Numéricos con algunos caracteres.
  
- Alfanuméricos, en los que se han representado todos los caracteres del alfabeto.

#### 1.1.6.2. EN FUNCION DEL TIPO DE CODIFICACION

- Por el ancho, en el cual los bits representados pueden ser barras o espacios y lo que los diferencia es su ancho, es decir que los ceros "0" están representados por barras o espacios delgados y los unos "1" por barras o espacios anchos, guardando una relación de magnitud fija que puede ser 2 a 1 o de 3 a 1, dependiendo de la aplicación.
  
- No retorno a cero en el que se asignan a las barras el dígito binario uno "1", correspondiendo a los espacios el dígito cero "0", las barras y espacios son del mismo ancho y

la etiqueta puede tener varias barras consecutivas o espacios. El código debe evitar secuencia prolongadas de barras o espacios, llamadas "no permitidas", tal que sea posible detectar los extremos de una etiqueta y recuperar el valor binario, sirviéndose de las marcas de guarda que son barras utilizadas como referencia de las dimensiones. Un ejemplo de este tipo es el código UPC (Universal Product Code).

#### 1.1.6.3. EN FUNCION DE LA DENSIDAD

- Alta densidad.
- Media densidad.
- Baja densidad.

La clasificación en función de la densidad toma en cuenta el número de caracteres que pueden ser impresos por unidad de longitud y los parámetros que debe considerarse son: el número de barras, los espacios necesarios por cada caracter, el ancho mínimo de las barras. Este último dependerá del desarrollo tecnológico en los dispositivos de impresión y lectura.

Puesto que un dispositivo de lectura comercial puede detectar barras de hasta 0.3 mm como mínimo, la densidad

dependerá del número de barras por caracter y del tipo de codificación, alcanzándose densidades de 3.6, 2.3 y 2 caracteres por centímetro.

#### 1.1.6.4. CARACTERISTICAS

En general las características más importantes que debe cumplir un código de barras son:

- Amplio conjunto de caracteres.
- Amplios límites de tolerancia para impresión como para la decodificación.
- Alta densidad de información.
- Capacidad de autoverificación.
- Bidireccionalidad.
- Mínimo número de barras por caracter.
- Seguridad y confiabilidad de las lecturas.

Por ende los códigos de barras deben llegar a un compromiso dependiendo de la aplicación, las limitaciones del hardware, las técnicas de decodificación y en muchos casos teniendo que sacrificar algunas de sus características.

### 1.1.7. SCANNERS

Dispositivos electrónicos y ópticos que barren los códigos de barras y son sensibles a la diferencia de la reflexión (intensidad de luz) entre barras y espacios.

#### 1.1.7.1. SCANNERS ESTACIONARIOS

- No requieren contacto con el material a ser leído, puesto que el símbolo que se desea leer es movido frente al rayo de luz del dispositivo.

#### 1.1.7.2. SCANNERS DE MANO

- Pueden o no requerir contacto con el material a ser leído.
- Los de lápiz lumínico (light pen) requieren contacto.
- Los de pistola láser (led) no requieren contacto.

La detección de las barras o espacios se consigue utilizando técnicas ópticas que valiéndose del diferente coeficiente de reflexión de los colores permiten diferenciar un nivel de intensidad de luz reflejada diferente para cada color ante una misma fuente de luz. El fotodetector al pasar sobre las regiones claras y oscuras presentará respuestas diferenciables con relativa facilidad, sin perder de vista



que la detección involucra otros problemas que requieren una solución tecnológica de alto nivel.

La luz ambiental es un factor que dificulta el proceso en forma significativa, sea esta de emisión natural o artificial. De ésta última existen muchos factores que la alteran como son: las variaciones de tensión de alimentación de la fuente producen cambios de intensidad y peor aun los emisores alimentados con corriente alterna en los que las variaciones dependen de la frecuencia de la alimentación, cambios imperceptibles para el hombre no así para un detector muy sensible y de alta ganancia como lo es un detector óptico. Las sombras producidas por reflexión producen variaciones y el movimiento realizado durante la exploración producirá sombras dependiendo de la posición relativa de la fuente y el detector. La presencia de la luz solar ocasiona también dificultades por el efecto de las sombras. Todas las variaciones cuyo origen está en las fuentes de luz son aleatorias y por tanto deben ser minimizadas, haciendose necesario contar con un dispositivo de lectura que incorpore tanto al emisor como al receptor.

Otro problema de la detección es la resolución del dispositivo de lectura, directamente asociado con la construcción física, pues la respuesta de reflexión de la

superficie con las barras será la suma de la respuesta de todos los puntos sobre los cuales incide el haz. Esto no es un problema cuando toda la superficie es del mismo color, pero si lo es cuando se acerca a los bordes en los que la respuesta varía conforme el campo de acción del lector, cuyo diámetro "d", pasa de un color a otro como se ve en la FIGURA 1.1, en que la señal resultante es proporcional a las áreas expuestas de uno y otro color. Donde "Id" es la corriente en oscuridad del detector.

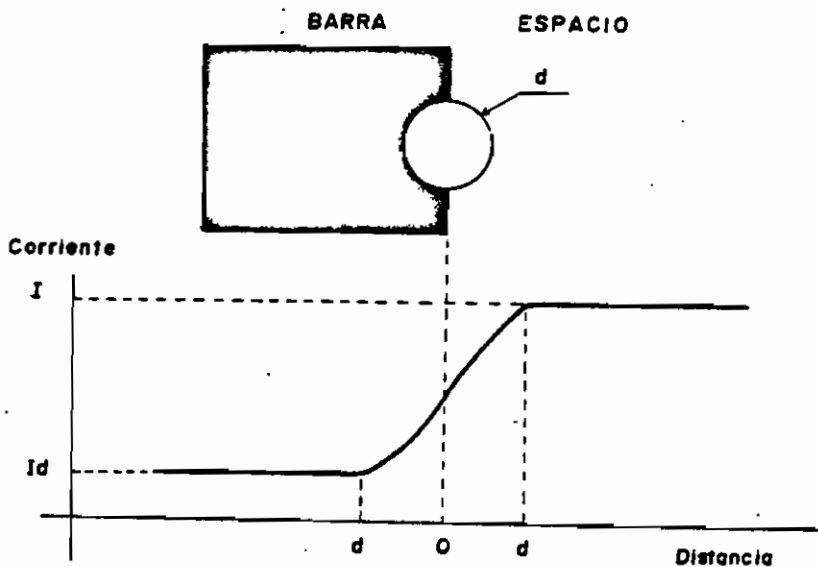


Fig 1.1. VARIACION DE LA CORRIENTE DEL DETECTOR.

Otro factor intimamente relacionado con la resolución es la velocidad de exploración, si el detector tiene una pobre

resolución y la exploración se efectúa muy rápidamente, algunas barras no pueden ser detectadas; siendo posible que la contribución de la barra no sea significativa en el conjunto, ya que el receptor considera la reflexión de todos los puntos expuestos.

## 1.2. LOS EMISORES Y DETECTORES OPTICOS.

El sistema de detección está constituido fundamentalmente por un emisor y un detector óptico cuyas características están determinadas por la técnica utilizada para la exploración de la etiqueta y son:

Haz móvil .- Esta técnica realiza la exploración de la etiqueta mediante desplazamientos sucesivos del haz. Estos desplazamientos se consiguen con sistemas mecánicos de manera que el haz recorra a lo largo de una línea recta que debe atravesar transversalmente las barras de la etiqueta, técnica conocida como "exploración unidireccional", alcanzando entre 50 y 700 barridos por segundo, a distancias entre 25 y 75 centímetros, utilizando emisores láser.

Otra alternativa para esta técnica es cambiar la dirección con cada exploración, de esta manera la etiqueta no tendrá que ser expuesta en una dirección predefinida sino que el

sistema se encargará de tomar aquella exploración en la que fue barrida transversalmente. Es evidente que no es necesario considerar todas las direcciones posibles para una exploración, pues para pequeños desplazamientos las variaciones son mínimas.

En sistemas prácticos se los hace a 0, 45 y 90 grados y se los conoce como de "exploración omnidireccional". Los errores de lectura con esta técnica se reducen, al poder contar con varias exploraciones en una misma dirección, siendo posible verificar dos consecutivas para detectarlas. Sin embargo la mayor dificultad radica en el sistema mecánico y óptico que permite desplazar el haz.

Haz fijo .- Los sistemas que utilizan un haz fijo son aquellos donde la exploración puede conseguirse desplazando la etiqueta frente al detector y es posible contar un sistema mecánico que realice esta labor, así se garantizaría que la velocidad sea constante en forma similar a los lectores de tarjetas magnéticas.

Otra variedad de este sistema consiste en mantener fija la etiqueta y desplazar el haz transversalmente sobre ésta, de forma similar a los sistemas de haz móvil, con la particularidad que la exploración se realiza una sola vez y

puede ser conseguida por medios mecánicos o en forma manual. De estos, los de desplazamiento manual, por la diferente velocidad a la que se exploran los puntos de una etiqueta, presentan las mayores dificultades para la decodificación. En lo referente a los sistemas de exploración que utilizan técnicas de haz fijo o móvil con la velocidad de exploración constante, la decodificación es mucho más confiable y más aun en aquellos que realizan varias exploraciones consecutivas de una misma etiqueta.

Todos los sistemas antes mencionados requieren que la resolución del detector sea lo más alta posible, para lo cual el haz debe ser muy fino, consiguiéndose con la utilización de un láser que permite alcanzar grandes distancias y limitando el alcance del sistema a la capacidad del receptor. Otra alternativa es enfocar el haz mediante lentes biconvexas, en cuyo caso la limitación será la distancia focal y la resolución se verá afectada si la etiqueta pasa a una distancia diferente a la focal.

Otros aspectos que deben ser tomados en cuenta para la elección del tipo de emisor de luz que ha escogerse son:

- Prolongada vida útil
- Alta eficiencia lumínica

- Bajo consumo de potencia
- Reducido ángulo de emisión
- Brillo
- Bajo costo

Deben hacerse consideraciones relacionadas con la aplicación y facilidades que dará al sistema. Por ejemplo un supermercado tendrá una variedad de productos, en los que las etiquetas se encuentran sobre superficies no necesariamente planas y direcciones aleatorias, esto obliga a tener un sistema de detección del tipo de haz móvil omnidireccional que utilice un láser de baja potencia como emisor. Pero en una biblioteca o librería todos los artículos tiene una superficie plana y las etiquetas pueden ubicarse en sitios predefinidos por los editores o bibliotecarios. Aun cuando es posible utilizar un sistema igual al del ejemplo anterior, puede reducirse el costo utilizando un sistema de haz fijo con desplazamiento manual.

Puesto que el principio de funcionamiento de un lector de código de barras es la diferencia de reflexión de espacios y barras que forman la etiqueta, los emisores y detectores de luz tiene mucha importancia en las consideraciones de diseño.

Existen varios tipos de emisores. Los más populares son aquellos contruidos utilizando la tecnología de los semiconductores. Los diodos emisores de luz LED por su bajo costo de fabricación y sus características físicas y eléctricas los hacen muy versátiles en múltiples aplicaciones.

La luminiscencia de los dispositivos semiconductores se origina en la recombinación de electrones y huecos con una radiación de energía lumínica cuando la juntura PN es polarizada directamente y esta radiación es generalmente lineal respecto a la corriente aplicada, con longitudes de onda que pueden estar en el rango de la luz visible entre 300 a 400 nano metros e infrarroja cercana al rango de 750 a 950 nano metros, dependiendo de las características de los elementos utilizados en la difusión del diodo. Longitudes más pequeñas se consiguen con semiconductores a base de arseniuro de galio Ga-As con longitudes de onda de 0.5 a 30 micro metros.

### 1.3. DESCRIPCION GENERAL DEL SISTEMA

El sistema está constituido por módulos que cumplen funciones específicas como son: el manejo del impresor, la comunicación serial con el equipo lector de barras y la decodificación de la información recuperada desde una

etiqueta, la creación o edición de las palabras código utilizadas. Estos módulos están enlazados por uno principal, el cual se encarga de transferir los parámetros requeridos por los módulos antes mencionados y de administrar una base de datos como ejemplo de aplicación.

### 1.3.1. MODULO DE IMPRESION DE ETIQUETAS.

Este módulo envía los comandos del modo gráfico apropiados al impresor seleccionado, realizando la impresión de las etiquetas con las dimensiones físicas así como el número de copias requeridas. Esta rutina es descrita con más detalle en el segundo capítulo.

### 1.3.2. MODULO DE COMUNICACION Y DECODIFICACION.

El módulo de comunicación administra el intercambio de información entre el computador y el lector utilizando el pórtico serial, en el que el protocolo empleado intercambia un byte a la vez de la siguiente forma: El módulo envía un byte que contiene exclusivamente comandos predefinidos, esperando del lector una respuesta adecuada. De acuerdo a la información recibida, si es interpretada como un comando puede desencadenar mensajes de error; o reconocido como byte que contienen en unos "1" y ceros "0" el código



binario correspondiente a cada caracter. El lector no enviará ninguna respuesta en caso que el comando recibido desencadene funciones que no requieren respuesta específica.

La información relativa a las lecturas es procesada y usada como índice en el archivo que contiene las palabras código; archivo que se utilizó como fuente para la impresión de las etiquetas, recuperando el caracter correspondiente. La rutina está descrita con más detalle en el tercer capítulo.

### 1.3.3. MODULO GENERADOR Y EDITOR DE CODIGOS

Este módulo se encarga de crear los archivos que contiene los correspondientes códigos binarios asignados a cada caracter, indexados de tal forma que la decodificación de un caracter sea sencilla y rápida. La edición se puede realizar tanto en modo manual como automático. Esta rutina se detalla en el segundo capítulo.

### 1.3.4. MODULO PRINCIPAL

El módulo principal está encargado de administrar una base de datos. Las opciones permitirán imprimir etiquetas, realizar la identificación utilizando un lector de código de barras, ingreso de datos, consultas de información.

### 1.3.5. CIRCUITO LECTOR

La parte física del sistema consta de los siguientes módulos:

- Scanner.- Es el elemento óptico mediante el cual es posible efectuar la exploración de una etiqueta, los claros y oscuros son transformados en variaciones de corriente que guardan estrecha relación con la diferencia de los coeficientes de reflexión de barras y espacios, así como de su ancho relativo.

- Digitalizador. Es el circuito electrónico acondicionador de señal que amplificará la señal proveniente del scanner. Mediante un proceso diferencial y un circuito con lazo de histéresis la señal de entrada previamente amplificada es comparada con una muestra anterior de la misma señal de entrada, de este modo es posible discriminar entre las barras y los espacios, haciendo corresponder un nivel de tensión de 5 voltios ó 0 voltios respectivamente.

- Procesador de señal digital.- Esta etapa del sistema está constituida por un circuito digital que utiliza un microcontrolador que desempeña las siguientes tareas:

- Establecer la comunicación serial con un computador PC-XT.
- Registrar las lecturas del scanner.
- Recuperar la información binaria a partir de las lecturas.
- Detectar errores de decodificación.
- Validar ó no una exploración.
- Enviar el correspondiente código de los mensajes de error o la información válida según sea el caso.

#### 1.3.6. PROCESADOR DE LA INFORMACION DEL CODIGO IMPRESO EN LA ETIQUETA

Esta etapa corresponde a la rutina desarrollada para un computador PC - XT. Presenta un ejemplo de aplicación desarrollado en un lenguaje de alto nivel (Quick Basic 4.0), en el cual se muestra las ventajas de la utilización de un sistema de código barras.

#### 1.4. SOFTWARE DE PROCESAMIENTO DE LA INFORMACION

La información generada por el lector, no es más que una secuencia de pulsos cuyos niveles de tensión han sido estandarizados a los de la tecnología TTL, siendo posible procesarlos en forma digital. Por esto el primer paso en lo que se refiere a procesamiento lo hace el microcontrolador

## 2.1. GENERACION DE CODIGO DE BARRAS

### 2.1.1. CARACTERISTICAS DEL CODIGO

Como se anota en el numeral 1.1.6.4, los códigos de barras deberán poseer características acordes a los requerimientos mínimos de una aplicación específica y a las facilidades que brinde el equipo que será utilizado en el proceso de decodificación. Deberán tomarse en consideración adicionalmente factores económicos y tecnológicos para el diseño de un código.

Los tópicos tomados en consideración con el fin de establecer los parámetros que definen las características de un código de barras se analizan seguidamente:

### 2.1.2. NUMERO DE CARACTERES

El número de caracteres del alfabeto latino está compuesto por 10 numéricos y 27 literales, haciendo un total de 37 caracteres, siendo necesarios al menos 6 bits con un total de 64 combinaciones diferentes para un código con éste número de caracteres, como se deduce de la Ecuación 2.1. Es claro que una gran cantidad de las combinaciones posibles no

son utilizadas de manera que algunos de ellos son asignados a otros caracteres de uso frecuente, como: \*, %, /, ?, etc.

$$\text{Número de combinaciones} = 2^n \quad \text{Ec 2.1}$$

### 2.1.3. TOLERANCIA DEL SISTEMA

Otro aspecto que debe considerarse en relación al número de bits necesarios para representar un carácter, es dotar al sistema de una única decodificación del carácter, ampliando los límites de tolerancia de impresión y decodificación. Esto se obtiene utilizando las secuencias binarias cuya distancia mínima sea la mayor posible. "Según Shu Lin & Costello la distancia mínima se define como la menor distancia que se puede encontrar al calcular la distancia de Hamming, de entre todas las palabras que forman el código."

Una manera bastante simple para obtener una distancia mínima sin utilizar un algoritmo muy complicado es tomar las secuencias con un número fijo de bits de valor uno "1". El número de secuencias útiles se calcula con la ecuación 2.2, y la FIGURA 2.1 presenta en forma tabulada el número de secuencias válidas para 8 bits por carácter.

---

<sup>1</sup> ERROR CONTROL CODING. SHU LIN - COSTELLO. Pág 65

$$\# \text{ comb. (c) } = \frac{n!}{m! (n - m)!} \quad \text{Ec 2.2}$$

n      Número de bits por caracter.

m      Número de bits con valor uno "1".

Bits. (n)	unos "1". (m)	Palabras (c)	D <sub>min.</sub>
8	1	8	2
8	2	28	2
8	3	56	2
8	4	70	2

Fig 2.1 COMBINACIONES PARA 8 BITS.

La densidad de caracteres por unidad de longitud del código impreso está determinada por las especificaciones de los sistemas lector e impresor y por la estructura del código. Debe llegarse a un compromiso que procure aprovechar de la mejor manera las características de unos y otros.

#### 2.1.4. DENSIDAD DE INFORMACION

En esta sección se analizan diferentes estructuras para un código de barras, tomando como ejemplos las utilizadas en sistemas prácticos y son:

a) Código cuya representación asigna a una barra delgada el dígito cero "0" y a una barra ancha el dígito uno "1" o viceversa, intercalando un espacio blanco entre las barras, como el código 2 de 5 de Hewlett Packard. TABLA 2.1.

Cada carácter es codificado por 5 dígitos binarios de los cuales solo dos pueden tomar el valor de uno "1", de esta manera se consigue que entre dos palabras del código la distancia mínima sea 2, reduciendo la probabilidad de error. El mayor número de bits de la palabra código toma el valor cero "0" y les corresponden las barras de menor ancho. Las barras anchas naturalmente corresponderán a unos "1". En éste código las barras anchas son el doble de las delgadas y los espacio blancos tiene el mismo ancho de una barra delgada. Con estas consideraciones la longitud total del código en la etiqueta se calcula usando la siguiente ecuación:

$$\text{Long. Total} = A * ( (\# "1" \times 3) + (\# "0" \times 2) - 1 ) \text{ Ec.2.3}$$

Donde:

A Es el ancho de una barra delgada.

# "1" Es el número de unos "1".

# "0" Es el número de ceros "0".



Fig 2.2 EJEMPLO DE CODIGO DE BARRAS PARA 2 de 5.



Como se puede observar en la FIGURA 2.2, el número de espacios blancos es igual al número total de barras menos uno. El conjunto de caracteres del código son los números de 0 a 9 y se lo clasificaría como un código numérico de baja densidad debido a la presencia de los espacios entre barras.

b) Código con una representación similar a la descrita en la sección 2.1.4.a, aprovechando los espacios entre barras para llevar información. El valor binario de barras y espacios está determinado por el ancho. Ejemplo de esta técnica es el código 3 de 9 desarrollado por Radio Shack. El formato 3 de 9 utiliza 9 bits para representar el código, con 3 bits de valor uno "1". 5 bits se representan por barras y 4 bits por espacios. De los 3 bits de valor uno "1", 2 son barras y 1 es espacio. Algunos de los caracteres de este código se muestran en la TABLA 2.2.

Una desventaja de este tipo de código es que cada caracter empieza y termina con una barra oscura, lo que implica que entre palabras código se requiere intercalar un espacio blanco que los separe, el cual no debe tener la dimensión de un cero "0" o de un uno "1", teniendo por obvias razones una dimensión mayor que la de un uno "1". La FIGURA 2.3, presenta una etiqueta de éste código.

## CODIGO 3 DE 9 DE RADIO SHACK

CARACTER	CODIGO
1	1 0 0 1 0 0 0 0 1
2	0 0 1 1 0 0 0 0 1
3	1 0 1 1 0 0 0 0 0
4	0 0 0 1 1 0 0 0 1
5	1 0 0 1 1 0 0 0 0
6	0 0 1 1 1 0 0 0 0
A	1 0 0 0 0 1 0 0 1
B	0 0 1 0 0 1 0 0 1
\$	0 1 0 1 0 1 0 0 0
*	0 1 0 0 1 0 1 0 0

Tabla 2.2 EJEMPLOS DE CARACTERES CODIFICADOS CON 3 de 9.

3 of 9



Fig 2.3 EJEMPLO DE CODIGO DE BARRAS PARA 3 de 9.

c) Código en el que las barras tienen un valor binario de uno "1" y los espacios cero "0", por tanto un caracter puede estar representado por secuencias con varios unos "1" o ceros "0" consecutivos formando barras y espacios de anchos no estandarizados. Esta técnica de decodificación toma como referencia el ancho de una barra o espacio simple y mediante un proceso de comparación es posible recuperar la información binaria a partir de las lecturas del "SCANNER" durante la exploración. La velocidad debe ser aproximadamente constante y deben incluirse periódicamente barras de simple ancho que actualicen la referencia. Ejemplo de esta técnica es el UPC " Universal Product Code", que corresponde a un código numérico con dos representaciones diferentes para cada dígito.

Las etiquetas del código están constituidas por dos grupos de 5 caracteres cada uno. La representación de un caracter depende del grupo en que se encuentre y cada dígito está codificado por 7 bits agrupados en dos barras y dos espacios. El número de bits de valor uno "1" es impar si está en el lado izquierdo y par cuando está en el lado derecho de la etiqueta, como se muestra en la TABLA 2.3. El formato de las etiquetas de éste código presenta: bits de guarda, bits de separación de grupos y dos caracteres adicionales ubicados entre las guardas y el conjunto de

caracteres. El primero identifica el sistema de codificación de productos farmacéuticos, alimenticios, etc. El segundo localizado en el extremo derecho, es un caracter de verificación que se obtendrá al pasar los 11 primeros caracteres por un algoritmo codificador. Las guardas o secuencias de inicio y fin son "101", y la de separación es "01010", siendo utilizadas para establecer la duración de una barra llamada "tiempo de barra"; lo cual establecerá la referencia para la decodificación. La FIGURA 2.4 ilustra una etiqueta del código UPC. ( Universal Product Code)

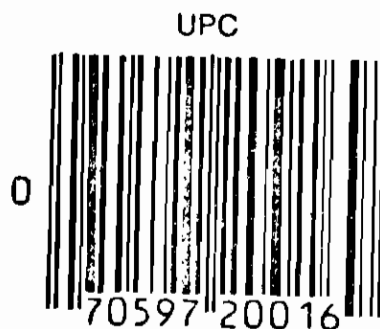


Fig 2.4 EJEMPLO DE CODIGO DE BARRAS PARA UPC.

## CODIGO UPC

CARACTER	IZQUIERDA	DERECHA
	IMPAR	PAR
0	0001101	1110010
1	0011001	1100110
2	0010011	1101100
3	0111101	1000010
4	0100011	1011100
5	0110001	1001110
6	0101111	1010000
7	0111011	1000100
8	0110111	1001000
9	0001011	1110100

Tabla 2.3 CARACTERES CODIFICADOS CON UPC.

La longitud total de una etiqueta para el código UPC, se puede calcular con la ecuación:

$$\text{Long. Total} = A * [ ((10 + 2) * 7) + 11 ] \quad \text{Ec.2.5}$$

$$\text{Long. Total} = A * 95 \quad \text{Ec.2.6}$$

Donde:

A Ancho de una barra o espacio.

El término  $10$  representa el número de dígitos del código, el  $2$  corresponde a los caracteres de control, el factor  $7$  es el número de bits por carácter y el término  $11$  es el número de barras de guarda y la separación entre los bloques izquierdo y derecho de dígitos.

#### 2.1.4.1. DESARROLLO DEL CODIGO 3 DE 8

El código alfanumérico desarrollado incluye: caracteres especiales, alta densidad, restricción del número de bits con valor "1", exploración bidireccional, no se incluyen dígitos adicionales, número de caracteres entre  $1$  y  $6$ , guardas de inicio y fin. La tolerancia a la variación de la velocidad está determinada por el algoritmo decodificador. Las características del sistema de impresión y exploración definen la tolerancia en las dimensiones de la etiqueta.

La impresión del código de un carácter se efectúa de manera similar a la descrita en la sección 2.1.4.b. El número de bits por carácter debe ser par garantizando que todos los caracteres inicien con un espacio y terminen con una barra. Es necesario que una etiqueta quede perfectamente delimitada por barras en sus extremos, de manera que el número de bits de las guardas sumado al de los caracteres debe ser impar. Las guardas proporcionan una plena identificación del sentido de exploración. Las secuencias binarias utilizadas

para las guardas son: "10111" y "1011" para izquierda y derecha respectivamente.

Las guardas utilizadas además de tener el número de bits requeridos son asimétricas en sí mismas, simplificando la lectura bidireccional. Es posible no utilizar guarda de final, en cuyo caso el proceso de lectura será unidireccional, requiriendo de los usuarios del sistema un conocimiento de la estructura de las etiquetas o que ésta tenga marcas adicionales indicando el sentido de lectura.

Las guardas de izquierda (10111) y derecha (1011) aparentemente son similares si las miramos en el mismo sentido, más si las consideramos con diferente sentido de exploración es evidente su diferencia, lo que permite establecer rápidamente el sentido de lectura.

La longitud de la etiqueta se calcula con la siguiente ecuación:

$$\text{Long. Total.} = A * [( \# "c" * ( \# "b" + \# "1" ) ) + 16] \quad \text{Ec 2.7}$$

Donde:

- A Es la dimensión del ancho de una barra o espacio, cuyo valor binario es cero "0".
- # "c" Número de caracteres.

#"b" Número de bits por caracter.

#"1" Número de bits de valor uno "1" por caracter.

La longitud de cada caracter se calcula por el número de bits por caracter y el número de bits de valor uno "1". El número de bits utilizados para las guardas es 9, de los cuales 7 son de valor uno "1".

## 2.2. PROGRAMA DE GENERACION DE CODIGO DE BARRAS.

La representación de caracteres en código de barras no implica solamente crear la tabla de equivalencias, es necesario también poder obtener la correspondiente representación gráfica en dimensiones físicas adecuadas, ajustandose a los requerimientos mínimos del sistema de exploración. El medio de impresión escogido es el impresor matricial trabajando en modo gráfico.

El programa desarrollado para la generación del código de barras está bajo el nombre de BSGEN.EXE y es una rutina desarrollada en Basic. Permite editar o modificar la tabla de equivalencias de los caracteres. El archivo tendrá la extensión "CBR" y es creado si no se encuentra presente en la unidad de trabajo definida.



### 2.2.1. TABLA DE EQUIVALENCIAS

El archivo debe ser de rápido acceso e indexado para facilitar su consulta. Tanto la tabla de equivalencia como los índices forman parte del mismo archivo, ocupando regiones bien definidas. Este es consultado para verificar si los caracteres digitados al crear una etiqueta tienen código asignado y así obtener el código binario de un carácter durante el proceso de impresión de la etiqueta, o para recuperar los caracteres cuando el lápiz lector ha efectuado una exploración.

Las equivalencias están almacenadas en registros del archivo cuya identificación es el valor ASCII del carácter. Esto facilita la consulta de la tabla cuando un carácter es digitado. Cuando el carácter es recuperado desde la etiqueta por el circuito lector, la información recuperada corresponde al código de barras, haciendo necesaria una región en la tabla cuyos registros contengan el valor ASCII del carácter y su identificación sea el código de barras. Por último, para facilitar el proceso de edición, debe existir información respecto a la posición del carácter en la pantalla del computador.

El archivo además debe guardar los parámetros utilizados para el código de barras para efectos de supervisión si la tabla de equivalencias es editada posteriormente.

La FIGURA 2.5 ilustra el mapa de memoria del archivo con la tabla de equivalencias.

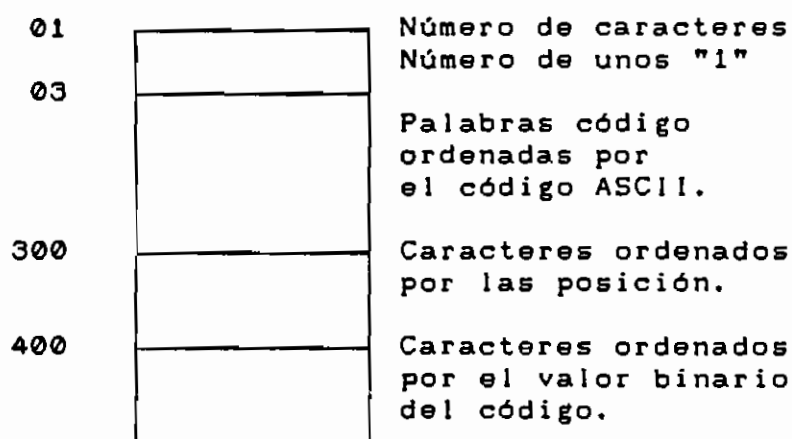


Fig 2.5 MAPA DEL ARCHIVO.

### 2.2.2. EJEMPLO DE TABLA DE EQUIVALENCIAS EDITADA UTILIZANDO EL PROGRAMA BSGEN.EXE

La TABLA 2.4 presenta un ejemplo generado con: palabras código de 8 bits y 3 bits de valor uno "1", por lo que el código generado se denominará 3 de 8. El nombre del archivo que contiene la tabla de equivalencia del código se denomina "TABLA.CBR", archivo al que se hará referencia para la creación de etiquetas y su decodificación.

ESCUELA POLITECNICA NACIONAL  
 FACULTAD DE INGENIERIA ELECTRICA  
 TABLA DE CARACTERES

#	CARACTER	ASCII	CODIGO
1	A	65	00000111
2	B	66	00001011
3	C	67	00001101
4	D	68	00001110
5	E	69	00010011
6	F	70	00010101
7	G	71	00010110
8	H	72	00011001
9	I	73	00011010
10	J	74	00011100
11	K	75	00100011
12	L	76	00100101
13	M	77	00100110
14	N	78	00101001
15	N	165	00101010
16	O	79	00101100
17	P	80	00110001
18	Q	81	00110010
19	R	82	00110100
20	S	83	00111000
21	T	84	01000011
22	U	85	01000101
23	V	86	01000110
24	W	87	01001001
25	X	88	01001010
26	Y	89	01001100
27	Z	90	01010001
28	1	49	01010010
29	2	50	01010100
30	3	51	01011000
31	4	52	01100001
32	5	53	01100010
33	6	54	01100100
34	7	55	01101000
35	8	56	01110000
36	9	57	10000011
37	0	48	10000101
38	/	47	10000110
39	\	92	10001001
40	@	64	10001010
41	#	35	10001100
42	%	37	10010001
43	&	38	10010010
44	*	42	10010100
45	-	45	10011000
46	+	43	10100001
47	?	63	10100010

08-08-1989

Ramiro E. Morejón T.

Tabla 2.4 GENERADA CON 8 BITS Y 3 UNOS.

### 2.2.3. PARAMETROS DE ENTRADA DE BSGEN.EXE

El programa BSGEN.EXE requiere de algunos parámetros para su normal ejecución, los cuales están predefinidos como parte de la rutina y pueden ser modificados pasandolos como argumentos de BSGEN y son:

Unidad de trabajo.- "U="

La unidad predefinida es "A" y puede ser alterada con la opción "U=x" donde "x" toma la identificación de alguna de las unidades disponibles.

Nombre del archivo.- "A="

Permite definir el nombre del archivo de la tabla del código. Si este no existe en la unidad de trabajo definida, es creado. El nombre tendrá caracteres aceptados por el sistema operativo DOS para nombres de archivos.

Número de bits.- "N="

Define el número de bits de las palabras código, el valor predefinido es 8 y el parámetro se altera con "N=x" donde

"x" debe ser un número par como se especifica en la sección 2.1.4.1, correspondiente a las características del código implantado.

Número de unos.- "M="

Esta opción fija el número de bits con valor uno "1" en cada palabra código. El parámetro se modifica con "M=x" donde "x" es menor o igual a "N". Predefinido 3.

#### 2.2.4. SINTAXIS

La sintaxis que invoca al programa BSGEN.EXE es:

BSGEN U=a/A=tabla/N=8/M=3

Donde las opciones deben estar separadas por una barra inclinada "/". La condición predefinida será tomada como válida si la opción correspondiente es omitida.

## 2.3. PROGRAMA DE IMPRESION DE ETIQUETAS

La impresión de las etiquetas la realiza el programa BSIMP.EXE. Este módulo maneja los comandos gráficos de impresores matriciales del tipo IBM Proprinter y EPSON, que son los de mayor difusión en el mercado.

El código implantado requiere que el ancho de barras o espacios mantenga una relación de 2 a 1, correspondiendo la mayor dimensión a los bits de valor uno "1", reduciendo la longitud total de la etiqueta por tener más ceros "0" que unos "1". Las dimensiones pueden ser alteradas debiendo tomarse en consideración la resolución del sistema detector "scanner" y la capacidad gráfica del impresor.

### 2.3.1. MODO GRAFICO DE UN IMPRESOR MATRICIAL

El modo gráfico de cuádruple densidad, controla cada una de las agujas de la cabeza de impresión para conseguir un gráfico punto a punto. En este modo se puede activar más de un punto a la vez.<sup>2</sup>

Cada aguja es activada en forma independiente y tiene asignado un valor decimal que depende de la posición que

---

<sup>2</sup>Printer User's Guide, Apex80 , pág 6-3 y 6-4.

ocupa en la cabeza del impresor como se muestra en la TABLA 2.5.

128	●	Punto superior
64	●	
32	●	
16	●	
8	●	
4	●	
2	●	
1	●	Punto inferior

Tabla 2.5 VALOR DE PONDERACION DE CADA AGUJA.

El número decimal que debe ser enviado como parte del comando gráfico que controla las agujas se lo obtiene sumando el valor correspondiente de los puntos que se desean imprimir.

En el caso particular para dibujar las barras se tienen sólo dos posibilidades: imprimir todos los puntos en el caso de una barra donde el número enviado es 255 o avanzar un espacio blanco cuando se envía el número 0. El ancho de las barras o espacios esta determinado por el comando gráfico utilizado y por el número de veces que consecutivamente se

envíen los comandos de impresión con el argumento correspondientes para barras "255" y espacios "0". La longitud vertical de la etiqueta está determinada por el número de veces que se envíe la secuencia de barras.

Los gráficos de cuádruple densidad con 240 puntos por pulgada (aproximadamente 9.4 puntos por milímetro), permite obtener barras cuyo ancho mínimo es 0.317 mm en las que son notorias las irregularidades en bordes, ocasionadas por el sistema mecánico que mueve la cabeza del impresor y reduciendo considerablemente la velocidad de impresión.

La densidad gráfica utilizada para barras de ancho mínimo 0.635 mm, presenta una impresión más uniforme haciendo menos notorias las imperfecciones en los bordes.

### 2.3.2. IMPRESION DE UNA ETIQUETA

Las etiquetas se construyen a partir de las secuencias binarias almacenadas en el archivo creado por BSGEN.EXE de extensión "CBR" (tabla de equivalencias) y los bits de inicio "10111" y fin "1011" (guardas). La secuencia binaria así formada se toma bit a bit y en el caso "1"s el comando gráfico será enviado el doble de veces que si fuese un cero



"0". El valor decimal en cada caso cambiará en forma alternada entre 255 para dibujar una barra y 0 para un espacio. Como se mencionó anteriormente las guardas utilizadas garantizan que una etiqueta inicie y termine con una barra.

### 2.3.3. PARAMETROS DE ENTRADA DE BSIMP.EXE

Los parámetros de entrada de BSIMP.EXE mantienen la sintaxis ya descrita para BSGEN.EXE, y son:

Unidad de trabajo.- "U="

Nombre del archivo.- "A="

Número de bits.- "N="

Los de uso específico de BSIMP son:

Impresor utilizado.- "I="

Permite definir el impresor y las opciones válidas son: EPSON e IBM. Cabe destacar que la principal diferencia en el manejo del modo gráfico se refiere a uso del comando de tabulación horizontal.

#### Longitud de las barras.- "L="

Las barras de una etiqueta deben tener una longitud vertical que se ajuste al espacio disponible en el objeto donde se adhiere, además la exploración de las barras bien puede no ser perpendicular a las barras por lo tanto es posible que si la longitud vertical es muy pequeña, durante la exploración el "scanner" puede salir del espacio cubierto por la etiqueta provocando una lectura errónea. Por lo anterior se debe poder imprimirlas en diferentes tamaños.

La opción "L=x" pasa a BSIMP.EXE en "x" un factor que multiplique la longitud de las barras, siendo la predefinida 24/72 de pulgada (8.46 mm).

#### Ancho de las barras.- "H="

En un sistema práctico puede ser necesario imprimir etiquetas cuyo ancho sea mayor al establecido; con el propósito de integrarlo en la portada de un libro, en una tarjeta de identificación, etc; para posteriormente ser reducida a un tamaño standard mediante procesos fotomecánicos. Esto permitiría conseguir barras con un ancho significativamente menor al obtenido con un impresor

matricial sin perder la resolución en los bordes ni las proporciones de barras y espacios. La opción "H=x" modifica esta dimensión, donde "x" es un número entero que multiplica el ancho mínimo del modo gráfico utilizado con: 3/240 de pulgada (0.317 mm) para ceros (0) y 6/240 de pulgada (0.635 mm) para unos (1), siendo el valor predefinido 2.

Número de copias.- "C="

En un sistema práctico se requiere etiquetar varios artículos de similares características, en estos casos se requiere imprimir automáticamente un número suficiente de ellas con la opción "C=x" donde "x" es un número entero.

Caracteres de la etiqueta.- "D="

El parámetro "D" permite pasar a la rutina BSIMP.EXE los caracteres que serán representados en código de barras en la etiqueta. Los caracteres válidos serán aquellos que tengan su equivalencia binaria en el archivo "TABLA.CBR".

## 2.4. EJEMPLOS DE GENERACION DE CODIGO DE BARRAS.

La TABLA 2.4 muestra los caracteres con sus respectivas equivalencias editadas con BSGEN.EXE, tomando los parámetros predefinidos. El archivo creado es "TABLA.CBR" y se utilizará como referencia para todos los ejemplos y pruebas en todas las partes del sistema implantado.

Los ejemplos siguientes muestran un mismo conjunto de caracteres (012345) representados en código de barras. En cada caso se altera el valor predefinido para los parámetros "L" (largo) y "H" (ancho) de BSIMP.EXE.

Las opciones que no aparecen como parte del argumento de BSIMP.EXE son los predefinidos: u=A / a=TABLA / n=8 / c=1 / i=EPSON.

### EJEMPLO # 1

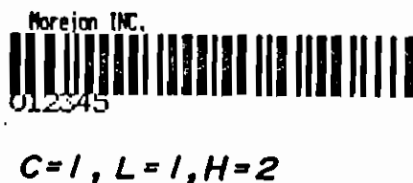


Fig 2.6 BSIMP l=1/h=2/d=012345

## EJEMPLO # 2



$$C=1, L=2, H=2$$

Fig 2.7 BSIMP  $l=2/h=2/d=012345$ 

## EJEMPLO # 3



$$C=1, L=2, H=3$$

Fig 2.8 BSIMP  $l=2/h=3/d=012345$ 

## EJEMPLO # 4



$$C=1, L=2, H=4$$

Fig 2.9 BSIMP  $l=2/h=4/d=012345$

### 3.1. PRINCIPIO DE FUNCIONAMIENTO DE UN LECTOR OPTICO.

El lector óptico es un sistema de detección y decodificación electrónico construido a partir de componentes semiconductores que responden a la luz, permitiendo detectar los cambios en la intensidad de un haz de luz incidente. Las variaciones de intensidad luminosa se consiguen cuando el detector es movido frente a una superficie que contiene barras de un código, sirviendo de transductor entre el código impreso y un circuito electrónico acondicionador de señal para un posterior procesamiento de la información codificada en la etiqueta.

Las respuesta del detector es una señal eléctrica cuya magnitud dependerá de varios factores tales como: el coeficiente de reflexión del area explorada, la fuente de luz incidente en la etiqueta, la velocidad de exploración, el elemento utilizado; siendo los de más frecuente uso los fotodiodos, las fotoresistencias, los fototransistores, etc.

En el caso particular del lector de código de barras manual "lector de lápiz", la exploración se realiza pasando el lápiz óptico frente a la etiqueta y detectando las regiones oscuras (barras) por una disminución de la corriente en el sensor, ocasionada por una disminución de intensidad de luz

reflejada. La magnitud es el resultado de la contribución de todos los puntos dentro del campo de acción del detector, de manera que la transición de un espacio claro a un oscuro no es abrupta como se muestra en la FIGURA 1.1. y la pendiente en el punto de cambio dependerá de la velocidad de exploración.

La fuente de luz y los coeficientes de reflexión de las áreas exploradas determinan la intensidad de corriente, con respuesta lineal según la Ec 3.1. Por lo tanto la variación de cualquiera de estos factores ocasiona variaciones de corriente. Es claro que si existe una perturbación en nivel de luz incidente, ésta se verá reflejada en la corriente. Las variaciones de intensidad en la luz incidente tiene su origen en la luz ambiental.

$$i = K (A^* * C^* + A^n * C^n) L \quad \text{Ec 3.1}$$

i	Corriente en el detector
K	Coefficiente de eficiencia cuántica del detector (constante)
L	Intensidad luminosa
A*	Area de exposición
C*	Coefficiente de reflexión

reflejada. La magnitud es el resultado de la contribución de todos los puntos dentro del campo de acción del detector, de manera que la transición de un espacio claro a un oscuro no es abrupta como se muestra en la FIGURA 1.1. y la pendiente en el punto de cambio dependerá de la velocidad de exploración.

La fuente de luz y los coeficientes de reflexión de las áreas exploradas determinan la intensidad de corriente, con respuesta lineal según la Ec 3.1. Por lo tanto la variación de cualquiera de estos factores ocasiona variaciones de corriente. Es claro que si existe una perturbación en nivel de luz incidente, ésta se verá reflejada en la corriente. Las variaciones de intensidad en la luz incidente tiene su origen en la luz ambiental.

$$i = K (A^b * C^b + A^n * C^n) L \quad \text{Ec 3.1}$$

i	Corriente en el detector
K	Coefficiente de eficiencia cuántica del detector (constante)
L	Intensidad luminosa
A <sup>n</sup>	Area de exposición
C <sup>n</sup>	Coefficiente de reflexión



El área total explorada es:

$$\pi r^2 = A^b + A^o \quad \text{Ec 3.2}$$

r                      Radio del area efectiva del detector

En la Ec 3.1, el coeficiente de las partes claras "C<sup>b</sup>" es cercano a la unidad y aproximadamente constante; el de las partes oscuras "C<sup>o</sup>" es de valor muy bajo debido a la opalescencia de las barras y puede no ser una constante si el sistema de impresión deja pequeñas imperfecciones y por tanto tiene una variación aleatoria. Estos coeficientes no pueden ser evaluados puesto que responden a factores mecánicos como la textura del material en el que se imprimen las etiquetas, el sistema y equipo de impresión; ya que no será igual la calidad de un impresor matricial que la de un impresor láser. El parámetro K determina la eficiencia cuántica del detector y relaciona la cantidad de energía luminosa incidente en el area efectiva del detector con la corriente generada por este concepto.

La FIGURA 3.1, ilustra cualitativamente la variación de la corriente en el detector cuando las barras en la etiqueta presentan imperfecciones.

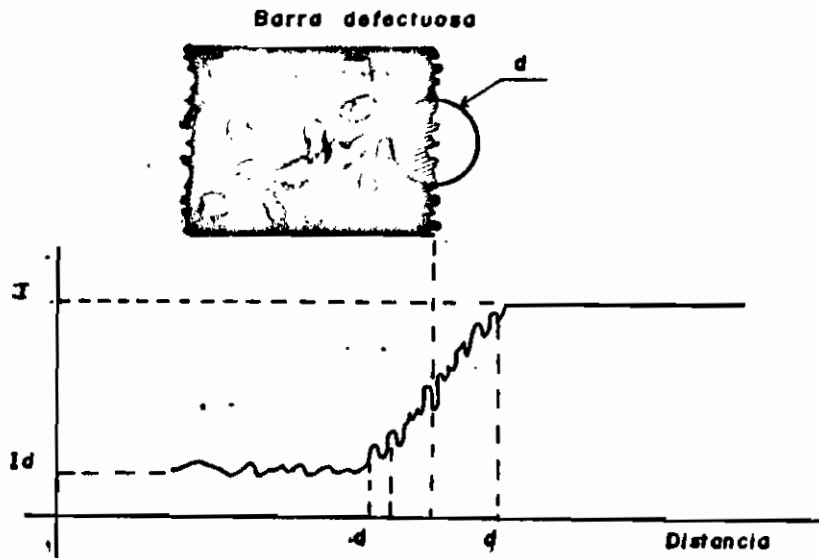


Fig 3.1 VARIACION DE LA CORRIENTE CON ETIQUETAS  
DEFECTUOSAS

El diseño del sistema detector reflectivo presenta obstáculos que deben ser salvados utilizando tecnología de muy alto nivel que permita construir un dispositivo que reúna las características adecuadas para cumplir con los requerimientos de un sistema práctico. El sistema detector debe tener las dimensiones adecuadas que permitan montarlo en un dispositivo de mano y que presente flexibilidad en su manejo. La resolución requerida hace necesario tener integrado un sistema de enfoque, que se consigue con pequeñas lentes biconvexas alcanzando definición de hasta 0.3 mm. La luz ambiental es filtrada mediante una máscara que permite el paso de luz infrarroja, con lo que se reduce la interferencia ocasionada por las fuentes de luz externa

al sistema detector. El emisor y detector son elementos semiconductores que operan en el rango de la luz visible y el infrarrojo cercano.

"HEWLETT PACKARD" ha desarrollado el "HEDS-1000", un detector reflectivo cuyo diseño se ajusta a las necesidades de un lector de código de barras manual, con una muy alta resolución y permitiendo reducir considerablemente las dimensiones del lápiz óptico y las etiquetas. Otras aplicaciones para componentes de estas características son: detección de objetos en cintas de transportadoras, reconocimiento de bordes, tacómetros, exploración de defectos en superficies, etc. La FIGURA 3.2 muestra un esquema de la estructura del HEDS-1000 en la que se aprecia claramente sus partes.

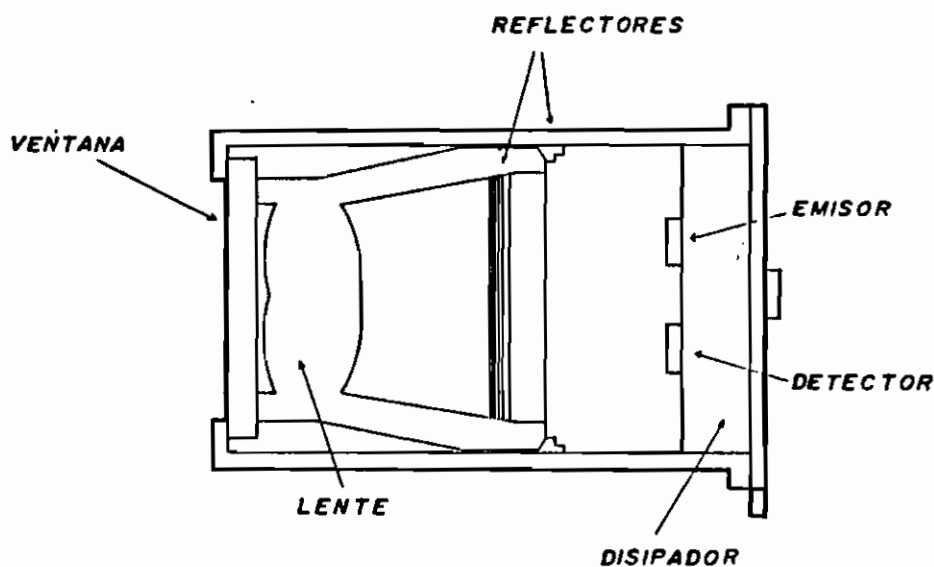
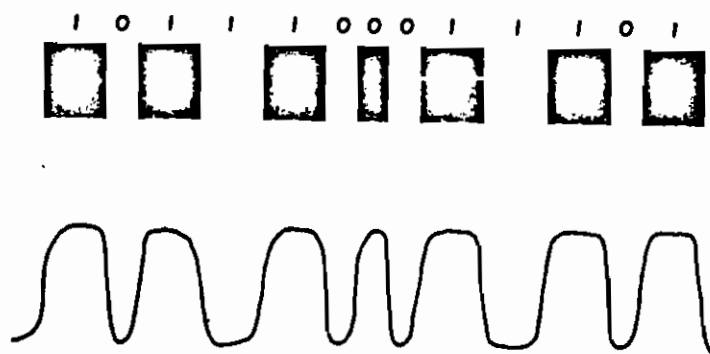


Fig 3.2 ESTRUCTURA DE HEDS - 1000.

La velocidad máxima y mínima de exploración está determinada por la rapidez con la que el sistema óptico responde a las variaciones de la intensidad de luz reflejada. La señal eléctrica así obtenida es procesada por un circuito acondicionador de señal, encargado de convertir una señal análoga en una señal discreta digital con el nivel alto (uno lógico) para representar las barras y el nivel bajo (cero lógico) para los espacios blancos, en la que los bordes coinciden con los flancos de la señal digital resultante como se muestra en la FIGURA 3.3.



El circuito que realiza esta tarea está constituido por tres etapas bien identificadas:

La primera es un amplificador de alta ganancia, cuya respuesta es una señal que será puramente alterna.

La segunda etapa del circuito, un comparador con lazo de histéresis, donde las señales comparadas son : la señal amplificada y una señal obtenida a partir de un circuito detector de picos alimentado con la misma señal de entrada, permitiendo discriminar los puntos de cambio de la señal que marcan las transiciones de regiones claras a oscuras de la etiqueta.

La parte final es un interfaz que acondiciona la señal para hacerla compatible con los niveles lógicos de la tecnología TTL. La información digital obtenida es alimentada a un sistema digital más complejo, realizando un procesamiento con el propósito de transformar los pulsos, cuya duración depende del tamaño de las barras o espacios, en información que sea factible utilizarse en un proceso de más alto nivel y así identificar los caracteres representados en las etiquetas.

El circuito digital mide el ancho de cada barra y espacio. La técnica empleada consiste en medir el tiempo transcurrido entre cada flanco de la señal de entrada asumiendo que la velocidad de exploración es constante. La velocidad se

define como la razón de desplazamiento al intervalo de tiempo.<sup>3</sup>

El valor binario de la etiqueta podrá ser recuperado gracias a la relación 2 a 1 existente entre unos "1" y ceros "0". La secuencia binaria obtenida será procesada por un algoritmo, que determinará si esta se ajusta a las características del código utilizado en la impresión de la etiqueta con el propósito de detectar cualquier anomalía durante la exploración y recuperar los caracteres allí representados.

Para reducir la circuitería al mínimo posible se utiliza el microcontrolador INTEL 8031, que tiene incorporadas rutinas para realizar las funciones de un UART (Universal Asynchronous Receiver Transmitter), permitiendo manejar un pórtico serial bajo la norma RS-232. Además su capacidad para ejecutar operaciones aritméticas de 8 bits, manejo de temporizadores, acceso a memoria interna o externa, etc; simplifican considerablemente el diseño.

---

<sup>3</sup>Física Francis W. Sears/Mark W. Zemansky pág 63 1975.

### 3.2. DESCRIPCION DEL PORTICO DE COMUNICACIONES CON EL COMPUTADOR

RS-232C es la norma de intercambio de información serial más simple y difundida para interconectar un equipo periférico no estandarizado a un computador personal. El protocolo de comunicaciones empleado puede ser ajustado a las necesidades propias de la aplicación en que se lo requiere.

El interfaz serial del circuito utilizado trabaja en modo asincrónico y el control del sentido de la comunicación se efectúa bajo supervisión del programa implantado. SBUF es el registro donde se almacena el byte transmitido o recibido utilizando dos registros independientes. P3.0 es el pin de recepción RXD, P3.1 para transmisión TXD, siendo esta una función alterna de los pines del pòrtico P3, cuyos niveles de tensión son compatibles con TTL.

El UART opera bajo el MOD0 1 con el siguiente formato: 8 bits de datos y velocidad de transmisión variable; el cual queda definido cuando se carga el registro SCON con el valor hexadecimal 50H. (ver apéndice B)

La velocidad binaria se fija utilizando el TIMER 1 y está determinada por la siguiente ecuación:

$$\text{velocidad} = \frac{2^{SMOD} \times \text{Frecuencia del cristal}}{384 \times [256 - (TH1)]} \quad \text{Ec 3.3}$$

Donde:

SMOD Es el bit más significativo del registro PCON (Power Control Register). En 1 la velocidad de transmisión se duplica.

TH1 El valor en decimal del contenido del byte más significativo del TIMER 1.

La ecuación descrita es válida únicamente en los modos 1 y 3 en los que la velocidad es ajustable.

"El modo 1, recibe o transmite 10 bits, un bit de inicio "0", 8 bits de datos, siendo el menos significativo el primero en ser enviado y un bit de parada "1", en recepción el bit de parada pasa a RB8 en el registro SCON".\* (ver apéndice B).

La transmisión de un byte se inicia cuando una instrucción utiliza el registro SBUF como destino, entonces se carga

---

\*MCS 51 ARCHITECTURE Pág 6 - 14



"1" en el bit 9 del registro de desplazamiento y la bandera del registro de control de TX se activa solicitando la transmisión. Esta se inicia en S1P1 (primer periodo del primer ciclo) de la siguiente instrucción.

La recepción se inicia cuando la transición 1-0 es detectada en el pin RXD. Para conseguir recuperar los datos la entrada RXD es muestreada este pin con un reloj que es 16 veces más rápido que la velocidad de transmisión definida, generada con un contador de este módulo. Cuando la transición es detectada el contador es inicializado y el registro de desplazamiento se carga con 1FFH. El valor del bit tomado como válido para ser cargado en SBUF es el presente cuando el contador se encuentra en los estados 7°, 8° y 9°.

### 3.3. DISEÑO DEL LECTOR DE CODIGO DE BARRAS.

#### 3.3.1. LAPIZ LECTOR

El diseño de un lector de código de barras como se dijo anteriormente requiere de un dispositivo detector construido con tecnología muy especializada, por tanto en ésta sección se limitará a exponer las características del HEDS-1000 que es el detector empleado y se propondrá un diseño de las

partes concernientes al circuito, la técnica utilizada para construir el amplificador y el circuito para discriminar las barras y espacios convirtiendolos en señales digitales que posteriormente serán procesadas por el microcontrolador INTEL 8031.

El HEDS-1000 es un detector en el que se encuentra integrado un emisor y un detector con un punto focal común, con una resolución de 0.190 mm. La longitud de onda del emisor es de 700 nm, está provisto de un filtro de rechazo a la luz ambiental y la salida puede tomarse del fotodiodo o a través de un transistor que puede ser configurado como un amplificador de alta ganancia. FIGURA 3.4.

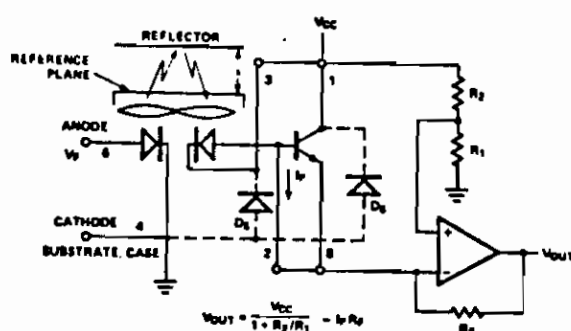


Fig 3.4 DIAGRAMA ESQUEMATICO DEL DETECTOR.

El circuito amplificador y discriminador esta diseñado en base a amplificadores operacionales y elementos discretos

teniendo como primera etapa un amplificador de alta ganancia y bajo ruido, diseñado con inversión de fase, puesto que esta no es una condición crítica en el circuito. La configuración del circuito se muestra en la FIGURA 3.5.

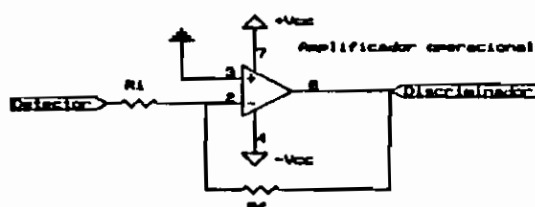


Fig 3.5 AMPLIFICADOR CON INVERSION DE FASE.

$$A_v = R_f / R_i$$

Ec 3.4

La señal que entrega el fotodiodo está en el orden de las décimas de voltio, por lo que la ganancia debe estar en el orden de 20, lo que define la relación de las resistencias  $R_f$  y  $R_i$  según la ecuación Ec 3.4 en el circuito de la FIGURA 3.5.

El circuito discriminador está implantado con la configuración de un comparador como lo muestra la FIGURA 3.6 (a). La señal de entrada  $f_{in}$  es la obtenida como respuesta del fotodiodo previamente amplificada. El valor de

tensión de referencia se consigue con un circuito detector de picos, cuya señal de entrada es la misma  $f_{in}$ . Cuando  $V_c$  es menor que  $f_{in}$ , D1 conduce y carga el condensador C1 hasta alcanzar un máximo, C1 permanece en  $V_{cmax}$  hasta que  $f_{in}$  es menor que  $V_{cmax}$ , entonces D2 conduce hasta que C1 llega a  $V_{cmin}$ . La FIGURA 3.6 (b) muestra en detalle los puntos de decisión, donde  $V_d$  es la caída de tensión en los diodos D1 y D2. El lazo de histéresis producido por R1 y R2, garantiza la conmutación del amplificador desplazando el punto de decisión. La red formada por R1 y R2 debe presentar muy alta impedancia haciendo mínima la descarga del condensador C1 y su relación debe garantizar que el punto de decisión no sea desplazado más de  $V_d$ .

$$V_{cmax} = f_{inmax} - 0.6 \quad \text{Ec 3.5}$$

$$V_+ = (V_o - V_c) * R1 / (R1 + R2) + V_c \quad \text{Ec 3.6}$$

Donde:

$V_+$  Es el voltaje de comparación.

$V_c$  Voltaje en C1.

$V_o$  Voltaje de saturación.

Para:

$V_{in} > V_+$  entonces  $V_o$  es  $-VCC$

$V_{in} < V_+$  entonces  $V_o$  es  $+VCC$

Los voltajes de comparación serán:

$$\text{Máximo} \quad V_+ = V_c - ((V_{CC} + V_c) * R_1 / (R_1 + R_2)) \quad \text{Ec 3.7}$$

$$\text{Mínimo} \quad V_+ = V_c + ((V_{CC} + V_c) * R_1 / (R_1 + R_2)) \quad \text{Ec 3.8}$$

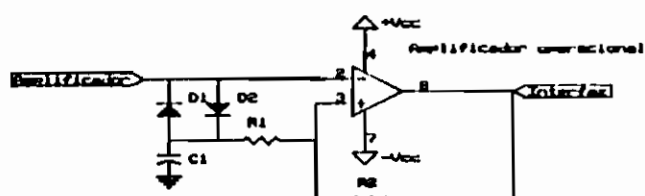


Fig 3.6 (a) CIRCUITO DISCRIMINADOR.

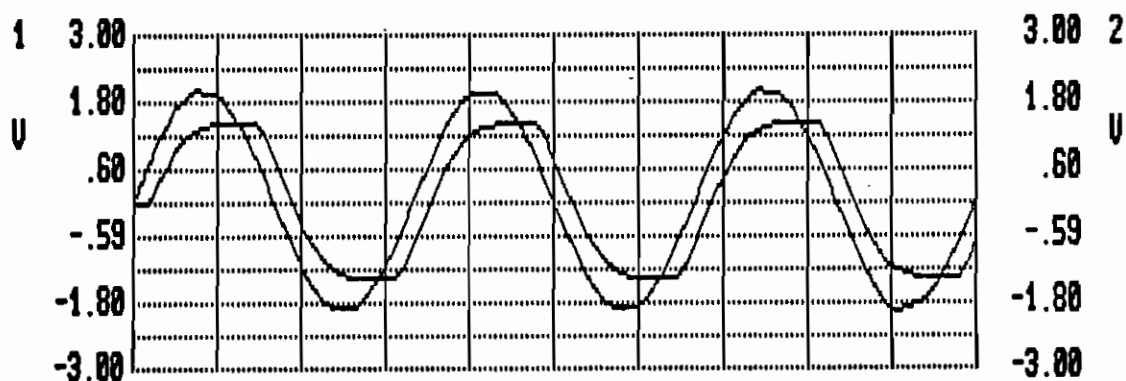


Fig 3.6 (b) RESPUESTA DEL DETECTOR DE PICOS.

La etapa final del circuito es un cuadrador de señal, que proporciona el adecuado interfaz con circuitos de tecnología TTL, haciendo que los flancos de la señal sean pronunciados y los niveles de voltaje adecuados. FIGURA 3.7

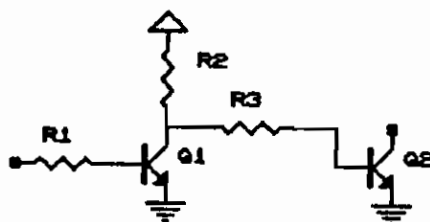


Fig 3.7 CIRCUITO CUADRADOR DE SENAL.

### 3.3.2. CIRCUITO DIGITAL.

El circuito digital implantado se realiza con una configuración mínima basada en un INTEL 8031, como se muestra en el diagrama de bloques de la FIGURA 3.8; en la cual se incluye a más del microcontrolador, la memoria EPROM que contiene el programa, un circuito SN7473<sup>14374</sup> que retiene la dirección de acceso a la memoria EPROM durante el ciclo de traida de la instrucción a ser ejecutada y un circuito MAX232 que efectúa el interfaz de los niveles TTL del INTEL 8031 con los niveles de tensión del pórtico serial del

computador personal. (ver apéndice E). El pórtico serial del equipo periférico es un conector DB-25P macho, del cual los pines 1 y 7 corresponden a los puntos de tierra física y lógica respectivamente, el pin 2 para transmisión "Tx", el pin 3 para recepción "Rx", las señales DTR y DSR son utilizadas para verificar que el equipo periférico está instalado colocando un puente de los pines 20 y 6. El lápiz se conecta en un conector DE-9P macho, cuyo pin 7 es tierra, el pin 9 polarización del lápiz de 5 voltios y el pin 2 es la entrada de la señal.

El lápiz óptico es un dispositivo comercial que ha sido desarrollado para sistemas lectores de código de barras del tipo manual, que incluye en su diseño un detector reflectivo y el circuito amplificador y discriminador. En la presente implantación se utiliza un equipo comercializado por RADIO SHACK que es compatible con el computador personal TRS-80. En su diseño emplea el QEDS-1099, detector reflectivo de HEWLETT PACKARD, con dimensiones físicas y características eléctricas que corresponden a las del HDES-3000, que es el lápiz lector de código de barras de HP. (ver apéndice G).

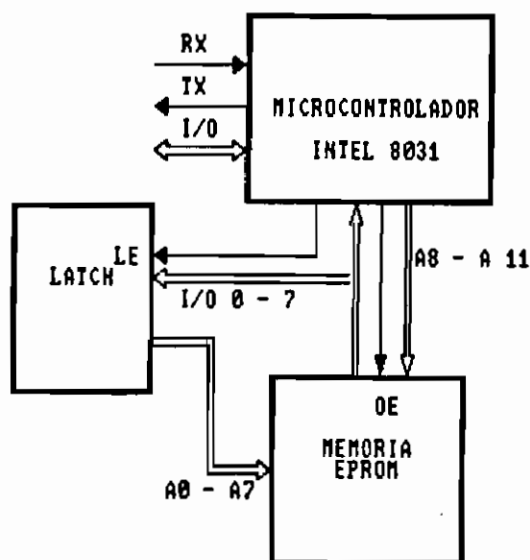


Fig 3.8 DIAGRAMA DE BLOQUES CON LA CONFIGURACION MINIMA DE UN INTEL 8031.

### 3.3.3. RUTINAS DEL MICROCONTROLADOR

El circuito INTEL 8031 presenta en su arquitectura una región de memoria RAM interna, la misma que es aprovechada para almacenar los datos y los registros necesarios para la normal operación de las rutinas implantadas. El mapa de memoria está dividido en tres regiones:



Las localidades de memoria entre 00H y 1FH reservadas para los registros propios del INTEL 8031, necesarios para su normal funcionamiento durante la atención a interrupciones.

La región localizada entre 20H y 59H es utilizada para almacenar los datos provenientes de las lecturas, su extensión es de 57 bytes para lecturas correspondientes a: 9 guardas, 6 caracteres representados por 8 bits cada uno. Este segmento opera como una pila tipo "FIFO", siendo el registro R0 el apuntador. A éste segmento de memoria se lo llamará "TABLA", por tabla de datos.

Las localidades de memoria entre 60H y 6DH son registros intermedios utilizados por las rutinas para: datos, parámetros y caracteres de comandos. Las restantes localidades no son utilizadas en ésta aplicación.

La FIGURA 3.9 ilustra el mapa de memoria y la localización de los registros.

El programa escrito para el INTEL 8031 está estructurado de manera que una vez inicializado el modo de operación de los contadores y el pórtico de comunicaciones, la rutina entra en un lazo que verifica el contenido de RX232. Este registro es cargado con el contenido de SBUF cuando la interrupción serial es atendida. DIAGRAMA 01.

00H		
20H		
59H		
	CONT	60H
	REF	61H
	BAND	62H
	DATOS	63H
	DAT1	64H
	TX232	65H
	RX232	66H
	AUX	67H
	MAYOR	68H
	MENOR	69H
	NUM	6AH
	INIC	6BH
	NUM1	6CH
	NUM2	6DH

Fig 3.9 MAPA DE MEMORIA DEL INTEL 8031.

**DIAGRAMA 01**  
**DIAGRAMA DE FLUJO DE INICIALIZACION Y ATENCION DE COMANDOS**

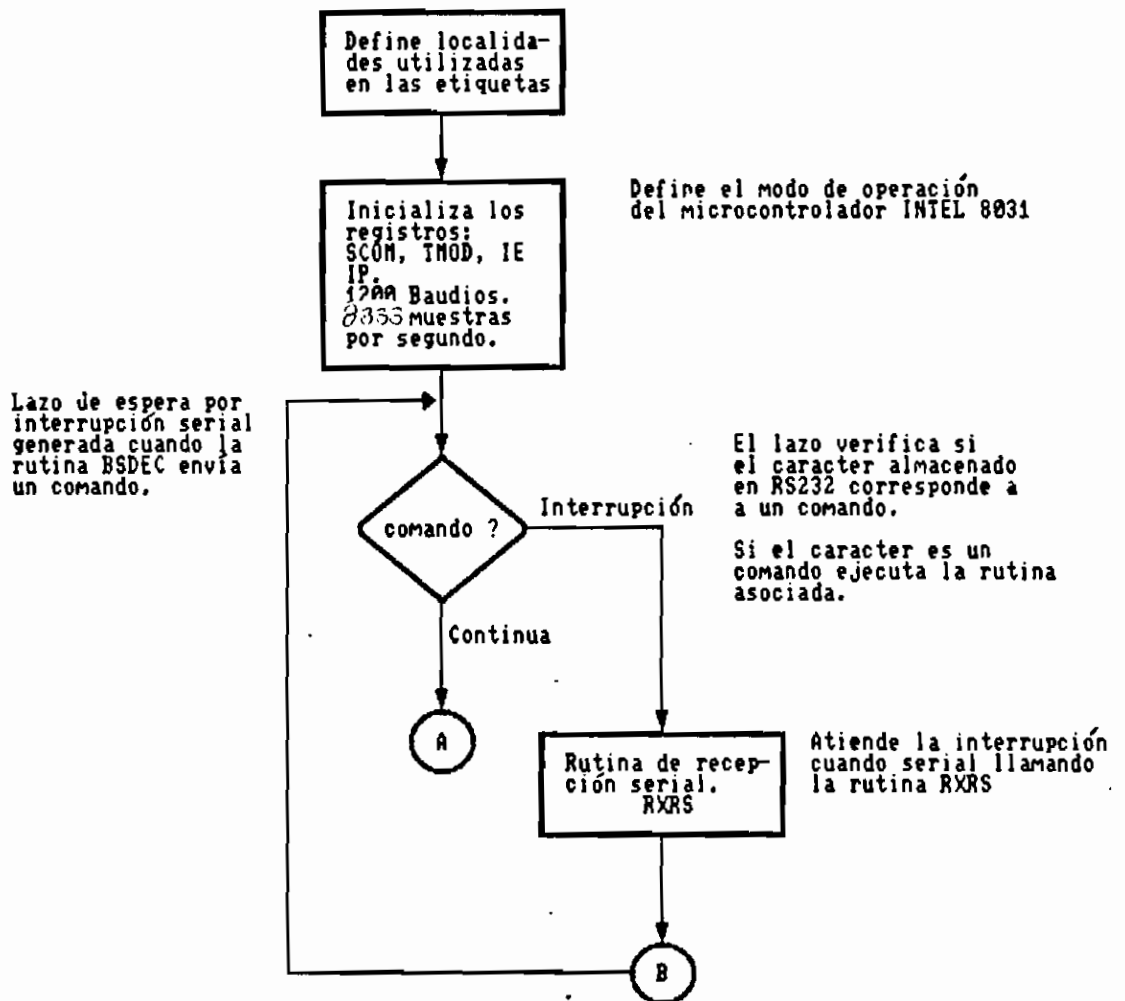
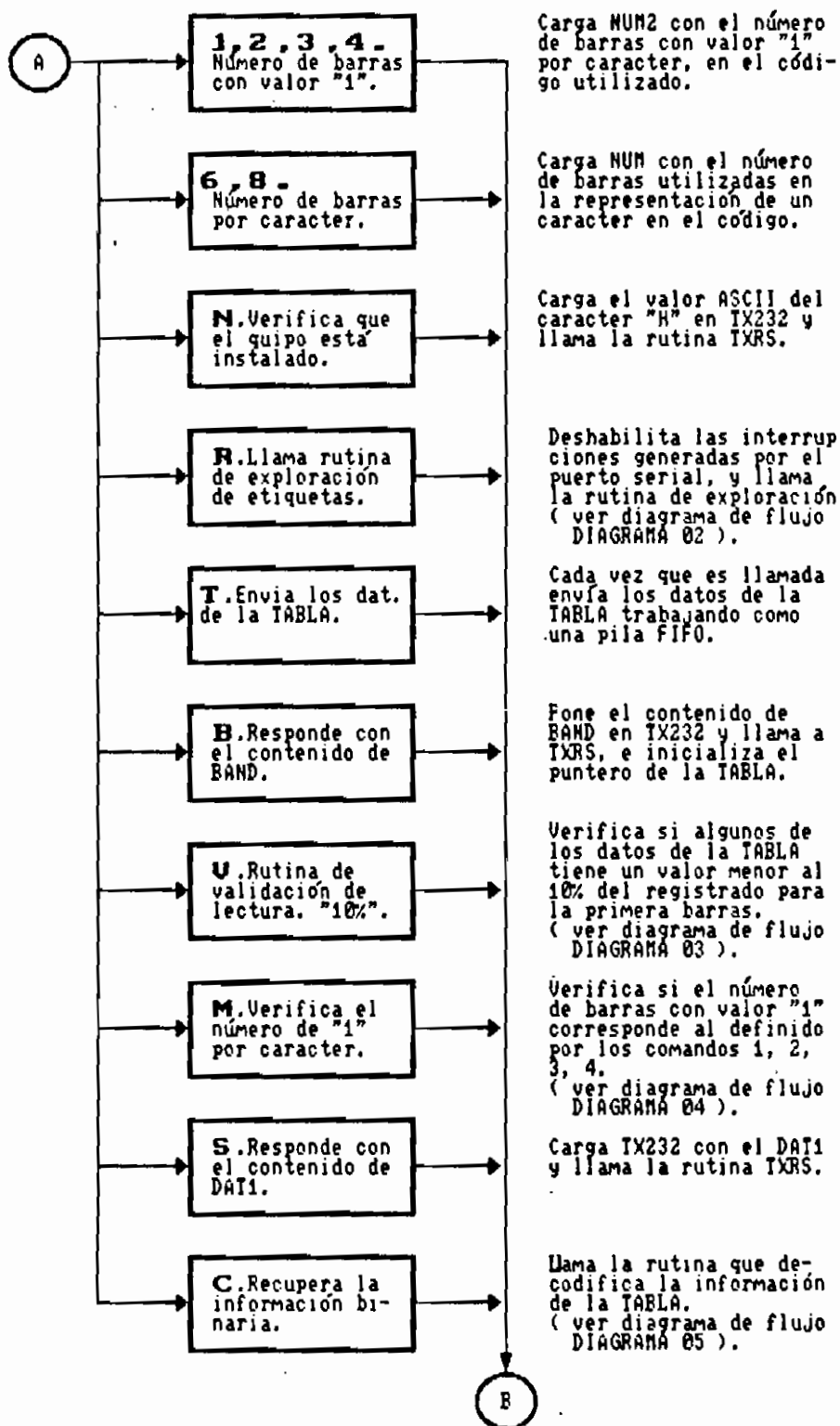


DIAGRAMA 01

DIAGRAMA DE FLUJO DE INICIALIZACION Y ATENCION DE COMANDOS  
CONTINUACION



Una rutina se ejecuta cuando el byte recibido en el registro RX232 es un comando, el registro RX232 se carga con el valor 00H. Si las rutinas tienen muchas instrucciones o son procedimientos recursivos, es necesario definir las como subrutinas y que serán llamadas después de definir sus parámetros de entrada.

Los comandos implantados son:

Comandos "1", "2", "3", "4", definen el número de bits de valor uno "1" por caracter y se presentan como comandos diferentes debido a la estructura del programa. Todas ellas modifican el registro NUM2.

"6", "8", definen el número de bits por caracter. Es un número mayor que el número de bits de valor uno "1", como se indica en la sección 2.1.3.d. Estos comandos modifican el valor del registro NUM.

"N", responde con un caracter conocido, verificando la comunicación en los dos sentidos. El error correspondiente será indicado si el caracter esperado no es recibido en el computador. Las posibles causas son: el caracter no corresponde al enviado debido a una falla en el soporte físico (cable, conectores, transientes); el circuito digital

está: desconectado, dañado o el programa se encuentra fuera del lazo de espera ejecutando una rutina previamente llamada en la que se ha deshabilitado la posibilidad de interrupciones.

"T", transmite un byte del segmento de datos utilizando como puntero el registro R0 y que se incrementa cada vez que "T" es ejecutado.

"B", carga con 20H el registro R0 retornando el puntero al inicio de la "TABLA". Transmite el contenido del registro BAND, que contiene un caracter generado por la última rutina ejecutada. La función de cada caracter será descrita como parte de la rutina que lo genere.

"S", transmite el contenido de DAT1, registro con una doble función. Durante el proceso de exploración se incrementa con cada barra o espacio, registrando el número de lecturas; en la decodificación registrará el número de caracteres representados en código de barras.

"R", rutina de exploración de la etiqueta. Cuando es llamada deshabilita las interrupciones, inicializa DAT1, CONT, BAND y R0. La función de ésta rutina es detectar los puntos de cambio entre barras y espacios. Las lecturas almacenadas en

la "TABLA" corresponden al valor alcanzado por CONT, registro que se incrementa con cada interrupción generada por el TIMER 0. El bit 0 de BAND llamado FB es la bandera de flanco, toma el valor del punto de prueba P1.0, bit 0 del pòrtico 1 en el INTEL 8031. FB conserva el valor de P1.0 hasta la siguiente interrupción y lo compara con el nuevo valor de P1.0. Cuando el valor binario no es igual, el contenido de CONT es almacenado en "TABLA", DAT1 y el puntero R0 se incrementan. DIAGRAMA 02

El valor de BAND toma el valor "0" para un espacio y uno "1" si es barra. La rutina termina cuando CONT toma el valor FFH, condición que ocurre cuando el lápiz es movido muy lentamente en relación a la velocidad de exploración fijada por las interrupciones del TIMER 0 ó cuando la etiqueta ha llegado a su fin. Es necesario distinguir entre uno y otro caso, por ello se válida la lectura si el último valor de BAND es uno "1", pues la etiqueta termina con una barra y es posible que CONT alcance el valor FFH con una barra que no sea la final. Una segunda verificación se efectúa para determinar si el número de barras tiene un mínimo de 9, correspondientes al número de guardas. No obstante la exploración puede ser tomada por válida aun cuando existe una condición de falla, la misma que será detectada con otras rutinas diseñadas para el efecto. En conclusión la

terminación normal ó anormal de "R" es válida si BAND toma el valor 01H y nula en otros casos.

"C", recupera la información binaria a partir de los datos de la "TABLA". El algoritmo implantado calcula el número de caracteres representados en código de barras, con DAT1 y NUM, utilizando Ec 3.9. Luego asume que la lectura se efectuó desplazando el "SCANNER" de izquierda a derecha, por lo tanto espera que las primeras cinco lecturas correspondan a la secuencia de inicio, la misma que debe ser identificada plenamente, para lo cual toma como referencia del segmento la media aritmética de las lecturas en las localidades 21H y 24H de la "TABLA" correspondientes a cero "0" y uno "1" respectivamente. El valor así obtenido es cargado en REF y será comparado con todas las lecturas del segmento considerado utilizando la subrutina llamada CURV, cuyos parámetros de entrada son: la referencia del segmento REF, la dirección de inicio en R0 y el número de lecturas consideradas en R6. DIAGRAMA 05

$$\text{DAT1} = (\text{DAT1} - 9) / \text{NUM} \qquad \text{Ec 3.9}$$

CURV compara una a una las lecturas con el valor de REF, si REF es mayor corresponderá a cero "0" y si es menor será uno "1". Los bits recuperados son cargados en DATOS,



desplazando los bits a la izquierda, formando un byte cuyos bits corresponden a los representados en barras. Por ésta razón el número de bits por caracter es menor o al menos igual a 8.

Si el valor devuelto por CURV en DATOS es diferente de 17H, el programa asume que la lectura se realizó en sentido contrario, es decir de derecha a izquierda. En este caso la referencia se calcula de igual forma que el caso anterior, con las lecturas de las localidades 22H y 23H. Si DATOS es diferente de 0DH se concluye la rutina cargando BAND con 45H para marcar el error correspondiente. La FIGURA 3.10 muestra la ubicación de las guardas en la "TABLA" para los dos sentidos de exploración.

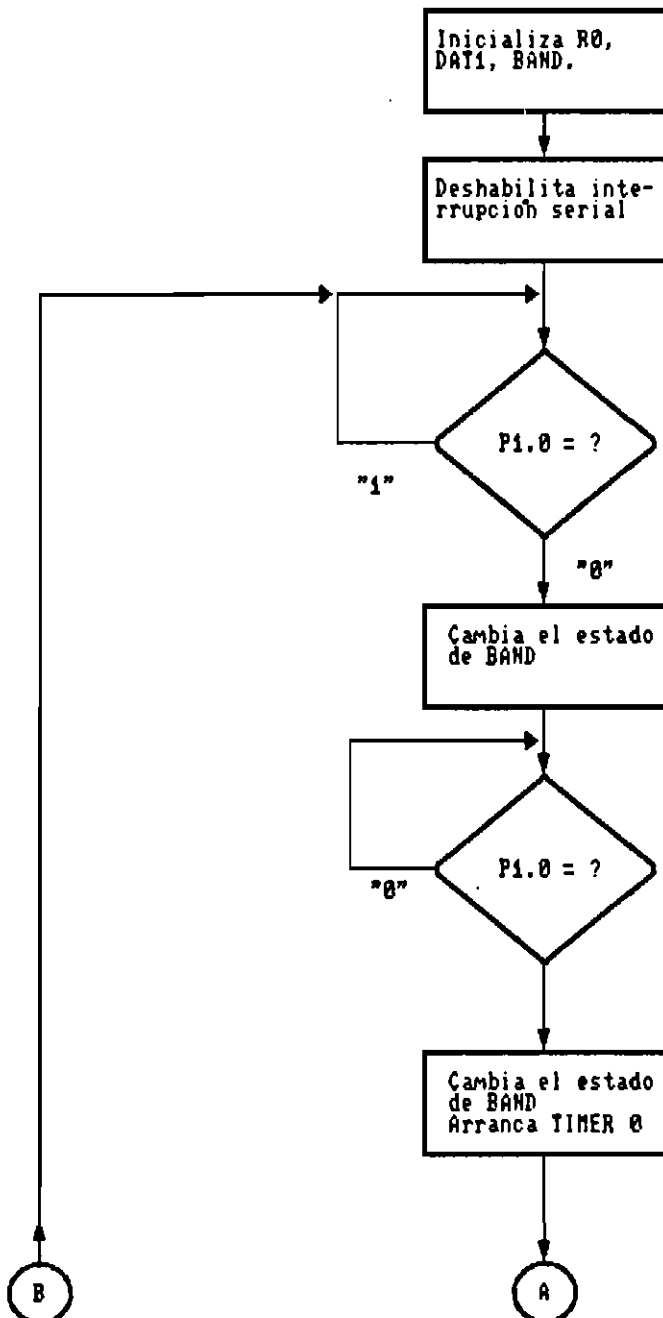
20H	21H	22H	23H	24H	25H
1	0	1	1	1	X

Fig 3.10.a. LOCALIDADES DE "TABLA" PARA LA GUARDA IZQUIERDA.

20H	21H	22H	23H	24H	25H
1	0	1	1	X	X

Fig 3.10.b. LOCALIDADES DE "TABLA" PARA LA GUARDA DERECHA.

**DIAGRAMA 02**  
**RUTINA DE EXPLORACION DE LA ETIQUETA (comando R)**



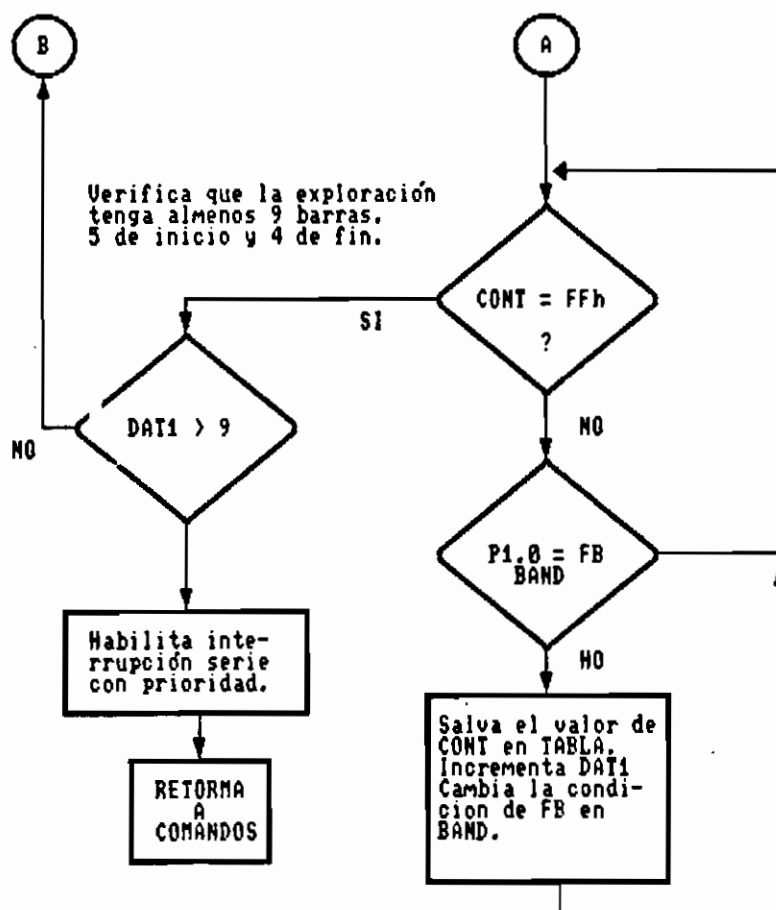
Define los valores iniciales de: R0 = 20h dirección de inicio de la TABLA, DAT1 = 00h el número de barras detectadas, BAND = 01h identifica la primera barra oscura.

Inhíbe las interrupciones generadas por el pórtilo serial, y habilita la interrupción generada por TIMER 0, dándole la máxima prioridad.

Lazo de espera de la rutina, lee el estado del bit 0 del pórtilo 1 el cambio de "1" a "0" ocurre cuando el lápiz es activado.

Lazo de espera por el primer flanco en la exploración de la etiqueta, este es generado por una barra.

DIAGRAMA 02  
 RUTINA DE EXPLORACION DE LA ETIQUETA (comando R)  
 CONTINUACION



**DIAGRAMA 05**  
**RECUPERA LA INFORMACION BINARIA A PARTIR DE LAS LECTURAS**

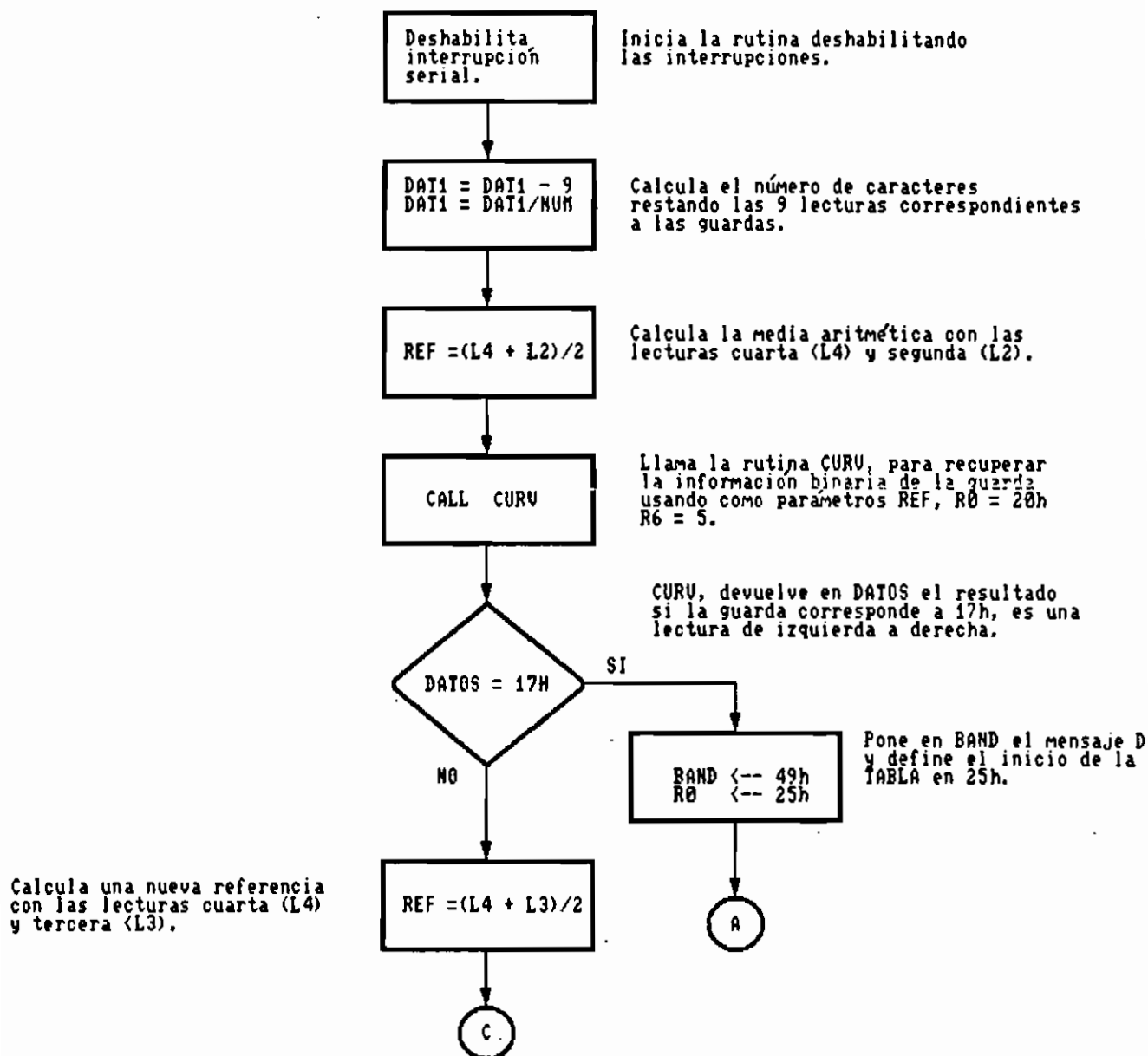


DIAGRAMA 05  
 RECUPERA LA INFORMACION BINARIA A PARTIR DE LAS LECTURAS  
 CONTINUACION

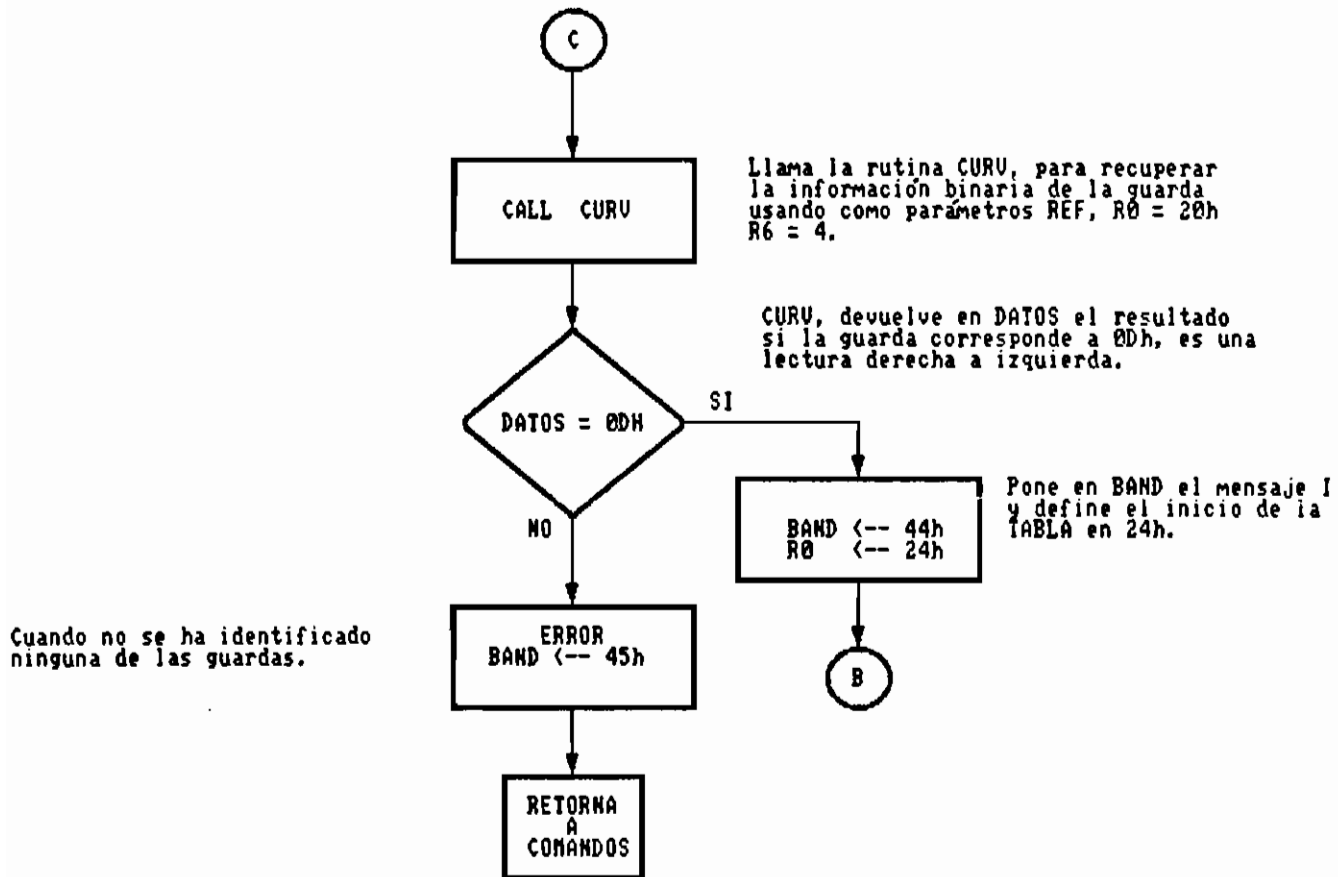


DIAGRAMA 85  
 RECUPERA LA INFORMACION BINARIA A PARTIR DE LAS LECTURAS  
 LECTURA DERECHA  
 CONTINUACION

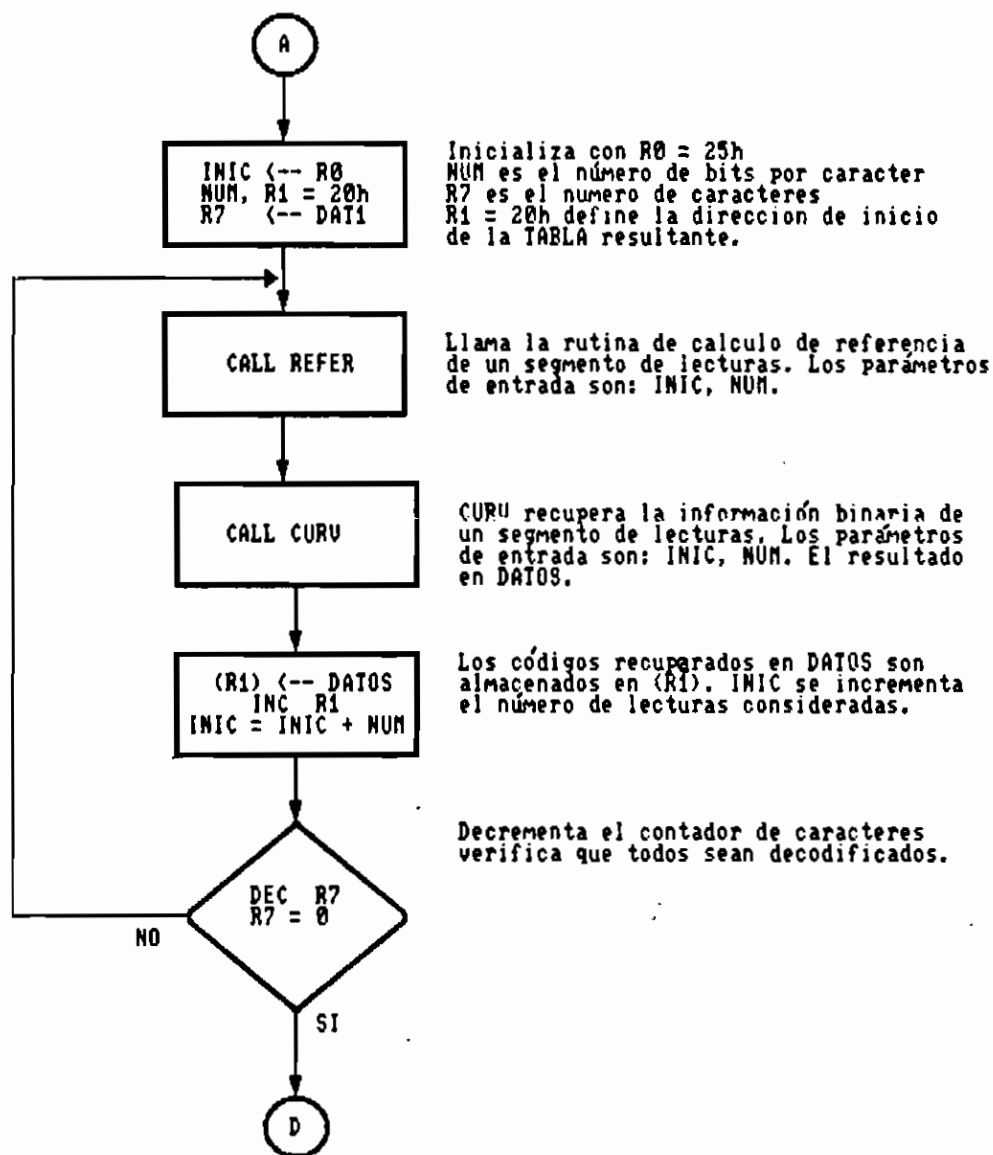


DIAGRAMA 85  
 RECUPERA LA INFORMACION BINARIA A PARTIR DE LAS LECTURAS  
 LECTURA DERECHA  
 CONTINUACION

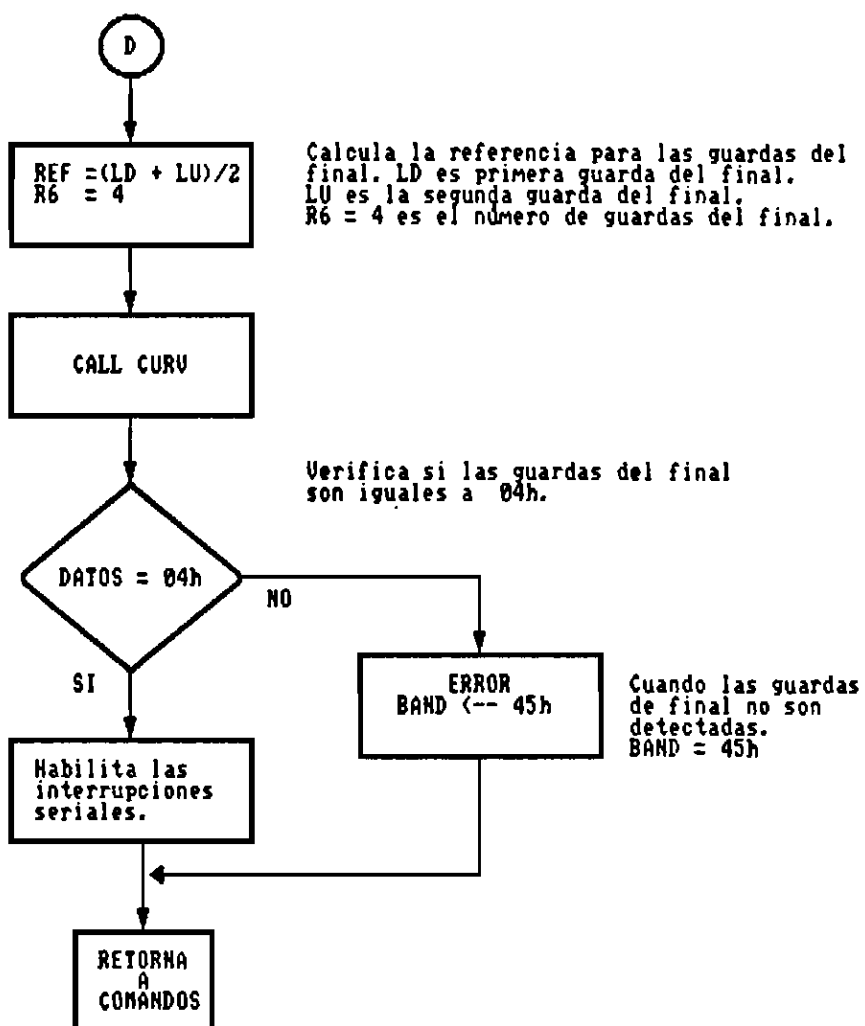


DIAGRAMA 05  
 RECUPERA LA INFORMACION BINARIA A PARTIR DE LAS LECTURAS  
 LECTURA IZQUIERDA  
 CONTINUACION

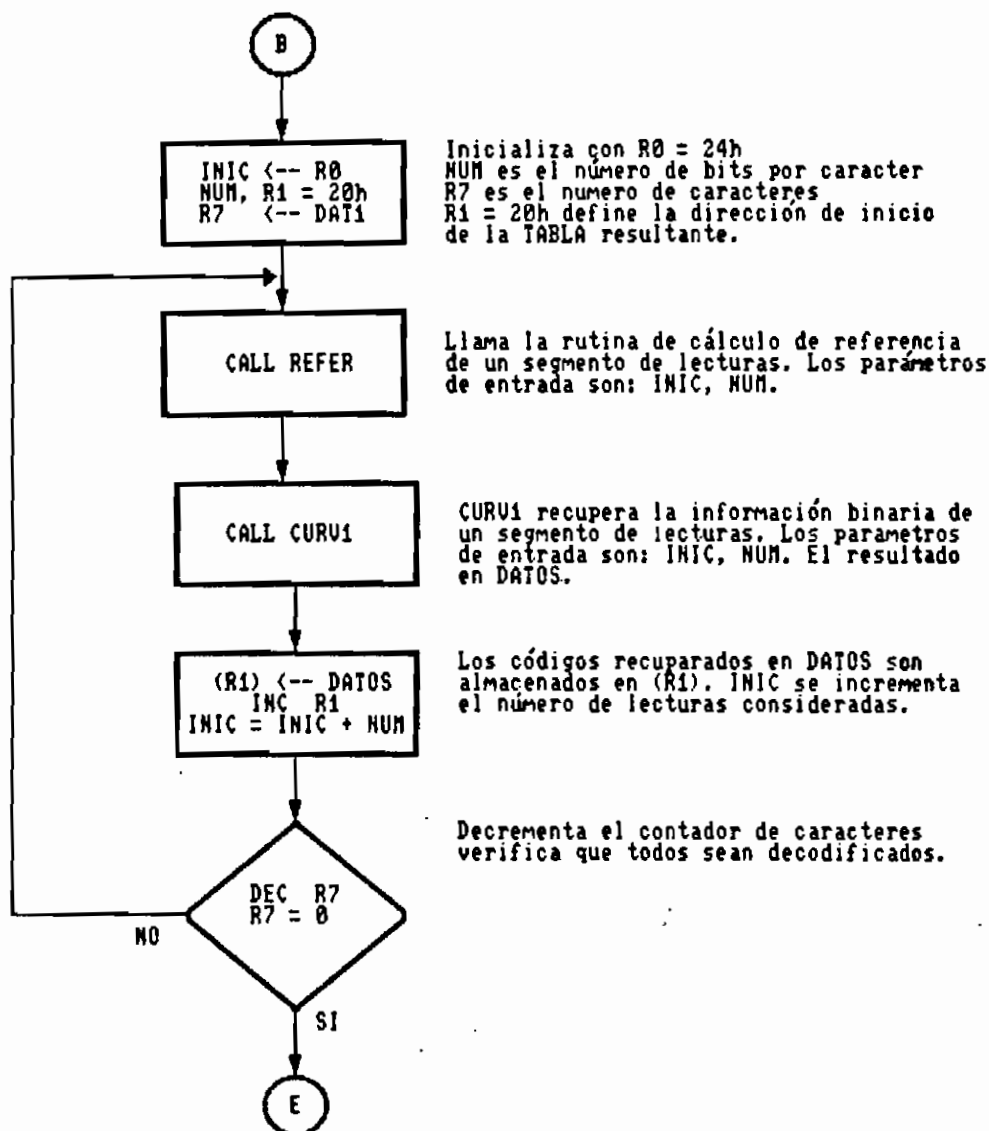
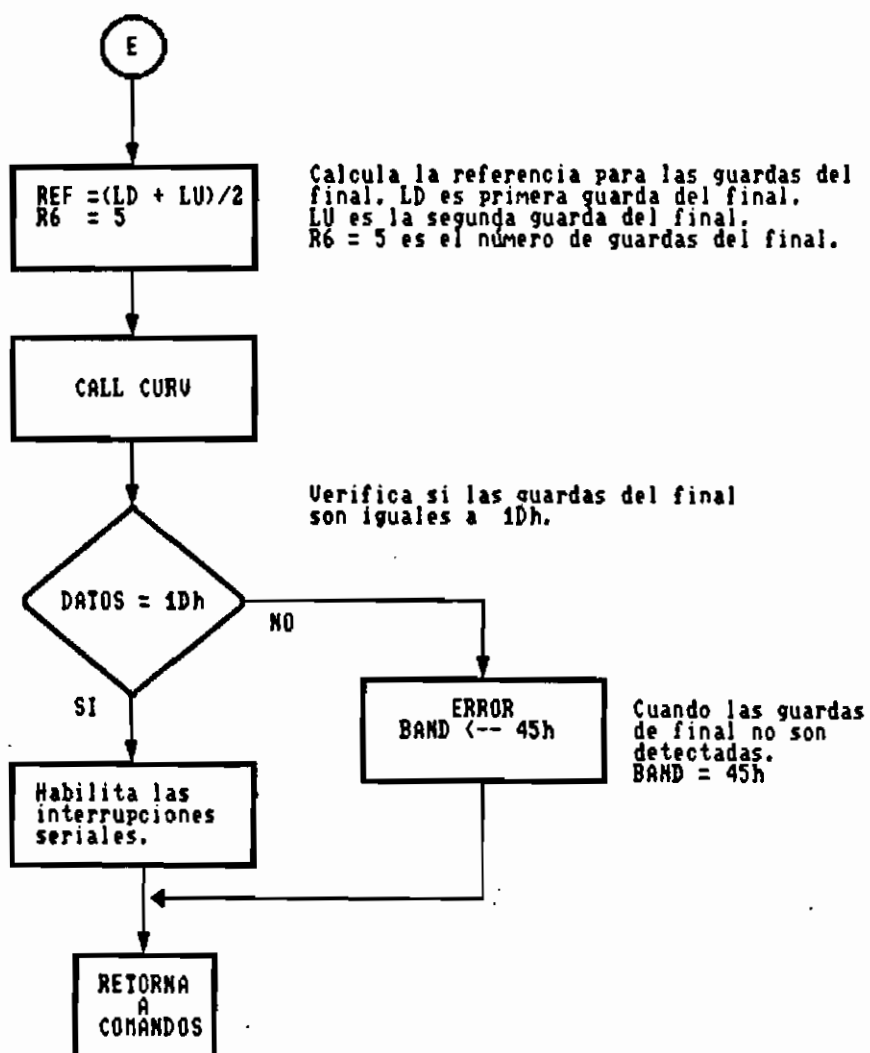




DIAGRAMA 05  
 RECUPERA LA INFORMACION BINARIA A PARTIR DE LAS LECTURAS  
 LECTURA IZQUIERDA  
 CONTINUACION



Cuando una de las secuencia de guarda es identificada, BAND es cargado con 49H si DATOS es 17H, 44H si es 0DH continuando la decodificación. El número de lecturas consideradas por cada segmento es el número de bits por caracter en NUM. La referencia de cada segmento se calcula llamando la rutina REFER, cuyos parámetros de entrada son: la dirección de inicio del segmento en R0, el número de bits por caracter en NUM1, registro que almacena el contenido de NUM de manera temporal (registro auxiliar). REFER devuelve la referencia en REF, calculada obteniendo la mayor y menor lectura del segmento, valores que corresponden a uno "1" y cero "0". Los registros AUX, MAYOR y MENOR son utilizados por REFER. Por comparación de cada lectura con las restantes se consigue separar el mayor en MAYOR y el menor en MENOR, donde REF será la media aritmética de estos dos valores. Este método es el más adecuado por las siguientes razones:

- La velocidad del "SCANNER" no es constante, iniciando con una mayor variación en las primeras lecturas y alcanzando una relativa estabilidad hacia el final de la etiqueta.
- Se conoce que el grupo de bits por caracter, cuenta con al menos un bit de valor uno "1" y uno de valor cero "0", con relación de 2 a 1 para el ancho de las

barras que los representan. Por tanto las lecturas deberán mantener esta misma relación.

- La variación de la velocidad en un segmento de la etiqueta es pequeña pudiendo asumirse que es constante. Es de esperar que lecturas correspondientes a bits del mismo valor sean similares: Si la variación está fuera del margen proporcionado por el algoritmo, el error es inevitable.

La normal terminación de "R", verifica los últimos bits que deberán corresponder a la guarda de final correspondiente, dependiendo de las guardas de inicio. Será 1DH si la de inicio es 0DH y 0BH si fue 17H. BAND tomará el valor 45H si la condición no se cumple. Las interrupciones del pórtico serial son habilitadas antes de retornar al lazo de espera de comandos.

"V", cuyos parámetros de entrada son: la dirección de inicio de la "TABLA" en R0, el número de lecturas en DAT1. La referencia de la verificación se calcula tomando el 10% del valor en la localidad 20H y cargandolo en REF. Todas las lecturas son comparadas con REF y una lectura menor denota un desperfecto en la etiqueta ó que la velocidad de exploración se hizo 5 veces mayor que la inicial. La

presencia de una lectura considerada errónea ocasiona que BAND tome el valor 46H, para identificar el tipo de error.

#### DIAGRAMA 03

"M", es la rutina más importante en la validez de una exploración. Si una lectura errónea ha logrado sobrepasar todas las demás trampas, ésto significa que al menos un bit de valor uno "1" ha sido interpretado como cero "0" ó viceversa, alterando el número de bits de valor uno "1" por caracter. Una lectura puede tomarse por válida sin serlo cuando un bit de valor uno "1" cambia a cero "0" y un bit cero "0" cambia a uno "1" en el mismo caracter. Los parámetros de entrada son: NUM2, NUM y DAT1. La presencia de un error es advertida cuando BAND toma el valor 45H.

#### DIAGRAMA 04

No obstante todas la verificaciones incorporadas, existe la posibilidad de tener una decodificación errónea tomada por válida. La confiabilidad del sistema está directamente relacionada por la ocurrencia de estos errores.

NOTA.- El comando "V" cumplirá eficientemente su tarea sólo si es ejecutado antes de modificar el contenido de la "TABLA", es decir después del comando "R" y antes del comando "C".

El mensaje del comando "M" sólo será útil si es ejecutado después de "C".

**DIAGRAMA 03**  
**LECTURAS MENORES 10% DEL LA INICIAL**

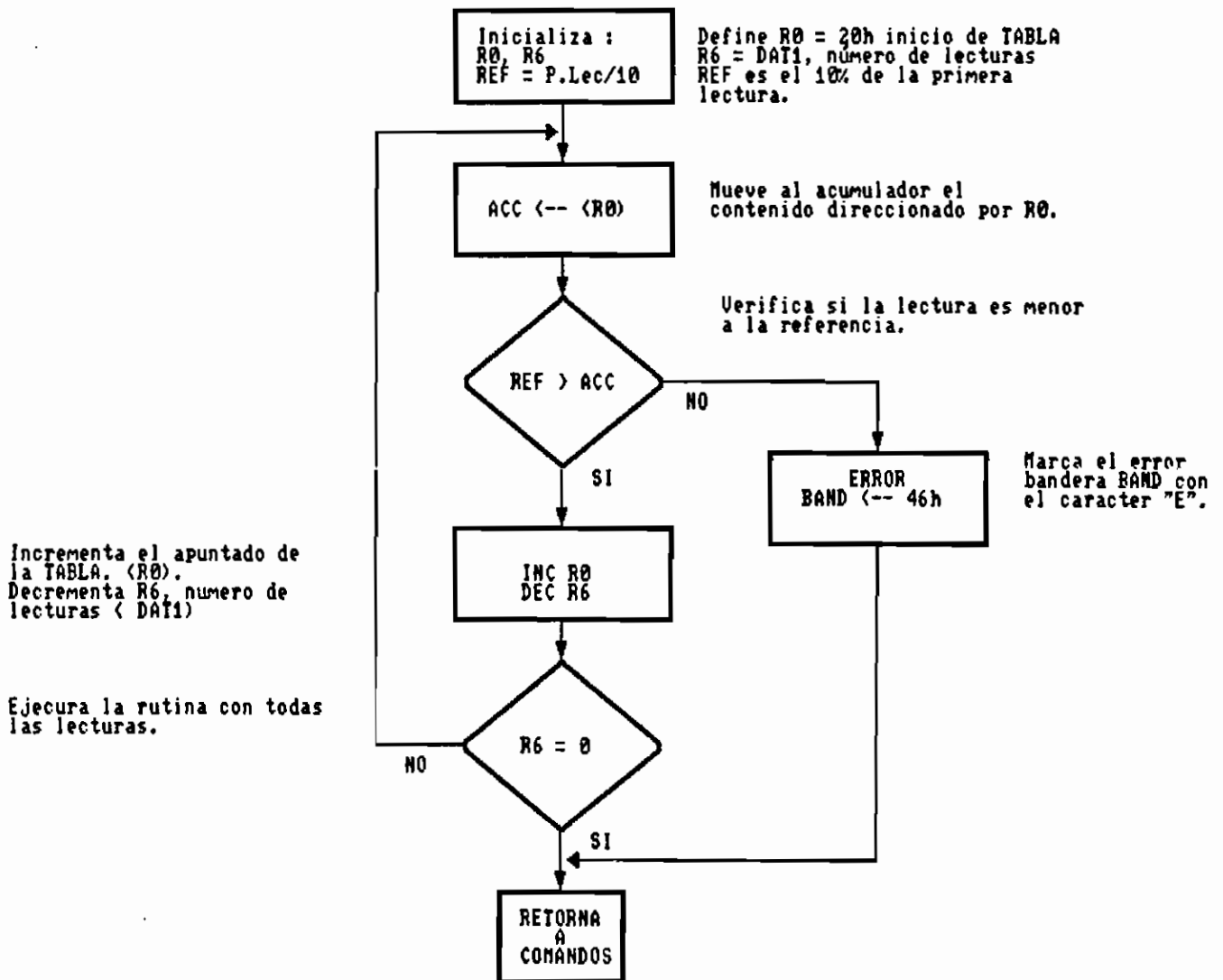


DIAGRAMA 04  
VERIFICA NUMERO DE BITS DE VALOR UNO

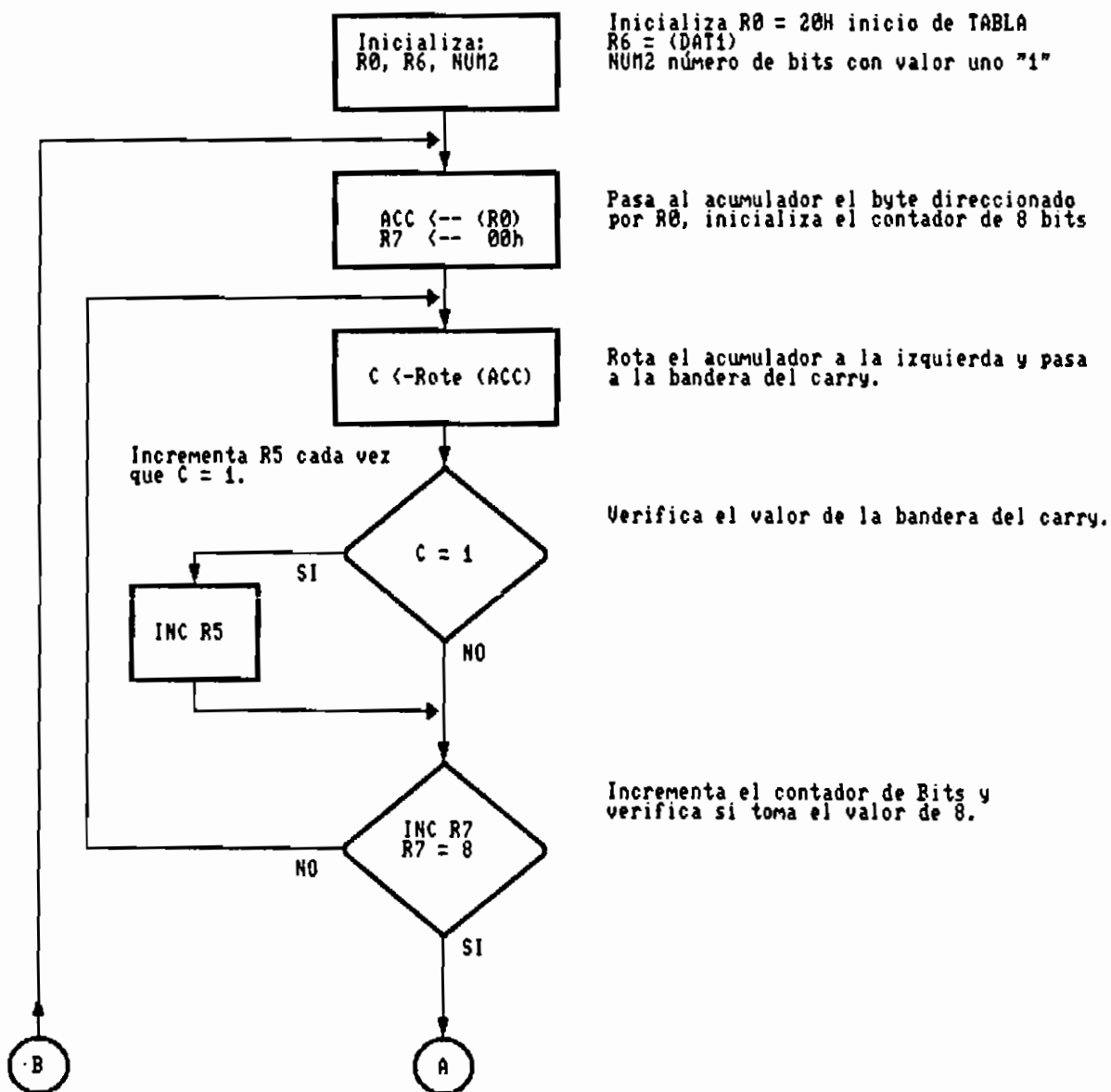
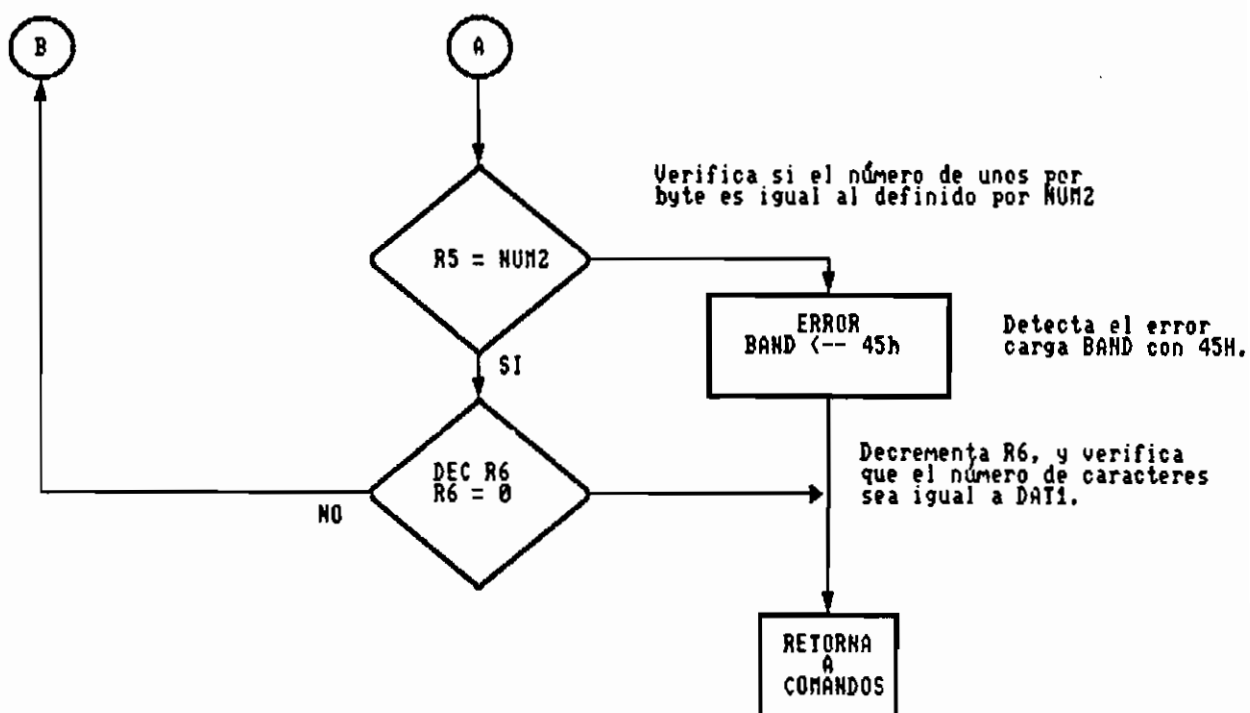


DIAGRAMA 04  
VERIFICA NUMERO DE BITS DE VALOR UNO  
CONTINUACION



## 3.4.1. DIAGRAMA FISICO.

La FIGURA 3.11 muestra una vista superior a escala 1:1 con la ubicación de los componentes.

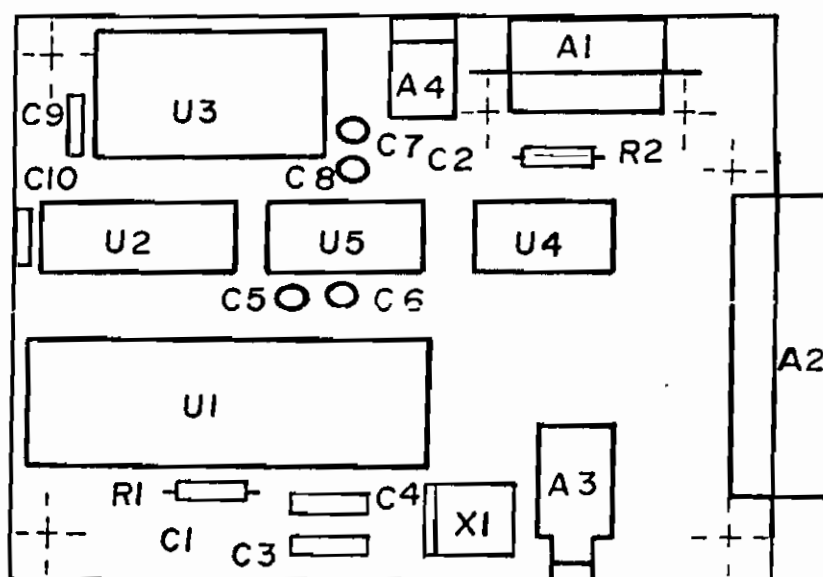


Fig 3.11. VISTA SUPERIOR DEL PROTOTIPO.

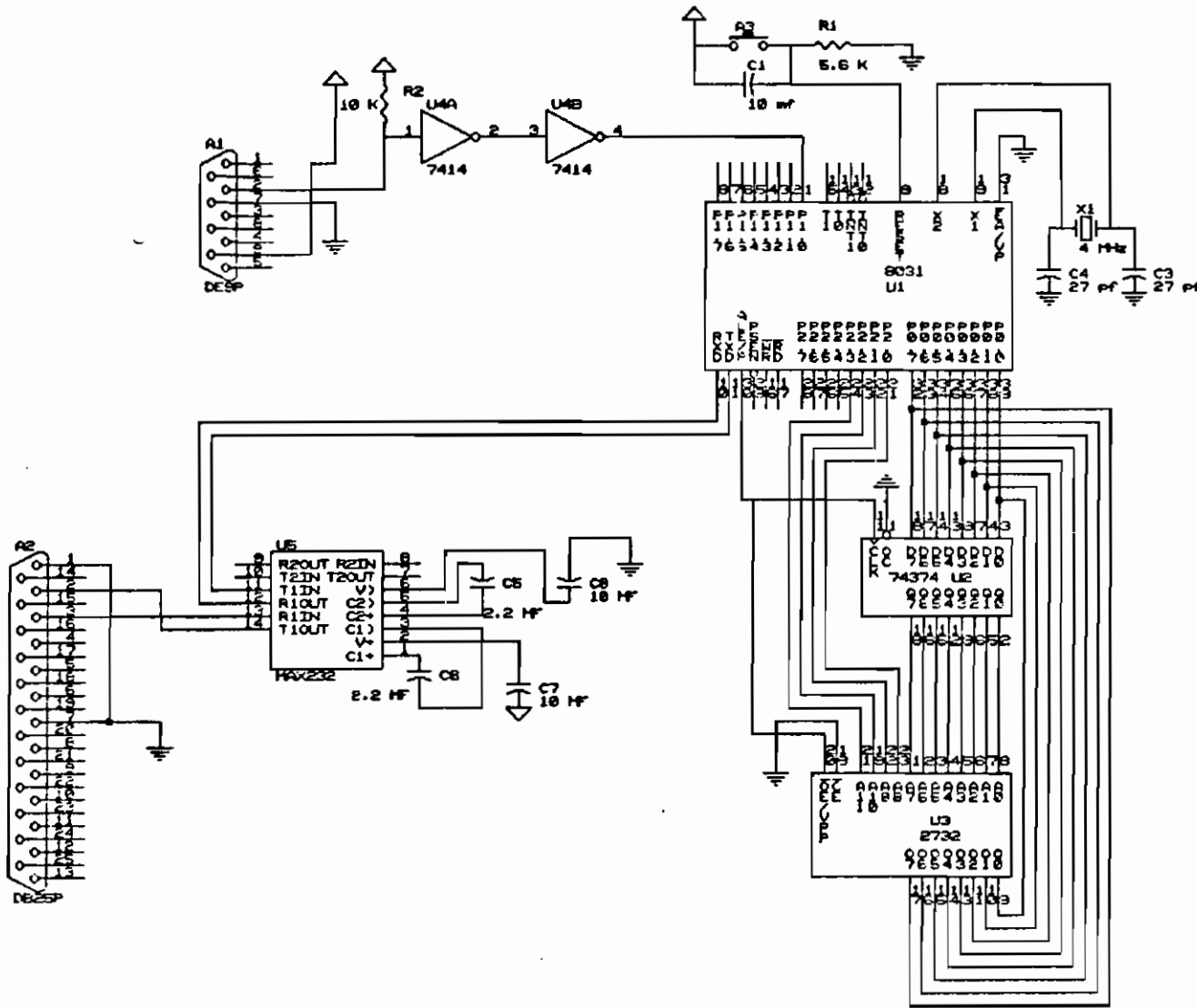


## 3.4.2. LISTA DE COMPONENTES.

U1	INTEL P8031	
U2	SN74LS374	
U3	HN462732G	
U4	DM7414N	
U5	MAX232CPE	
R1	6.8 K $\Omega$ - 1/4 W	
R2	10 K $\Omega$ - 1/4 W	
C1	10 $\mu$ F - 50V	electrolítico
C2	47 $\mu$ F - 50V	electrolítico
C3	27 pF - 5%	cerámico
C4	27 pF - 5%	cerámico
C5	4.7 $\mu$ F - 10V	tantalio
C6	4.7 $\mu$ F - 10V	tantalio
C7	10 $\mu$ F - 25V	tantalio
C8	10 $\mu$ F - 25V	tantalio
C9	33 nF	cerámico
C10	33 nF	cerámico
X1	4 MHz	
A1	Conector DE-9P	macho
A2	Conector DB-25P	macho
A3	Pulsante	
A4	Conector Vdc.	
A5	1.5 Mts. de cable con terminales DB-25S hembra. normalizado RS-232.	
A6	Transformador 110 Vac - 60 Hz a 5 Vdc. 300 mA. regulada.	
A7	Digital Wand. TRS - 80. modelo 100.	

## 3.4.3. CARACTERISTICAS ELECTRICAS.

Voltaje de alimentación	5 Vdc.
Consumo de corriente durante la exploración	300 mA.
Consumo medio	270 mA.
Potencia media.	1.35 W.



3.4.4. DIAGRAMA ESQUEMATICO

Title		
LECTOR DE CODIGO DE BARRAS		
Size	Document Number	REV
B	1	B1
Date:	November 29, 1998	Sheet 1 of 1

#### 4.1. DESCRIPCION DE LAS RUTINAS DE COMUNICACIONES

##### 4.1.1. RUTINA DE RECEPCION

En el microcontrolador INTEL 8031, la detección de una transición en el pin RXD, pone en "1" el bit RI del registro de SCOM, generando la interrupción serial cuya dirección de inicio es 0023H. La rutina inicia preservando el status, el bit RI se cambia de "1" a "0", (aun cuando esta instrucción no es necesaria puesto que la bandera es cambiada por el hardware una vez atendida la interrupción). El byte recibido en SBUF se transfiere a un registro auxiliar en la localidad 66H de la memoria RAM interna, con la etiqueta RX232.

```

RXRS:    PUSH    PSW
          CLR     RI
          MOV     RX232, SBUF
          POP     PSW
          RETI
```

##### 4.1.2. RUTINA DE TRANSMISION

La rutina de transmisión es utilizada para enviar un caracter almacenado en TX232 cada vez que es llamada.

## 4.2. SOFTWARE DE COMUNICACIONES PARA UN COMPUTADOR PC-XT

Un computador personal cuenta con circuitos especializados, para comunicaciones seriales como el USART (Universal Synchronous / Asynchronous Recieve / Transmitter). Circuitos que requieren de un proceso de inicialización. (establecido por el lenguaje utilizado para crear las rutinas de comunicaciones). Los parámetros definidos para el USART son: velocidad de transmisión, chequeo de paridad, número de bits de parada, número de bits transmitidos y determina las condiciones de operación de las señales de intercambio "handshake".

" Handshake. Es el proceso por el cual un aparato percibe el estado de otro y responde adecuadamente"<sup>6</sup>

### 4.2.1. INSTRUCCIONES UTILIZADAS EN COMUNICACIONES

Como se mencionó anteriormente, el presente trabajo está desarrollado en Basic. Las sentencias e instrucciones disponibles para intercambio de información, con pórticos y archivos son de propósito general. Las rutinas implantadas para la comunicación entre el computador y el circuito digital utilizan las siguientes sentencias e instrucciones:

---

<sup>6</sup>El libro del RS 232 Pag 46.

"OPEN COM(n)" permite inicializar el p3rtico, definiendo los par3metros y el modo de operaci3n. (ver apendice D)

"PRINT #(n),{variable}" escribe en la cola de transmisi3n asociada al p3rtico. Los caracteres son transmitidos inmediatamente a la velocidad binaria definida. (ver apendice D)

INPUT\$(x,#(n)) lee un n3mero "x" de caracteres de la cola de recepci3n del p3rtico. (ver apendice D).

Para definir el modo de intercambio de informaci3n debe considerarse la cantidad de datos que intervienen en la comunicaci3n entre el lector y el computador. Siendo muy pocos los datos que identifican una etiqueta y tomando en cuenta que la informaci3n contiene los elementos necesarios para validaci3n de la misma, no se hace imprescindible utilizar un protocolo de comunicaciones de alto nivel; siendo el protocolo implantado basado en el formato del RS232-C.

El n3mero m3ximo de caracteres representados en el c3digo de barras implantado es 6, requiriendo su decodificaci3n de un n3mero igual de bytes de informaci3n. Sin embargo como puede

deducirse de la sección 3.3.3, la comunicación serial entre el computador personal y el circuito digital mantiene un intercambio de comandos y mensajes de error, que permiten hacer decisiones sobre la validez de una lectura.

Los comandos son caracteres enviados por el computador personal, que generarán una interrupción serial en el circuito digital si está habilitada ésta posibilidad ó en caso contrario son ignorados. La secuencia de comandos está determinada por BSDEC.EXE, que es la rutina de decodificación.

#### 4.2.2. INICIALIZACION DEL PORTICO SERIAL

La comunicación serial en el computador se inicia cuando el pòrtico de comunicaciones es abierto, utilizando la sentencia OPEN COM(n), con la cual se definen características como: velocidad binaria, número de bits transmitidos, bits de parada, paridad, tipo de acceso y la utilización de las señales de "handshake". (ver apéndice D)

Las características deben ser iguales a las definidas para el puerto serial disponible en el circuito digital. Por tanto el pòrtico de comunicacion serial es inicializado como sigue: 8 bits de datos, sin verificación de paridad, un bit

de parada, 1200 bits por segundo, acceso aleatorio, chequeo de la señal "request to send". Esta última es utilizada para realizar una verificación de la instalación física del circuito gracias al puente entre los pines 6 y 20 del conector en el equipo periférico. (ver apéndice E)

Sentencia OPEN COM(n):

```
OPEN "com1:1200,n,8, , rs,cs0,ds0," FOR RANDOM AS #1
```

#### 4.2.3. RUTINA DE COMUNICACIONES

La rutina de comunicación "ENV", escribe y lee un caracter de la cola respectiva (transmisión o recepción) cada vez que es llamada. "ENV" envía el valor ASCII correspondiente a la letra mayúscula del comando deseado, utilizando la instrucción "PRINT #1, Ke\$". El proceso de transmisión requiere de un tiempo de aproximadamente 10 mS por byte. Si el comando obtiene como respuesta un caracter, se requerirá de otros 10 mS además del tiempo que el circuito digital tome en la ejecución de la rutina antes de responder. Por ésta razón se considera un retardo, el cual se genera con valor aproximado de 50 mS. El retardo no deberá depender de la velocidad del computador y se lo consigue leyendo el reloj del computador con la sentencia "TIMER", tomando éste valor como referencia en un lazo y cuya condición de salida

será el nuevo valor sumado al tiempo de retardo que "TIMER" deberá alcanzar antes de retornar de la subrutina de retardo.

Retardo:

```
Inicio = TIMER
DO UNTIL (TIMER - Inicio > 0.05)
LOOP
```

El tiempo es suficiente para que dos bytes a una velocidad de 1200 bits por segundo sean transmitidos y cerca de 500 instrucciones del programa del INTEL 8031 sean ejecutadas utilizando un cristal de 4MHz en el circuito lector. Esto permite que antes que la siguiente instrucción de BSDEC.EXE se ejecute, las rutinas desencadenadas por los comandos "N", "S", "B" y "T"; hayan concluido. Cabe destacar que sólo los comandos mencionados responden con la transmisión de un byte y que son las únicas que permiten al computador indagar sobre el estado del circuito así como sobre los códigos recuperados.

La instrucción LOC(n), recupera el número de bytes presentes en la cola de recepción. Si el valor de LOC es cero indica que no existe respuesta del circuito digital, situación normal si el comando no es alguno de los antes



mencionados. Cuando LOC(n) es diferente de cero el caracter es recuperado con "INPUT\$".

#### 4.2.4. PROTOCOLO DE COMUNICACION

El proceso de comunicación se basa en el conocimiento de la función que realizan las rutinas del circuito digital cuando un comando es enviado.

Los comandos: "1", "2", "3", "4", "6" y "8" no tiene una respuesta especifica y no requieren confirmación.

El comando "N" utilizado para verificar que la comunicación en los dos sentidos ha sido establecida, tiene como respuesta el caracter "H".

Los comandos "R", "C", "V" y "M", si bien no tiene una respuesta propia, como resultado de su ejecución pueden ser utilizados posteriormente los comandos "S", "B" o "T" para indagar el estado del circuito digital. Los comandos "R" y "C" por la función que realizan y por ser los más extensos, deshabilitan la posibilidad de interrupción serial. BSDEC.EXE entra en un lazo que envia consecutivamente el comando "B" hasta recibir respuesta; lo cual ocurrirá cuando

las interrupciones sean habilitadas una vez concluida la rutina.

El caracter recuperado con el comando "B" muestra la condición de la rutina ejecutada en el circuito digital. La acción a tomarse está relacionada con la validez de la lectura.

La FIGURA 4.1 presenta en forma resumida los caracteres utilizados para los comandos enviados por el computador al circuito digital y la respuesta de éste en cada caso.

Rutina	Respuesta	Significado
"R"	1	Lectura no válida
"R"	0	Lectura válida
"V"	F	Etiqueta defectuosa
"C"	D	Lectura de Der - Izq
"C"	I	Lectura de Izq - Der
"C"	E	Guardas no detectadas
"M"	E	ERROR en formato

Fig 4.1 CARACTERES DE RESPUESTA PARA COMANDO "B"

En el microcontrolador las rutinas de comunicaciones discutidas en el capítulo tercero, están estructuradas de

forma que sólo un caracter es recibido o transmitido cuando RXRS o TXRS son llamadas.

#### 4.3. IDENTIFICACION DE CODIGOS EN EL COMPUTADOR

Para entender de una mejor manera la decodificación es necesario antes tener una visión global del problema, esto es conocer la estructura de una etiqueta, características del código utilizado, el mapa de memoria del archivo con la tabla de equivalencias, el circuito digital que adapta la señal, el soporte de transmisión de datos y el lenguaje de programación.

Una primera etapa en el proceso de decodificación consiste en la adquisición de datos. La información que se irá almacenando corresponderá a la obtenida en la exploración de las etiquetas con el lápiz óptico, registrando el tiempo de duración de cada barra ó espacio.

El algoritmo implantado elimina las guardas y recupera la información binaria. Esta etapa requiere del concurso de BSDEC.EXE como se explicó en la sección 3.3.4 y determina las acciones a tomarse. Cuando la exploración es errónea inmediatamente envia los comandos correspondientes para que

un nuevo intento se realice. La detección de una exploración válida determina la terminación normal de ésta etapa y el inicio de la siguiente, en la cual los caracteres enviados por el circuito digital son procesados para obtener el valor decimal correspondiente al código ASCII; los cuales serán utilizados como índices del archivo "TABLA.CBR". Una de las regiones del mapa de memoria asocia los caracteres código y el valor decimal de la secuencia asignada durante su edición.

Los caracteres decodificados por este método se someten a un procesamiento literal para formar una palabra código, que podrá ser utilizada por una base de datos como código alfanumérico.

En el ejemplo se ilustra la identificación del carácter A, su código ASCII es 65h, el código binario asociado es 00000111; siendo además el primer carácter en la tabla.

Para comprender de mejor manera el ejemplo, se detalla la estructura del mapa del archivo "TABLA.CBR" (tabla de equivalencias). La información en las tres primeras localidades del archivo "TABLA.CBR" es:

nombre del archivo	TABLA
# de bits por caracter	8
# de unos por caracter	3
# de caracteres	*

El código ASCII (65) correspondiente al caracter "A" se encuentra almacenado en las localidades 301 y 407 del archivo "TABLA.CBR", como se aprecia en la FIGURA 4.2. La asignación de estas localidades se la realiza de la siguiente manera: al segmento del archivo cuyo inicio está en la dirección 300, se le suma la posición del caracter escogido "A", que para el ejemplo la dirección resultante es 301. La posición del código ASCII en el segmento cuyo inicio es 400 se determina sumando el valor decimal del código binario "00000111", / "7", asociado al caracter "A" escogido. figura 4.2

El código binario "00000111" asignado al caracter "A", se encuentra en la localidad 65 de la tabla, FIGURA 4.2.

La sección de la etiqueta correspondiente al caracter "A", se muestra en la FIGURA 4.3 y la lectura promedio efectuada de izquierda a derecha da como resultado: 36, 34, 29, 30, 28, 73, 70, 69. La ubicación en la memoria del circuito INTEL 8031 dependerá la posición que ocupe el caracter en la

etiqueta, pudiendo tomar las ubicaciones siguientes: 26h, 2Eh, 36h, 3Eh, 46h o 4Eh que se obtienen para este ejemplo sumando a la dirección de inicio del segmento "TABLA" (20h), el número de bits de la guarda izquierda y el número de bits por caracter.

01		Número de caracteres
03		Número de unos "1"
65	. 00000111 .	Palabras código ordenadas por el código ASCII.
300		Caracteres ordenados por las posición.
301	65 .	
400		Caracteres ordenados por el valor binario del código.
407	65 .	

Fig 4.2 MAPA DEL ARCHIVO.

El algoritmo de decodificación en el circuito digital calcula la referencia a partir de la mayor y la menor lecturas del grupo, obteniendo la media aritmética:

$$\text{REF} = ( 28 + 73 ) / 2$$

$$\text{REF} = 4Dh$$

Nota.- Todas las operaciones y los valores de las lecturas son números hexadecimales.

La comparación se realizará de cada lectura con el valor de REF, si REF es mayor el bit será cero "0" y si es menor será uno "1".

REF	Lectura	Bit
4D	36	0
4D	32	0
4D	29	0
4D	30	0
4D	28	0
4D	73	1
4D	70	1
4D	69	1

El valor recuperado en "DATOS" es "00000111", ó 07,, donde el número de unos "1" es tres. Este valor es transmitido al computador, que lo recupera como caracter cuyo código ASCII es 07 " ". El valor decimal 7 sumado al inicio del bloque de dirección 400 da como resultado 407, que constituye el índice en el archivo "TABLA.CBR", localidad donde está almacenado el código ASCII del caracter "A".

## 5.1. PRUEBAS DE FUNCIONAMIENTO

Se realizan dos pruebas de evaluación: la primera permite establecer el grado de confiabilidad en la decodificación de una etiqueta y la segunda evalúa los resultados obtenidos al trabajar con un impresor matricial en la creación de etiquetas de códigos de barras, ilustrando la variación de la velocidad de exploración durante la lectura.

### 5.1.1. PRUEBA #1

#### 5.1.1.1. OBJETIVO

Establecer en forma porcentual la frecuencia de ocurrencia de errores tipificados.

#### 5.1.1.2. CONSIDERACIONES

- Las etiquetas utilizadas (ocho) en ésta prueba son las mismas para todos los encuestados.
- El número de caracteres de cada etiqueta es variable y el orden en que son muestreadas es aleatorio.
- El sentido de exploración es indistintamente escogido por el usuario.
- El número de muestras por usuario no es fijo.



### 5.1.1.3. ERRORES TIPIFICADOS

- "ER1".- Terminación anormal de la exploración de la etiqueta la cual ocurre cuando el lápiz óptico no cumple el ciclo de exploración, es decir: El detector sale del campo de la etiqueta, es levantado o apagado antes de terminar el barrido, o la velocidad de exploración está fuera del rango mínimo.
  
- "ER2".- Lectura con una dimensión registrada de menos de 10% del ancho de la primera barra, producido por una variación significativa de la velocidad durante el muestreo, o por una etiqueta defectuosa.
  
- "ER3".- Ausencia de secuencias de inicio y/o fin. Las secuencias establecidas para inicio o fin no son reconocidas correctamente.
  
- "ER4".- Caracter no compatible con reglas de decodificación, alguno o algunos de los caracteres recuperados tienen una representación binaria diferente a la establecida en la impresión. Es muy importante verificar que el formato de decodificación sea el mismo de impresión.

- "ER5".- Decodificación errónea, la cual ocurre cuando caracteres son cambiados unos por otros.

La FIGURA 5.1 muestra las etiquetas utilizadas por los usuarios durante la prueba.



Fig 5.1 ETIQUETAS USADAS EN LA PRUEBA #1

## 5.1.1.4. RESULTADOS

La TABLA 5.1 presenta los resultados obtenidos al ser leídas las etiquetas por 8 usuarios diferentes. El número de lecturas realizadas y el sentido de exploración consideradas por los usuarios para cada una de las 8 etiquetas disponibles para la prueba, han sido consideradas con el propósito de simular condiciones similares a las existentes en una aplicación práctica.

# de L	ER1	%	ER2	%	ER3	%	ER4	%	ER5	%
98	1	1.02	1	1.02	9	9.18	18	18.3	1	1.02
44	5	11.3	0	0.00	4	9.09	7	15.9	0	0.00
53	0	0.00	0	0.00	3	5.66	5	9.43	0	0.00
65	1	1.53	0	0.00	10	15.3	8	12.3	0	0.00
46	4	8.69	2	4.34	8	17.3	0	0.00	0	0.00
37	1	2.7	0	0.00	7	18.9	5	13.5	0	0.00
25	0	0.00	1	4.00	3	12.0	2	8.00	0	0.00
13	2	15.3	0	0.00	3	23.0	0	0.00	0	0.00
TOTALES										
381	14	3.67	4	1.04	47	12.3	45	11.8	1	0.26

Tabla 5.1 RESULTADOS DE LA PRUEBA #1.

Analizando la FIGURA 5.2 con los resultados obtenidos de la prueba #1 se concluye:

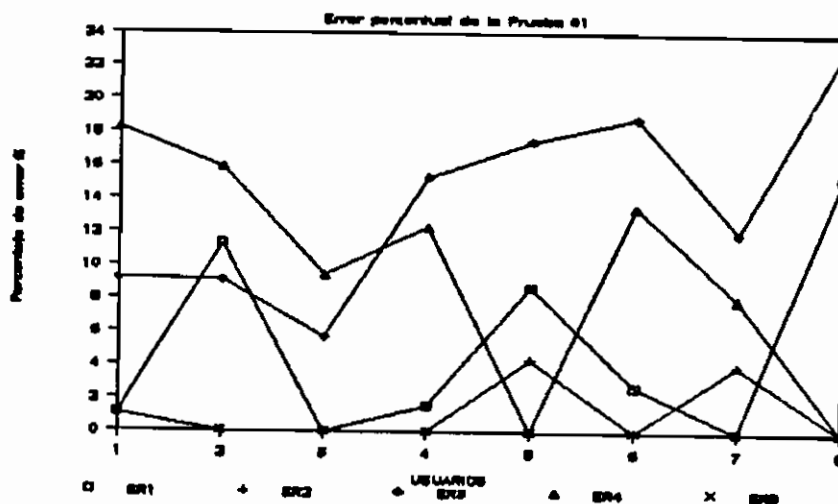


Fig 5.2 RESULTADOS DE LA PRUEBA #1

- El porcentaje de error por terminación anormal (ER1), de la exploración muestra que la altura de las barras (8 mm), es suficiente para que los usuarios permanezcan dentro del campo de la etiqueta.
- Las etiquetas utilizadas para la realización de la prueba no presentan defectos considerables, pues el porcentaje de error es muy reducido. (ER2)
- La tasa de error relacionada con la detección de las guardas de inicio y fin (ER3) es la más alta respecto a las demás. Esto se explica en razón de que los

errores tipificados como ER1, (terminación anormal) puede ser sobrepasado si la terminación se realiza en una barra, convirtiéndose éste criterio de error en uno mucho más efectivo que los anteriores.

- El error tipificado como caracter fuera de formato (ER4) es el segundo en ocurrencia. La frecuencia está directamente relacionada con el algoritmo de detección y la variación de la velocidad. Una gran variación de la velocidad en pequeños tramos de la etiqueta es la causa más frecuente para este error.
  
- El error en la detección (ER5) presenta el porcentaje más bajo en relación a los demás. La confiabilidad del sistema en general está determinado por la ocurrencia estos errores, cuyo porcentaje es de 0.26%.

## 5.1.2. PRUEBA #2

### 5.1.2.1. OBJETIVO

Determinar la característica de impresión de las etiquetas y la variación de la velocidad de exploración.

## 5.1.2.2. CONSIDERACIONES

- Las etiquetas están formadas por barras y espacios del mismo ancho.
- El período entre cada interrupción generada por el temporizador "TIMER 0" se calcula tomando un doceavo de la frecuencia del cristal utilizado por el INTEL 8031 (4 MHz) y el valor hexadecimal cargado en TH0 (D7H). La Ec 5.1 permite calcular el período.

$$t = T * 12 * (FFH - TH0) \quad \text{Ec. 5.1}$$

Donde:

t Período de las interrupciones (120  $\mu$ S)

T Período del oscilador (.25  $\mu$ S)

TH0 Valor de inicio del TIMER 0 (D7H)

- La longitud total de la etiqueta se calcula multiplicando el número de barras y espacios por el ancho de una barra. Ec 5.2

$$\text{Long.} = A * \# "b"$$

Ec. 5.2

Donde:

Long.      Longitud total de la etiqueta  
 A          Ancho de una barra o espacio.  
 # "b"      Número de barras y espacios.



Fig 5.3 (a) ETIQUETA CON BARRAS DE ANCHO 2.544 mm.



Fig 5.3 (b) ETIQUETA CON BARRAS DE ANCHO 1.276 mm.



Fig 5.3 (c) ETIQUETA CON BARRAS DE ANCHO 0.636 mm.

- La velocidad de exploración se calcula registrando el número de interrupciones por cada barra o espacio, puesto que es conocido el periodo entre ellas y la distancia recorrida. La velocidad promedio se determina por la Ec. 5.3.

$$\text{Velc} = A / ( \# "i" * t ) \quad \text{Ec.5.3}$$

Donde:

Velc.	Velocidad de exploración del lápiz óptico.
A	Dimensión del ancho de una barra o espacio.
# "i"	Número de interrupciones por barra o espacio.
t	Periodo de una interrupción.

- La estructura de las rutinas desarrolladas para el INTEL 8031 permite configurarlas de manera que es posible recuperar las lecturas correspondientes a barras y espacios sin procesarlas, permitiendo contar con la información necesaria para calcular la velocidad promedio para cada barra o espacio.

Los resultados de las pruebas realizadas se presentan en forma gráfica en las FIGURAS 5.4, 5.5 y 5.6.



La FIGURAS 5.4 (a) y 5.4 (b) se obtienen realizando la exploración de la etiqueta de la FIGURA 5.3 (a) donde la dimensión de barras y espacios es 2.544 mm. El número de barras es 17.

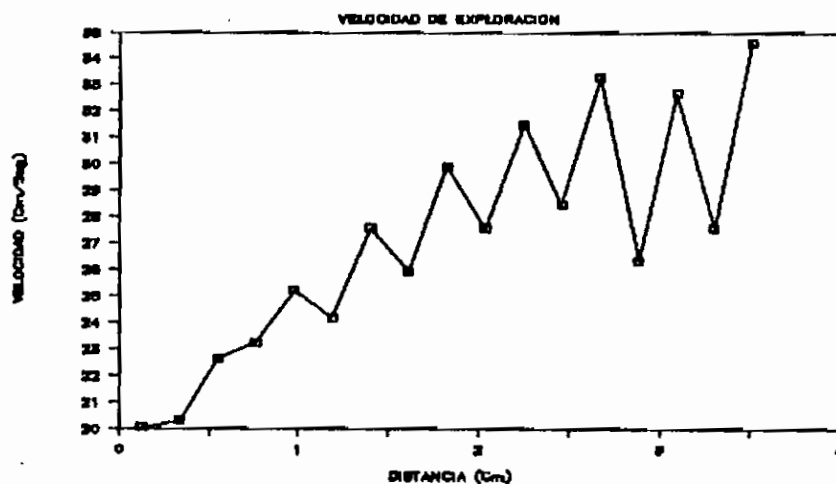


Fig 5.4 (a) RESPUESTA DE VELOCIDAD

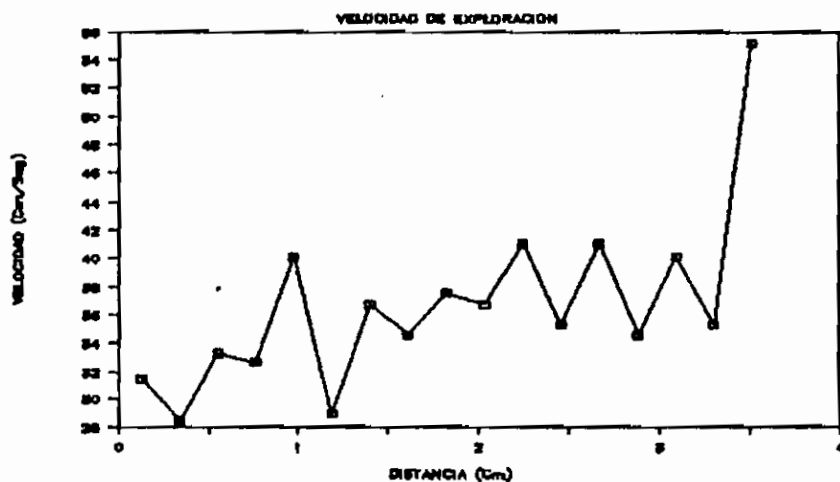


Fig 5.4 (b) RESPUESTA DE VELOCIDAD

LEETRA DE CODIGO DE BARRAS

Las FIGURAS 5.5 (a) y 5.5 (b) corresponden a lecturas efectuadas en la etiqueta de la FIGURA 5.3 (b), la dimensión de barras y espacios es 1.276 mm. El total de lecturas es 33.

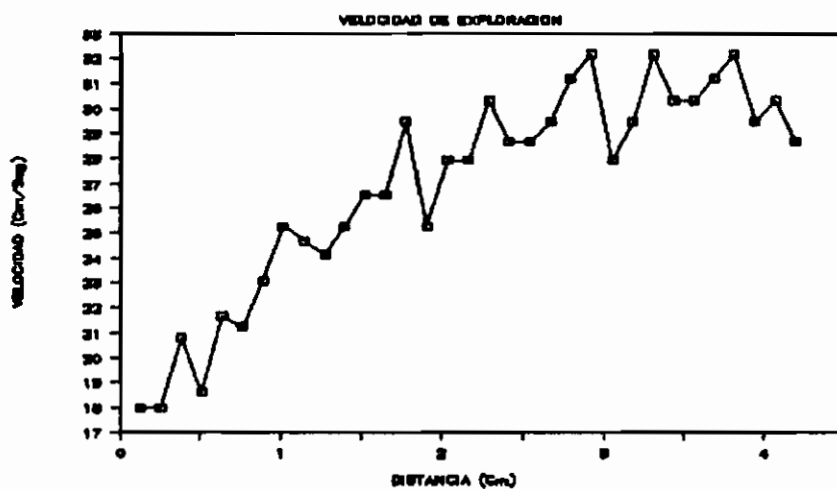


Fig 5.5 (a) RESPUESTA DE VELOCIDAD

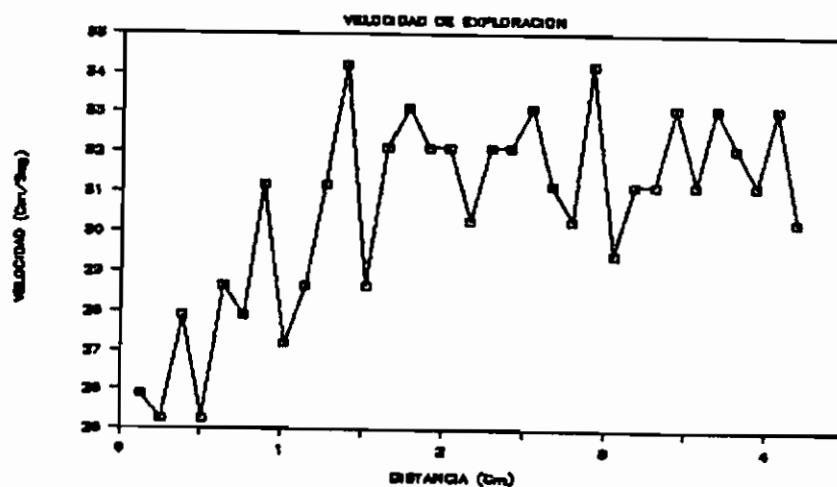


Fig 5.5 (b) RESPUESTA DE VELOCIDAD

Las FIGURAS 5.6 (a) y 5.6 (b), representan la velocidad media durante la exploración de 33 barras y espacios, cuya dimensión es 0.636 mm, obtenidas de las exploraciones de la etiqueta de la FIGURA 5.3 (c).

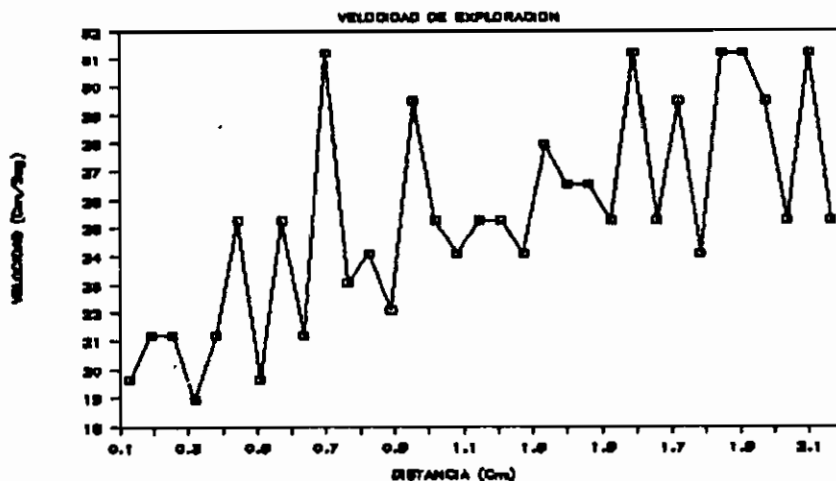


Fig 5.6 (a) RESPUESTA DE VELOCIDAD

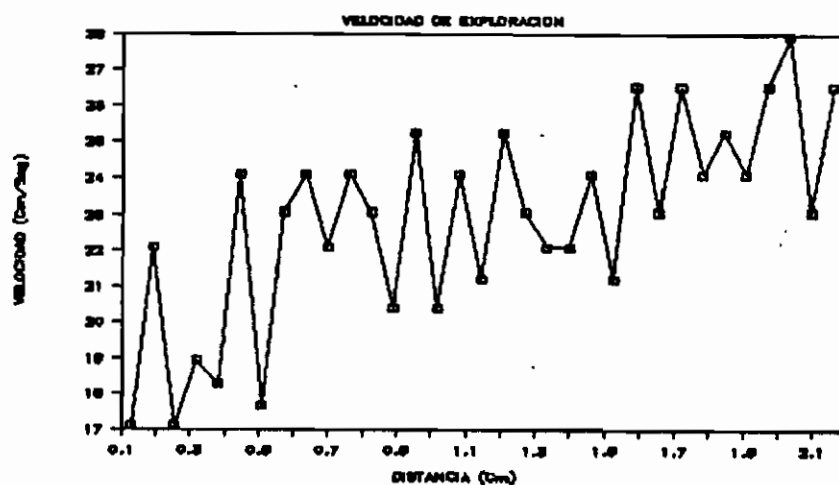


Fig 5.6 (b) RESPUESTA DE VELOCIDAD

Del análisis de resultados, se concluye:

- La velocidad media de exploración no es constante, observándose un nivel *bajo* en la parte inicial de la etiqueta y alcanzando una relativa estabilidad muy rápidamente.
  
- La estructura de las etiquetas presenta un formato que empieza y termina con una barra, por tanto se puede determinar exactamente cuales lecturas corresponden a barras y cuales a espacios. Las gráficas muestran un valor menor para los espacios en comparación al registrado para las barras adyacentes. Este fenómeno tiene relación con el sistema de impresión, pues las barras resultantes son de un mayor ancho que los espacios, aun cuando el mecanismo de impresión se desplaza con intervalos iguales de 0.317 mm. Esto se explica porque la acción de una aguja en la cabeza de impresión sobre la cinta, dibuja un punto cuyo diámetro es mayor que el de la aguja mismo, aumentando el ancho de las barras en forma casi imperceptible a simple vista.
  
- Se observa en algunas de las gráficas discontinuidades en la velocidad, con valores que superan en mucho a los

obtenidos con el ensanchamiento de las barras. Tiene su origen en los defectos de impresión. Refiérase a la sección 3.1 y la FIGURA 3.2.

- La velocidad para los ejemplos varia en el rango entre 17 Cm/Seg y 55 Cm/Seg, que están dentro del rango permitido por el lápiz óptico. (apéndice G).

## 5.2. ANALISIS DE RESULTADOS

Todos los archivos ejecutables utilizados por el sistema incluyen rutinas para la detección automática de las unidades disponibles. Con el propósito de tener un rápido acceso al archivo con la tabla de equivalencias "TABLA.CBR", se lo copia a la unidad virtual (disco en memoria), incrementando la velocidad durante las consultas a "TABLA.CBR". Por ésta razón se escogerá siempre la unidad de disco virtual, la cual es creada modificando el archivo de configuración "CONFIG.SYS" si no se encuentra presente al ejecutar el programa BSCOD.EXE.

La presentación de mensajes de error y la posibilidad de corregirlos sin tener que abandonar la rutina, incrementan de manera considerable la extensión de los programas fuentes

y de los ejecutables; los cuales han sido realizados como módulos independientes con el propósito de utilizarlos en forma individual.

El procesamiento literal de los argumentos en cada módulo incrementa aun más su tamaño, convirtiendo rutinas relativamente sencillas en programas extensos.

La presencia de las instrucciones a las que se ha hecho referencia en los párrafos anteriores bien pueden ser considerados redundantes como parte de todos los módulos ejecutables, más permiten la utilización de BSDEC.EXE, BSIMP.EXE y BSGEN.EXE en forma totalmente independiente, no así BSCOD.EXE que requiere de BSDEC.EXE para la decodificación de una etiqueta y de BSIMP.EXE para la creación de la misma.

Las pruebas realizadas muestran un bajo porcentaje de error y una confiabilidad global del sistema aceptable.

Se comprobó además que el número de exploraciones consideradas erróneas disminuye apreciablemente cuando el usuario del sistema adquiere la destreza necesaria para mantener la velocidad aproximadamente constante durante la exploración.

### 5.3. ANALISIS ECONOMICO

Los costos efectuados en el desarrollo de cualquier prototipo no son exclusivamente aquellos ocasionados por materiales y componentes utilizados sino también el tiempo invertido en su desarrollo.

En esta sección se hace referencia en forma comparativa a los costos efectuados para implementación del prototipo del sistema lector de código de barras con perspectivas a una producción masiva de equipos de similares características.

Con el propósito de ilustrar el costo de un sistema lector de código de barras se incluyen datos referenciales de equipos comerciales tales como:

La compañía "International Technologies and Systems Corp. (ITS)", ha puesto en el mercado un sistema lector de código de barras cuyas características y costos se detallan seguidamente.

- Software para generación de código de barras usado en impresor matricial para los códigos: 3 de 9, CODA BAR, 2 de 5, UPC-A y 128. Cuyo precio de lista es 125 dólares y un costo de 69 dólares para distribuidores.

- Lápiz lector de código de barras con las siguientes características: emulación de teclado, software compatible con IBM PC/XT, PC/AT y clones. Con un precio de lista de 495 dólares, y 295 dólares para distribuidores.\*

Por su parte "Worthington Data Solutions" ha propuesto un sistema comercial para edición de todo tipo de etiquetas, cuyas características se presentan a continuación.

- Software para etiquetas, usado en impresores láser y matriciales como: Epson, IBM, Okidata y Laser Jet. Facilidades: flexibilidad de formato, múltiples tamaños, etiquetas de seguridad, códigos de barras como: 2 de 5, UPC/EAN, MIL-STD, AIAG, Código 39, etc. Su costo es de 279 dólares y 49 dólares adicionales por otros códigos de barras.
- Lector de código de barras cuyo software es compatible con PC6XT, AT y PS/2. Conectado como un segundo teclado con un costo de 399 dólares para la versión compatible con PS/2 y 385 dólares para la compatible con PC/XT/AT.†

---

\*PC MAGAZINE MARZO 29, 1989.

†PC MAGAZINE MARZO 29, 1989.



Seguidamente se presenta un detalle de los componentes utilizados y el precio unitario en dólares, dado que no son fabricados en el país a la fecha. El valor unitario de los componentes se vería reducido considerablemente hasta en un 50% para una producción masiva.

La construcción del prototipo utilizando la técnica conocida como "Wire Wrap" para el circuito requiere de los siguientes componentes:

Lista de componentes	Precio unitario
S1 Socket 40 pines (wire wrap)	2.29
S2 Socket 24 pines (wire wrap)	1.35
S3 Socket 20 pines (wire wrap)	1.19
S4 Socket 16 pines (wire wrap)	0.49
S5 Socket 14 pines (wire wrap)	0.45
S6 Pc Board	3.75
TOTAL (Dólares)	9.52

Los costos ocasionados en la elaboración del circuito impreso dependen de la tecnología utilizada.

## Lista de componentes.

Precio unitario  
de catálogo.\*

U1	INTEL P8031	3.95
U2	SN74LS374	1.49
U3	HN462732G	4.25
U4	DM7414N	0.49
U5	MAX232CPE	6.00
R1	6.8 K $\Omega$ - 1/4 W	----
R2	10 K $\Omega$ - 1/4 W	----
C1	10 $\mu$ F - 50V      electrolítico	0.18
C2	47 $\mu$ F - 50V      electrolítico	0.18
C3	27 pF - 5%      cerámico	0.08
C4	27 pF - 5%      cerámico	0.08
C5	4.7 $\mu$ F - 10V      tantalio	0.31
C6	4.7 $\mu$ F - 10V      tantalio	0.31
C7	10 $\mu$ F - 25V      tantalio	0.39
C8	10 $\mu$ F - 25V      tantalio	0.39
C9	33 nF      cerámico	----
C10	33 nF      cerámico	----
X1	4 MHz	1.59
A1	Conector DE-9P macho	1.39
A2	Conector DB-25P macho	2.39
A3	Pulsante	0.39
A4	Conector Vdc.	----
A5	1.5 Mts. de cable con terminales DB-25S hembra. (normalizado RS-232.)	7.00
A6	Transformador 110 Vac - 60 Hz a 5 Vdc. 300 mA. regulada.	2.95
A7	Digital Wand. TRS - 80. modelo 100.	80
TOTAL (Dólares)		111.09

Considerando los valores del equipo comercial en forma comparativa a los costos ocasionados en la implementación del prototipo, se concluye que:

---

\*Catálogo JAMECO Electronics. 1989.

- La mayor parte del costo de un equipo comercial tiene relación con los gastos incurridos en su desarrollo y los costos de operación de una compañía de diseño electrónico.
- La producción masiva del sistema lector de código de barras reduciría los costos de producción, manteniendo las características del sistema.
- La reducida circuitería requerida para el circuito decodificador, justificaría una producción a gran escala de circuitos integrados VSLI de propósito específico.
- Es posible obtener un alto margen de utilidad en la comercialización del sistema lector de código de barras si la producción es masiva.

#### 5.4. CONCLUSIONES Y COMENTARIOS

El diseño y construcción de la parte física del sistema lector de código de barras requiere del uso imprescindible de un elemento cuyas características permitan detectar perturbaciones muy pequeñas, logrando así disminuir

considerablemente el ancho de las barras del código, limitando el uso de dispositivos similares que no ofrecen estas bondades.

La parte correspondiente al circuito digital se ve reducido considerablemente merced a la introducción de circuitos integrados VLSI que son capaces de ejecutar instrucciones de temporización, aritméticas, acceso a memoria y comunicaciones entre otras.

El uso del pórtico serial de comunicaciones presenta ventajas en cuanto a la construcción del circuito digital y posibles aplicaciones futuras, pues el protocolo de intercambio es muy sencillo y las rutinas de control de comunicaciones pueden ser escritas para diversos lenguajes de programación, con el propósito de incorporar las facilidades del lector de código de barras en aplicaciones específicas.

Las rutinas desarrolladas son de propósito general para todas las etiquetas editadas con el formato establecido. Adicionalmente se puede modificar algunas características tales como: número de bits por carácter, número de bits de valor uno "1"; manteniendo en todos los casos la misma

estructura para los bits de guardas y un máximo de 6 caracteres por etiqueta.

Los tópicos considerados en el desarrollo del sistema lector de código de barras proporcionan las herramientas y el criterio suficiente para diseñar una versión de propósito específico con pequeñas modificaciones.

Utilizando la misma configuración circuital es posible diseñar un sistema que realice la identificación total de los caracteres representados en las etiquetas si previamente se definen los parámetros siguientes: número de bits por caracter, número de bits de valor uno "1", conjunto de caracteres y número máximo de caracteres por etiqueta. El sistema resultante podría trabajar en forma autónoma.

Un diseño de estas características presenta algunas ventajas tales como:

- El tamaño del archivo "TABLA.CBR" que contiene la tabla de equivalencias tanto en la codificación como en la decodificación podría ser reducido en forma significativa.

- El equipo puede modificarse para convertirlo en un sistema portátil de identificación.

Las principales limitaciones encontradas son:

- Requiere un sistema de alimentación con baterías que dependiendo de la tecnología utilizada por los componentes VLSI requeridos, constituye un limitante en cuanto al tiempo de operación continua.
- El soporte mecánico del sistema debe ser elaborado de tal forma de garantizar que su tamaño y resistencia sea el más adecuado para su característica de portabilidad.

Las tareas de impresión y decodificación ejecutadas por los módulos BSIMP.EXE y BSDEC.EXE, pueden ser integradas como parte de un sistema completo que utilice las facilidades asociadas al código de barras.

El desarrollo de bases de datos para el control de inventarios, elaboración de facturas, clasificación de partes y piezas, control de personal, administración estudiantil, etc; son aplicaciones en las cuales la

Introducción de sistemas de identificación automática es necesaria.

Se puede deducir del presente trabajo que la aplicación de códigos de barras en sistemas prácticos aumenta la confiabilidad para el manejo de la información gracias a la reducida tasa de error en la decodificación.

En desarrollos futuros relacionados con el tema, podría introducirse como parte del sistema de codificación el uso de técnicas para detección y corrección de errores, con el propósito de aumentar la confiabilidad y reducir el número de exploraciones desechadas por contener algún error.

Los sistemas para automatización de información son cada vez más difundidos con el uso de códigos numéricos en la identificación de usuario, transacciones, etc; siendo necesario el desarrollo de un código numérico con una muy alta confiabilidad, seguridad y flexibilidad en el formato, pues la exclusividad del código mejora la seguridad de un sistema.

El mayor obstáculo para el sistema lector de código de barras diseñado consiste en la velocidad de exploración y su solución en mucho depende del formato utilizado para el

código. Las pruebas realizadas muestran un incremento de la velocidad y una característica diferente para cada lectura efectuada a una misma etiqueta, siendo el uso de referencias parciales obtenidas de cada segmento el método más concluyente, que uno en el cual se consideren referencias iniciales o algoritmos que consideren todas las lecturas para con ellas establecer la referencia en cada punto.

Referencias iniciales serán suficientes cuando el sistema de impresión garantice barras uniformes con bordes muy regulares para un detector de resolución 0.19mm.

Será posible utilizar detectores ópticos de menor resolución incrementando el ancho mínimo de las barras y espacios.

El protocolo utilizado convierte al circuito digital en un circuito esclavo, pues requiere de comandos y decisiones controladas por el programa BSDEC.EXE. La estructura descrita es transparente para el usuario debido al reducido volumen de información a intercambiar en los dos sentidos, permitiendo con pequeñas modificaciones del programas maestro BSDEC.EXE obtener la información de una etiqueta con diferentes niveles de procesamiento, los cuales podrán ser utilizados para decodificar etiquetas creadas con otros códigos siempre y cuando el número de barras y espacios sea



menor o máximo igual a 57, que es el número de localidades de memoria destinado a las lecturas obtenidas de barras y espacios.

La utilización de herramientas para el desarrollo de circuitos basados en microcontroladores implica una familiarización con dichas herramientas, debiendo guardarse las respectivas restricciones sobre los resultados de la simulación, pues condiciones de operación normales para el circuito práctico deben ser reproducidas por el software, tal es el caso de las rutinas de transmisión y recepción desarrolladas para el INTEL 8031.

La reciente introducción en el mercado nacional de una cantidad considerable de computadores personales en diversas aplicaciones y de sistemas de automatización utilizando la técnica de código de barras, hace imprescindible el establecimiento de patrones y normas relacionadas con la codificación impresa, para estandarizar su uso en la identificación de productos de consumo masivo. Los criterios expuestos, las consideraciones realizadas y las conclusiones obtenidas en el presente trabajo pueden ser utilizados en la elección de códigos de barras que se ajusten a las necesidades y condiciones tecnológicas actuales del País.

## **APENDICES**

- A. MODO GRAFICO DEL IMPRESOR
- B. REGISTROS DE CONTROL DEL INTEL 8031
- C. INSTRUCCIONES PEEK, POKE DEL BASIC
- D. COMUNICACION SERIAL CON QUICK BASIC 4.0
- E. CARACTERISTICAS FISICAS DEL PORTICO SERIAL
- F. DIAGRAMAS DE FLUJO Y LISTADO DE PROGRAMAS
- G. DATOS TECNICOS DE LOS COMPONENTES
- H. MANUAL DE USO

## MODO GRAFICO DEL IMPRESOR

En un impresor matricial trabajando en el modo gráfico se controla en forma individual cada una de las agujas de la cabeza del impresor para conseguir un gráfico punto a punto. El número de puntos depende del comando y los parámetros utilizados.

La mayor densidad de un gráfico es 240 x 72 puntos por pulgada cuadrada. Las agujas en la cabeza del impresor están separadas  $1/72$  de pulgada. En modo de texto el carro avanza  $12/72$  de pulgada a simple espacio cuando el comando de fin de línea es enviado. En el modo gráfico el tabulador vertical debe ser de  $8/72$  de pulgada.

Los comandos gráficos definen la densidad horizontal y es posible imprimir más de un punto a la vez controlando simultáneamente todos las agujas de la cabeza de impresión. Cada punto tiene un valor de ponderación binario correspondiente a la posición siendo el de menor ponderación el inferior. La aguja será activada si el bit correspondiente es "1", haciendo muy fácil controlar si se imprimen o no. El argumento en los comandos puede ser un número hexadecimal o decimal, de manera que será más útil

trabajar con el valor decimal de la ponderación, sumando los valores correspondientes de las agujas que se desee activar.

Ejemplo A.1.

128 o 128	128 o	128 o 128
64 o	64 o 64	64 o
32 o 32	32 o	32 o
16 o	16 o	16 o
8 o 8	8 o 8	8 o
4 o	4 o	4 o 4
2 o 2	2 o 2	2 o 2
1 o	1 o	1 o
<hr/>		
170	74	134

Eje A.1 Control de agujas de cabeza de impresión.

COMANDOS GRAFICOS:

Gráficos de Simple densidad.-

ASCII:	ESC	K	n1	n2
Decimal:	27	75	n1	n2
Hex:	1B	4B	n1	n2

Con 60 puntos por pulgada, utilizando 8 agujas, donde el número de columnas está determinado por:

$$n1 + ( n2 \times 256 )$$

Gráficos de Doble densidad.-

ASCII:	ESC	L	n1	n2
Decimal:	27	76	n1	n2
Hex:	1B	4C	n1	n2

Con 120 puntos por pulgada, utilizando 8 agujas y baja velocidad de impresión, donde el número de columnas está determinado por:

$$n1 + ( n2 \times 256 )$$

Gráficos de Doble densidad y alta velocidad.-

ASCII:	ESC	Y	n1	n2
Decimal:	27	89	n1	n2
Hex:	1B	59	n1	n2

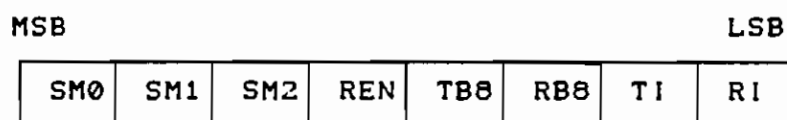
Con 120 puntos por pulgada, utilizando 8 agujas y alta velocidad de impresión, donde el número de columnas está determinado por:

$$n1 + ( n2 \times 256 )$$

## REGISTROS DE CONTROL DEL INTEL 8031

El modo de operación del pòrtico serial del microcontrolador 8031 queda determinado por los registros SCOM, IE, IP, PCON, TCON, TMOD.

SCOM Es el registro de control del pòrtico serial y tiene la siguiente estructura.



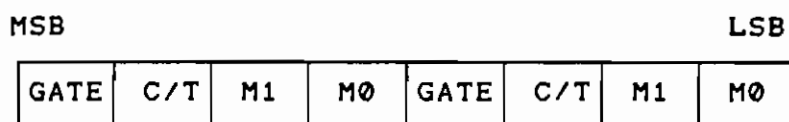
SM0, SM1 Definen el modo de trabajo del pòrtico

SM0	SM1	MODO	DESCRIPCION	VELOCIDAD
0	0	0	Registro de desplazamiento	f/12
0	1	1	8-bits UART	variable
1	0	2	9-bits UART	f/64 o f/32
1	1	3	9-bits UART	variable

Tabla B.1

SM2 Habilita la comunicación entre microcontroladores.

TMOD Es el registro de control de los temporizadores y contadores; tiene la siguiente estructura.(TIMER/COUNTER)



GATE Este bit permite habilitar la señal de interrupción asociada a un TIMER.

C/T Cuando este bit es igual a 1 el contador está habilitado y si es 0 el temporizador está activo.

M1,M0 La combinación de estos bits define el modo de operación del contador o temporizador de acuerdo a la tabla B.1.

Las interrupciones del MCS 51 están controladas por dos registros: IE contiene el estado de habilitación o no de las fuentes de interrupción, IP la prioridad de las interrupciones.

## REGISTRO IE

MSB				LSB			
EA	X	X	ES	ET1	EX1	ET0	EX0

EA Cuando EA = 1 las interrupciones son reconocidas, si EA = 0 las fuentes de interrupción no son reconocidas estén o no habilitadas individualmente.

ES Habilitación de interrupción del pórtico serial, si ES = 0 es deshabilitado.

ET1 Si ET1 = 0 la interrupción generada por el TIMER1 es deshabilitada.

EX1 Cuando EX1 = 0 la interrupción externa INT1 es deshabilitada.

ET0 Si ET0 = 0 la interrupción generada por el TIMER0 es deshabilitada.

EX0 Cuando EX0 = 0 la interrupción externa INT0 es deshabilitada.

El registro IE tendrá el valor de 90H para habilitar la atención de las interrupciones y permitir ejecutar los comandos que por el pórtico serial son enviados desde el computador durante el lazo de espera.



## REGISTRO IP

MSB						LSB	
X	X	X	PS	PT1	PX1	PT0	PX0

- PS Prioridad de interrupción serial.
- PT1 Prioridad de interrupción generada por TIMER1.
- PX1 Prioridad de interrupción externa INT1.
- PT0 Prioridad de interrupción generada por TIMER0.
- PX0 Prioridad de interrupción externa INT0.

La más alta prioridad de las interrupciones se obtiene cuando el bit correspondiente es uno "1" y en el caso de tener interrupciones de igual prioridad que aparecen en el mismo instante la secuencia de atención está predefinida de la siguiente manera:

PRIORIDAD	FUENTE	VECTOR	
1.-	INT0	0003H	Externa 0
2.-	TF0	000BH	Timer 0
3.-	INT1	0013H	Externa 1
4.-	TF1	001BH	Timer 1
5.-	RI+TI	0023H	Pórtico serial

## INSTRUCCIONES PEEK, POKE DEL BASIC

Las instrucciones permiten leer o escribir directamente en localidades de memoria del computador, el argumento define la localidad expresado en forma decimal entre 0 y 65535 o en su correspondiente hexadecimal con el formato &Hxxxx.

Previa la ejecución de PEEK o POKE debe definirse el segmento de memoria con la instrucción DEF SEG.

PEEK.- Lee el dato almacenado en una localidad específica de memoria.

```
A = PEEK(&H400)/(1024)
```

Retorna un número decimal entre 0 y 255 que se encuentre en la localidad desplazada &H400 (1024) del origen del segmento.

POKE.- Escribe un byte en una localidad determinada.

```
POKE &H5A00,&HFF/(23040,255)
```

Escribe &HFF (255) en la localidad &H5A00 (23040), las direcciones pueden tomar valores entre 0 y 65535 y los datos entre 0 y 255.

Mucha de las operaciones del PC están controladas por datos que se encuentran en las posiciones bajas de la memoria, particularmente en dos áreas adyacentes cuya longitud es 256 bytes, que empiezan en 400 HEX y 500 HEX; que contienen información referente a: equipo instalado, estado de las unidades de disco, etc. Estos datos son generados por el ROM BIOS durante el procedimiento de inicialización y es posible inspeccionarlos y/o cambiarlos.

Las direcciones utilizadas para la instalación son:

Número de unidades de disco rígido presentes.-

PEEK(&H475)/(1141)

El equipo instalados se puede encontrar leyendo las localidades (&H410)/(1040) y (&H411)/(1041).

Para (&H410)/(1040)

MSB

LSB

D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----

D7,D6 Indica el número de pórticos paralelos instalados.

D5 1 si está instalado un pórtico para impresor serial.

D4 1 si está instalado el Joystick.

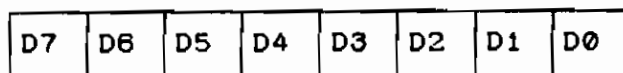
D3,D2,D1 Número de pórticos serie.

D0 0 si está instalado el circuito de DMA.

Para (&H411)/(1041)

MSB

LSB



D7,D6 Número de unidades de disco flexible.

00 = 1, 01 = 2, 10 = 3, 11 = 4

D5,D4 Modo de video.

01 = 40 columnas, 10 = 60 columnas, 11 = 80 columnas monocromático.

D3,D2 RAM del sistema instalada.

D1 No utilizado.

D0 1 si está instalada alguna unidad de disco.

Monitor.-

Tipo de adaptador de video.

PEEK(410) AND 48 = 0 No tiene monitores instalados.

= 16 40 X 25 adaptador gráfico.

= 32 80 X 25 adaptador gráfico.

= 48 Monitor monocromático

## COMUNICACION SERIAL CON QUICK BASIC 4.0

Puesto que las rutinas de comunicaciones del computador están desarrolladas utilizando Quick Basic versión 4.0, deben definirse los parámetros de intercambio de información a través del pòrtico serial. La sentencia OPEN COMn permite definir el modo de operación del pòrtico.

```
OPEN "COM(n):[velocidad],[paridad],[número de bits]
,[bits de parada],[RS],[CS(n)],[DS(n)],[CD(n)],[LF]
,[PE]" AS (# [LEN = número de bytes]
```

COM(n) Define el dispositivo utilizado COM1 o COM2.

Los parámetros pueden ser:

[Velocidad de transmisión]: 75, 110, 150, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200 bits por segundo.

[Paridad]: S (space) siempre introduce un bit con valor cero como paridad en Tx y Rx. M (mark) introduce un bit con valor uno como paridad en Tx y Rx. O (odd) chequea que el byte recibido tenga paridad impar. E (even) verifica la paridad par. N (none) no transmite paridad y no la chequea.

[Número de bits]: 4, 5, 6, 7 o 8. Para 4 bits no se admite uso de paridad.

[Bits de parada]: pueden ser uno o dos.

Nota.- Los valores predefinidos son: 300 bits por segundo, paridad par, 7 bits de datos, dos bits de parada en 75 o 110 bits por segundo y un bit en las restantes velocidades.

Se pueden definir condiciones adicionales sobre el modo de operación de las señales de intercambio como:

[RS] suprime señal RTS ( request to send)

[CS(n)] control de CTS (clear to send) y (n) es tiempo de espera en milisegundos por esta señal y puede ser un valor entre 0 y 65535.

[DS(n)] controla DSR (data set ready) similar a la anterior.

[CD(n)] controla CD (carrier detect).

[LF] envía un Line feed cada return.

[PE] deshabilita chequeo de paridad.

Nota.- la predefinición en estas señales es : CS1000, DS1000 y CD0. Si RS es definido CS0 es predefinido y si (n) es omitido toma el valor (0).

Es posible definir el número máximo de bits que puede leer o escribir en la cola del pórtico utilizando la opción LEN. El número máximo de caracteres en la cola es 512 si LEN es omitida en la sentencia OPEN. El uso de LEN es útil cuando se utiliza las instrucciones GET y PUT.

GET #{pórtico},{número de caracteres}

La instrucción GET utilizada con CDM(n) retorna el número caracteres especificado y no puede exceder el valor determinado con LEN. #{pórtico} es el número con el que se abrió el pórtico serial.

PUT #{pórtico},{número de caracteres}

PUT escribe en la cola de transmisión el número de bytes indicado y debe ser menor o igual que lo determinado por LEN.

Las funciones EOF, LOC, LOF permiten realizar un control de la cola de comunicaciones que es de mucha utilidad en transmisión asincrónica.

- EOF(#) (-1) indica que la cola de recepción esta vacía; (0) hay caracteres esperando ser leídos.
- LOF(#) Muestra el espacio libre en la cola de transmisión.
- LOC(#) Retorna el número de caracteres presentes en la cola de recepción. La cola puede ser de extensión mayor a 512 caracteres, valor determinado con la opción /c:xxx cuando se carga el Quick Basic.

En modo asincrónico se utiliza los comandos XOFF (&H13) para suspender momentaneamente la transmisión cuando la cola de recepción está llena y XON (&H11) para reiniciar. Estos comandos son generados por el programa del equipo receptor.

La instrucción INPUT\$ es preferida para leer una línea puesto que todos los caracteres ASCII pueden ser significativos en la comunicación. INPUT\$ asigna a una variable literal el número de caracteres indicados en el comando.

A\$ = INPUT\$ (n, #)



n es el número de caracteres leídos y puede reemplazarse con LOC(#) si se desea descargar la cola con cada instrucción de lectura.

ON COM(n) permite detectar cuando un caracter ha sido escrito en la cola de recepción, interrumpiendo la ejecución del programa principal para atender la rutina de lectura. Esta facilidad no es aconsejable para leer mensajes pequeños puesto que cada vez que el pòrtico es accedido se interrumpe la ejecución del programa y puede ocasionar sobreflujo en el stack del computador.

La instrucción PRINT puede ser utilizada para escribir en la cola de transmisión del pòrtico serial.

PRINT #,A\$

Donde A\$ es la variable que contiene los caracteres que se desean transmitir.

## CARACTERISTICAS FISICAS DEL PORTICO SERIAL

El p rtico serial de un computador personal opera bajo la norma de intercambio de informaci n serial EIA RS-232, con un conector macho del tipo DB-25. A continuaci n se describe la funci n asignada a cada uno de sus pines:

- Pin 1      Punto de tierra f sica o no utilizado.
- Pin 2      Tx. canal de transmisi n de datos.
- Pin 3      Rx. canal de recepci n de datos.
- Pin 4      RTS (Request to send), en cero l gico significa que el p rtico est  inactivo. En uno l gico pregunta al receptor si se encuentra listo para recibir los datos.
- Pin 5      CTS (Clear to send), se al enviada del receptor al transmisor que indica que la cola de recepci n est  libre para recibir informaci n. La transferencia ser  efectiva si esta se al es detectada en uno l gico.
- Pin 6      DSR (Data set ready), se al de respuesta a RTS (Request to send); indica que el terminal receptor est  disponible para ser accedido.
- Pin 7      GND (Signal ground), referencia de las l neas de datos y control.

- Pin 8 DCD (received line signal detector), detector de portadora de datos.
- Pin 9-19 No utilizados.
- Pin 20 DTR (Data terminal ready), indica que la próxima palabra de información está lista.
- Pin 22 RI (Ring indicator), similar a RTS (Request to send), para el dispositivo externo.
- Pin 23-25 No utilizados.

#### CARACTERISTICAS ELECTRICAS DEL RS-232C

Los niveles de tensión utilizados por ésta norma de comunicación serial, son: +12 y -12 voltios.

Las líneas de datos utilizan -12V para uno lógico y +12V para cero lógico.

Las señales de control para la transferencia es +12V para uno lógico y -12V para cero lógico.

Nota.- Los valores de tensión reconocidos están entre +/- 5 y +/- 15 voltios.

**DIAGRAMA 01**  
**DIAGRAMA DE FLUJO DE INICIALIZACION Y ATENCION DE COMANDOS**

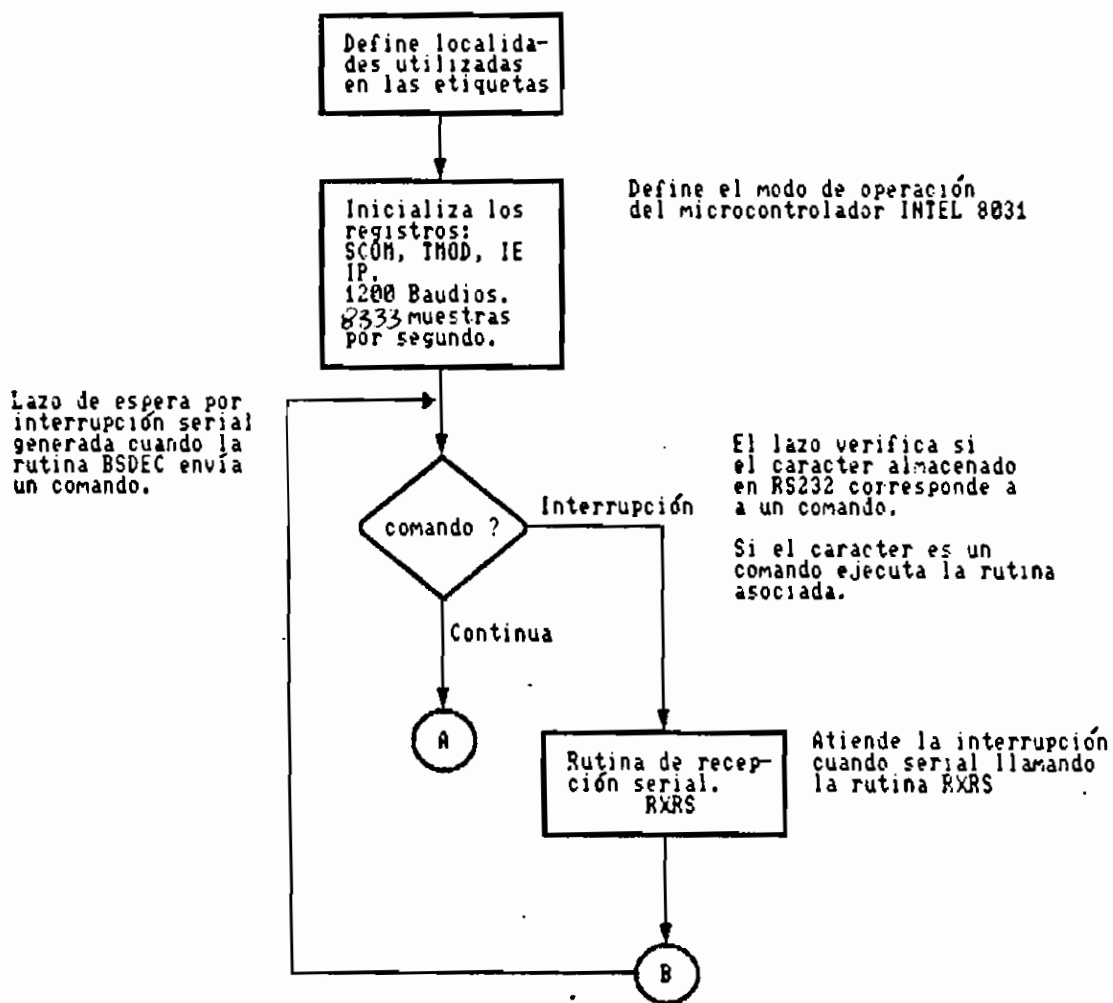


DIAGRAMA B1

DIAGRAMA DE FLUJO DE INICIALIZACION Y ATENCION DE COMANDOS  
CONTINUACION

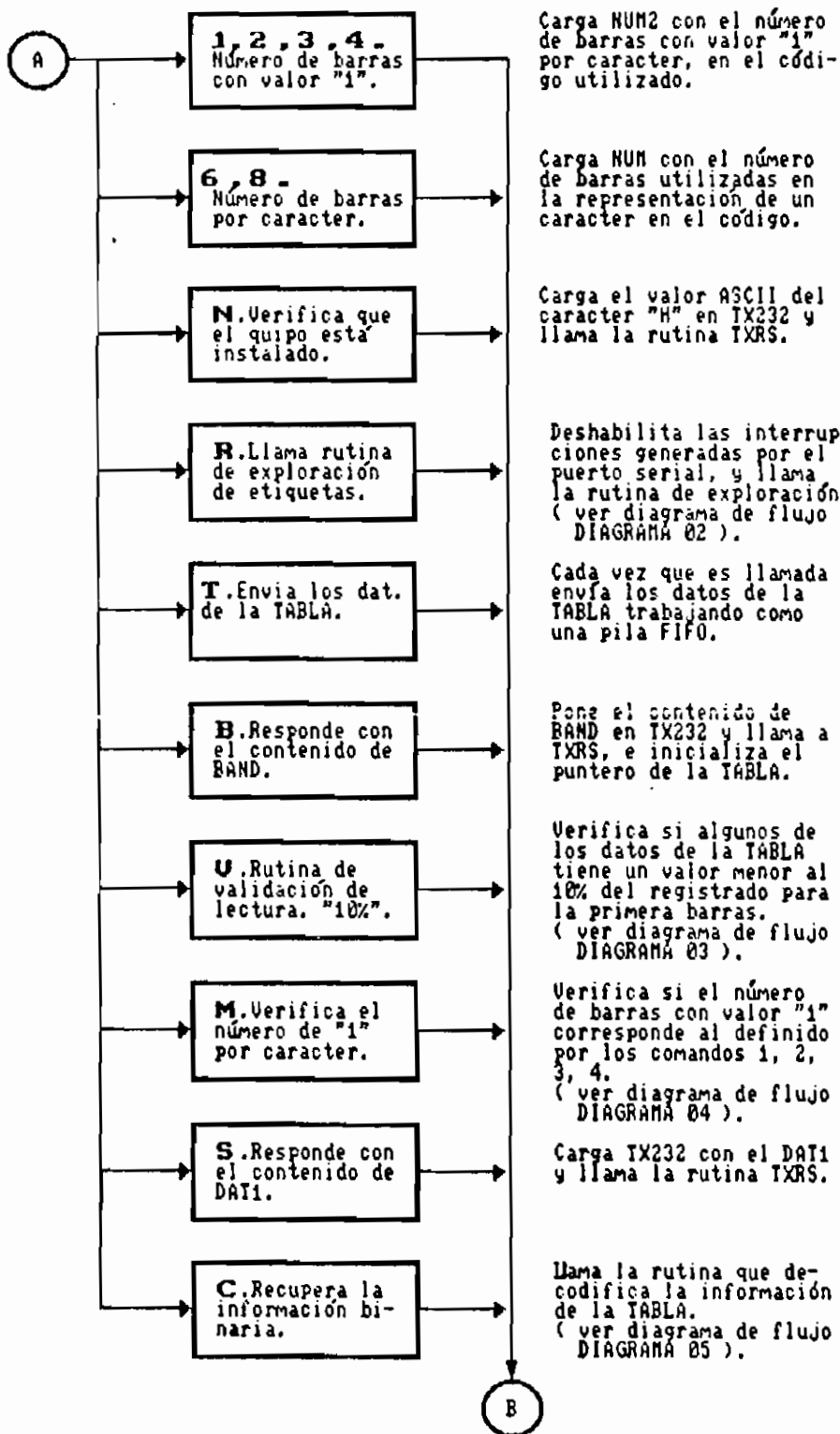
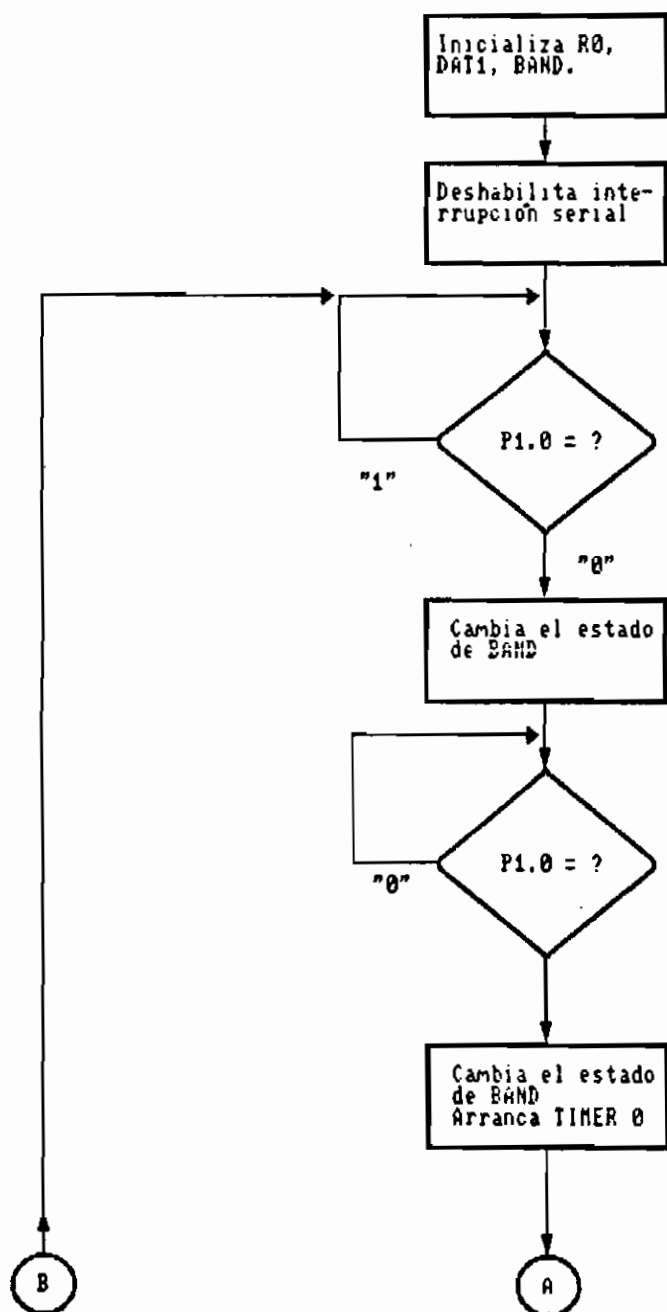


DIAGRAMA 02  
 RUTINA DE EXPLORACION DE LA ETIQUETA (comando R)



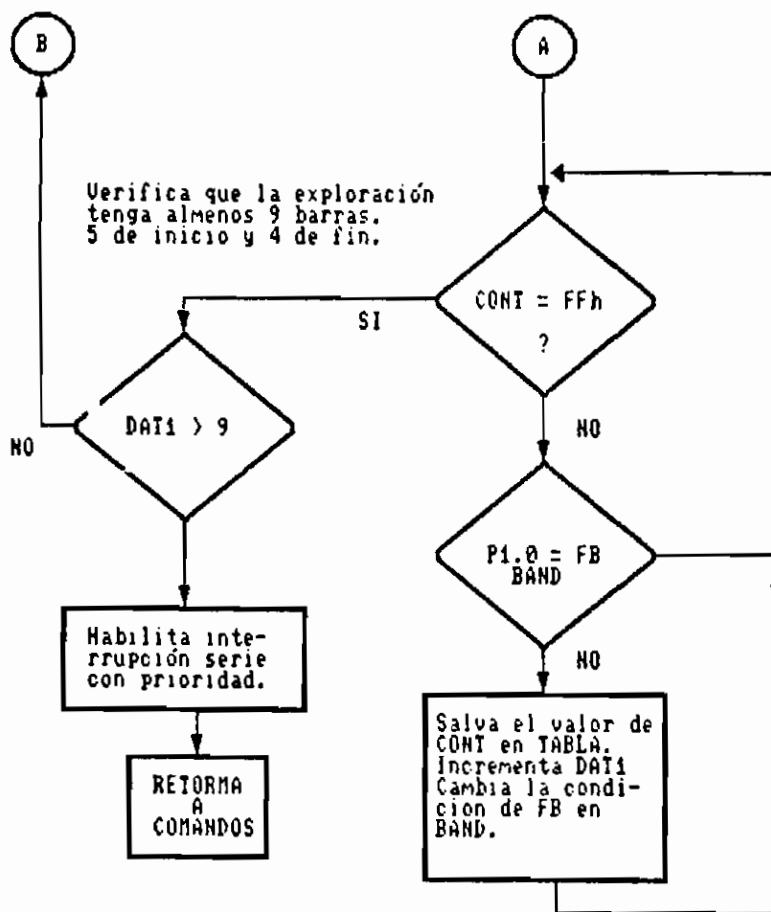
Define los valores iniciales de: R0 = 20h dirección de inicio de la TABLA, DAT1 = 00h el número de barras detectadas, BAND = 01h identifica la primera barra oscura.

Inhibe las interrupciones generadas por el pòrtico serial, y habilita la interrupción generada por TIMER 0, dándole la máxima prioridad.

Lazo de espera de la rutina, lee el estado del bit 0 del pòrtico 1 el cambio de "1" a "0" ocurre cuando el lápiz es activado.

Lazo de espera por el primer flanco en la exploración de la etiqueta, este es generado por una barra.

DIAGRAMA 82  
 RUTINA DE EXPLORACION DE LA ETIQUETA (comando R)  
 CONTINUACION



Verifica que la exploración tenga al menos 9 barras. 5 de inicio y 4 de fin.

Verifica si el contador no ha llegado al valor máximo. Esto ocurre cuando el lápiz sale de la etiqueta.

Comprueba el estado de la entrada detectando los cambios de estado. Esto corre en los cambios de barras a espacio.

Genera la TABLA con las lecturas tomadas. Incrementa DATi registrando la longitud de la TABLA. Cambia el bit FB de BAND, donde FB = "1" para barras, y FB = "0" para espacios.

DIAGRAMA 03  
LECTURAS MENORES 10% DEL LA INICIAL

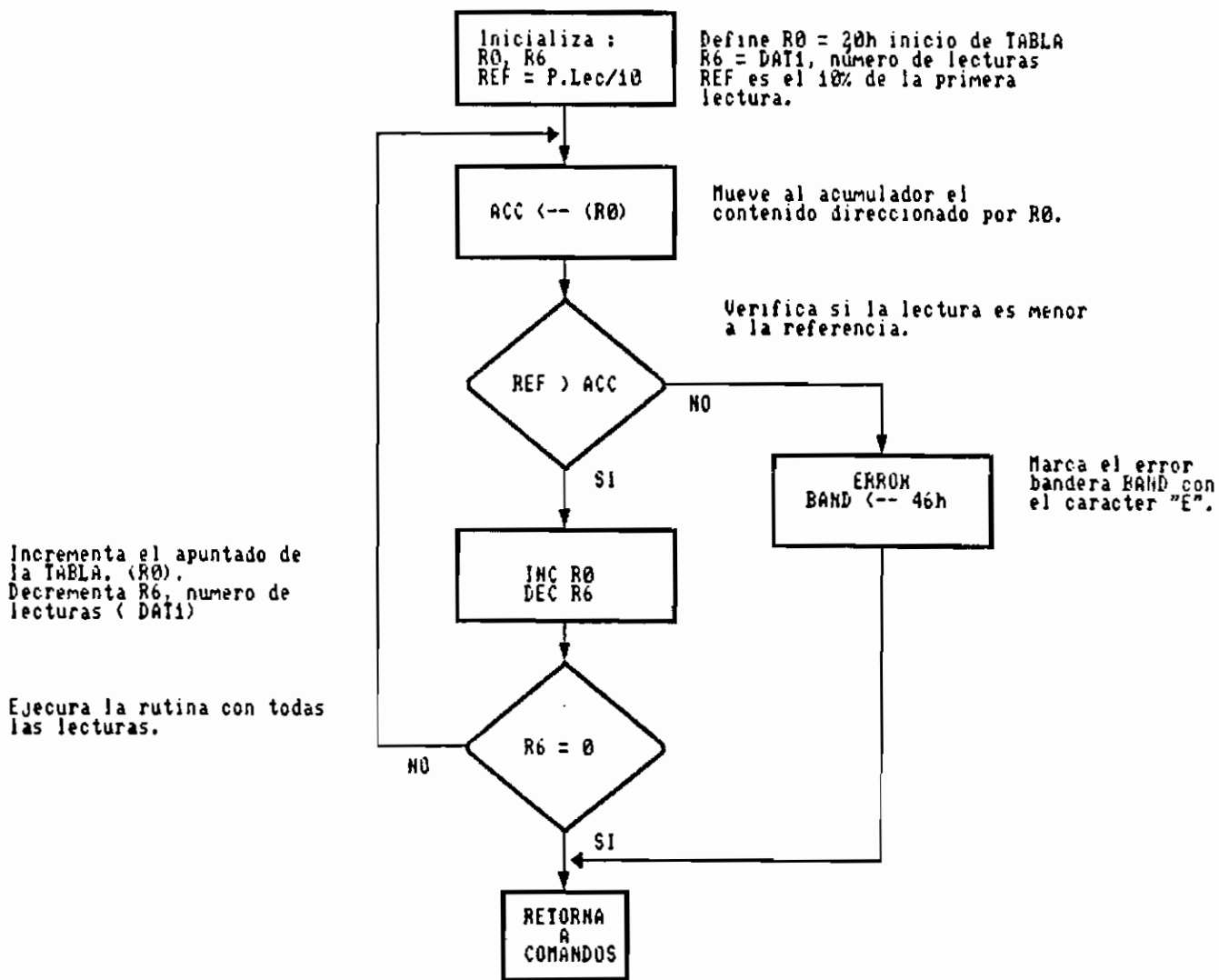




DIAGRAMA 04  
 VERIFICA NUMERO DE BITS DE VALOR UNO

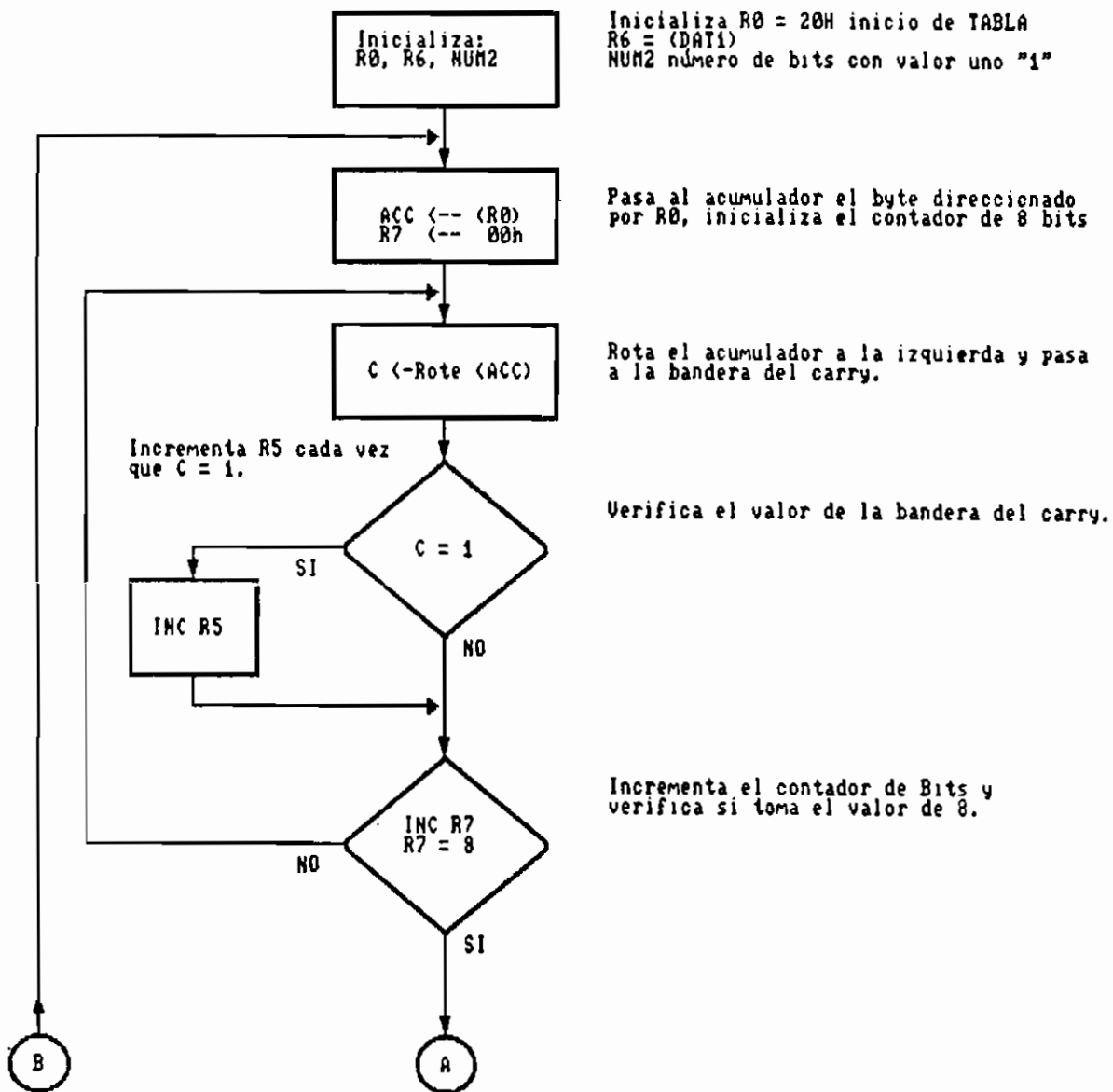


DIAGRAMA 04  
VERIFICA NUMERO DE BITS DE VALOR UNO  
CONTINUACION

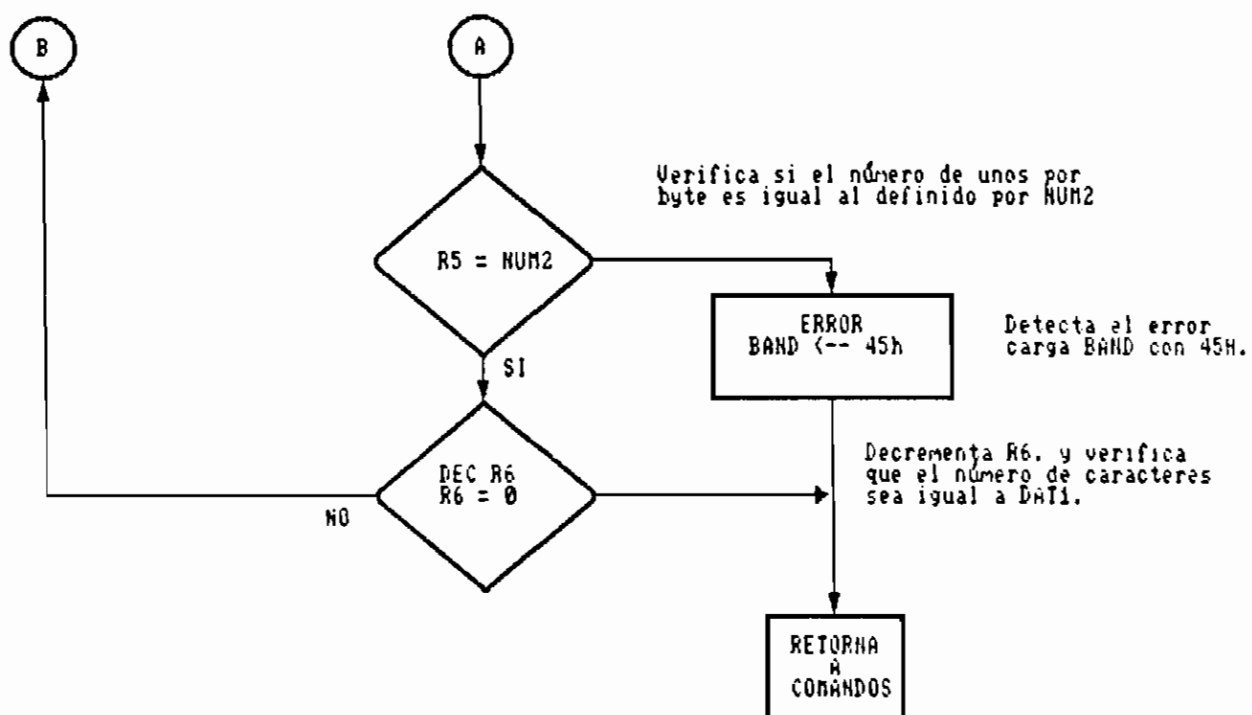


DIAGRAMA 05  
 RECUPERA LA INFORMACION BINARIA A PARTIR DE LAS LECTURAS

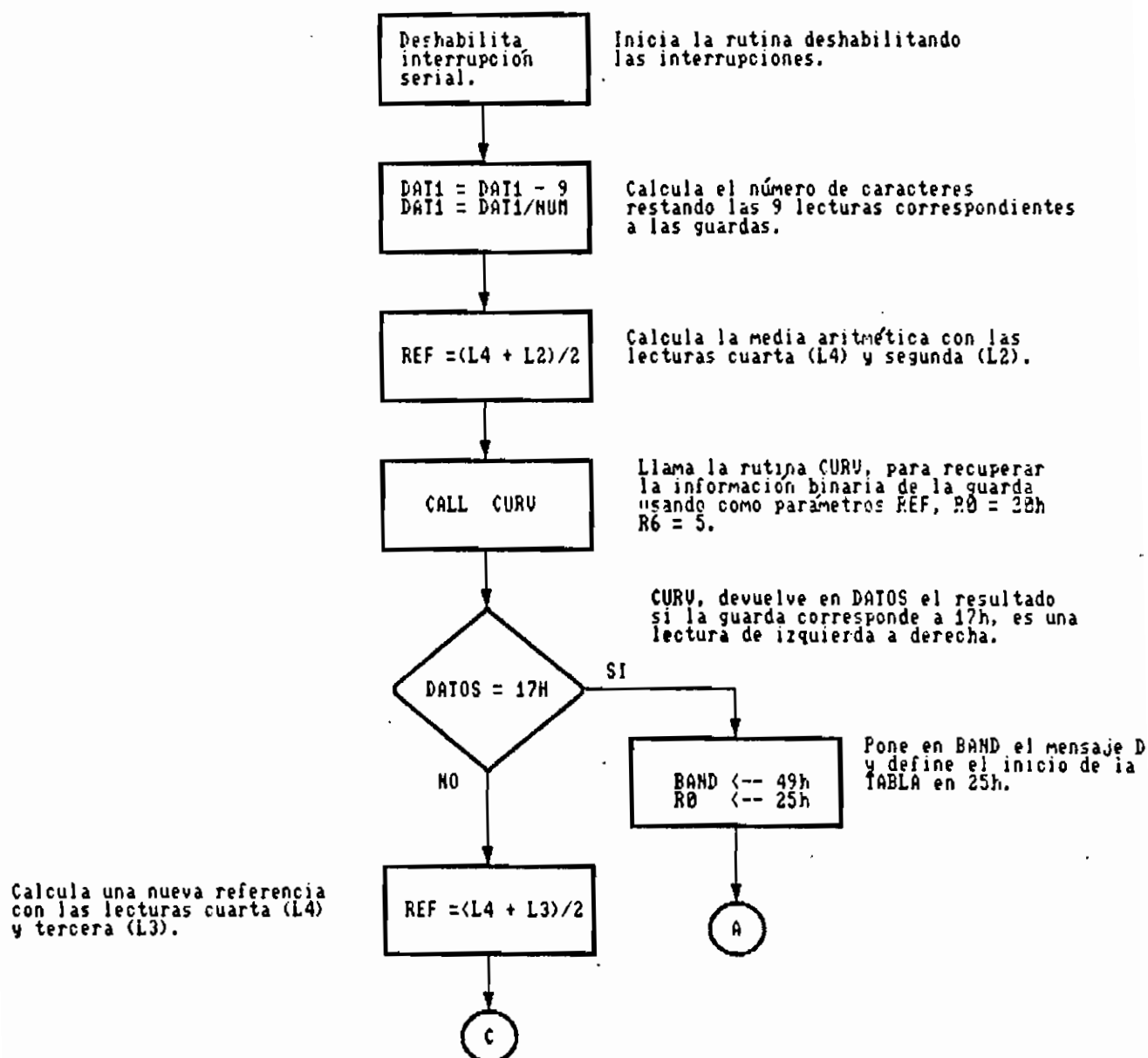


DIAGRAMA 05  
 RECUPERA LA INFORMACION BINARIA A PARTIR DE LAS LECTURAS  
 CONTINUACION

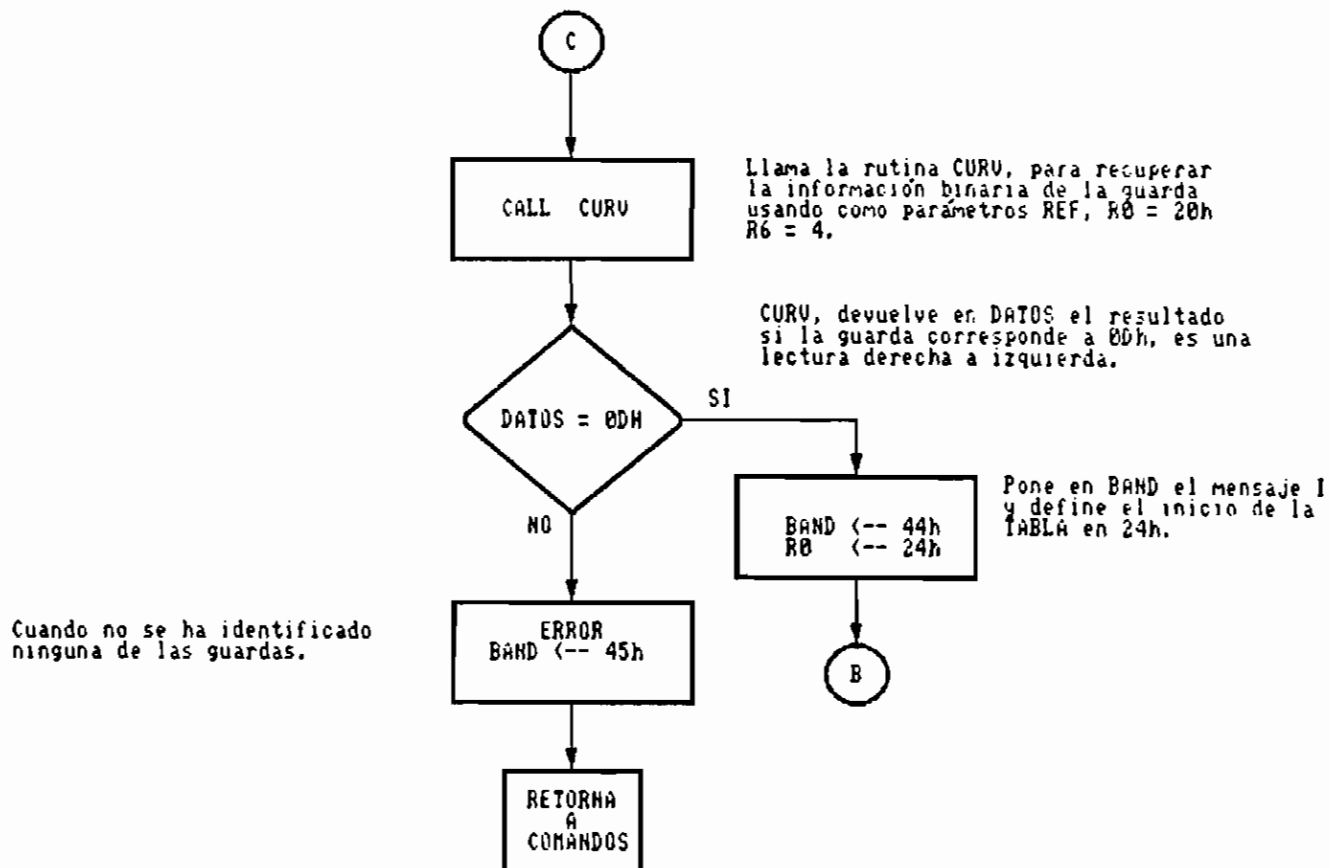


DIAGRAMA B5  
 RECUPERA LA INFORMACION BINARIA A PARTIR DE LAS LECTURAS  
 LECTURA DERECHA  
 CONTINUACION

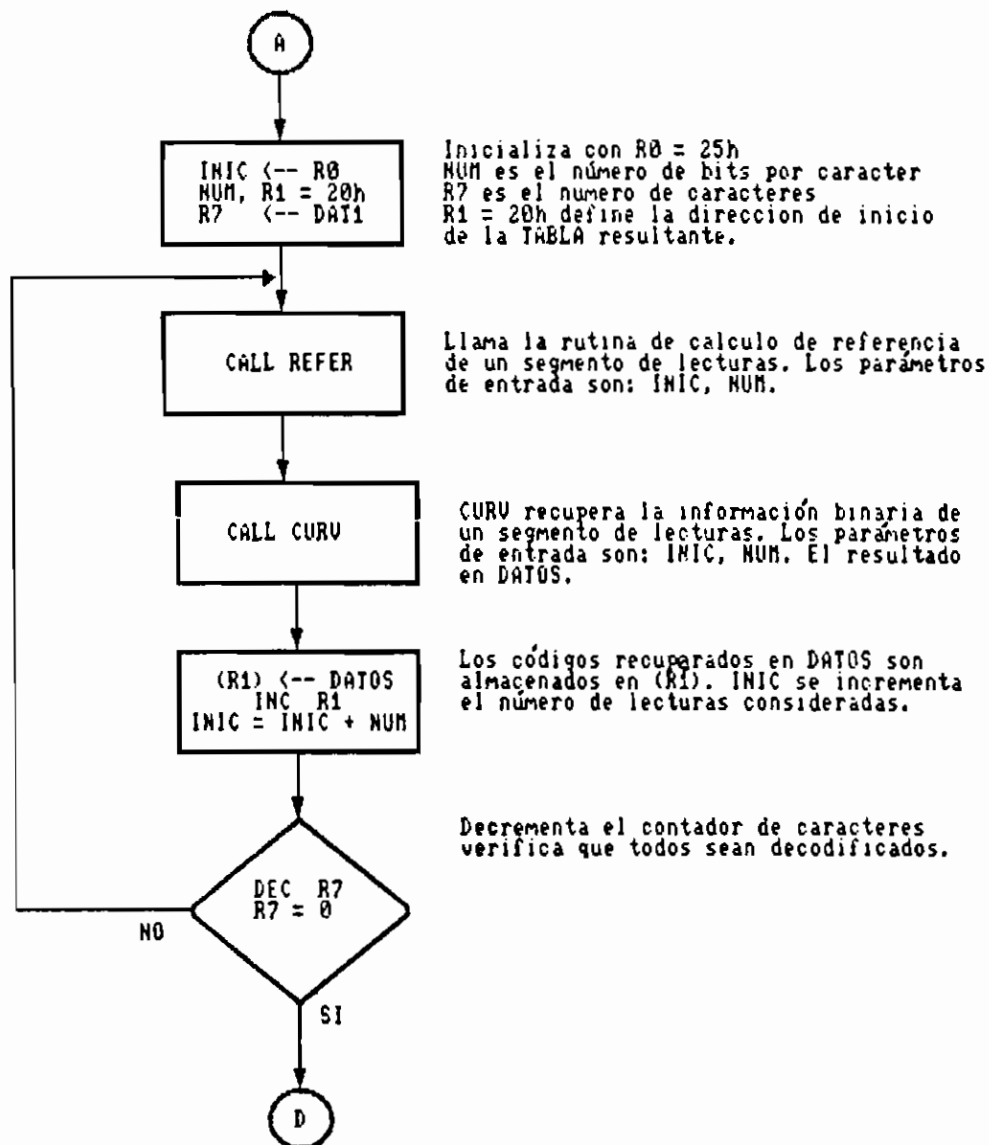


DIAGRAMA 05  
 RECUPERA LA INFORMACION BINARIA A PARTIR DE LAS LECTURAS  
 LECTURA DERECHA  
 CONTINUACION

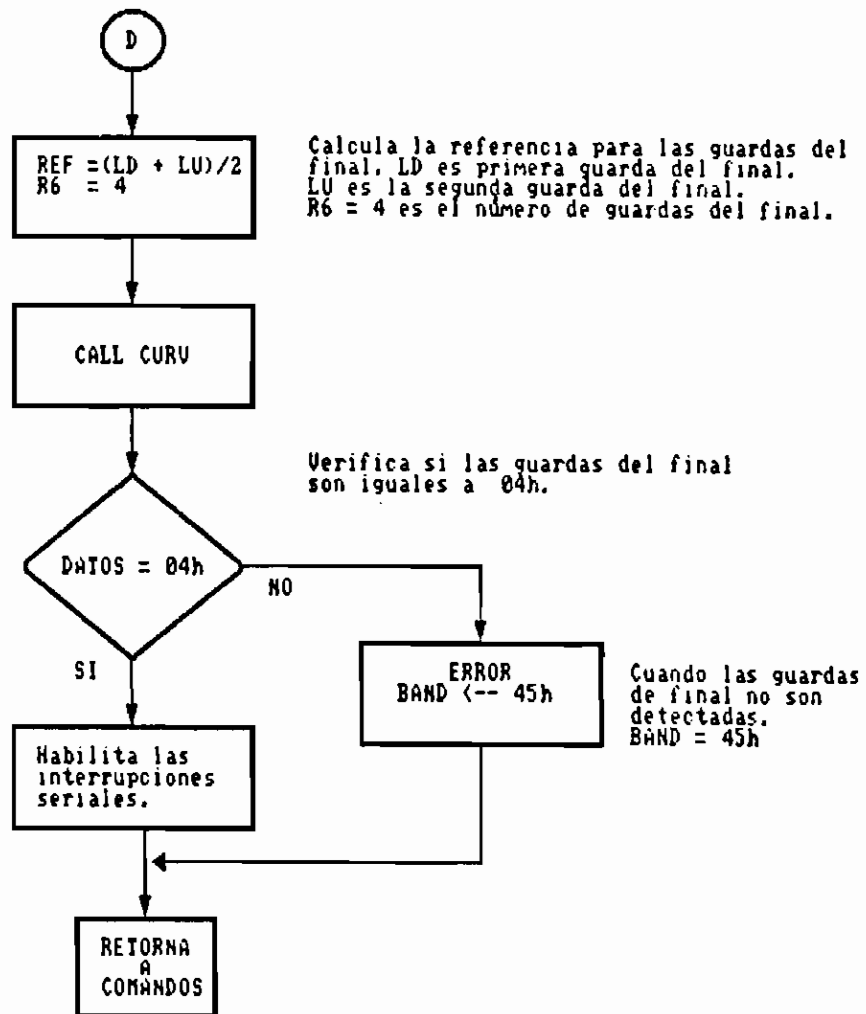


DIAGRAMA B5  
 RECUPERA LA INFORMACION BINARIA A PARTIR DE LAS LECTURAS  
 LECTURA IZQUIERDA  
 CONTINUACION

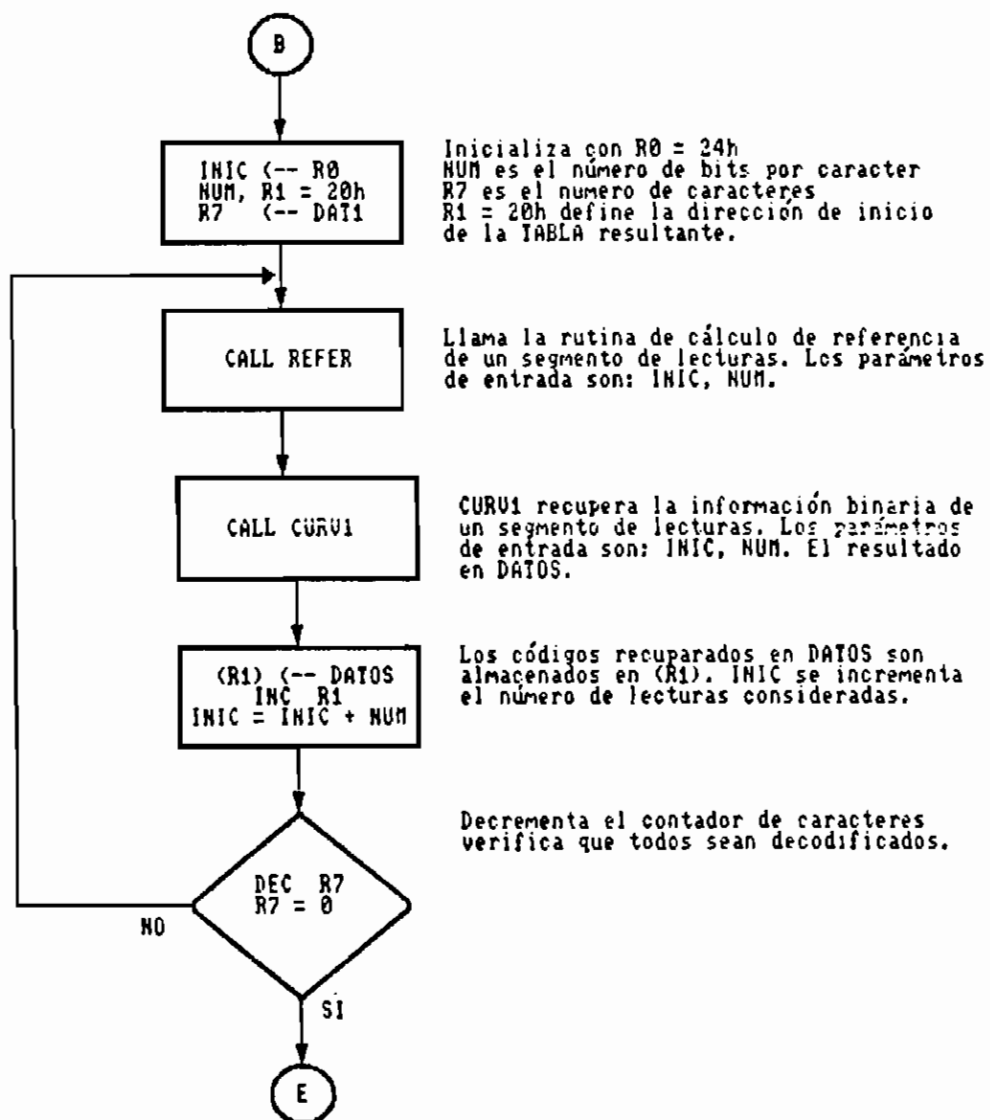
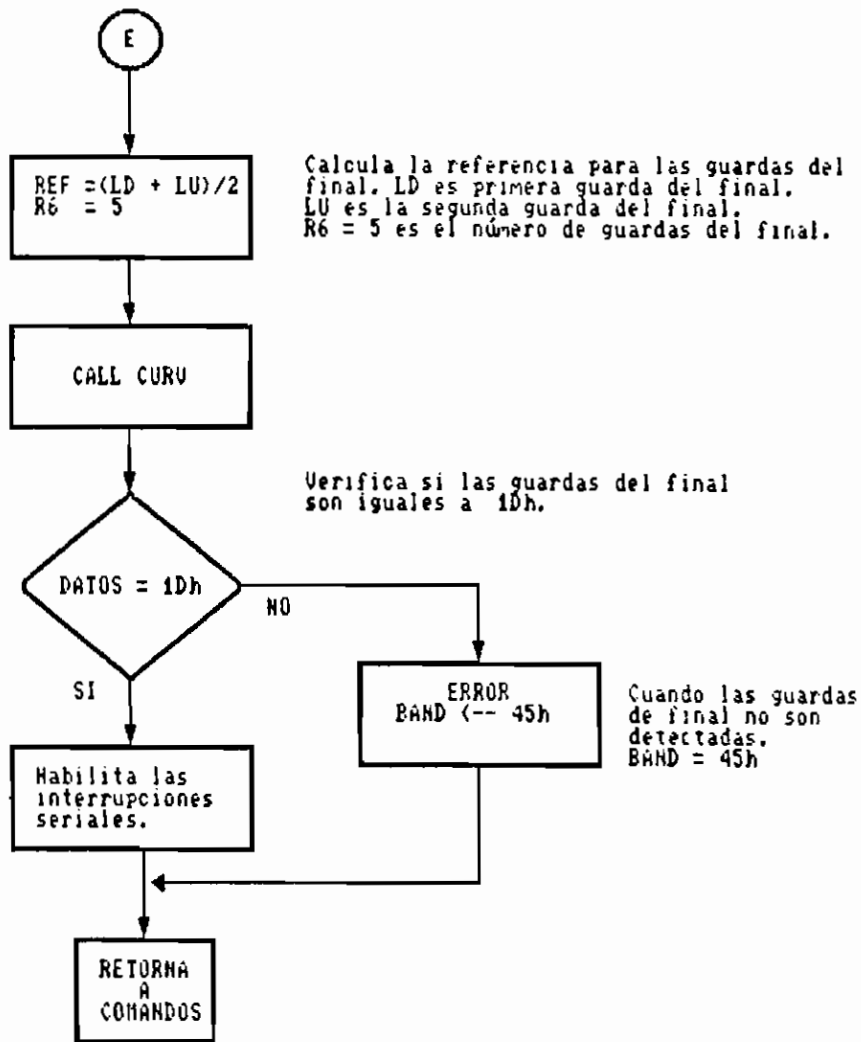


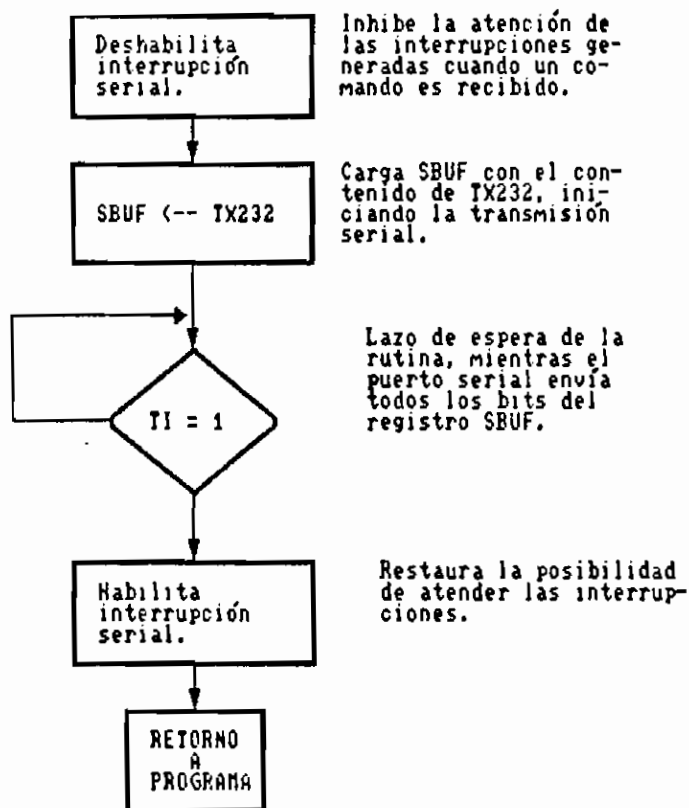
DIAGRAMA 05  
 RECUPERA LA INFORMACION BINARIA A PARTIR DE LAS LECTURAS  
 LECTURA IZQUIERDA  
 CONTINUACION





## DIAGRAMA 06

## ROUTINA DE TRANSMISION SERIAL DE UN BYTE



## DIAGRAMA 87

## ROUTINA DE RECEPCION SERIAL DE UN BYTE

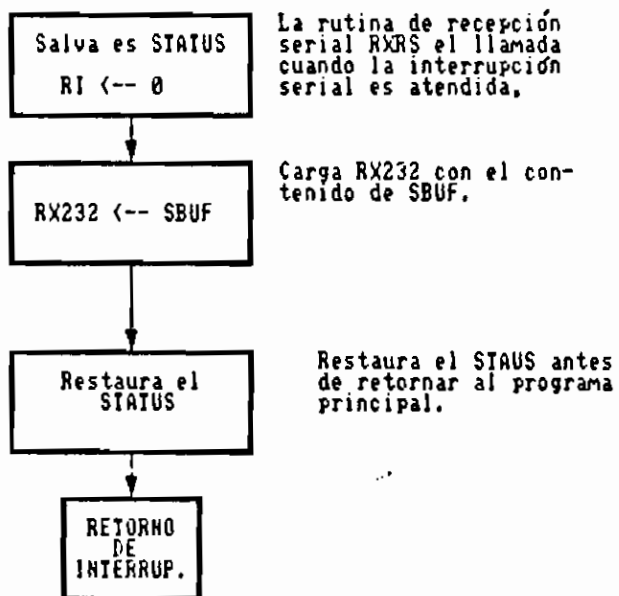


DIAGRAMA 88  
CURV COMPARA LAS LECTURAS DE UN SEGMENTO CON REF

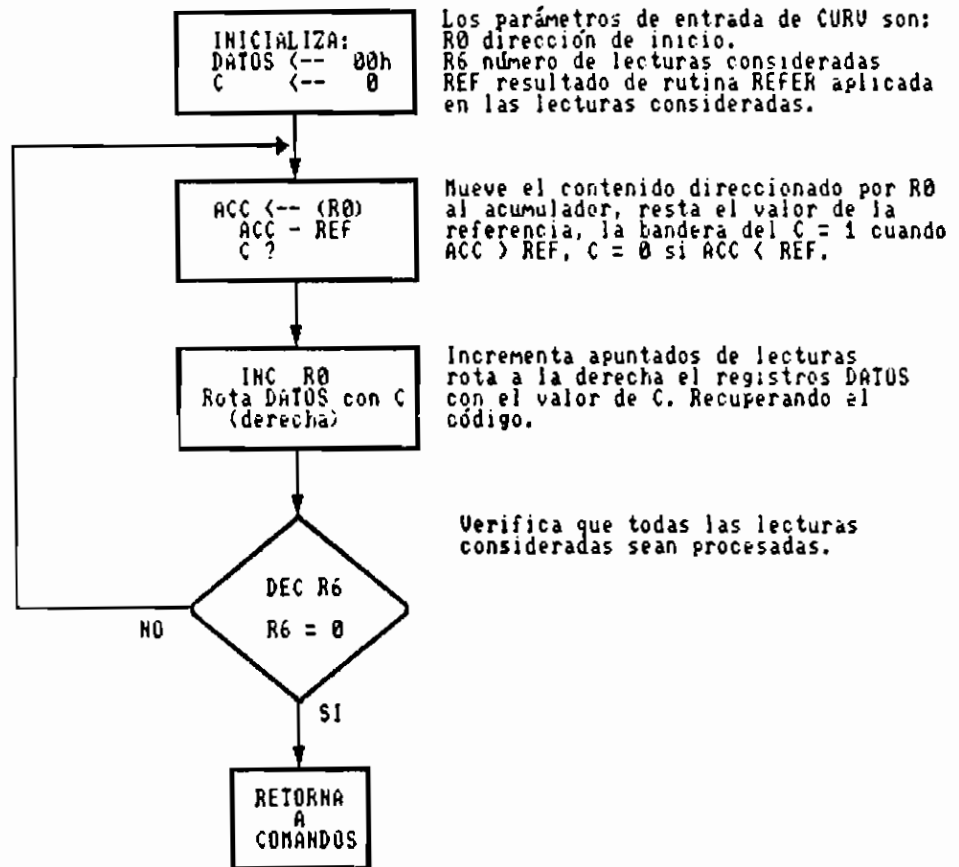


DIAGRAMA 89  
CURV1 COMPARA LAS LECTURAS DE UN SEGMENTO CON REF

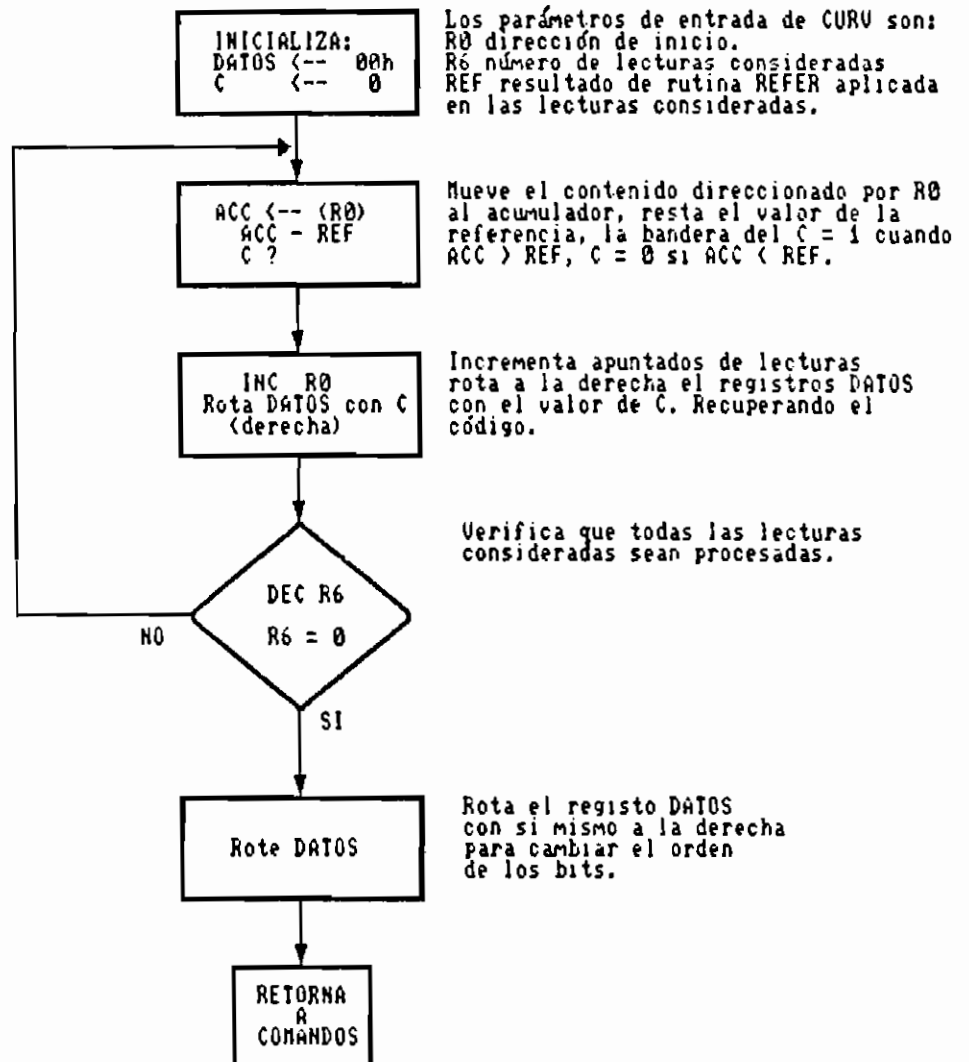


DIAGRAMA 18  
 CALCULO DE LA REFERENCIA DE UN SEGMENTO DE LECTURAS

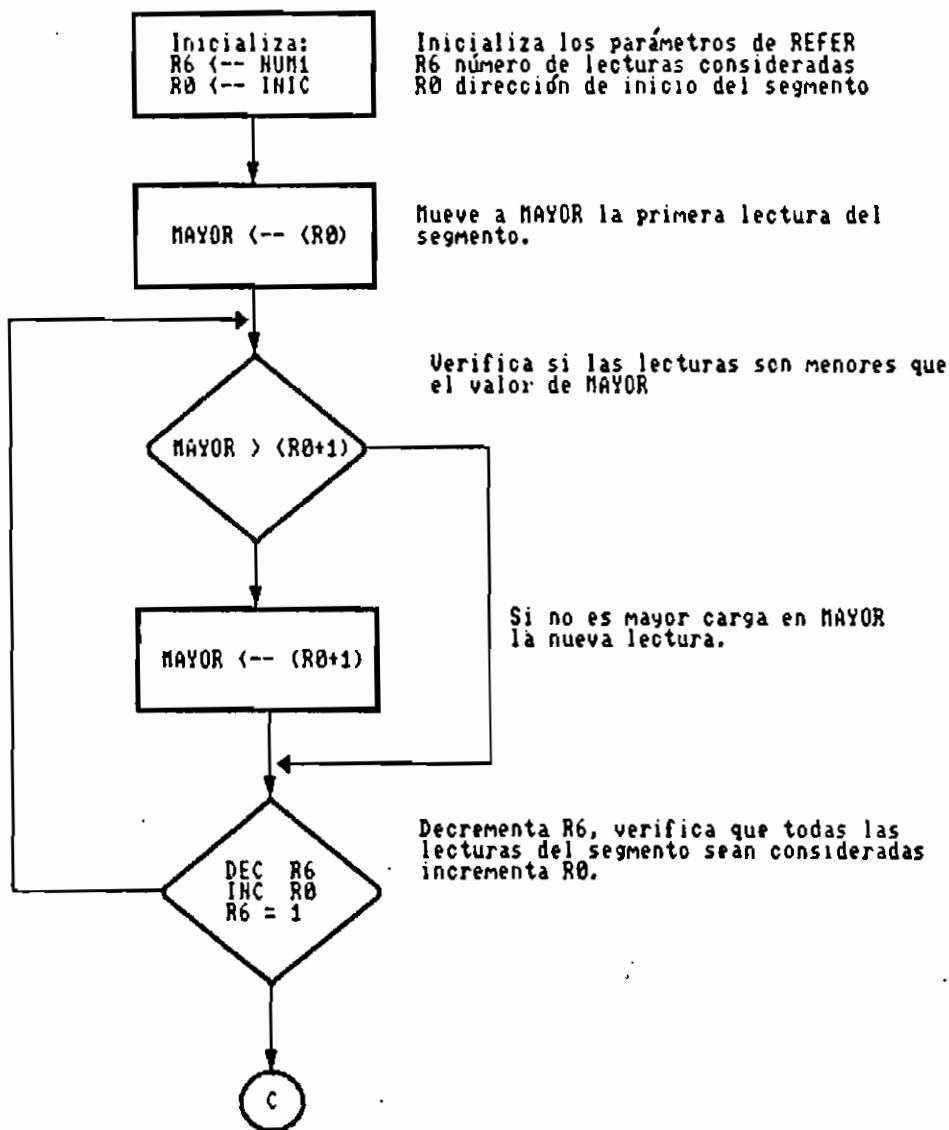
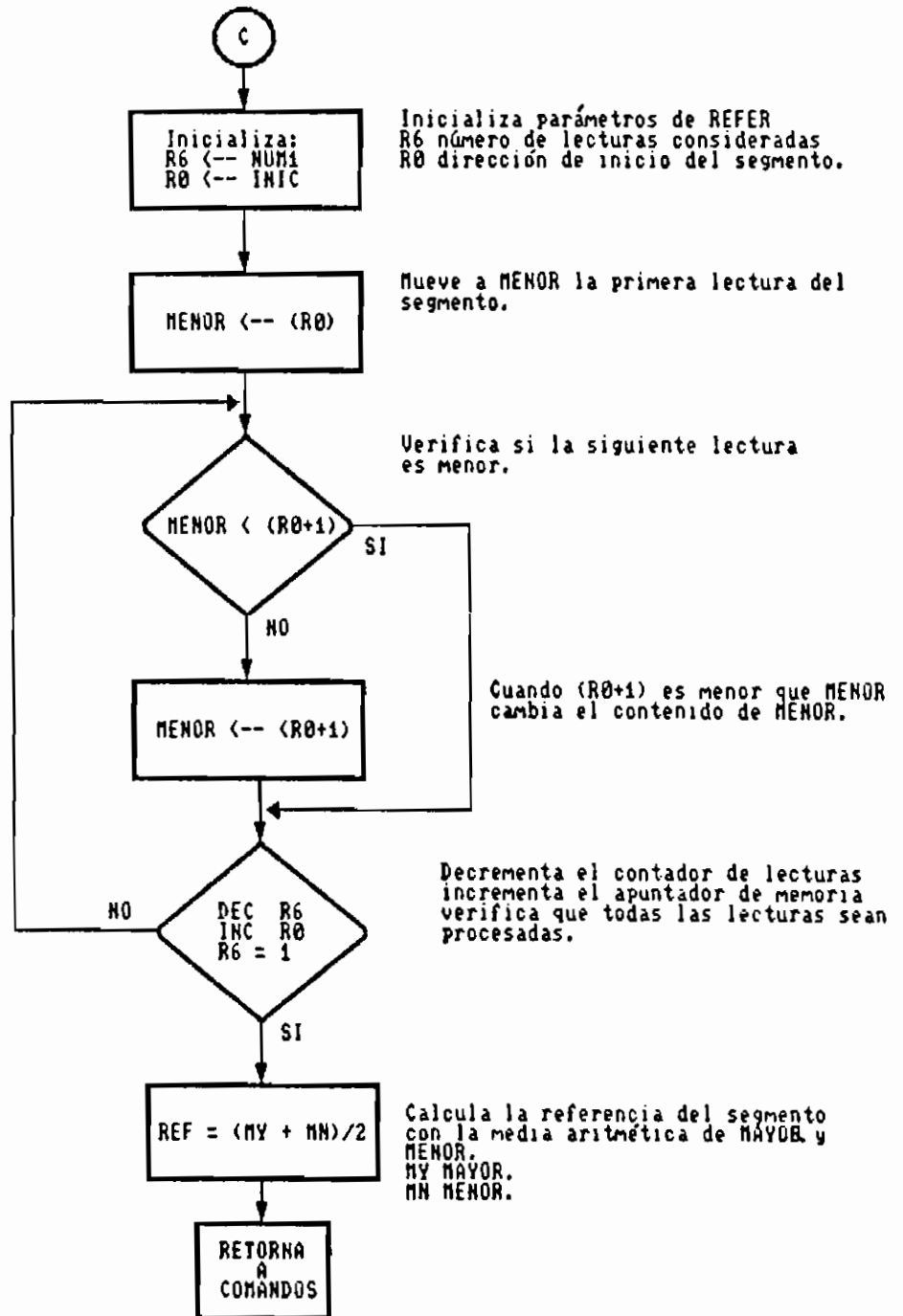


DIAGRAMA 10  
 CALCULO DE LA REFERENCIA DE UN SEGMENTO DE LECTURAS  
 CONTINUACION



```

;
; *****
;                               Ramiro E Morejón T.
;                               Mayo de 1989
;
; Rutina para decodificar las barras de código
; compatible
; con BSDEC.
; Contador   --- CONT
; Referencia --- REF
; EXTERN     VALIDO1
; EXTERN     VALIDO
; EXTERN     CURVA
;
; DEFSEG     BSCOD, START=00H, CLASS=CODE
; SEG        BSCOD
; * * * * *
;
; LJMP       30H           ; Salto a inicio de
;                               ; programa.
; ORG        0BH           ; Salto a inicio de
;                               ; rutina de incremento
;                               ; del contador.
; PUSH       PSW           ; Salva el STATUS.
; LJMP       OVFCNT
; * * * * *
;
; ORG        23H           ; Salto para atención de
;                               ; interrupción serial.
; LJMP       XRS
; * * * * *
;
; CONT       EQU          30H
; REF        EQU          60H           ; Contador.
; BAND       EQU          61H           ; Referencia.
; DATOS      EQU          62H           ; Banderas.
;                               ; flanco FB.
; DAT1       EQU          63H           ; Contiene secuencias de
;                               ; 8 bits recuperados de
;                               ; las etiquetas.
; TX232      EQU          64H           ; Contador de número de
;                               ; barras máximo 6.
; RX232      EQU          65H           ; Parámetro de entrada de
;                               ; la rutina de
;                               ; transmisión.
; NUM        EQU          66H           ; Parámetro de salida de
;                               ; la rutina de recepción.
; NUM2       EQU          6AH           ; Número de barras por
;                               ; caracter.
;                               ; Número de unos "1".

```





```

;      * * * * *
;      Rutina de recepción serial RS232
;      Mayo 1989.
;
RXRS:      PUSH          PSW          ; Salva el STATUS
           CLR          RI          ; Bandera de interrupción
           MOV          RX232,SBUF
           POP          PSW         ; Restaura el STATUS.
           RETI

;      * * * * *
;      Recibe los comandos del computador y la rutina
;      apropiada.
;      Julio 1989.
;
COMAND:
;      Define el código con una barras de valor "1" por
;      caracter.
;
COM1:      MOV          ACC,RX232
           CJNE        A,#031H,COM2  ; Comando 1
           MOV          RX232,#00H
           MOV          NUM2,#01H
;      Define el código con dos barras de valor "1" por
;      caracter.
;
COM2:      MOV          ACC,RX232
           CJNE        A,#032H,COM3  ; Comando 2
           MOV          RX232,#00H
           MOV          NUM2,#02H
;      Define el código con tres barras de valor "1" por
;      caracter.
;
COM3:      MOV          ACC,RX232
           CJNE        A,#033H,COM4  ; Comando 3
           MOV          RX232,#00H
           MOV          NUM2,#03H
;      Define el código de cuatro barras de valor "1" por
;      caracter.
;
COM4:      MOV          ACC,RX232
           CJNE        A,#034H,COM6  ; Comando 4
           MOV          RX232,#00H
           MOV          NUM2,#04H
;      Define un código con seis barras por caracter para su
;      representación.
;
COM6:      MOV          ACC,RX232
           CJNE        A,#036H,COM8  ; Comando 6

```

```

MOV          RX232,#00H
MOV          NUM,#06H

;   Define un código con ocho barras por caracter para su
;   representación.
;
COMB:        MOV          ACC,RX232
              CJNE       A,#038H,COMN   ; Comando B
              MOV        RX232,#00H
              MOV        NUM,#08H

;   Solicita que el circuito lector responda con el
;   caracter "H" para verificar la función de
;   comunicaciones.
COMN:        MOV          ACC,RX232
              CJNE       A,#04EH,COMR   ; Comando N
              MOV        RX232,#00H
              MOV        TX232,#48H    ; Responde con "H"
                                          ; para verificar la
                                          ; instalación del
                                          ; equipo.
              LCALL      TXRS

;   Llama la rutina de exploración de la etiqueta.
;
COMR:        MOV          ACC,RX232
              CJNE       A,#052H,COMT   ; Comando R
              MOV        RX232,#00H
              LCALL      BRINI          ; Inicia rutina de
                                          ; lectura de barras
COMT:        MOV          ACC,RX232
              CJNE       A,#054H,COMB   ; Comando T
              MOV        RX232,#00H
              CALL       TXDATOS        ; Cada vez que es
                                          ; llamada envia los
                                          ; bytes almacenados
                                          ; en la TABLA
                                          ; usando como
                                          ; puntero R0.

;   Envía el estado de BAND, con caracteres predefinidos,
;   que son interpretados por el computador:
;   E   error (error detectado al decodificar).
;   D   derecha ( sentido de lectura).
;   I   izquierda ( sentido de lectura).
;   0   Validación al terminar rutina BRINI.
;   1   Error por exploración incompleta.
COMB:        MOV          ACC,RX232
              CJNE       A,#042H,COMC   ; Comando B
              MOV        RX232,#00H

```

```

CALL          TXBAND          ; Envia el registro
                                ; BAND e inicializa
                                ; R0 con #20H.

```

```

; Llama la rutina para recuperar la información binaria,
; con el uso del algoritmo implementado en CURVA.
; (llamado a rutina externa).
;

```

```

COMC:        MOV          ACC,RX232
              CJNE       A,#043H,COMV  ; Comando C
              MOV        RX232,#00H
              LCALL      CURVA

```

```

; Verifica que ninguna lectura sea menor al 10% de la
; obtenida en la primera barra.
; (llamado a rutina externa).
;

```

```

COMV:        MOV          ACC,RX232
              CJNE       A,#056H,COMM  ; Comando V
              MOV        RX232,#00H
              LCALL      VALIDO        ; Chequea si es
                                         ; válida la lectura,
                                         ; en caso contrario
                                         ; marca el error a
                                         ; través de BAND.

```

```

; Verifica si el número de barras de valor "1"
; corresponde al definido por el código utilizado.
; (llamado a rutina externa).
;

```

```

COMM:        MOV          ACC,RX232
              CJNE       A,#4DH,COMS   ; Comando M
              MOV        RX232,#00H
              LCALL      VALIDO1

```

```

; Responde con el número de barras detectadas si es
; inmediatamente después del comando R. El número de
; caracteres si es utilizado después del comando C.
;

```

```

COMS:        MOV          ACC,RX232
              CJNE       A,#053H,COMI  ; Comando S
              MOV        RX232,#00H
              MOV        TX232,DAT1    ; envia el valor de

```

```

DAT1
              LCALL      TXRS
; Continúa lazo de espera de comandos.
;

```

```

COMI:        LJMP        COMAND

```

```

;      * * * * *
;      Ramiro E. Morejón T.
;      Mayo 1989
;      Rutina de transmisión serial RS232
;

```

TXRS:

```

      CLR          TI
      CLR          ES          ; Deshabilita la
                               ; posibilidad de
                               ; interrupción serial.

      MOV          SBUF,TX232
      JNB          TI,$        ; Espera hasta terminar
                               ; la transmisión del byte
                               ; almacenado en TX232.

      CLR          TI
      SETB         ES          ; Habilita la posibilidad
                               ; de interrupción serial.

      RET

```

```

;      * * * * *
;      Transmisión de datos con el comando T.
;      Mayo 1989.
;      Cada vez que es llamada envia un dato e incrementa el
;      puntero. (R0)
;

```

TXDATOS:

```

      MOV          TX232,@R0
      CALL         TXRS        ; Llama la rutina de
                               ; transmisión de un byte.

      INC          R0          ; Incrementa el apuntador
                               ; de la TABLA.

      RET

```

```

;      * * * * *
;      Transmisión de registro de banderas comando BAND.
;      Mayo 1989
;

```

TXBAND:

```

      MOV          R0,#20H     ; Inicializa el puntero
                               ; de lecturas. (R0)

      MOV          TX232,BAND
      CALL         TXRS        ; Llama la rutina de
                               ; transmisión de un byte.

      RET

```

```

;      * * * * *
;      Rutina de muestreo de lector de barras
;      deshabilita las interrupciones del p6rtico serial
;      arranca el contador 0
BRINI:
      MOV          R0,#020H
      MOV          DAT1,#000H; Inicializa DAT1.
      MOV          BAND,#001H; Inicializa BAND.
      MOV          IE,#82H   ; Habilita interrupciones
                               ; del TIMER 0.
      MOV          IP,#02H   ; Prioridad de
                               ; interrupci3n del
                               ; TIMER 0.
;      Inicia Lazo de espera por se1al del lector.
;
BRPRI:
      JB          P1.0,BRPRI
      CALL        BAND0
BRSEG:
      JNB         P1.0,BRSEG
      CALL        BAND0
      MOV         CONT,#000H; Inicializa el Contador.
      MOV         TL0,TH0   ; Inicializa TIMER 0.
      SETB        TR0       ; Arranca TIMER 0.

;      * * * * *
;      Inicia cuenta de las barras.
;
REFN:
      MOV         ACC,#0FFH
      SUBB        A,CONT
      JZ          BRFIN
      MOV         ACC,P1    ; Chequea el valor de
                               ; P1.0.
      XRL         A,BAND    ; Verifica la bandera FB.
      JB          ACC.0,FREFN
      LJMP        REFN
FREFN:
      CALL        TABLA
      MOV         CONT,#000H
      INC         DAT1
      CALL        BAND0    ; Cambia la bandera
      LJMP        REFN

```

```

;          * * * * *
BRFIN:    CLR          C
          MOV          ACC,DAT1
          SUBB         A,#04H
          JC           BRINI      ; Verifica si al menos 4
                                   ; barras han sido leídas.
          MOV          IE,#90H    ; Cambia IE e IP antes de
          MOV          IP,#10H    ; retornar al lazo de
                                   ; espera, habilita la
                                   ; posibilidad de
                                   ; interrupción serial.

          RET

;          * * * * *
;          Complemento de bandera FB , Cuando una transición ha
;          ocurrido, y contiene el estado de P1.0 la entrada de
;          datos provenientes del lector.
;          Mayo 1989.
;
;
BAND0:    MOV          ACC,BAND
          CPL          ACC.0
          MOV          BAND,ACC
          RET

TABLA:    MOV          @R0,CONT
          INC          R0
          RET
          END

```

```

; * * * * *
; Define como públicas las rutinas. Serán utilizadas por
; la rutina LECC.
        PUBLIC      VALIDO1
        PUBLIC      VALIDO
        PUBLIC      CURVA
        DEFSEG      AJUSTE, CLASS=CODE
        SEG         AJUSTE
BAND     EQU        62H
DATOS    EQU        63H ; Secuencias de 8 bits.
DAT1     EQU        64H ; Número de lecturas.
REF      EQU        61H ; Referencia de comparación.
CONT     EQU        60H ; Parámetro de entrada de
; CALREF.
AUX      EQU        67H ; Registro auxiliar.
MAYOR    EQU        68H ; Mayor de un grupo.
MENOR    EQU        69H ; Menor de un grupo.
NUM      EQU        6AH ; Número de barras por
; caracter.
INIC     EQU        6BH ; Inicio del segmento.
NUM1     EQU        6CH ; Parámetro de REFER.
NUM2     EQU        6DH ; Parámetro de ERROR.
; * * * * *
; Rutina para recuperar los la información almacenada en
; las etiquetas. Como un código binario.
; R1 contiene la dirección de inicio de las localidades
; asignadas a los caracteres recuperados.
;
CURVA:
        MOV         IE, #00H ; Deshabilita todas las
; interrupciones.
        MOV         IP, #00H
        MOV         ACC, NUM ; Toma los valores
; definidos para el
número
; de bits.
        MOV         NUM1, ACC
; Calcula el número de segmentos de la lectura y lo
; almacena en DAT1.
;
        MOV         ACC, DAT1
        SUBB        A, #09H ; Resta el número de
; barras de inicio y fin.
        MOV         B, NUM ; Divide para el número
de
; barras por caracter.
        DIV         AB
        MOV         DAT1, ACC
; Inicializa la rutina con el número de caracteres y la

```

```

; dirección de inicio del segmento de datos.
;
      MOV          R7,DAT1    ; Número de segmentos.
      MOV          R1,#20H    ; Dirección de inicio
                                ; de datos.
; Verifica que la secuencia de inicio corresponda a 17H,
; cuando la etiqueta es leída de izquierda a derecha
; tomando como referencia los caracteres que acompañan a
; las barras.
      MOV          ACC,24H    ; Cuarta barra.
      ADD          A,21H      ; Segunda barra.
      MOV          B,#02H    ; Calcula la referencia.
      DIV          AB
      MOV          REF,ACC
      MOV          R0,#20H    ; Define la dirección de
                                ; inicio, y el número de
                                ; barras como parámetros
                                ; de CURV.

      MOV          R6,#05H
      LCALL        CURV
      MOV          ACC,#17H  ; Verifica la secuencia
de                                     ; inicio. ( 17H ).

      SUBB         A,DATOS
      MOV          BAND,#49H ; BAND = 1 , si es
                                ; identificada
plenamente.
      JZ           CNT1      ; Continua
decodificación.
; Si la primera secuencia no es detectada, asume que la
; exploración se ejecutó en el otro sentido, en este caso
; debe verificarse 0DH.
;
      MOV          ACC,23H    ; Cuarta barra.
      ADD          A,22H      ; Tercera barra.
      MOV          B,#02H    ; Calcula la referencia.
      DIV          AB
      MOV          REF,ACC
      MOV          R0,#20H
      MOV          R6,#04H
      LCALL        CURV
      MOV          ACC,#0DH  ; Verifica la secuencia
de                                     ; inicio. ( 0DH ).

      SUBB         A,DATOS
      MOV          BAND,#44H ; BAND = D
      JNZ          CNT2
      MOV          R0,#24H
      CLR          P1.1
LCUR1:

```



```

MOV          INIC,R0
MOV          NUM,NUM1
LCALL       REFER          ; Calcula la referencia
                                ; del segmento.

MOV          R6,NUM
MOV          R0,INIC
LCALL       CURV1
MOV          @R1,DATOS
INC         R1
MOV          ACC,INIC
ADD         A,NUM1
MOV          R0,ACC
DJNZ       R7,LCUR1
; Verifica secuencia de final
;

MOV          INIC,R0
MOV          ACC,@R0
INC         R0
INC         R0
INC         R0
ADD         A,@R0
MOV          B,#02H
DIV         AB
MOV          REF,ACC
MOV          R0,INIC
MOV          R6,#05H
LCALL       CURV
MOV          ACC,#1DH      ; verifica la secuencia
de                                     ; final.

SUBB       A,DATOS
JZ         FIN2
MOV          BAND,#45H
FIN2:      MOV          IE,#90H      ; Habilita interrupción
                                ; serial antes de
regresar                                     ; a lazo de espera de
                                                ; comandos.

MOV          IP,#10H
SETB       P1.1
RET

CNT2:      MOV          IE,#90H      ; Habilita interrupción
                                ; serial antes de
regresar                                     ; a lazo de espera de
                                                ; comandos.

MOV          IP,#10H
MOV          BAND,#45H
RET

```

```

CNT1:      MOV          R0, #25H
LCUR:      MOV          INIC, R0
           MOV          NUM, NUM1
           LCALL       REFER
           MOV          R6, NUM
           MOV          R0, INIC
           LCALL       CURV
           MOV          @R1, DATOS
           INC          R1
           MOV          ACC, INIC
           ADD          A, NUM1
           MOV          R0, ACC
           DJNZ        R7, LCUR
;          Verifica secuencia de final
;
           MOV          INIC, R0
           MOV          ACC, @R0
           INC          R0
           ADD          A, @R0
           MOV          B, #02H
           DIV          AB
           MOV          REF, ACC
           MOV          R0, INIC
           MOV          R6, #04H
           LCALL       CURV
           MOV          ACC, #0BH ; Verifica la secuencia
;                               ; de final.
           SUBB        A, DATOS
           JZ          FIN1
           MOV          BAND, #45H
FIN1:      MOV          IE, #90H ; Habilita interrupción
;                               ; serial antes de
regresar ; a lazo de espera de
;                               ; comandos.
           MOV          IP, #10H
           RET
;          * * * * *
;          CUR: requiere de parámetros REF con el valor de
;          referencia del segmento.
;          R6 contiene la longitud del segmento.
;          R0 contiene la dirección de inicio del segmento.
;          REF es el parámetro de salida de REFER
;
CURV:      MOV          DATOS, #00H
CUR:       CLR          C
           MOV          ACC, @R0

```

```

        SUBB          A, REF
        MOV           ACC, DATOS
        CPL          C
        RLC          A
        MOV           DATOS, ACC
        INC          R0
        DJNZ         R6, CUR
        RET
;      * * * * *
;
CURV1:  MOV           DATOS, #00H
CUR1:   CLR          C
        MOV           ACC, @R0
        SUBB         A, REF
        MOV           ACC, DATOS
        CPL          C
        RRC          A
        MOV           DATOS, ACC
        INC          R0
        DJNZ         R6, CUR1
        MOV           ACC, #08H
        SUBB         A, NUM
        JZ            ROTE1
ROTE:   MOV           R6, ACC
        CLR          C
        MOV           ACC, DATOS
        RRC          A
        MOV           DATOS, ACC
        DJNZ         R6, ROTE
ROTE1:  RET
;      * * * * *
;      Verifica si no existen lecturas menores al 10%
;      consideradas erroneas.
;
VALIDO: MOV           R0, #20H
        MOV           R6, DAT1
        MOV           B, #0AH
        MOV           ACC, @R0
        DIV          AB
        MOV           REF, ACC
CHEQ:   MOV           ACC, @R0
        SUBB         A, REF
        JNC          CHEF
        MOV           BAND, #46H
CHEF:   INC          R0
        DJNZ         R6, CHEQ

```

```

      RET
;    * * * * *
;    Rutina para verificar el número de unos "1"
;    de cada código recuperado.
;    Nota.- Debe ser ejecutada después de realizar
;    la decodificación.
;
VALIDO1:
      MOV          R0,#20H
      MOV          R6,DAT1
VAL4:  MOV          R7,#08H
      MOV          ACC,@R0
      MOV          R5,#00H
      CLR          C
VAL2:  RLC          A
      JNC          VAL1
      INC          R5
VAL1:  DJNZ         R7,VAL2
      MOV          ACC,NUM2
      CLR          C
      SUBB         A,R5
      JZ           VAL5
      MOV          BAND,#45H
      LJMP        VAL6
VAL5:  INC          R0
      DJNZ         R6,VAL4
VAL6:  RET
;    * * * * *
;    Cálculo de la referencia de un segmento con el número
;    mayor y menor detectados.
;    Parámetros de entrada: NUM longitud del segmento.
;                           INIC dirección de inicio.
;
REFER:
      MOV          R6,NUM1
      MOV          R0,INIC
;
;    Obtiene el mayor de un segmento de memoria cuyo inicio
;    está en R0 y la longitud está en R6.
;
MAY:  DEC          R6
      MOV          MAYOR,@R0
MAY2: INC          R0
      MOV          AUX,@R0
      MOV          ACC,MAYOR
      CLR          C
      SUBB         A,AUX
      JNC          MAY1
      MOV          ACC,AUX
      XCH         A,MAYOR

```

```

MAY1:    DJNZ      R6,MAY2
          MOV      R6,NUM1
          MOV      R0,INIC
;
;      Obtiene el menor de un segmento de memoria cuyo
;      inicio está en R0 y la longitud está en R6.
;
MEN:      DEC      R6
          MOV      MENOR,@R0
MEN2:    INC      R0
          MOV      AUX,@R0
          MOV      ACC,MENOR
          CLR      C
          SUBB    A,AUX
          JC      MEN1
          MOV      ACC,AUX
          XCH     A,MENOR
MEN1:    DJNZ     R6,MEN2
;
;      Calcula la referencia del segmento en consideración
;      tomando el mayor y el menor , obteniendo la media
;      aritmética.
          MOV      A,MAYOR
          ADD     A,MENOR
          MOV     B,#02H
          DIV     AB
          MOV     REF,ACC
          RET
          END

```

DIAGRAMA 11  
PROGRAMA ESDEC.EXE

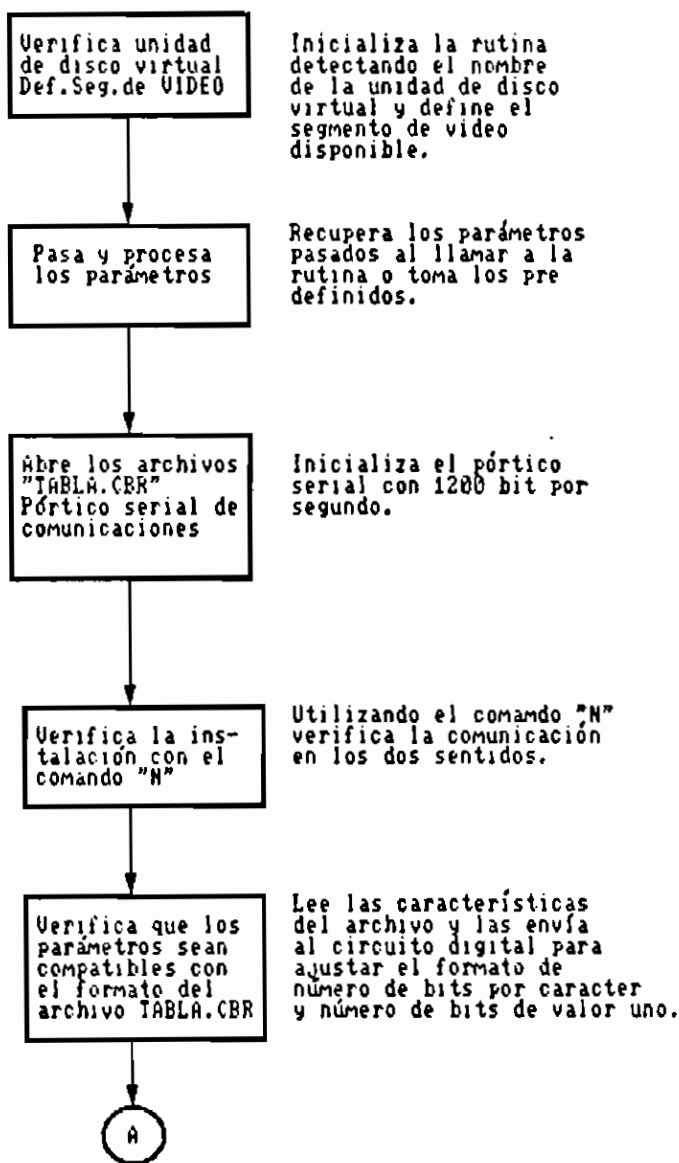


DIAGRAMA 11  
PROGRAMA BSDEC.EXE  
CONTINUACION

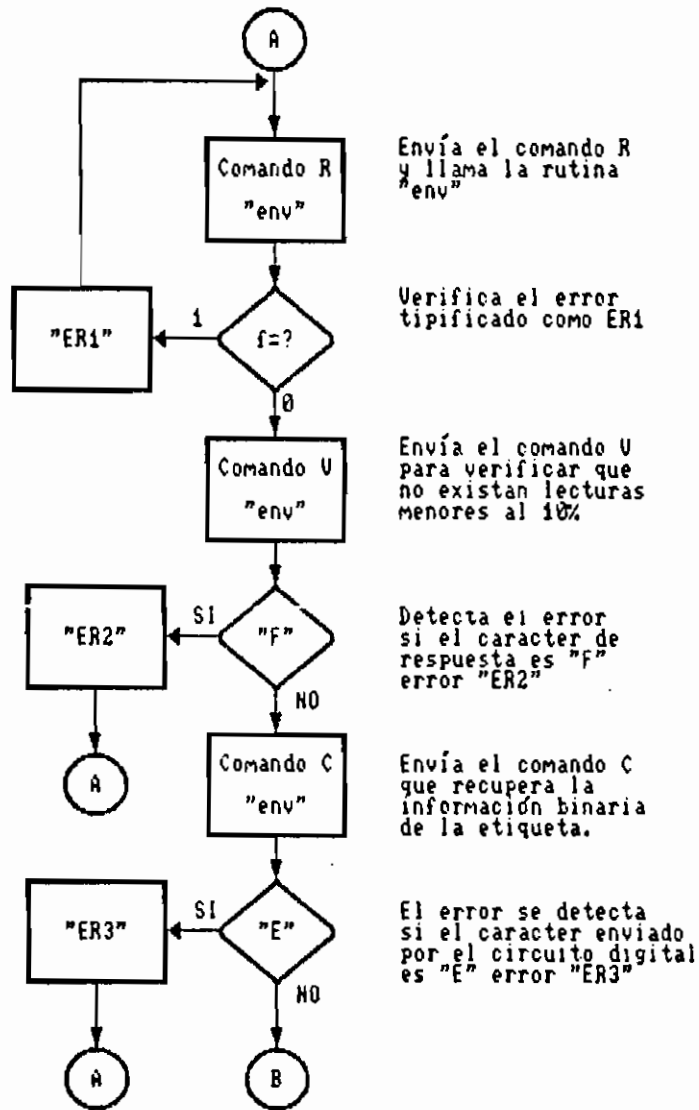


DIAGRAMA 11  
PROGRAMA BSDEC.EXE  
CONTINUACION

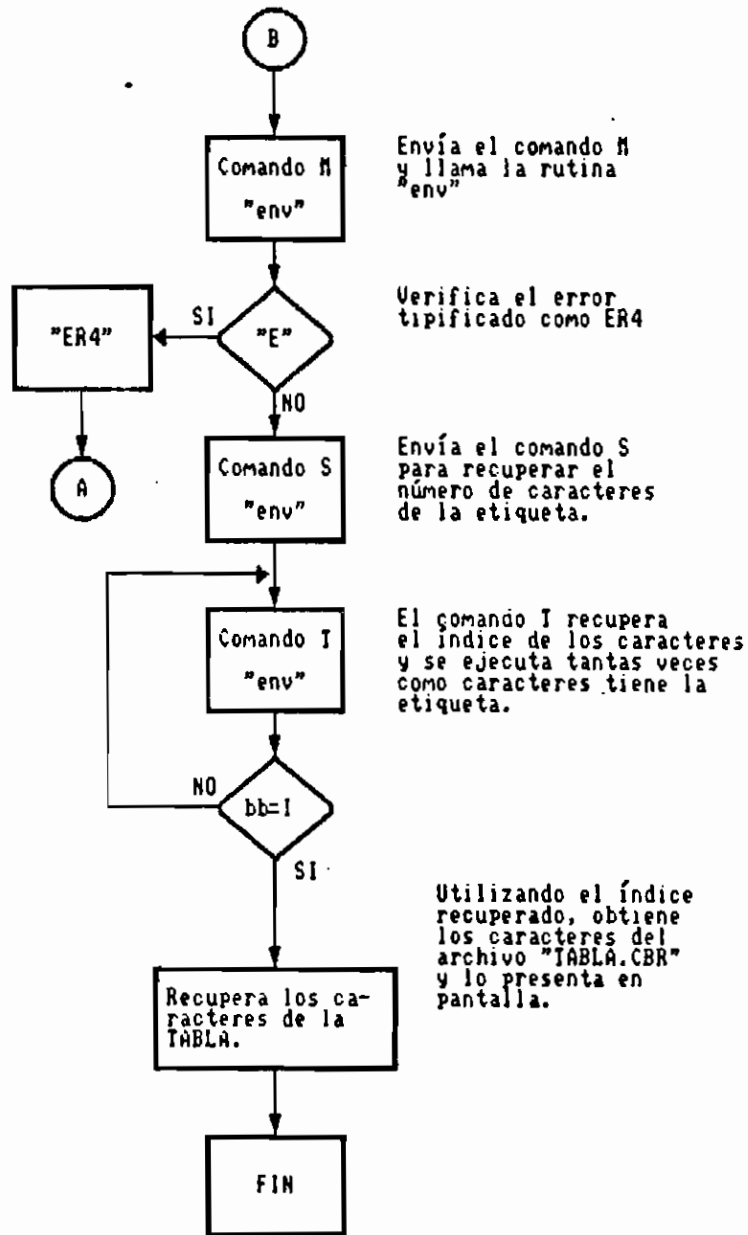
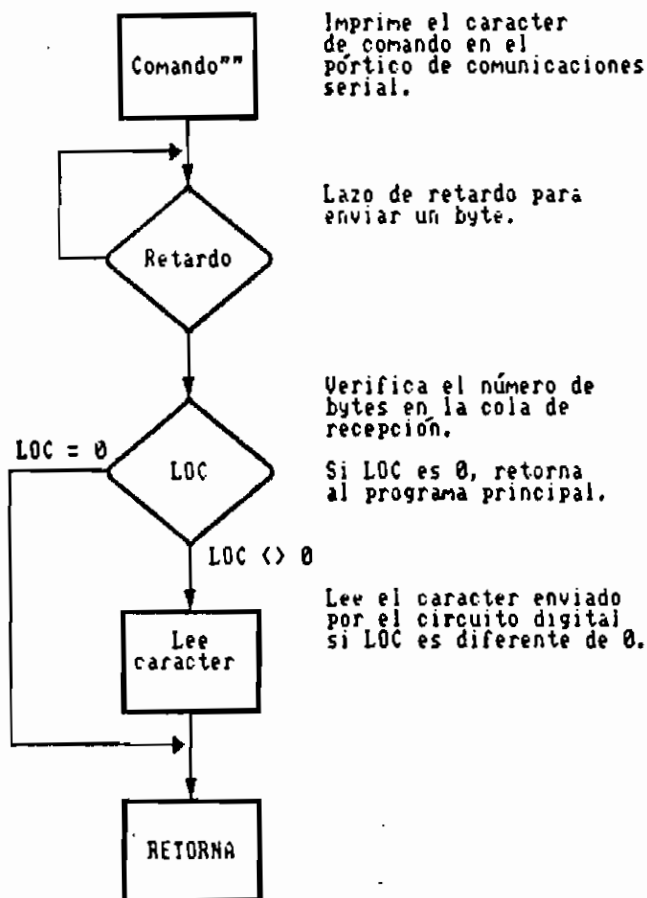




DIAGRAMA 11  
 PROGRAMA ESDEC.EXE  
 CONTINUACION  
 SUBROUTINA "ENV"



```

DECLARE SUB cuadro (ix!, iy!, jx!, jy!)
ON ERROR GOTO Falla
DEF SEG = 0

' ===== OPCIONES =====
'   La rutina Opciones recupera los argumentos del
'   programa, los que deben ser escritos separados por "/"
'   ,además no es admisible un caracter blanco como parte
'   de un nombre.
'=====
'

A = PEEK(1141)
IF A = 0 THEN
  d$ = "C"
ELSEIF A = 1 THEN
  d$ = "D"
ELSEIF A = 2 THEN
  d$ = "E"
END IF
  d1$ = d$ + ":lec"
,
'   LEC.CBR es le archivo de salida del código recuperado
'   valido cuando la opción l=a.
,
d2$ = d$ + ":LECT.CBR"

,
'   Determina el segmento de video
,

A = PEEK(1097)
IF A < 6 THEN
DEF SEG = &HB800
ELSEIF A = 7 THEN
DEF SEG = &HB000
ELSEIF A > 13 THEN
DEF SEG = &HA800
END IF

A$ = COMMAND$
IF UCASE$(A$) = "S" THEN
  PRINT "BSDEC U=x/A=xxxxx/N=x/M=x/l=x"
  RESET
  END
END IF

k = 1

```

```

A = LEN(A$)
DIM A$(A + 5), b$(10)
in = 1
FOR I = 1 TO A
A$(I) = MID$(A$, I, 1)
IF ASC(A$(I)) = 47 THEN
fi = I - 1
A$(I) = ""
FOR j = in TO fi
IF A$(j) <> " " THEN
A$(I) = A$(I) + A$(j)
END IF
NEXT
in = I + 1
b$(k) = A$(I)
k = k + 1
END IF
NEXT
fi = A
b$ = ""
FOR j = in TO fi
b$(k) = b$(k) + A$(j)
NEXT

```

' Para control del lector la opciones admisibles son:  
' U= X ,unidad en la que se encuentra el archivo \*\*.CBR  
que contiene los códigos binarios utilizados. Las  
unidades permitidas son: a:,b:, c:, d: y e:.  
Predefinido A:  
' A= XXXX.CBR , nombre del archivo que contiene los  
códigos binarios. Predefinido TABLA.CBR.  
' N= X ,número de bits del código EPN que pueden ser 6,  
8. Predefinido 8.  
' M = X ,n mero de unos del código binario  
' I = X, si x es I entonces muestra en la pantalla el  
código recuperado si x es A el código ser almacenado  
en un archivo LECT.CBR en el disco virtual.

```

FOR I = 1 TO k
b = LEN(b$(I))
c$ = MID$(b$(I), 1, 2)
IF c$ = "U=" THEN
u$ = MID$(b$(I), 3, 1)
ELSEIF c$ = "A=" THEN
tabl$ = MID$(b$(I), 3, b - 2)
ELSEIF c$ = "N=" THEN
num$ = MID$(b$(I), 3, b - 2)
ELSEIF c$ = "M=" THEN
munos$ = MID$(b$(I), 3, b - 2)

```

```

ELSEIF c$ = "I=" THEN
imp$ = MID$(b$(1), 3, b - 2)
END IF
NEXT

```

```

IF u$ = "" THEN
u$ = "A"
END IF
IF tabl$ = "" THEN
tabl$ = "TABLA"
END IF
IF num$ = "" THEN
num$ = "8"
END IF
IF muno$ = "" THEN
muno$ = "3"
END IF
IF imp$ = "" THEN
imp$ = "I"
END IF

```

```

tabl$ = u$ + ":" + tabl$ + ".CBR"
camp = VAL(num$)
muno = VAL(muno$)

```

```

,
' Rutina de ingreso utilizando el lector de código de
barras en este modo se envian caracteres de comado al
circuitio 8031
,

```

```

IF imp$ = "I" THEN
LOCATE 22, 45
PRINT "Lector on"
END IF
,
,

```

```

' Inicializa el pörtico serial con 1200 baudios, no
chequeo de paridad 8 bits de datos
,

```

```

OPEN "com1:1200,n,8,,rs,cs0,ds0" FOR RANDOM AS #1
,

```

```

' Se abre el archivo que contiene la tabla de los códigos
binarios de los caracteres.
,

```

```

OPEN tab1$ FOR RANDOM AS #20
FIELD #20, camp AS cod5$
,
,   verifica las características del archivo
,
GET #20, 1
cmp = VAL(codbr$)
GET #20, 2
mxu = VAL(codbr$)

t = 0
,
,   f es utilizado como bandera de recepción cuando se
envían comandos en los que se espera una respuesta
después de que la acción desencadenada no tiene una
respuesta inmediata.
,
f = 0
ke$ = STR$(camp)
GOSUB env
ke$ = STR$(muno)
GOSUB env

,
,   El comando N para verificar que el equipo está
instalado.
,
inst:
ij = 0
DO
ke$ = "n"
GOSUB env
IF ra$ = "H" THEN EXIT DO
ij = ij + 1
IF ij = 10 THEN
BSAVE d1$, 0, 4000
CALL cuadro(20, 7, 60, 14)
LOCATE 10, 24
PRINT "Equipo no disponible"
LOCATE 11, 24
PRINT "Verifique las instalaciones"
LOCATE 12, 24
PRINT "Presione ENTER para continuar"
DO
IF INKEY$ = CHR$(13) THEN BLOAD d1$, 0: GOTO inst
IF INKEY$ = CHR$(27) THEN RESET: END
LOOP

```

```
END IF
LOOP
```

```
lecr:
,
' El comando "r" inicia la rutina de lectura.
,
ke$ = "R"
GOSUB env
,
' Lazo de espera por una respuesta del lector, la que
indica que una exploración se concluyó.
,
```

```
DO
ke$ = "b"
GOSUB env
IF f = 1 THEN
IF ASC(ra$) = 1 THEN SOUND 200, 10: GOTO lecr
IF ASC(ra$) = 0 THEN EXIT DO
END IF
LOOP
```

```
ke$ = "v"
GOSUB env
ke$ = "b"
GOSUB env
IF ra$ = "F" THEN SOUND 200, 10: GOTO lecr
```

```
,
' Inicia la rutina que contiene el algoritmo de
recuperación de unos (1) y ceros (0).
,
```

```
ke$ = "c"
GOSUB env
,
' Lazo de espera por fin de la rutina.
,
```

```
DO UNTIL f = 1
ke$ = "b"
GOSUB env
ch$ = ra$
LOOP
IF ch$ = "E" THEN SOUND 200, 10: GOTO lecr
,
```

```
' chequea los errores cuando se aplica el algoritmo para
recuperar los bits de información.
```

```

'
ke$ = "m"
GOSUB env
ke$ = "b"
GOSUB env
IF ra$ = "E" THEN SOUND 200, 10: GOTO lecr
'
' Comando "s" pide se responda con el número de lecturas
' si es ejecutado antes de "c" y si es posterior ser el
' número de caracteres las respuesta
'
ke$ = "s"
GOSUB env
bb = INT(b)
'
' Regresa el apuntador de memoria al inicio y pide se
' envíen los valores de las lecturas o los
' correspondientes bits recuperados.
'
ke$ = "b"
GOSUB env
DIM p(100), car$(100)
FOR l = 1 TO bb
'
' "t" pide se envíe un byte cada vez que es utilizado
' este comando.
'
ke$ = "T"
PRINT #1, ke$

'===== Retardo=====
' Retardo de 10 ms para cualquier procesador, utilizado
' para garantizar que todos los bits en una transmisión
' serial sean recibidos cuando se tabaje a una velocidad
' de 1200 baudios.
'
=====
'

Inicio! = TIMER
DO UNTIL (TIMER - Inicio! > .001) OR (TIMER - Inicio! < 0)
LOOP

A = LOC(1)
IF A = 0 THEN
f = 0
ELSE
f = 1
ra$ = INPUT$(LOC(1), 1)

```

```

b = ASC(ra$)
p(i) = b
END IF
NEXT

```

```

IF ch$ = "D" THEN
FOR i = 1 TO INT(bb / 2)
chx = p(i)
p(i) = p(bb - i + 1)
p(bb - i + 1) = chx
NEXT
END IF

```

```

FOR i = 1 TO bb
GET #20, 400 + p(i)
car$(i) = cod5$
NEXT
car$ = ""
FOR i = 1 TO bb
car$ = car$ + MID$(car$(i), 8, 1)
NEXT
inp$ = car$
COLOR 4, 3
IF imp$ = "I" THEN
LOCATE 20, 10
PRINT inp$
END IF
IF imp$ = "A" THEN
OPEN d2$ FOR OUTPUT AS 100
PRINT #100, inp$
END IF
GOTO iend

```

```

'
' Rutina que permite enviar los caracteres de comandos y
' espera una respuesta por parte del lector.
'

```

```

env:
ke$ = UCASE$(ke$)
PRINT #1, ke$

```

```

'=====Retardo=====
' Retardo de 10 ms para cualquier procesador, utilizado
' para garantizar que todos los bits en una transmisión
' serial sean recibidos cuando se tabaje a una velocidad
' de 1200 baudios.
'
'=====
'

```



```

Inicio! = TIMER
DO UNTIL (TIMER - Inicio! > .005) OR (TIMER - Inicio! < 0)
LOOP

,
'   verifica si algun caracter fue enviado como respuesta y
'   caso afirmativo define f =1, y lee del p rtico el
'   primer caracter.
,
A = LOC(1)
IF A = 0 THEN
f = 0
ELSE
f = 1
ra$ = INPUT$(LOC(1), 1)
b = ASC(ra$)
END IF
RETURN
,
'   Cierra el p rtico y el archivo que fueron abiertos
,

.iend:
CLOSE #20
CLOSE #1
BEEP
END

```

Falla:

```

arr = ERR
BSAVE dl$, 0, 4000
PRINT arr
CALL cuadro(20, 7, 60, 14)
IF arr = 57 THEN
'   Device I/O error
LOCATE 9, 25
PRINT "Error de Entreda/Salida"
LOCATE 11, 23
PRINT "C - continua \ ESC - abandona"
ELSEIF arr = 68 THEN
'   Device Unavailable
LOCATE 9, 25
PRINT "Recurso no disponible"
LOCATE 11, 23
PRINT "C - continua \ ESC - abandona"
ELSEIF arr = 24 THEN
'   Device timeout
LOCATE 9, 22
PRINT "Recurso no disponible por el momento"

```

```

LOCATE 11, 23
PRINT "C - continua \ ESC - abandona"
ELSEIF arr = 25 THEN
'      Device fault
LOCATE 9, 25
PRINT "El impresor está fuera de linea"
LOCATE 11, 23
PRINT "C - continua \ ESC - abandona"
ELSEIF arr = 27 THEN
'      Out of paper
LOCATE 9, 25
PRINT "El impresor no tiene papel"
LOCATE 11, 23
PRINT "C - continua \ ESC - abandona"
ELSEIF arr = 53 THEN
'      File no found
LOCATE 9, 25
PRINT "Archivo no encontrado"
LOCATE 11, 23
PRINT "C - continua \ ESC - abandona"
ELSEIF arr = 58 THEN
'      File already exists
LOCATE 9, 25
PRINT "Archivo existente"
LOCATE 11, 23
PRINT "C - continua \ ESC - abandona"
ELSEIF arr = 71 THEN
'      Disk not ready
LOCATE 9, 25
PRINT "Disco no está listo"
LOCATE 11, 23
PRINT "C - continua \ ESC - abandona"
ELSEIF arr = 70 THEN
'      Disk write protected
LOCATE 9, 25
PRINT "Potegido conta escritura"
LOCATE 11, 23
PRINT "C - continua \ ESC - abandona"

```

END IF

DO

```

Kei$ = INKEY$
IF UCASE$(Kei$) = "C" THEN
EXIT DO
END IF
IF Kei$ = CHR$(27) THEN
BLOAD d1$, 0
END
END IF

```

```

LOOP
BLOAD d1$, 0
RESUME

```

```

SUB cuadro (ix, iy, jx, jy)
'===== cuadro =====
'   Dibuja un recuadro entre los limites  iniciales (ix,iy)
'   y los limites finales (jx,jy).
'
'                               Ramiro E. Morejon T.
'                               Enero 1989
'=====
'
nx = ABS(jx - ix)
ny = ABS(jy - iy)
LOCATE iy, ix
PRINT "┌" ; STRING$(nx - 2, "-") ; "┐"
FOR I = 1 TO ny - 1
LOCATE iy + I, ix
PRINT "| " ; STRING$(nx - 2, " ") ; "| "
NEXT
LOCATE jy, ix
PRINT "└" ; STRING$(nx - 2, "-") ; "┘"
END SUB

```

DIAGRAMA 12  
PROGRAMA BSIMP.EXE

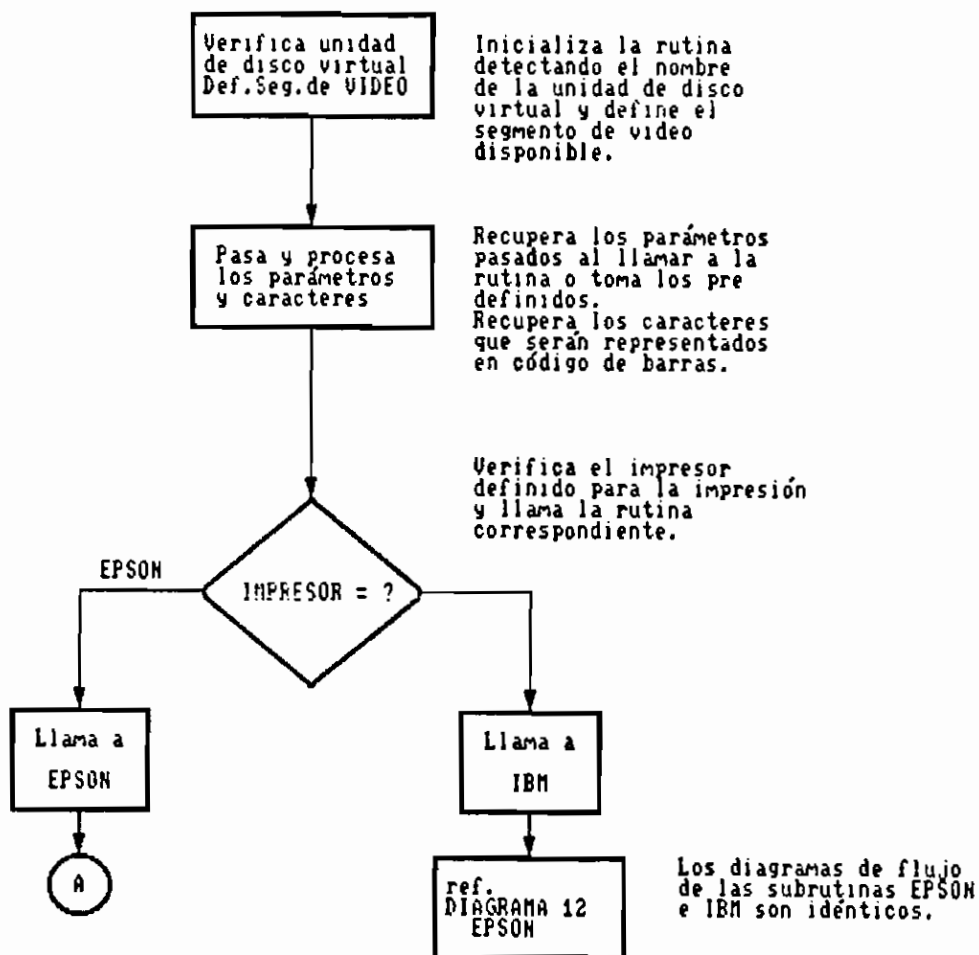


DIAGRAMA 12  
PROGRAMA BSIMP.EXE  
EPSON

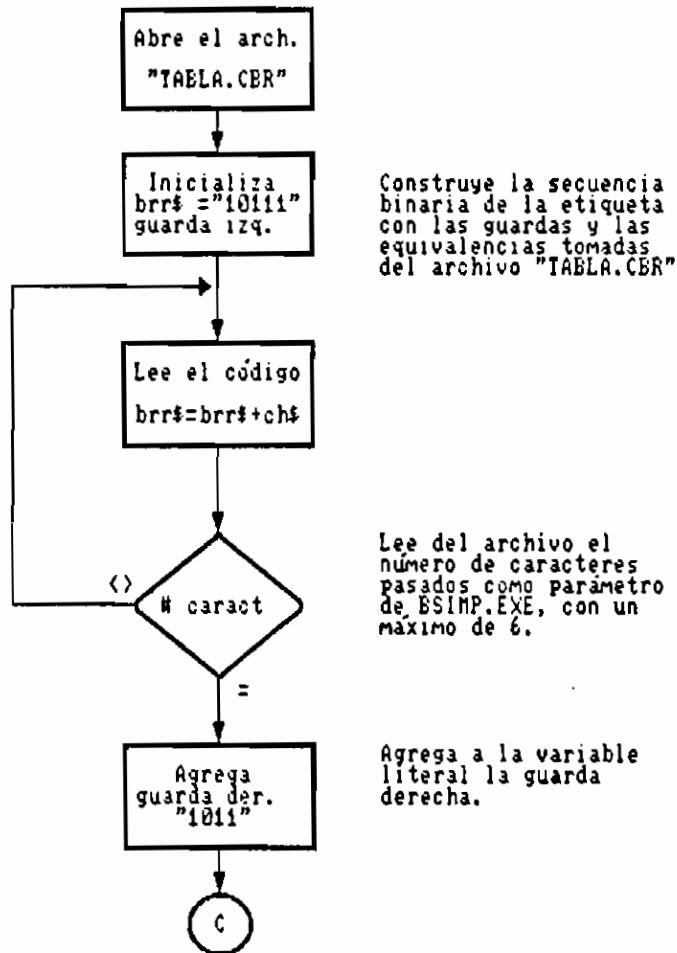
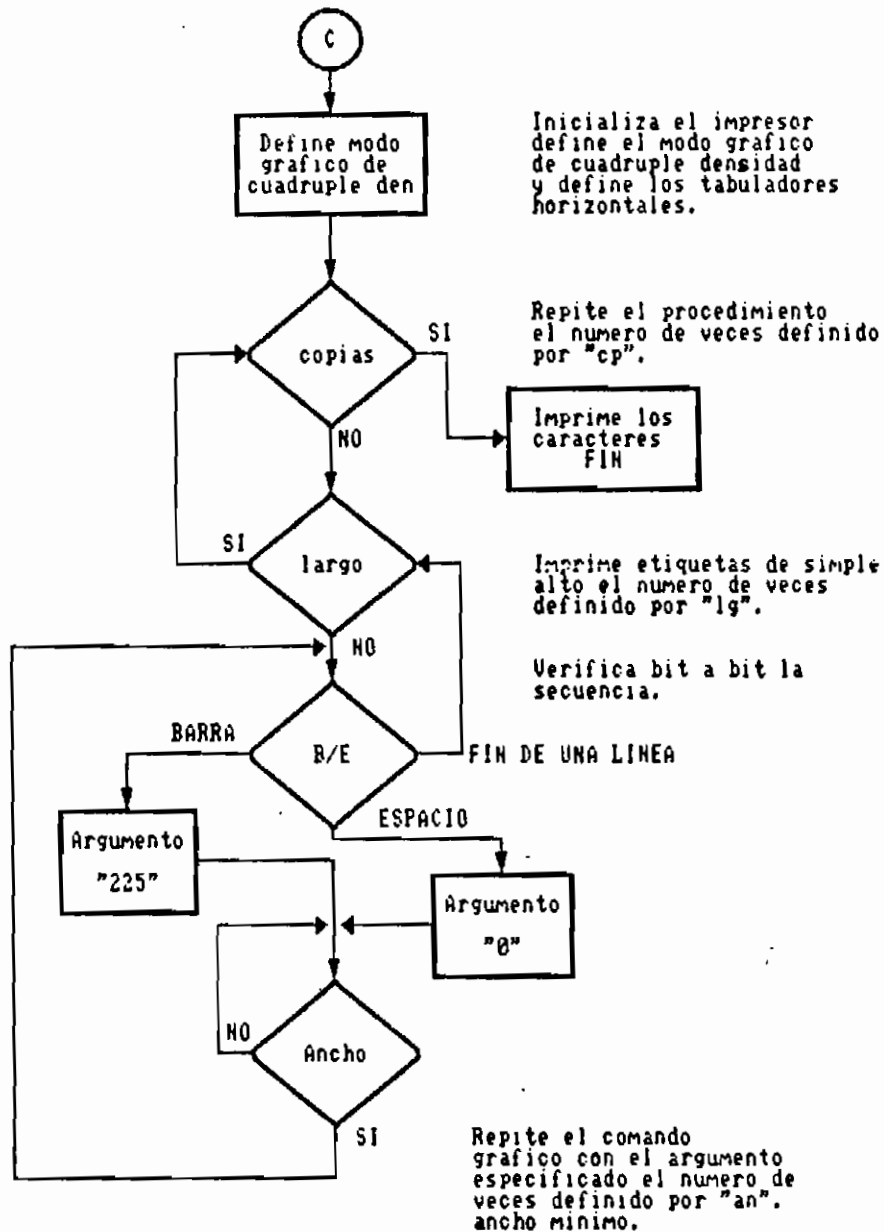


DIAGRAMA 12  
 PROGRAMA BSIMP.EXE  
 EPSON  
 CONTINUACION



```

DECLARE SUB cuadro (ix!, iy!, jx!, jy!)
DECLARE SUB IBM (inp$, lg!, an!, cp!)
DECLARE SUB EPSON (inp$, lg!, an!, cp!)
COMMON SHARED tabl$, ncam
ON ERROR GOTO Falla

```

```

DEF SEG = 0

```

```

' Define la unidad y el archivo para usar con las
' instrucciones BSAVE y BLOAD de rutina de error

```

```

a = PEEK(1141)
IF a = 0 THEN
    d$ = "C"
ELSEIF a = 1 THEN
    d$ = "D"
ELSEIF a = 2 THEN
    d$ = "E"
END IF
d1$ = d$ + ":imp"

```

```

' Determina el segmento de video

```

```

a = PEEK(1097)
IF a < 6 THEN
    DEF SEG = &HB800
ELSEIF a = 7 THEN
    DEF SEG = &HB000
ELSEIF a > 13 THEN
    DEF SEG = &HA800
END IF

```

```

'===== OPCIONES =====
' La rutina Opciones recupera los argumentos del
' programa, los que deben ser escritos separados por "/"
' ,además no es admisible un caracter blanco como parte
' de un nombre.
'=====

```

```

a$ = COMMAND$
k = 1
a = LEN(a$)
DIM a$(a + 5), b$(10)
in = 1
FOR I = 1 TO a

```

```

a$(1) = MID$(a$, 1, 1)
IF ASC(a$(1)) = 47 THEN
  fi = 1 - 1
  a$(1) = ""
  FOR j = in TO fi
    IF a$(j) <> " " THEN
      a$(1) = a$(1) + a$(j)
    END IF
  NEXT
  in = 1 + 1
  b$(k) = a$(1)
  k = k + 1
END IF
NEXT
fi = a
b$ = ""
FOR j = in TO fi
  b$(k) = b$(k) + a$(j)
NEXT

'      Para control del impresión las opciones admisibles
son:
'      U= X ,unidad en la que se encuentra el  archivo **.CBR
que
'      contiene los códigos binarios utilizados. Las unidades
'
'      permitidas son: a:, c: y d:. Predefinido A:
'      A= XXXX.CBR , nombre del archivo que contiene los
códigos
'      binarios. Predefinido TABLA.CBR.
'      l= XXXX ,impresor utilizado para crear las etiquetas.
'      Predefinido IBM.
'      L = X ,longitud de las barras de una etiqueta en
'      M = X ,ancho de las barras
'      C = X ,número de copias de una etiqueta. Predefinido 1
'
'      N = X ,número de bits del  código binario. Predefinido
8
'      D = XXXXXX, caracteres para codificar
FOR I = 1 TO k
b = LEN(b$(I))
C$ = MID$(b$(I), 1, 2)
IF C$ = "U=" THEN
  u$ = MID$(b$(I), 3, 1)
ELSEIF C$ = "A=" THEN
  tabl$ = MID$(b$(I), 3, b - 2)
ELSEIF C$ = "C=" THEN
  cp$ = MID$(b$(I), 3, b - 2)
ELSEIF C$ = "M=" THEN
  an$ = MID$(b$(I), 3, b - 2)

```



```

ELSEIF C$ = "L=" THEN
    lg$ = MID$(b$(1), 3, b - 2)
ELSEIF C$ = "I=" THEN
    inp$ = MID$(b$(1), 3, b - 2)
ELSEIF C$ = "N=" THEN
    ncam$ = MID$(b$(1), 3, b - 2)
ELSEIF C$ = "D=" THEN
    cod$ = MID$(b$(1), 3, b - 2)
END IF
NEXT

```

```

,
' Opciones predefinidas
,
' Unidad
,
IF u$ = "" THEN
    u$ = "A"
END IF
,
' Archivo
,
IF tabl$ = "" THEN
    tabl$ = "TABLA"
END IF
,
' Impresor
,
IF inp$ = "" THEN
    inp$ = "IBM"
END IF
,
' Largo
,
IF lg$ = "" THEN
    lg$ = "1"
END IF
,
' Ancho
,
IF an$ = "" THEN
    an$ = "2"
END IF
,
' Copias
,

```

```

IF cp$ = "" THEN
  cp$ = "1"
END IF
'
' Número de bits
'
IF ncam$ = "" THEN
  ncam$ = "8"
END IF
'
' Caracteres
'
IF cod$ = "" THEN
  cod$ = "EPN"
END IF

```

```

tabl$ = u$ + ":" + tabl$ + ".CBR"
cp = VAL(cp$)
an = VAL(an$)
lg = VAL(lg$)
ncam = VAL(ncam$)

```

```

'===== TIPO DE IMPRESOR =====
'                               Seleciona el tipo de impresor de entre los
definidos

```

```

'                               Ramiro E. Morejón T.
'                               MAYO 1989
'=====
'

```

```

IF inp$ = "IBM" THEN
  CALL IBM(cod$, lg, an, cp)
ELSEIF inp$ = "EPSON" THEN
  CALL EPSON(cod$, lg, an, cp)
END IF
END

```

```

Falla:
arr = ERR
BSAVE di$, 0, 4000
CALL cuadro(20, 7, 60, 14)
IF arr = 57 THEN
  ' Device I/O error
  LOCATE 9, 25
  PRINT "Error de Entreda/Salida"

```

```

LOCATE 11, 23
PRINT "C - continua \ ESC - abandona"
ELSEIF arr = 68 THEN
' Device Unavailable
LOCATE 9, 25
PRINT "Recurso no disponible"
LOCATE 11, 23
PRINT "C - continua \ ESC - abandona"
ELSEIF arr = 24 THEN
' Device timeout
LOCATE 9, 22
PRINT "Recurso no disponible por el momento"
LOCATE 11, 23
PRINT "C - continua \ ESC - abandona"
ELSEIF arr = 25 THEN
' Device fault
LOCATE 9, 25
PRINT "El impresor est fuera de linea"
LOCATE 11, 23
PRINT "C - continua \ ESC - abandona"
ELSEIF arr = 27 THEN
' Out of paper
LOCATE 9, 25
PRINT "El impresor no tiene papel"
LOCATE 11, 23
PRINT "C - continua \ ESC - abandona"
ELSEIF arr = 53 THEN
' File no found
LOCATE 9, 25
PRINT "Archivo no encontrado"
LOCATE 11, 23
PRINT "C - continua \ ESC - abandona"
ELSEIF arr = 58 THEN
' File already exists
LOCATE 9, 25
PRINT "Archivo existente"
LOCATE 11, 23
PRINT "C - continua \ ESC - abandona"
END IF

DO
Kei$ = INKEY$
IF UCASE$(Kei$) = "C" THEN
EXIT DO
END IF
IF Kei$ = CHR$(27) THEN
BLOAD d1$, 0
END
END IF
LOOP

```

```

SUB          EPSON          (inp$,          lg,          an,          cp)
'-----EPSON-----
'
'          Ramiro E. Morejón T.
'          Febrero 1989
'          APX - 80 / FX-286
'-----
'
      OPEN tabl$ FOR RANDOM AS #17
      FIELD #17, ncam AS codbr$
      l = LEN(inp$)
      brr$ = "10111"
      FOR j = 1 TO l
      ch$ = MID$(inp$, j, 1)
      ast% = ASC(ch$)
      GET #17, ast%
      brr$ = brr$ + codbr$
      NEXT j
      brr$ = brr$ + "1011"
      CLOSE #17

WIDTH LPRINT 255
OPEN "lpt1:" FOR OUTPUT AS #16
PRINT #16, CHR$(27); CHR$(68); CHR$(3); CHR$(5); CHR$(7);
FOR jnum = 1 TO cp
PRINT #16, CHR$(13)
PRINT #16, CHR$(27); CHR$(15);
PRINT #16, "          Morejon INC.";
LOCATE 22, 25
PRINT "espere por favor"
PRINT #16, CHR$(18);
FOR l = 1 TO 10
PRINT #16, CHR$(8);
NEXT l
PRINT #16, CHR$(27); "J"; CHR$(24);
FOR ila = 1 TO lg * 3
PRINT #16, CHR$(9);
pil = 0
pi0 = 255
lbr = LEN(brr$)
FOR ii = 1 TO lbr
bit$ = LEFT$(brr$, ii)
bit$ = RIGHT$(bit$, 1)
SWAP pil, pi0
IF bit$ = "0" THEN il = 1: GOTO lin
IF bit$ = "1" THEN il = 2:
lin:
l = il * an * 3
IF pil = 0 THEN
l = l

```

```
ELSEIF pi0 = 255 THEN
l = l - an
END IF

PRINT #16, CHR$(27); "Z"; CHR$(1); CHR$(0);
FOR i = 1 TO l
PRINT #16, CHR$(pi1);
NEXT
NEXT i
PRINT #16, CHR$(27); "J"; CHR$(22);
FOR ibs = 1 TO 1 + (l * lbrr) / 7
PRINT #16, CHR$(8);
NEXT ibs
NEXT ila
PRINT #16, CHR$(27); "X"; CHR$(1);
PRINT #16, CHR$(27); "M";
PRINT #16, CHR$(9);
PRINT #16, inp$;
NEXT jnum
PRINT #16, CHR$(13)
CLOSE #16
LOCATE 22, 25
PRINT " "

END SUB
```

```

SUB IBM (inp$, lg, an, cp)
'=====IBM=====
'
'           Ramiro E. Morejón T.
'           Febrero 1989
'           MSP - 45
'=====
      OPEN tab1$ FOR RANDOM AS #17
      FIELD #17, ncam AS codbr$
      l = LEN(inp$)
      brr$ = "10111"
      FOR j = 1 TO l
      ch$ = MID$(inp$, j, 1)
      ast% = ASC(ch$)
      GET #17, ast%
      brr$ = brr$ + codbr$
      NEXT j
      brr$ = brr$ + "1011"
      CLOSE #17

WIDTH LPRINT 255
OPEN "lpt1:" FOR OUTPUT AS #16
PRINT #16, CHR$(27); CHR$(88); CHR$(5); CHR$(100);
FOR jnum = 1 TO cp
PRINT #16, CHR$(13)
PRINT #16, CHR$(27); CHR$(15);
PRINT #16, "      Morejon INC.";
PRINT #16, CHR$(27); "J"; CHR$(22);
LOCATE 22, 25
PRINT "espere por favor"
PRINT #16, CHR$(18);
FOR ila = 1 TO lg * 3
PRINT #16, CHR$(27); CHR$(88); CHR$(5); CHR$(100);
pi1 = 0
pi0 = 255
lbr = LEN(brr$)
FOR ii = 1 TO lbr
bit$ = LEFT$(brr$, ii)
bit$ = RIGHT$(bit$, 1)
SWAP pi1, pi0
IF bit$ = "0" THEN il = 1: GOTO mlin
IF bit$ = "1" THEN il = 2:
mlin:
l = il * an * 3
PRINT #16, CHR$(27); "Z"; CHR$(1); CHR$(0);
FOR l = 1 TO l
PRINT #16, CHR$(pi1);
NEXT
NEXT ii
PRINT #16, CHR$(27); "J"; CHR$(22);

```

```

NEXT ila
PRINT #16, CHR$(27); CHR$(88); CHR$(5); CHR$(100);
PRINT #16, inp$;
NEXT jnum
PRINT #16, CHR$(13)
CLOSE #16
LOCATE 22, 25
PRINT "
END SUB

```

```

SUB cuadro (ix, iy, jx, jy)
'===== cuadro =====
'   Dibuja un recuadro entre los limites iniciales (ix,iy)
'   y los limites finales (jx,jy).
'
'                               Ramiro E. Morejón T.
'                               Enero 1989
'=====
'
nx = ABS(jx - ix)
ny = ABS(jy - iy)
LOCATE iy, ix
PRINT "┌" ; STRING$(nx - 2, "-") ; "┐"
FOR l = 1 TO ny - 1
LOCATE iy + l, ix
PRINT "| " ; STRING$(nx - 2, " ") ; "| "
NEXT
LOCATE jy, ix
PRINT "└" ; STRING$(nx - 2, "-") ; "┘"
END SUB

```


**HEWLETT  
PACKARD**

# HIGH RESOLUTION OPTICAL REFLECTIVE SENSOR

**HEDS-1000**

TECHNICAL DATA MARCH 1980

## Features

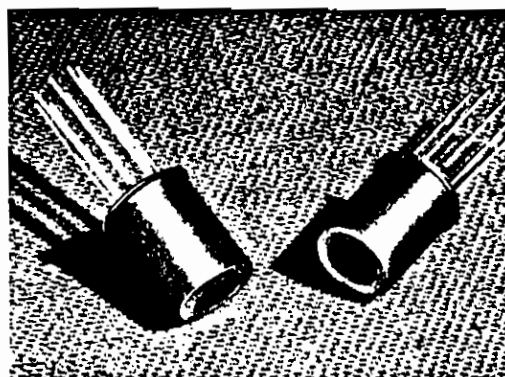
- FOCUSED EMITTER AND DETECTOR IN A SINGLE PACKAGE
- HIGH RESOLUTION — .190mm SPOT SIZE
- 700nm VISIBLE EMITTER
- LENS FILTERED TO REJECT AMBIENT LIGHT
- TO-5 MINIATURE SEALED PACKAGE
- PHOTODIODE AND TRANSISTOR OUTPUT
- SOLID STATE RELIABILITY

## Description

The HEDS-1000 is a fully integrated module designed for optical reflective sensing. The module contains a .178mm (.007 in.) diameter 700nm visible LED emitter and a matched I.C. photodetector. A bifurcated aspheric lens is used to image the active areas of the emitter and the detector to a single spot 4.27mm (.168 in.) in front of the package. The reflected signal can be sensed directly from the photodiode or through an internal transistor that can be configured as a high gain amplifier.

## Applications

Applications include pattern recognition and verification, object sizing, optical limit switching, tachometry, textile thread counting and defect detection, dimensional monitoring, line locating, mark, and bar code scanning, and paper edge detection.

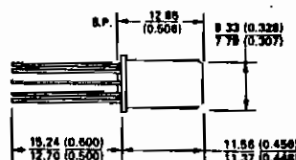
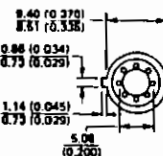
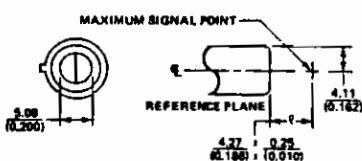


## Mechanical Considerations

The HEDS-1000 is packaged in a high profile 8 pin TO-5 metal can with a glass window. The emitter and photodetector chips are mounted on the header at the base of the package. Positioned above these active elements is a bifurcated aspheric acrylic lens that focuses them to the same point.

The sensor can be rigidly secured by commercially available two piece TO-5 style heat sinks, such as Thermalloy 2205, or Aavid Engineering 3215. These fixtures provide a stable reference platform and their tapped mounting holes allow for ease of affixing this assembly to the circuit board.

## Package Dimensions



### NOTES:

1. ALL DIMENSIONS IN MILLIMETERS AND (INCHES).
2. ALL UNTOLERANCED DIMENSIONS ARE FOR REFERENCE ONLY.
3. THE REFERENCE PLANE IS THE TOP SURFACE OF THE PACKAGE.
4. NICKEL CAN AND GOLD PLATED LEADS.
5. S.P. SEATING PLANE.
6. THE LEAD DIAMETER IS 0.46mm (0.018in.) TYP.



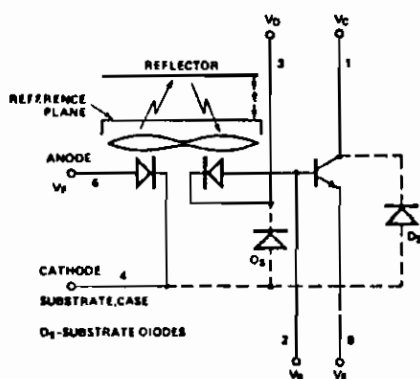
## Electrical Operation

The detector section of the sensor can be connected as a single photodiode, or as a photodiode transistor amplifier. When photodiode operation is desired, it is recommended that the substrate diodes be defeated by connecting the collector of the transistor to the positive potential of the power supply and shorting the base-emitter junction of the transistor. Figure 15 shows photocurrent being supplied from the anode of the photodiode to an inverting input of the operational amplifier. The circuit is recommended to improve the reflected photocurrent to stray photocurrent ratio by keeping the substrate diodes from acting as photodiodes.

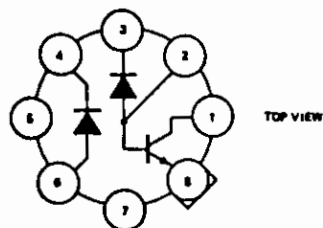
The cathode of the 700nm emitter is physically and electrically connected to the case-substrate of the device. Applications that require modulation or switching of the LED should be designed to have the cathode connected to the electrical ground of the system. This insures minimum capacitive coupling of the switching transients through the substrate diodes to the detector amplifier section.

The HEDS-1000 detector also includes an NPN transistor which can be used to increase the output current of the sensor. A current feedback amplifier as shown in Figure 6 provides moderate current gain and bias point stability.

SCHEMATIC DIAGRAM



CONNECTION DIAGRAM



Pin	FUNCTION
1	TRANSISTOR COLLECTOR
2	TRANSISTOR BASE, PHOTO DIODE ANODE
3	PHOTO DIODE CATHODE
4	LED CATHODE, SUBSTRATE CASE
5	NC
6	LED ANODE
7	NC
8	TRANSISTOR EMITTER

## Absolute Maximum Ratings at $T_A = 25^\circ\text{C}$

Parameter	Symbol	Min.	Max.	Units	Fig.	Notes
Storage Temperature	$T_S$	-40	+75	$^\circ\text{C}$		
Operating Temperature	$T_A$	-20	+70	$^\circ\text{C}$		
Lead Soldering Temperature 1.6mm from Sealing Plane			200 for 10 sec.	$^\circ\text{C}$		11
Average LED Forward Current	$I_F$		60	mA		2
Peak LED Forward Current	$I_{FPK}$		75	mA	1	1
Reverse LED Input Voltage	$V_R$		5	V		
Package Power Dissipation	$P_P$		120	mW		3
Collector Output Current	$I_O$		8	mA		
Supply and Output Voltage	$V_O, V_C, V_E$	-0.5	20	V		10
Transistor Base Current	$I_B$		5	mA		
Transistor Emitter Base Voltage	$V_{EB}$		5	V		

System Electrical/Optical Characteristics at  $T_A=25^\circ\text{C}$ 

Parameter	Symbol	Min.	Typ.	Max.	Units	Conditions	Fig.	Note	
Total Photocurrent ( $I_{PR}+I_{PS}$ )	$I_P$			375	nA	$T_A=-20^\circ\text{C}$	$I_F=35\text{mA}$ , $V_D=V_C=5\text{V}$	2,3	4
		100	140	250		$T_A=25^\circ\text{C}$			
		50				$T_A=70^\circ\text{C}$			
Reflected Photocurrent ( $I_{PR}$ ) to Internal Stray Photocurrent ( $I_{PS}$ )	$\frac{I_{PR}}{I_{PS}}$	4	6.5			$I_F=35\text{mA}$ , $V_C=V_D=5\text{V}$	3		
Transistor DC Static Current Transfer Ratio	$h_{FE}$	50				$T_A=-20^\circ\text{C}$	$V_{CE}=5\text{V}$ , $I_C=10\mu\text{A}$	4,5	
		100	200		$T_A=25^\circ\text{C}$				
Slew Rate			.08		V/ $\mu\text{s}$	$R_L=100\text{K}$ $I_{PK}=50\text{mA}$ $R_F=10\text{M}$ $I_{ON}=100\mu\text{s}$ , Rate = 1kHz	6		
Image Diameter	d		.17		mm	$I_F=35\text{mA}$ , $\ell=4.27\text{mm}$ (0.168in)	8,10	8,9	
Maximum Signal Point	$\ell$	4.02	4.27	4.52	mm	Measured from Reference Plane	9		
50% Modulation Transfer Function	MTF		2.5		$I_{PR}/\text{mm}$	$I_F=35\text{mA}$ , $\ell=4.27\text{mm}$	10,11	5,7	
Depth of Focus	$\Delta\ell$ FWHM		1.2		mm	50% of $I_P$ at $\ell=4.27\text{mm}$	9	5	
Effective Numerical Aperture	N.A.		.3						
Image Location	D		.51		mm	Diameter Reference to Centerline $\ell=4.27\text{mm}$		8	
Thermal Resistance	$\theta_{JC}$		85		$^\circ\text{C/W}$				

Detector Electrical/Optical Characteristics at  $T_A=25^\circ\text{C}$ 

Parameter	Symbol	Min.	Typ.	Max.	Units	Conditions	Fig.	Note	
Dark Current	$I_{PD}$		5	120	pA	$T_A=25^\circ\text{C}$	$I_F=0$ , $V_D=5\text{V}$ ; Reflection=0%		
				10	nA	$T_A=70^\circ\text{C}$			
Capacitance	$C_D$		45		pF	$V_D=0\text{V}$ , $I_P=0$ , $f=1\text{MHz}$			
Flux Responsivity	$R_\phi$		.22		$\frac{\text{A}}{\text{W}}$	$\lambda=700\text{nm}$ , $V_D=5\text{V}$	12		
Detector Area	$A_D$		.160		$\text{mm}^2$	Square, with Length=.4mm/Side			

Emitter Electrical/Optical Characteristics at  $T_A=25^\circ\text{C}$ 

Parameter	Symbol	Min.	Typ.	Max.	Units	Conditions	Fig.	Note
Forward Voltage	$V_F$		1.8	1.8	V	$I_F=35\text{mA}$	13	
Reverse Breakdown Voltage	$BV_R$	5			V	$I_R=100\mu\text{A}$		
Radiant Flux	$\phi_E$	5	9.0		$\mu\text{W}$	$I_F=35\text{mA}$ , $\lambda=700\text{nm}$	14	
Peak Wavelength	$\lambda_P$	680	700	720	nm	$I_F=35\text{mA}$	14	
Thermal Resistance	$\theta_{JC}$		150		$^\circ\text{C/W}$			
Temperature Coefficient of $V_F$	$\Delta V_F/\Delta T$		-1.2		$\text{mV}/^\circ\text{C}$	$I_F=35\text{mA}$		

## Transistor Electrical Characteristics at $T_A = 25^\circ\text{C}$

Parameter	Symbol	Min.	Typ.	Max.	Units	Conditions	Fig.	Note
Collector-Emitter Leakage	$I_{CEO}$		1		$\mu\text{A}$	$V_{CE}=5\text{V}$		
Base-Emitter Voltage	$V_{BE}$		.8		V	$I_C=10\mu\text{A}, I_B=70\text{nA}$		
Collector-Emitter Saturation Voltage	$V_{CE(SAT)}$		.4		V	$I_B=1\mu\text{A}, I_E=10\mu\text{A}$		
Collector-Base Capacitance	$C_{CB}$		.3		pF	$f=1\text{MHz}, V_{CB}=5\text{V}$		
Base-Emitter Capacitance	$C_{BE}$		.4		pF	$f=1\text{MHz}, V_{BE}=0\text{V}$		
Thermal Resistance	$\theta_{JC}$		200		$^\circ\text{C/W}$			

**NOTES:**

- 300 $\mu\text{s}$  pulse width, 1 kHz pulse rate.
- Derate Maximum Average Current linearly from 65 $^\circ\text{C}$  by 6mA/ $^\circ\text{C}$ .
- Without heat sinking from  $T_A = 65^\circ\text{C}$ , derate Maximum Average Power linearly by 12mW/ $^\circ\text{C}$ .
- Measured from a reflector coated with a 99% reflective white paint (Kodak 6080) positioned 4.27mm (0.166 in) from the reference plane.
- Peak-to-Peak response to black and white bar patterns.
- Center of maximum signal point image lies within a circle of diameter D relative to the center line of the package. A second emitter image (through the detector lens) is also visible. This image does not affect normal operation.
- This measurement is made with the lens cusp parallel to the black-white transition.
- Image size is defined as the distance for the 10%-90% response as the sensor moves over an abrupt black-white edge.
- (+) indicates an increase in the distance from the reflector to the reference plane.
- All voltages referenced to Pin 4
- CAUTION:** The thermal constraints of the acrylic lens will not permit the use of conventional wave soldering procedures. The typical preheat and post cleaning temperatures and dwell times can subject the lens to thermal stresses beyond the absolute maximum ratings and can cause it to defocus.

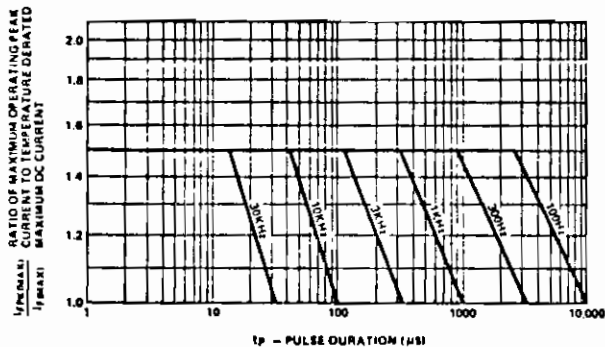


Figure 1. Maximum Tolerable Peak Current vs. Pulse Duration

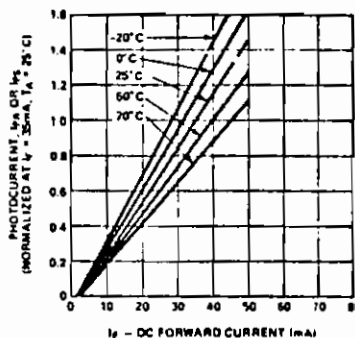


Figure 2. Relative Total Photocurrent vs. LED DC Forward Current

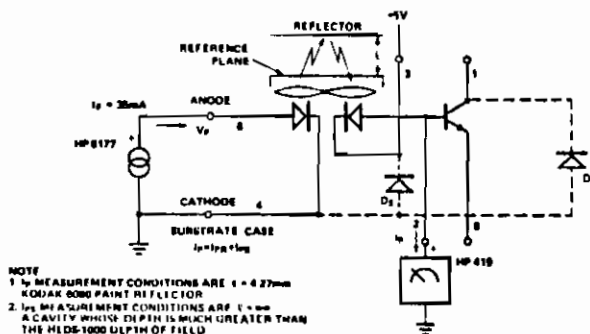


Figure 3.  $I_B$  Test Circuit

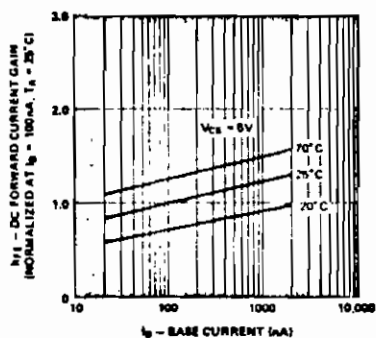


Figure 4. Normalized Transistor DC Forward Current Gain vs. Base Current at Temperature

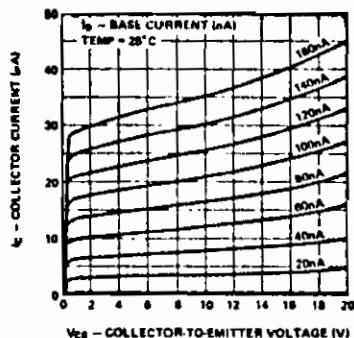


Figure 5. Common Emitter Collector Characteristics

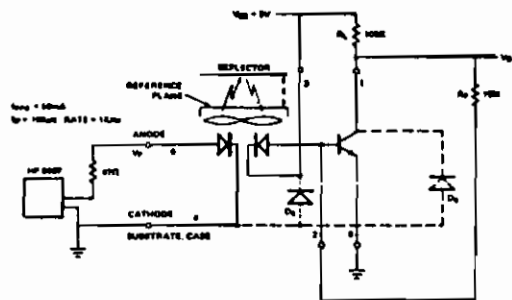


Figure 6. Slew Rate Measurement Circuit

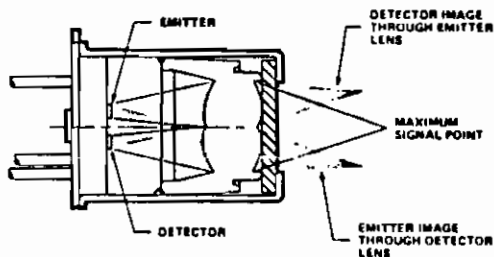


Figure 7. Image Location

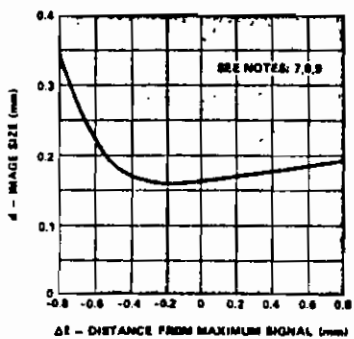


Figure 8. Image Size vs. Maximum Signal Point

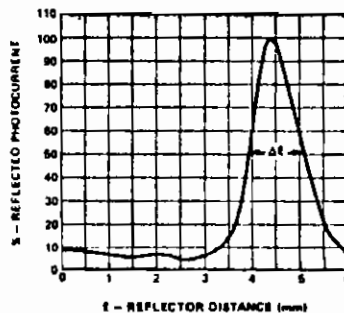


Figure 9. Reflector Distance vs. % Reflected Photocurrent

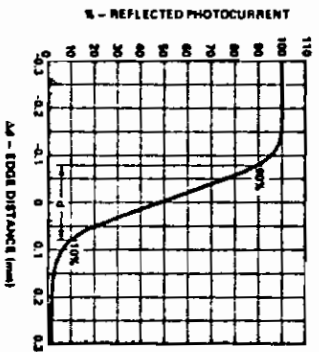


Figure 10. Step Edge Response

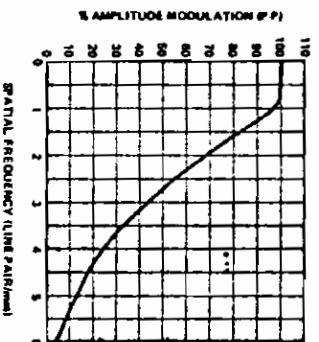


Figure 11. Modulation Transfer Function

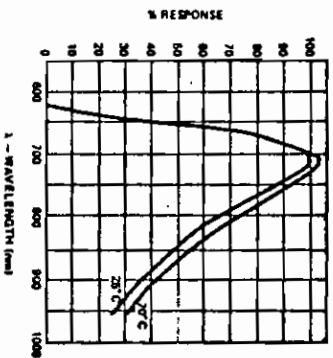


Figure 12. Detector Spectral Response

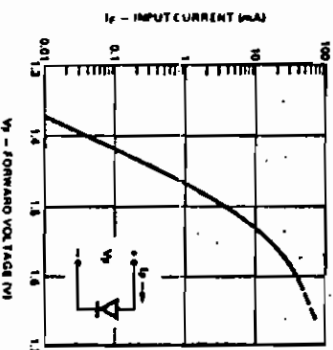


Figure 13. LED Forward Current vs. Forward Voltage Characteristics

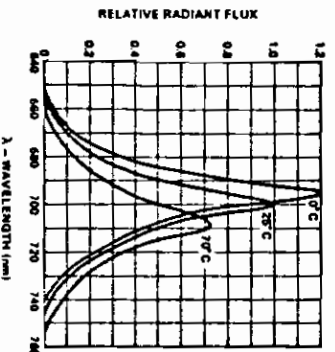


Figure 14. Relative Radiant Flux vs. Wavelength

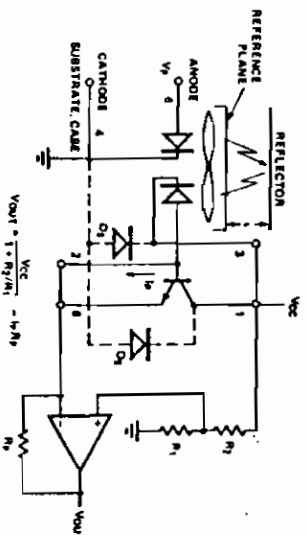


Figure 15. Photodiode Interconnection



**HEWLETT  
PACKARD**

## DIGITAL BAR CODE WAND

**HEDS-3000**

TECHNICAL DATA MARCH 1980

### Features

- **0.3 mm RESOLUTION**  
Enhances the Readability of dot matrix printed bar codes
- **DIGITAL OUTPUT**  
Open Collector Output Compatible with TTL and CMOS
- **PUSH-TO-READ SWITCH**  
Wand Consumes Power Only When Switch is Depressed
- **SINGLE SUPPLY OPERATION**
- **STYLIZED CASE**
- **DURABLE LOW FRICTION TIP**
- **SOLID STATE RELIABILITY**  
Uses LED and IC Technology



### Description

The HEDS-3000 Digital Bar Code Wand is a hand held scanner with integral push-to-read switch. It is designed to read all common bar code formats that have the narrowest bars printed with a nominal width of 0.3 mm (0.012 in). The wand contains an optical sensor with a 700 nm visible light source, photo IC detector, and precision aspheric optics. Internal signal conditioning circuitry converts the optical information into a logic level pulse width representation of the bars and spaces.

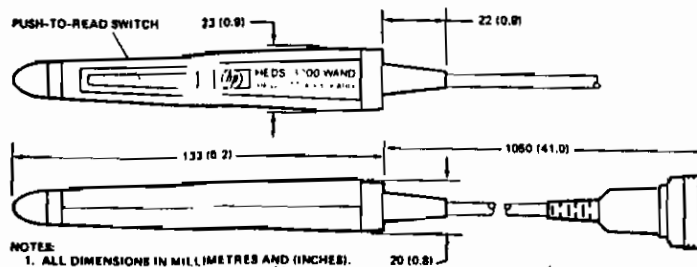
The HEDS-3000 comes equipped with a push-to-read switch which is used to activate the electronics, and strain relieved 104 cm (41 in.) cord with nine-pin subminiature D-style connector.

### Applications

The Digital Bar Code Wand is an effective alternative to the keyboard when used to collect information in self-contained blocks. Bar code scanning is faster than key entry and also more accurate since most codes have check-sums built-in to prevent incorrect reads from being entered.

Applications include remote data collection, ticket identification systems, security checkpoint verification, file folder tracking, inventory control, identifying assemblies in service, repair, and manufacturing environments, and programming appliances, intelligent instruments and personal computers.

### Wand Dimensions



## Recommended Operating Conditions

Parameter	Symbol	Min.	Max.	Units
Bar Width	a, b	0.3		mm
Scan Velocity	$v_{scan}$	7.6	76	cm/s
Contrast	PCS	70		%
Supply Voltage	$V_S$	3.6	5.75	V
Temperature	$T_A$	0	55	$^{\circ}\text{C}$
Orientation	See Figure 1			

## Electrical Operation

The HEDS-3000 consists of a precision optical sensor, an analog amplifier, a digitizing circuit, and an output transistor. These elements provide a TTL compatible output from a single voltage supply range of 3.6V to 5.75V. A non-reflecting black bar results in a logic high (1) level, while a reflecting white space will cause a logic low (0) at the  $V_O$  connection (pin 2). The output of the HEDS-3000 is an open collector transistor.

A push-to-read switch is used to energize the 700 nm LED emitter and electronic circuitry. When the switch is initially depressed, its contact bounce may cause a series of random pulses to appear at the output,  $V_O$ . This pulse train will typically settle to a final value within 0.5 ms.

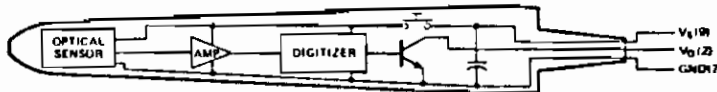
## Absolute Maximum Ratings

Parameter	Symbol	Min.	Max.	Units	Notes
Storage Temperature	$T_S$	-20	55	$^{\circ}\text{C}$	1
Operating Temperature	$T_A$	0	55	$^{\circ}\text{C}$	
Supply Voltage	$V_S$	-0.5	6.0	V	2
Output Transistor Power	$P_T$		200	mW	
Output Collector Voltage	$V_O$		20	V	

## Electrical Characteristics ( $V_S = 3.6\text{V}$ to $5.75\text{V}$ at $T_A = 25^{\circ}\text{C}$ , $R_L = 2.2\text{k}\Omega$ , unless otherwise noted)

Parameter	Symbol	Min.	Typ.	Max.	Units	Conditions	Fig.	Notes
Switch Bounce	$t_{sb}$		0.5	5	ms			3
High Level Output Current	$I_{OH}$			-400	$\mu\text{A}$	$V_{OH} = 2.4\text{V}$ , Bar Condition (Black)	3	
Low Level Output Current	$I_{OL}$			16	mA	$V_{OL} = 0.4\text{V}$ , Space Condition (White)	3	
Output Rise Time	$t_r$		2		$\mu\text{s}$	10%-90% Transition	3	
Output Fall Time	$t_f$		2		$\mu\text{s}$	90%-10% Transition	3	
Supply Current	$I_S$			50	mA	$V_S = 5\text{V}$ , Bar Condition (Black)		2.4

## Block Diagram



**GUARANTEED WIDTH ERROR PERFORMANCE**

( $V_S = 5V$ ,  $T_A = 0^\circ C$  to  $55^\circ C$ ,  $R_L = 2.2k\Omega$ ), unless otherwise noted:

Parameter	Symbol	Min.	Typ.	Max.	Units	Conditions	Fig.	Notes	
Bar Width Error	1st	$\Delta b_1$	0.08 (3.2)	0.11 (4.5)	mm (in. $\times 10^{-3}$ )	$T_A = 25^\circ C$	Margin $\geq 5$ mm Height = 0.25mm Tilt = $0^\circ$ $V_{scan} = 50$ cm/s Standard Test Tag Preferred Orientation $b = s = 0.3$ mm (0.012 in.) $2b = 2s = 0.6$ mm (0.024 in.)	1	5
			0.10 (3.8)	0.14 (5.5)		$T_A = 0^\circ$ to $55^\circ C$		2.6	7.8
	Interior	$\Delta b$	-0.01 (-0.2)	0.05 (1.8)	0.09 (3.5)	$T_A = 25^\circ C$		1.2	6.7
			-0.02 (-0.6)	0.05 (2.0)	0.10 (3.9)	$T_A = 0^\circ$ to $55^\circ C$		6.11	8.9 10.11
Space Width Error	Interior	$\Delta s$	0.0 (0.0)	-0.05 (-1.8)	-0.09 (-3.5)	$T_A = 25^\circ C$	1.2	6.7	
			0.0 (0.0)	-0.05 (-2.0)	-0.10 (-3.9)	$T_A = 0^\circ$ to $55^\circ C$	6.11	8.10 11	
Tag Scan Velocity	$V_{scan}$	76		76	cm/s		9	7	
Emitter Peak Wavelength	$\lambda$		700		nm	$T_A = 25^\circ C$			

**TYPICAL WIDTH ERROR PERFORMANCE** ( $V_S = 5V$ ,  $T_A = 25^\circ C$ ,  $R_L = 2.2k\Omega$ ), unless otherwise noted:

Parameter	Symbol		Typical WE Tilt = $0^\circ$ Height = 0.25mm		Typical WE Tilt = $30^\circ$ Height = 0.6mm		Units	Conditions	Fig.	Notes
	From	To								
Bar Width Error	Margin	1st	$\Delta b_1$	0.08 (3.2)	0.11 (4.2)		mm (in. $\times 10^{-3}$ )	Margin $\geq 5$ mm $1b = 1s = 0.3$ mm $2b = 2s = 0.6$ mm $T_A = 25^\circ C$ $V_S = 5V$ $V_{scan} = 50$ cm/s Preferred Orientation Standard Test Tag	1.2	5.7.8
	1s	1b	$\Delta b_{1-1}$	0.03 (1.2)	0.04 (1.6)		mm (in. $\times 10^{-3}$ )		1.2	6.7.8
	2s	1b	$\Delta b_{2-1}$	0.06 (2.5)	0.07 (2.9)		mm (in. $\times 10^{-3}$ )		1.2	6.7.8
	1s	2b	$\Delta b_{1-2}$	0.02 (0.9)	0.02 (0.7)		mm (in. $\times 10^{-3}$ )		1.2	6.7.8
	2s	2b	$\Delta b_{2-2}$	0.05 (1.9)	0.05 (2.1)		mm (in. $\times 10^{-3}$ )		1.2	6.7.8
Space Width Error	1b	1s	$\Delta s_{1-1}$	-0.04 (-1.4)	-0.04 (-1.4)		mm (in. $\times 10^{-3}$ )	1.2	6.7.8	
	2b	1s	$\Delta s_{2-1}$	-0.03 (-1.0)	-0.03 (-1.1)		mm (in. $\times 10^{-3}$ )	1.2	6.7.8	
	1b	2s	$\Delta s_{1-2}$	-0.07 (-2.7)	-0.08 (-3.3)		mm (in. $\times 10^{-3}$ )	1.2	6.7.8	
	2b	2s	$\Delta s_{2-2}$	-0.06 (-2.4)	-0.06 (-2.4)		mm (in. $\times 10^{-3}$ )	1.2	6.7.8	

**Notes:**

- Storage Temperature is dictated by Wand case.
- Power supply ripple and noise should be less than 100 mV
- Switch bounce causes a series of sub-millisecond pulses to appear at the output,  $V_o$
- Push-to-Read switch is depressed, and the Wand is placed on a non-reflecting (black) surface
- The margin refers to the reflecting (white) space that precedes the first bar of the bar code
- The interior bars and spaces are those which follow the first bar of bar code tag.
- The standard test tag consists of black bars, white spaces 0.3 mm (0.012 in.) min. photographed on Kodagraph Transtar TC5<sup>®</sup> paper with a print contrast signal greater than 0.9
- The print contrast signal (PCS) is defined as  $PCS = (R_w - R_b) / R_w$ , where  $R_w$  is the reflectance at 700 nm from the white spaces, and  $R_b$  is the reflectance at 700 nm for the bars
- 1.0 in. = 25.4 mm 1 mm = 0.0394 in.
- The Wand is in the preferred orientation when the surface of the switch button is parallel to the height dimension of the bar code



### OPERATION CONSIDERATIONS

The HEDS-3000 resolution is specified in terms of a bar and space Width Error, WE. The width error is defined as the difference between the calculated bar (space) width, B, (S), and the optically measured bar (space) widths, b (s). When a constant scan velocity is used, the width error can be calculated from the following:

$$B = t_b \cdot v_{scan}$$

$$S = t_s \cdot v_{scan}$$

$$\Delta b = B - b$$

$$\Delta s = S - s$$

Where

$\Delta b, \Delta s$  = bar, space Width Error (mm)

b, s = optical bar, space width (mm)

B, S = calculated bar, space width (mm)

$v_{scan}$  = scan velocity (mm/s)

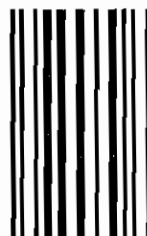
$t_b, t_s$  = wand pulse width outputs

The magnitude of the width error is dependent upon the width of the bar (space) preceding the space (bar) being measured. The Guaranteed Width Errors are specified as a maximum for the margin to first bar transition, as well as, maximums and minimums for the bar and space width errors resulting from transitions internal to the body of the bar code character. The Typical Width Error Performance specifies all possible transitions in a two level code (e.g. 2 of 5). For example, the  $\Delta b_{2-1}$  Width Error specifies the width error of a single bar module (0.3 mm) when preceded by a double space module (0.6 mm).

The Bar Width Error  $\Delta b$ , typically has a positive polarity which causes the calculated bar, B, to appear wider than its printed counterpart. The typical negative polarity of the Space Width Error  $\Delta s$ , causes the measured spaces to appear narrower. The consistency of the polarity of the bar and space Width Errors suggest decoding schemes which average the measured bars and measured spaces

within a character. These techniques will produce a higher percentage of good reads.

The Wand will respond to a bar code with a nominal module width of 0.3 mm when it is scanned at tilt angles between 0° and 30°. The optimum performance will be obtained when the Wand is held in the preferred orientation (Figure 1), tilted at an angle of 10° to 20°, and the Wand tip is in contact with the tag. The Wand height, when held normal to the tag, is measured from the tip's aperture, and when it is tilted it is measured from the tip's surface closest to the tag. The Width Error is specified for the preferred orientation, and using a Standard Test Tag consisting of black bars and white spaces. Figure 2 illustrates the random two level bar code tag. The Standard Test Tag is photographed on Kodagraph Transtar TC5® paper with a nominal module width of 0.3 mm (0.012 in.) and a Print Contrast Signal (PCS) of greater than 90%.



BAR WIDTH 0.3 mm (0.012 in.) BLACK & WHITE  
 PAPER - 79% PCS > 0.9 KODAGRAPH TRANSTAR TC5® PAPER

Figure 2. Standard Test Tag Format.

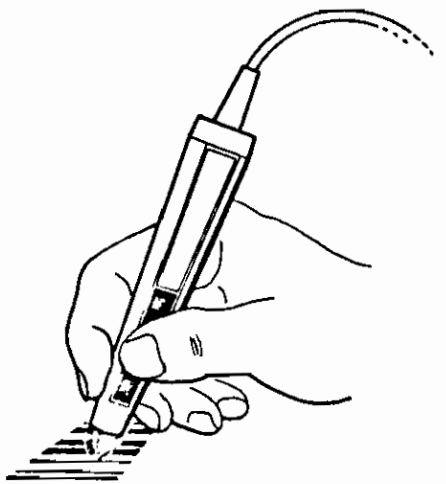


Figure 1. Preferred Wand Orientation.

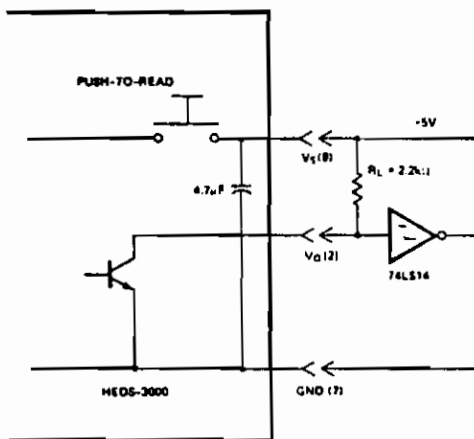


Figure 3. Recommended Logic Interface.

Typical Performance Curves ( $R_L = 2.2k\Omega$ )

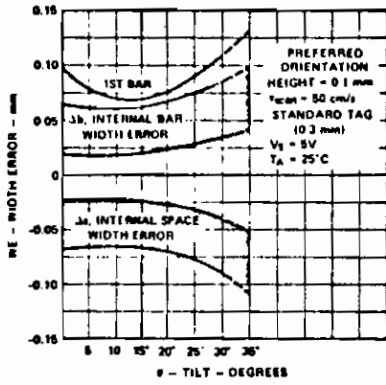


Figure 4. Width Error vs. Tilt (Preferred Orientation).

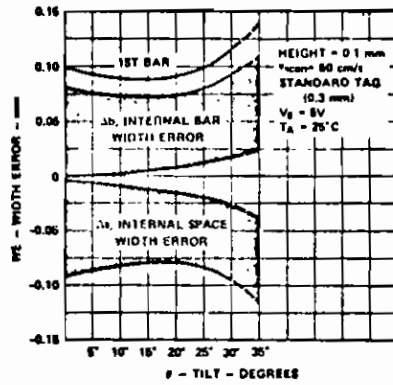


Figure 5. Width Error vs. Tilt (Any Orientation).

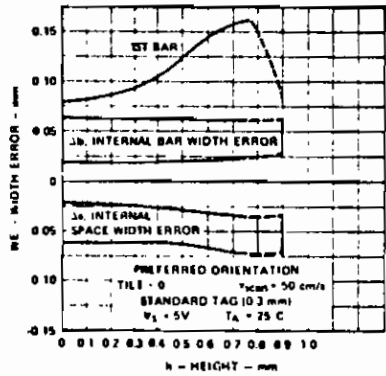


Figure 6. Width Error vs. Height (Preferred Orientation).

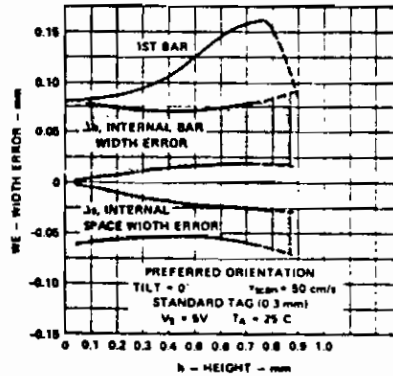


Figure 7. Width Error vs. Height (Any Orientation).

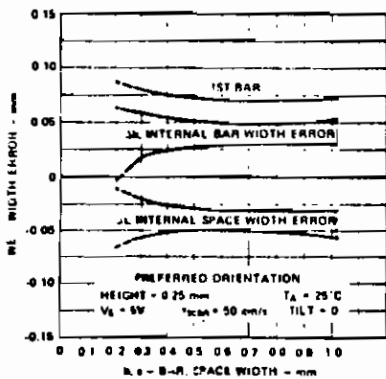


Figure 8. Width Error vs. Bar Width.

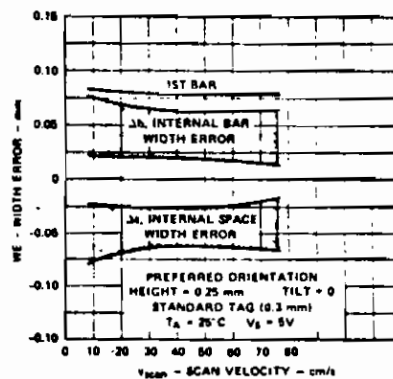


Figure 9. Width Error vs. Scan Velocity.

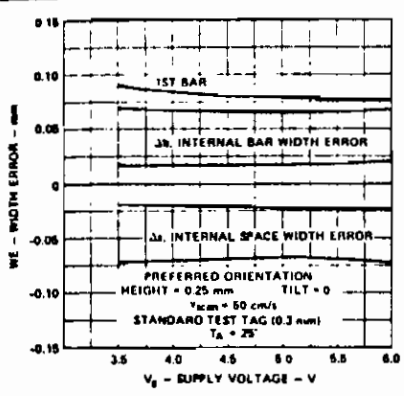


Figure 10. Width Error vs. Supply Voltage.

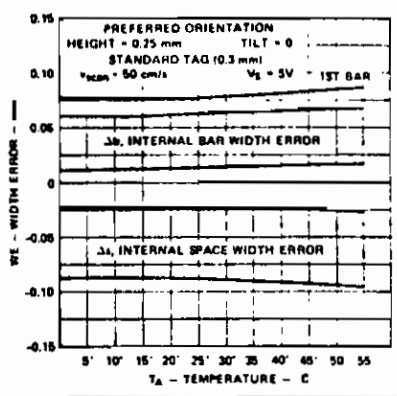


Figure 11. Width Error vs. Temperature.

**MECHANICAL CONSIDERATIONS**

The HEDS-3000 includes a standard nine pin D-style connector with integral squeeze-to-release retention mechanism. Two types of receptacles compatible with the retention mechanism are available from AMP Corp. (Printed circuit header: 745001-2 Panel mount: 745018, body, 66570-3, pins). Panel mount connectors that are compatible with the HEDS-3000 connector, but do not include the retention mechanism, are the Molex A7224, and AMP 2074-56-2.

**MAINTENANCE CONSIDERATIONS**

While there are no user serviceable parts inside the Wand, the tip should be checked periodically for wear and dirt, or obstructions in the aperture. The tip aperture is designed to reject particles and dirt but a gradual degradation in performance will occur as the tip wears down, or becomes obstructed by foreign materials.

Before unscrewing the tip, disconnect the Wand from the system power source. The aperture can be cleaned with a cotton swab or similar device and a liquid detergent.

The glass window on the sensor should be inspected and cleaned if dust, dirt, or fingerprints are visible. To clean the sensor window dampen a lint free cloth with a liquid cleaner, then clean the window with the cloth taking care not to disturb the orientation of the sensor. **DO NOT SPRAY CLEANER DIRECTLY ON THE SENSOR OR WAND.**

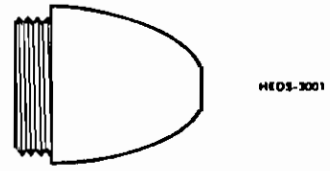


Figure 12. Wand Tip.

After cleaning the tip aperture and sensor window the tip should be gently and securely screwed back into the Wand assembly. The tip should be replaced if there are visible indications of wear such as a disfigured, or distorted aperture. The part number for the Wand tip is HEDS-3001. It can be ordered from any Hewlett-Packard parts center or franchised Hewlett-Packard distributor.

**OPTIONAL FEATURES**

The wand may also be ordered with the following special features:

- 193 color options
- Customer specified label
- No label
- Heavy duty retractable coiled cord
- No connector
- No switch button

For more information, call your local Hewlett-Packard sales office or franchised distributor

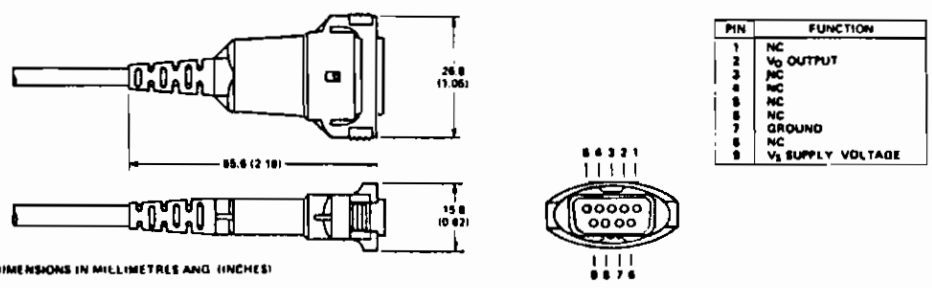


Figure 13. Connector Specifications.



## 2732 32K (4K x 8) UV ERASABLE PROM



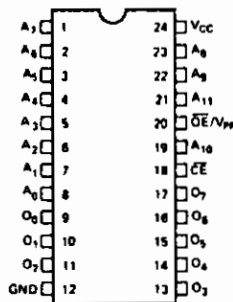
- Single +5V ± 5% Power Supply
- Output Enable for MCS-85™ and MCS-86™ Compatibility
- Fast Access Time: 450ns Max.
- Low Power Dissipation:  
150mA Max. Active Current  
30mA Max. Standby Current
- Pin Compatible to Intel® 2716 EPROM and 2332/2364 ROMs
- Completely Static
- Simple Programming Requirements  
— Single Location Programming  
— Programs with One 50ms Pulse
- Three-State Output for Direct Bus Interface

The Intel® 2732 is a 32,768-bit ultraviolet erasable and electrically programmable read-only memory (EPROM). The 2732 operates from a single 5-volt power supply, has a standby mode, and features an output enable control. For production, the pin compatible 2332 and 2364 ROMs are available. The total programming time for all bits is three and a half minutes. All these features make designing with the 2732 in microcomputer systems faster, easier, and more economical.

An important 2732 feature is the separate output control, Output Enable ( $\overline{OE}$ ), from the Chip Enable control ( $\overline{CE}$ ). The  $\overline{OE}$  control eliminates bus contention in multiple bus microprocessor systems. Intel's Application Note AP-30 describes the microprocessor system implementation of the  $\overline{OE}$  and  $\overline{CE}$  controls on Intel's 2716 and 2732 EPROMs. AP-30 is available from Intel's Literature Department.

The 2732 has a standby mode which reduces the power dissipation without increasing access time. The maximum active current is 150mA, while the maximum standby current is only 30mA, an 80% savings. The standby mode is achieved by applying a TTL-high signal to the  $\overline{CE}$  input.

### PIN CONFIGURATION



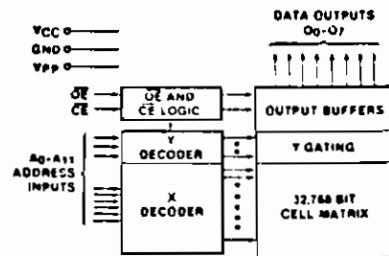
### PIN NAMES

A <sub>0</sub> -A <sub>11</sub>	ADDRESSES
$\overline{CE}$	CHIP ENABLE
$\overline{OE}$	OUTPUT ENABLE
O <sub>0</sub> -O <sub>7</sub>	OUTPUTS

### MODE SELECTION

MODE \ PINS	$\overline{CE}$ (18)	$\overline{OE/V_{pp}}$ (20)	V <sub>CC</sub> (24)	OUTPUTS (9, 11, 13-17)
Read	V <sub>IL</sub>	V <sub>IL</sub>	+5	D <sub>OUT</sub>
Standby	V <sub>IH</sub>	Don't Care	+5	High Z
Program	V <sub>IL</sub>	V <sub>pp</sub>	+5	D <sub>IN</sub>
Program Verify	V <sub>IL</sub>	V <sub>IL</sub>	+5	D <sub>OUT</sub>
Program Inhibit	V <sub>IH</sub>	V <sub>pp</sub>	+5	High Z

### BLOCK DIAGRAM



## PROGRAMMING

The programming specifications are described in the Data Catalog PROMIROM Programming Instructions Section.

### ABSOLUTE MAXIMUM RATINGS\*

Temperature Under Bias	-10°C to +80°C
Storage Temperature	-65°C to +125°C
All Input or Output Voltages with Respect to Ground	+6V to -0.3V

#### \*COMMENT

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stressing rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

## D.C. AND OPERATING CHARACTERISTICS

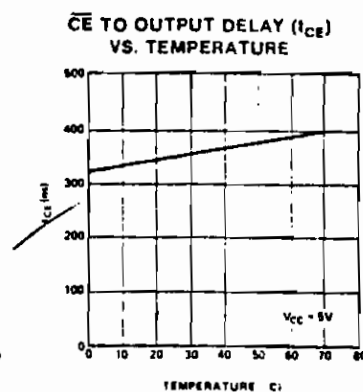
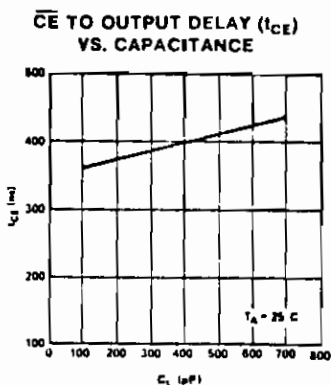
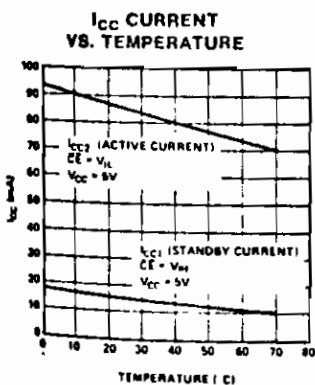
T<sub>A</sub> = 0°C to 70°C, V<sub>CC</sub> = +5V ± 5%

### READ OPERATION

Symbol	Parameter	Limits			Unit	Conditions
		Min.	Typ. <sup>(1)</sup>	Max.		
I <sub>LI1</sub>	Input Load Current (except $\overline{OE}/V_{PP}$ )			10	μA	V <sub>IN</sub> = 5.25V
I <sub>LI2</sub>	$\overline{OE}/V_{PP}$ Input Load Current			300	μA	V <sub>IN</sub> = 5.25V
I <sub>LO</sub>	Output Leakage Current			10	μA	V <sub>OUT</sub> = 5.25V
I <sub>CC1</sub>	V <sub>CC</sub> Current (Standby)		15	30	mA	$\overline{CE} = V_{IH}, \overline{OE} = V_{IL}$
I <sub>CC2</sub>	V <sub>CC</sub> Current (Active)		85	150	mA	$\overline{OE} = \overline{CE} = V_{IL}$
V <sub>IL</sub>	Input Low Voltage	-0.1		0.8	V	
V <sub>IH</sub>	Input High Voltage	2.0		V <sub>CC</sub> +1	V	
V <sub>OL</sub>	Output Low Voltage			0.45	V	I <sub>OL</sub> = 2.1mA
V <sub>OH</sub>	Output High Voltage	2.4			V	I <sub>OH</sub> = -400μA

Note: 1 Typical values are for T<sub>A</sub> = 25°C and nominal supply voltages.

## TYPICAL CHARACTERISTICS



**A.C. CHARACTERISTICS**
 $T_A = 0^\circ\text{C to } 70^\circ\text{C}, V_{CC} = +5\text{V} \pm 5\%$ 

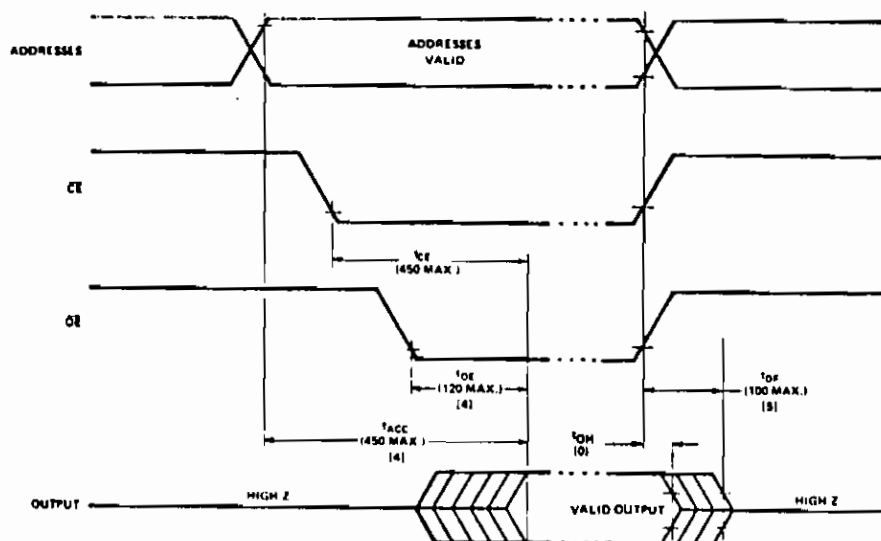
Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ. <sup>(1)</sup>	Max.		
$t_{ACC}$	Address to Output Delay			450	ns	$\overline{CE} = \overline{OE} = V_{IL}$
$t_{CE}$	$\overline{CE}$ to Output Delay			450	ns	$\overline{OE} = V_{IL}$
$t_{OE}$	Output Enable to Output Delay			120	ns	$\overline{CE} = V_{IL}$
$t_{DF}$	Output Enable High to Output Float	0		100	ns	$\overline{CE} = V_{IL}$
$t_{OH}$	Address to Output Hold	0			ns	$\overline{CE} = \overline{OE} = V_{IL}$

**CAPACITANCE** <sup>(2)</sup>  $T_A = 25^\circ\text{C}, f = 1\text{MHz}$ 

Symbol	Parameter	Typ.	Max.	Unit	Conditions
$C_{IN1}$	Input Capacitance Except $\overline{OE}/V_{PP}$	4	6	pF	$V_{IN} = 0\text{V}$
$C_{IN2}$	$\overline{OE}/V_{PP}$ Input Capacitance		20	pF	$V_{IN} = 0\text{V}$
$C_{OUT}$	Output Capacitance		12	pF	$V_{OUT} = 0\text{V}$

**A.C. TEST CONDITIONS**

Output Load: 1 TTL gate and  $C_L = 100\text{pF}$   
 Input Rise and Fall Times:  $\leq 20\text{ns}$   
 Input Pulse Levels: 0.8V to 2.2V  
 Timing Measurement Reference Level:  
 Inputs 1V and 2V  
 Outputs 0.8V and 2V

**A.C. WAVEFORMS** <sup>(3)</sup>**NOTES**

- 1 TYPICAL VALUES ARE FOR  $T_A = 25^\circ\text{C}$  AND NOMINAL SUPPLY VOLTAGES
- 2 THIS PARAMETER IS ONLY SAMPLED AND IS NOT 100% TESTED
- 3 ALL TIMES SHOWN IN PARENTHESES ARE MINIMUM TIMES AND ARE NSEC UNLESS OTHERWISE SPECIFIED
- 4  $t_{OH}$  MAY BE DELAYED UP TO 130ns AFTER THE FALLING EDGE OF  $\overline{CE}$  WITHOUT IMPACT ON  $t_{ACC}$
- 5  $t_{DF}$  IS SPECIFIED FROM  $\overline{OE}$  OR  $\overline{CE}$  WHICHEVER OCCURS FIRST

TTL  
MSI

**TYPES SN54LS373, SN54LS374, SN54S373, SN54S374,  
SN74LS373, SN74LS374, SN74S373, SN74S374  
OCTAL D-TYPE TRANSPARENT LATCHES AND  
EDGE-TRIGGERED FLIP-FLOPS**

BULLETIN NO. DL'S 12350 OCTOBER 1975 -- REVISED JUNE 1979

- Choice of 8 Latches or 8 D-Type Flip-Flops In a Single Package
- 3-State Bus-Driving Outputs
- Full Parallel-Access for Loading
- Buffered Control Inputs
- Clock/Enable Input Has Hysteresis to Improve Noise Rejection
- P-N-P Inputs Reduce D-C Loading on Data Lines ('S373 and 'S374)
- SN54LS363 and SN74LS364 Are Similar But Have Higher  $V_{OH}$  For MOS Interface

'LS373, 'S373  
FUNCTION TABLE

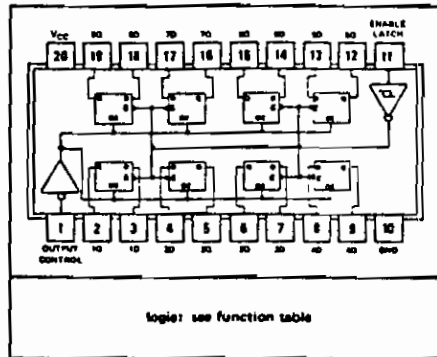
OUTPUT ENABLE	ENABLE LATCH	D	OUTPUT
L	H	H	H
L	H	L	L
L	L	X	$Q_0$
H	X	X	Z

'LS374, 'S374  
FUNCTION TABLE

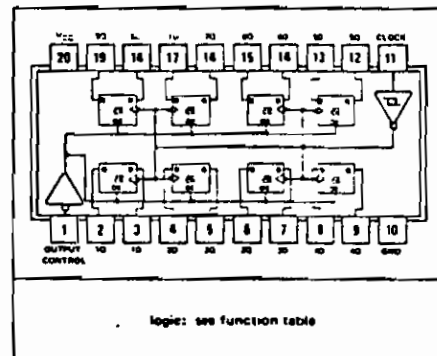
OUTPUT ENABLE	CLOCK	D	OUTPUT
L	↑	H	H
L	↑	L	L
L	L	X	$Q_0$
H	X	X	Z

See explanation of function tables on page 1-13.

SN54LS373, SN54S373 ... J PACKAGE  
SN74LS373, SN74S373 ... J OR N PACKAGE  
(TOP VIEW)



SN54LS374, SN54S374 ... J PACKAGE  
SN74LS374, SN74S374 ... J OR N PACKAGE  
(TOP VIEW)



**description**

These 8-bit registers feature totem-pole three-state outputs designed specifically for driving highly-capacitive or relatively low-impedance loads. The high-impedance third state and increased high-logic-level drive provide these registers with the capability of being connected directly to and driving the bus lines in a bus-organized system without need for interface or pull-up components. They are particularly attractive for implementing buffer registers, I/O ports, bidirectional bus drivers, and working registers.

The eight latches of the 'LS373 and 'S373 are transparent D-types latches meaning that while the enable (G) is high the Q outputs will follow the data (D) inputs. When the enable is taken low the output will be latched at the level of the data that was set up.

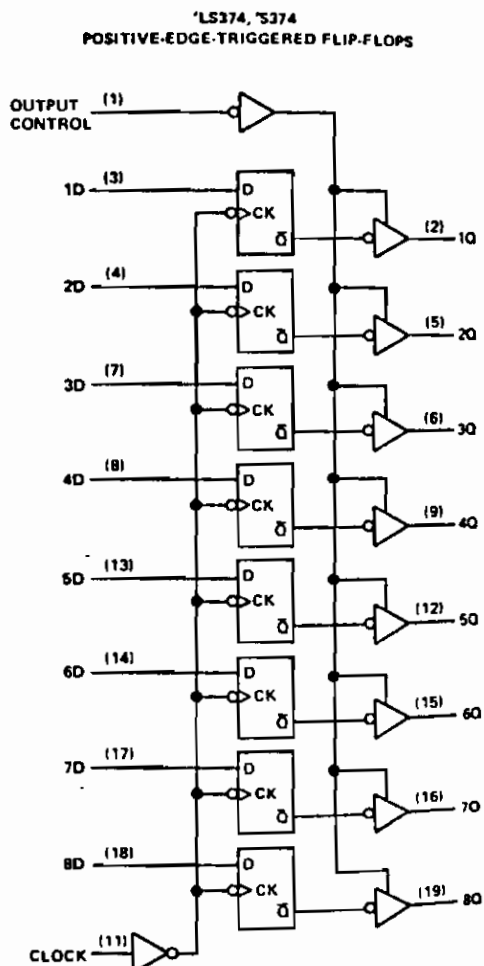
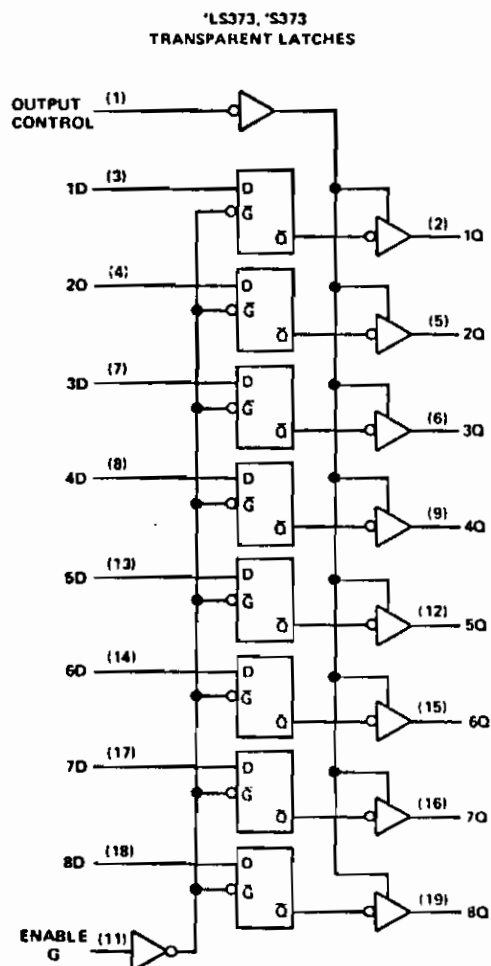
**TYPES SN54LS373, SN54LS374, SN54S373, SN54S374,  
SN74LS373, SN74LS374, SN74S373, SN74S374  
OCTAL D-TYPE TRANSPARENT LATCHES AND  
EDGE-TRIGGERED FLIP-FLOPS**

description (continued)

The eight flip-flops of the 'LS374 and 'S374 are edge-triggered D-type flip-flops. On the positive transition of the clock, the Q outputs will be set to the logic states that were setup at the D inputs.

Schmitt-trigger buffered inputs at the enable/clock lines simplify system design as ac and dc noise rejection is improved by typically 400 mV due to the input hysteresis. A buffered output control input can be used to place the eight outputs in either a normal logic state (high or low logic levels) or a high-impedance state. In the high-impedance state the outputs neither load nor drive the bus lines significantly.

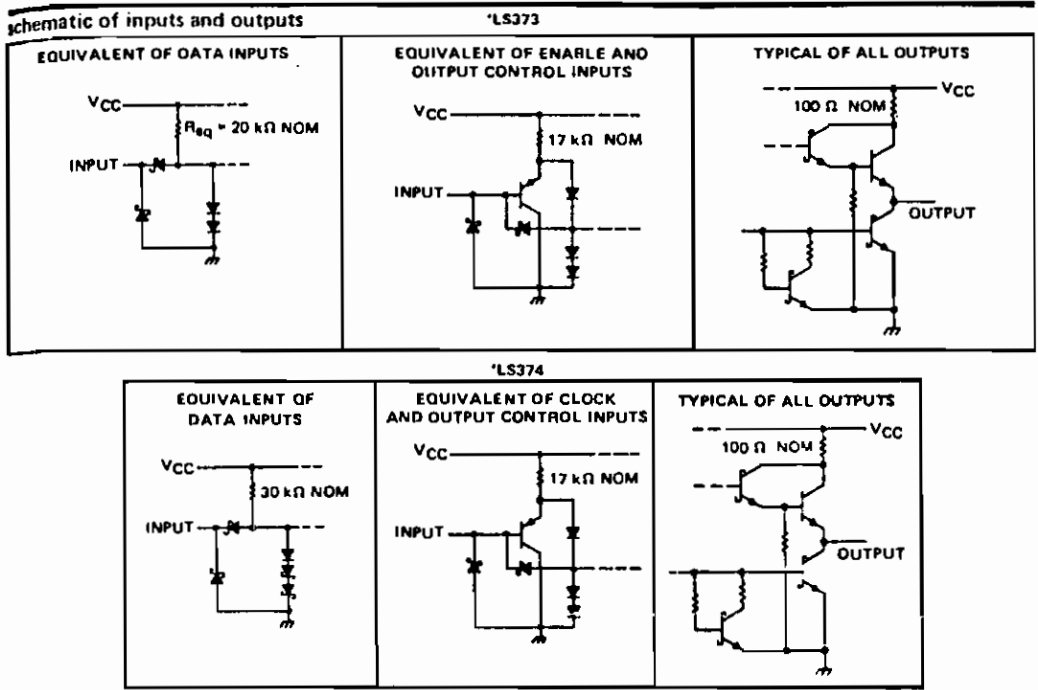
The output control does not affect the internal operation of the latches or flip-flops. That is, the old data can be retained or new data can be entered even while the outputs are off.





## TYPES SN54LS373, SN54LS374, SN74LS373, SN74LS374 OCTAL D-TYPE TRANSPARENT LATCHES AND EDGE-TRIGGERED FLIP-FLOPS

REVISED DECEMBER 1980



absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, $V_{CC}$ (see Note 1)	7 V
Input voltage	7 V
Off-state output voltage	7 V
Operating free-air temperature range: SN54LS*	-55°C to 125°C
SN74LS*	0°C to 70°C
Storage temperature range	-65°C to 150°C

NOTE 1: Voltage values are with respect to network ground terminal.

recommended operating conditions

	SN54LS*			SN74LS*			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, $V_{CC}$	4.5	5	5.5	4.75	5	5.25	V
High-level output voltage, $V_{OH}$			5.5			5.5	V
High-level output current, $I_{OH}$			-1			-2.6	mA
Width of clock/enable pulse, $t_w$	High	15		15			ns
	Low	15		15			
Data setup time, $t_{su}$	'LS373	5↓		5↓			ns
	'LS374	20↑		20↑			
Data hold time, $t_h$	'LS373	20↓		20↓			ns
	'LS374	0↑		0↑			
Operating free-air temperature, $T_A$		-55	125		0	70	°C

† The arrow indicates the transition of the clock/enable input used for reference: ↑ for the low-to-high transition, ↓ for the high-to-low transition.

## TYPES SN54LS373, SN54LS374, SN74LS373, SN74LS374 OCTAL D-TYPE TRANSPARENT LATCHES AND EDGE-TRIGGERED FLIP-FLOPS

REVISED AUGUST 1977

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER	TEST CONDITIONS <sup>†</sup>	SN54LS <sup>*</sup>		SN74LS <sup>*</sup>		UNIT			
		MIN	TYP <sup>‡</sup>	MAX	MIN		TYP <sup>‡</sup>	MAX	
V <sub>IH</sub>	High-level input voltage	2		2		V			
V <sub>IL</sub>	Low-level input voltage	0.7		0.8		V			
V <sub>IK</sub>	Input clamp voltage	V <sub>CC</sub> = MIN, I <sub>I</sub> = -18 mA		-1.5		V			
V <sub>OH</sub>	High-level output voltage	V <sub>CC</sub> = MIN, V <sub>IH</sub> = 2 V, V <sub>IL</sub> = V <sub>ILmax</sub> , I <sub>OH</sub> = MAX		2.4	3.4	2.4	3.1	V	
V <sub>OL</sub>	Low-level output voltage	V <sub>CC</sub> = MIN, V <sub>IH</sub> = 2 V, V <sub>IL</sub> = V <sub>ILmax</sub>		I <sub>OL</sub> = 12 mA	0.25	0.4	0.25	0.4	V
I <sub>OZH</sub>	Off-state output current, high-level voltage applied	V <sub>CC</sub> = MAX, V <sub>IH</sub> = 2 V, V <sub>O</sub> = 2.7 V		20		20		μA	
I <sub>OZL</sub>	Off-state output current, low-level voltage applied	V <sub>CC</sub> = MAX, V <sub>IH</sub> = 2 V, V <sub>O</sub> = 0.4 V		-20		-20		μA	
I <sub>I</sub>	Input current at maximum input voltage	V <sub>CC</sub> = MAX, V <sub>I</sub> = 7 V		0.1		0.1		mA	
I <sub>IH</sub>	High-level input current	V <sub>CC</sub> = MAX, V <sub>I</sub> = 2.7 V		20		20		μA	
I <sub>IL</sub>	Low-level input current	V <sub>CC</sub> = MAX, V <sub>I</sub> = 0.4 V		-0.4		-0.4		mA	
I <sub>OS</sub>	Short-circuit output current <sup>‡</sup>	V <sub>CC</sub> = MAX		-30	-130	-30	-130	mA	
I <sub>CC</sub>	Supply current	V <sub>CC</sub> = MAX, Output control at 4.5 V		'LS373 24 40		'LS374 24 40		mA	
				27 40		27 40			

<sup>†</sup> For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.<sup>‡</sup> All typical values are at V<sub>CC</sub> = 5 V, T<sub>A</sub> = 25°C.<sup>‡</sup> Not more than one output should be shorted at a time and duration of the short circuit should not exceed one second.switching characteristics, V<sub>CC</sub> = 5 V, T<sub>A</sub> = 25°C

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	'LS373			'LS374			UNIT	
				MIN	TYP	MAX	MIN	TYP	MAX		
f <sub>max</sub>							35	50		MHz	
t <sub>PLH</sub>	Data	Any Q	C <sub>L</sub> = 45 pF, R <sub>L</sub> = 687 Ω, See Notes 2 and 3	12	18					ns	
t <sub>PHL</sub>				12	18						
t <sub>PLH</sub>	Clock or enable	Any Q		20	30	15	28				ns
t <sub>PHL</sub>				18	30	19	28				
t <sub>PZH</sub>	Output Control	Any Q		15	28	20	28				ns
t <sub>PZL</sub>				25	36	21	28				
t <sub>PHZ</sub>	Output Control	Any Q	C <sub>L</sub> = 5 pF, R <sub>L</sub> = 667 Ω, See Note 3	12	20	12	20			ns	
t <sub>PLZ</sub>			15	25	14	25					ns

NOTES: 2. Maximum clock frequency is tested with all outputs loaded.

3. See load circuits and waveforms on page 3-11.

f<sub>max</sub> = maximum clock frequencyt<sub>PLH</sub> = propagation delay time, low-to-high-level outputt<sub>PHL</sub> = propagation delay time, high-to-low-level outputt<sub>PZH</sub> = output enable time to high levelt<sub>PZL</sub> = output enable time to low levelt<sub>PHZ</sub> = output disable time from high levelt<sub>PLZ</sub> = output disable time from low level



## TYPES SN54S373, SN54S374, SN74S373, SN74S374 OCTAL D-TYPE TRANSPARENT LATCHES AND EDGE-TRIGGERED FLIP-FLOPS

Electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER		TEST CONDITIONS <sup>†</sup>		MIN	TYP <sup>‡</sup>	MAX	UNIT
V <sub>IH</sub>	High-level input voltage			2			V
V <sub>IL</sub>	Low-level input voltage					0.8	V
V <sub>IK</sub>	Input clamp voltage		V <sub>CC</sub> = MIN, I <sub>I</sub> = -18 mA			-1.2	V
V <sub>OH</sub>	High-level output voltage	SN54S*	V <sub>CC</sub> = MIN, V <sub>IH</sub> = 2 V, V <sub>IL</sub> = 0.8 V, I <sub>OH</sub> = MAX	2.4	3.4		V
		SN74S*		2.4	3.1		V
V <sub>OL</sub>	Low-level output voltage		V <sub>CC</sub> = MIN, V <sub>IH</sub> = 2 V, V <sub>IL</sub> = 0.8 V, I <sub>OL</sub> = 20 mA			0.5	V
I <sub>OZH</sub>	Off-state output current, high-level voltage applied		V <sub>CC</sub> = MAX, V <sub>IH</sub> = 2 V, V <sub>O</sub> = 2.4 V			60	μA
I <sub>OZL</sub>	Off-state output current, low-level voltage applied		V <sub>CC</sub> = MAX, V <sub>IH</sub> = 2 V, V <sub>O</sub> = 0.5 V			-50	μA
I <sub>I</sub>	Input current at maximum input voltage		V <sub>CC</sub> = MAX, V <sub>I</sub> = 5.5 V			1	mA
I <sub>IH</sub>	High-level input current		V <sub>CC</sub> = MAX, V <sub>I</sub> = 2.7 V			50	μA
I <sub>IL</sub>	Low-level input current		V <sub>CC</sub> = MAX, V <sub>I</sub> = 0.5 V			-250	μA
I <sub>OS</sub>	Short-circuit output current <sup>§</sup>		V <sub>CC</sub> = MAX	-40		-100	mA
I <sub>CC</sub>	Supply current		V <sub>CC</sub> = MAX		105	160	mA
					90	140	mA

<sup>†</sup> For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

<sup>‡</sup> All typical values are at V<sub>CC</sub> = 5 V, T<sub>A</sub> = 25°C.

<sup>§</sup> Not more than one output should be shorted at a time and duration of the short circuit should not exceed one second.

Switching characteristics, V<sub>CC</sub> = 5 V, T<sub>A</sub> = 25°C

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	S373			S374			UNIT
				MIN	TYP	MAX	MIN	TYP	MAX	
f <sub>max</sub>							75	100		MHz
t <sub>PLH</sub>	Data	Any Q	C <sub>L</sub> = 15 pF, R <sub>L</sub> = 280 Ω, See Notes 2 and 4		7	12				ns
t <sub>PHL</sub>					7	12				
t <sub>PLH</sub>	Clock or enable	Any Q			7	14	8	15		ns
t <sub>PHL</sub>					12	18	11	17		ns
t <sub>PZH</sub>	Output	Any Q			8	15	8	15		ns
t <sub>PZL</sub>	Control				11	18	11	18		ns
t <sub>PHZ</sub>	Output	Any Q	C <sub>L</sub> = 5 pF, R <sub>L</sub> = 280 Ω, See Note 3		6	9	5	9		ns
t <sub>PLZ</sub>				Control		8	12	7	12	

NOTES: 2. Maximum clock frequency is tested with all outputs loaded.

4. See load circuits and waveforms on page 3-10.

f<sub>max</sub>: maximum clock frequency

t<sub>PLH</sub>: propagation delay time, low to high-level output

t<sub>PHL</sub>: propagation delay time, high to low-level output

t<sub>PZH</sub>: output enable time to high level

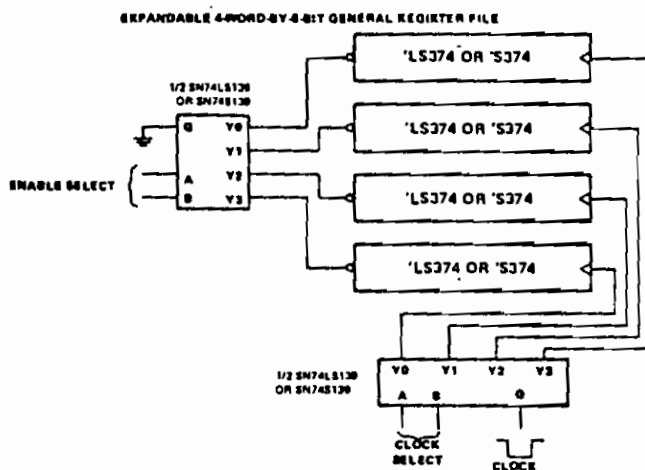
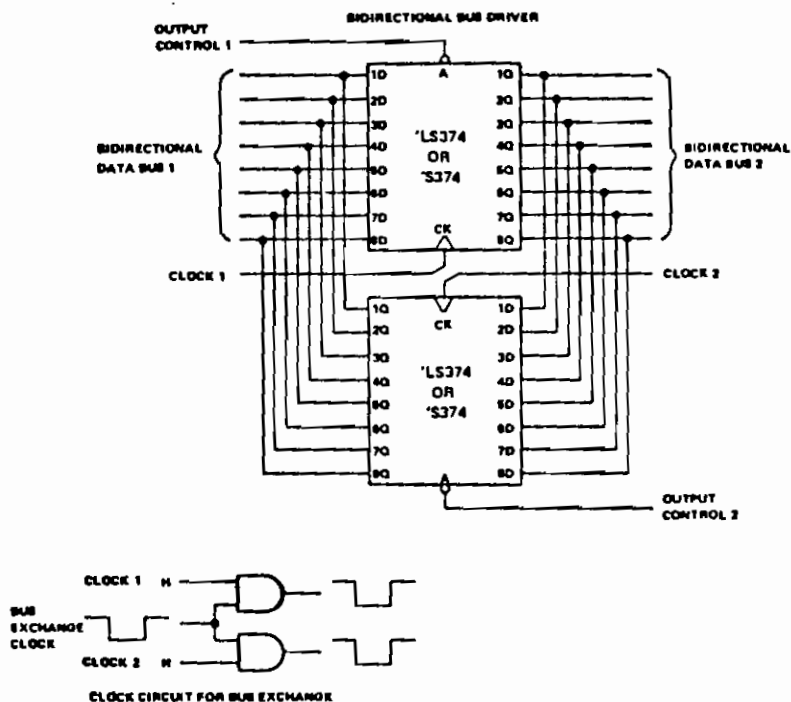
t<sub>PZL</sub>: output enable time to low level

t<sub>PHZ</sub>: output disable time from high level

t<sub>PLZ</sub>: output disable time from low level

## TYPES SN54LS374, SN54S374, SN74LS374, SN74S374 OCTAL D-TYPE TRANSPARENT LATCHES AND EDGE-TRIGGERED FLIP-FLOPS

### TYPICAL APPLICATION DATA





supply current<sup>†</sup>

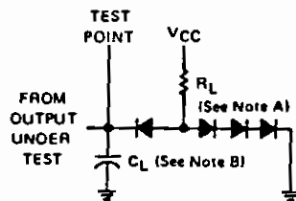
TYPE	I <sub>CC</sub> H (mA) Total with outputs high		I <sub>CCL</sub> (mA) Total with outputs low		I <sub>CC</sub> (mA) Average per gate (50% duty cycle)
	TYP	MAX	TYP	MAX	TYP
'13	14	23	20	32	8.5
'14	22	36	39	60	5.1
'132	15	24	26	40	5.1
'LS13	2.9	6	4.1	7	1.75
'LS14	8.6	16	12	21	1.72
'LS132	5.9	11	8.2	14	1.76
'S132	28	44	44	68	9

<sup>†</sup>Maximum values of I<sub>CC</sub> are over the recommended operating ranges of V<sub>CC</sub> and T<sub>A</sub>. Typical values are at V<sub>CC</sub> = 5 V, T<sub>A</sub> = 25°C.

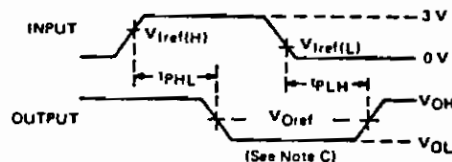
switching characteristics, V<sub>CC</sub> = 5 V, T<sub>A</sub> = 25°C

TYPE	TEST CONDITIONS	t <sub>PLH</sub> (ns) Propagation delay time, low-to-high-level output			t <sub>PLH</sub> (ns) Propagation delay time, high-to-low-level output		
		MIN	TYP	MAX	MIN	TYP	MAX
		'13		18		27	15
'14, '132	C <sub>L</sub> = 15 pF, R <sub>L</sub> = 400 Ω	15		22	15	22	
'LS13		15		22	18	27	
'LS14	C <sub>L</sub> = 15 pF, R <sub>L</sub> = 2 kΩ	15		22	15	22	
'LS132		15		22	15	22	
'S132	C <sub>L</sub> = 15 pF, R <sub>L</sub> = 280 Ω	7	10.5		8.5	13	

PARAMETER MEASUREMENT INFORMATION



LOAD CIRCUIT



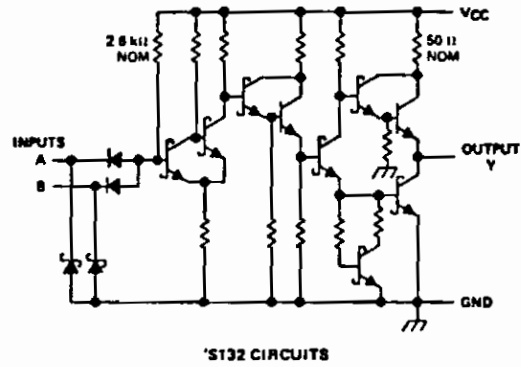
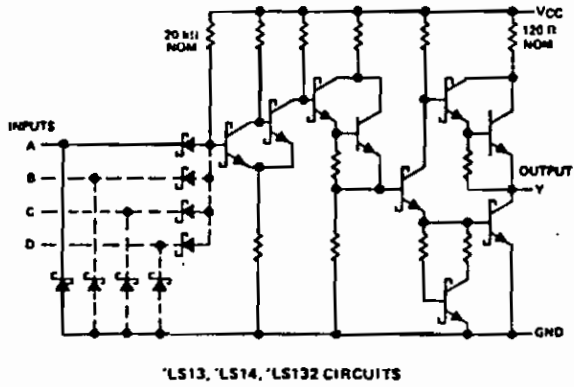
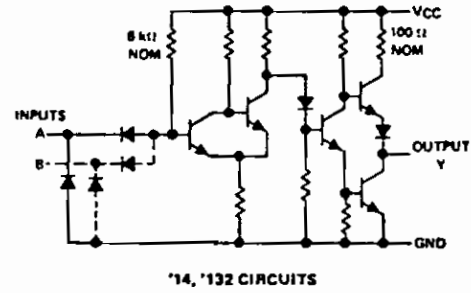
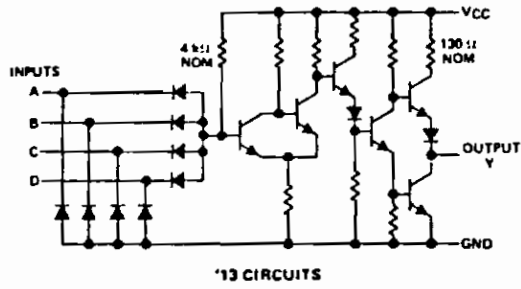
VOLTAGE WAVEFORMS

- NOTES. A. All diodes are 1N916 or 1N3064.  
 B. C<sub>L</sub> includes probe and jig capacitance.  
 C. Generator characteristics and reference voltages are:

	Generator Characteristics				Reference Voltages		
	Z <sub>out</sub>	PRR	t <sub>r</sub>	t <sub>f</sub>	V <sub>Iref(H)</sub>	V <sub>Iref(L)</sub>	V <sub>Oref</sub>
SN54/SN74 <sup>†</sup>	50 Ω	1 MHz	10 ns	10 ns	1.7 V	0.9 V	1.5 V
SN54LS/SN74LS <sup>†</sup>	50 Ω	1 MHz	15 ns	6 ns	1.6 V	0.8 V	1.3 V
'S132	50 Ω	1 MHz	2.5 ns	2.5 ns	1.8 V	1.2 V	1.5 V

SCHMITT-TRIGGER POSITIVE-NAND GATES AND INVERTERS WITH TOTEM-POLE OUTPUTS

schematics (each gate)



Resistor values shown are nominal.

**SCHMITT-TRIGGER POSITIVE-NAND GATES AND INVERTERS WITH TOTEM-POLE OUTPUTS**



## SCHMITT-TRIGGER POSITIVE-NAND GATES AND INVERTERS WITH TOTEM-POLE OUTPUTS

### TYPICAL CHARACTERISTICS OF '13, '14, AND '132 CIRCUITS†

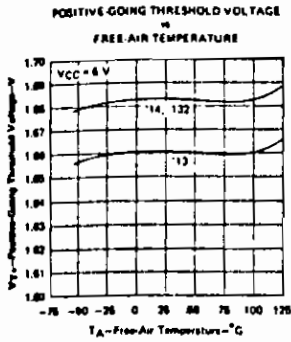


FIGURE 1

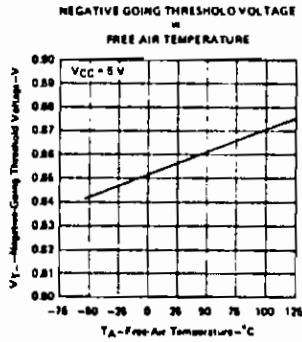


FIGURE 2

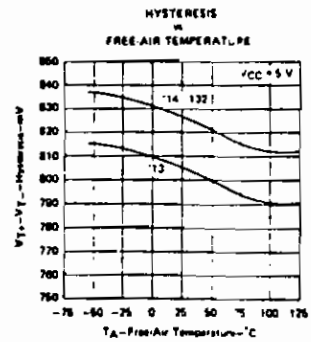


FIGURE 3

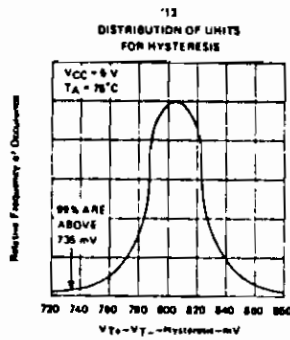


FIGURE 4

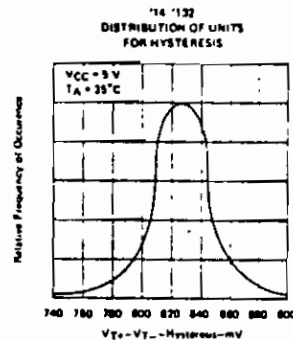


FIGURE 5

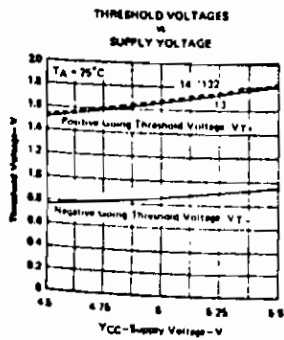


FIGURE 6

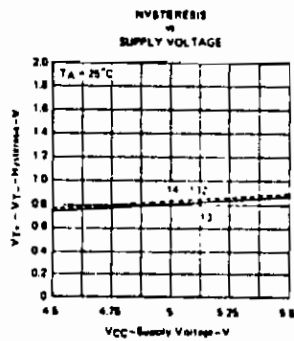


FIGURE 7

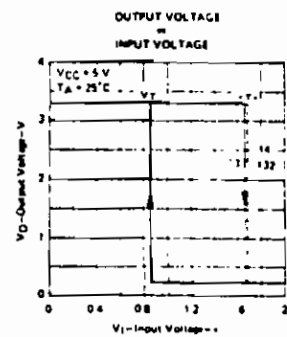
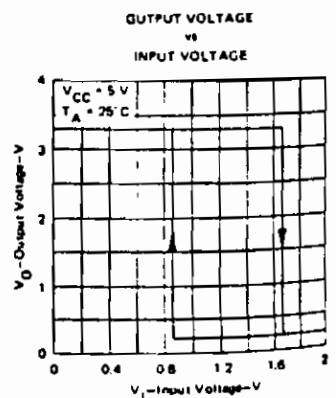
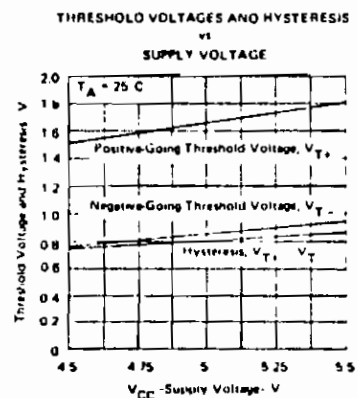
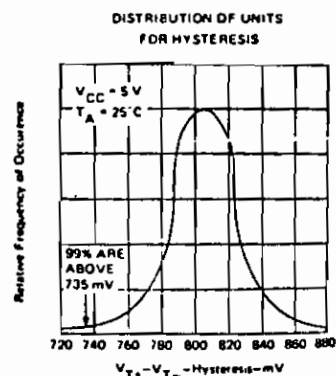
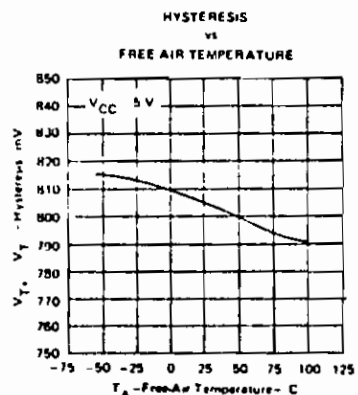
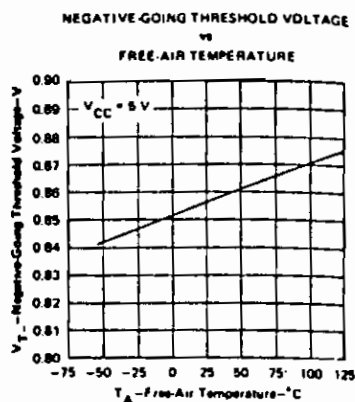
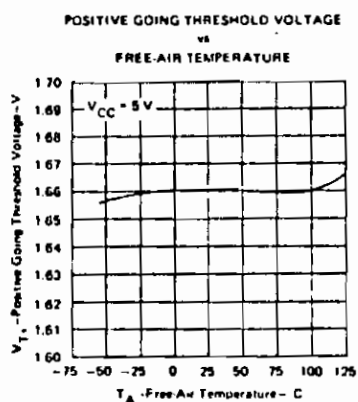


FIGURE 8

†Data for temperatures below 0°C and 70°C and supply voltages below 4.75V and above 5.25 V are applicable for SN5413, SN5414, and SN54172 only.

## SCHMITT-TRIGGER POSITIVE-NAND GATES AND INVERTERS WITH TOTEM-POLE OUTPUTS

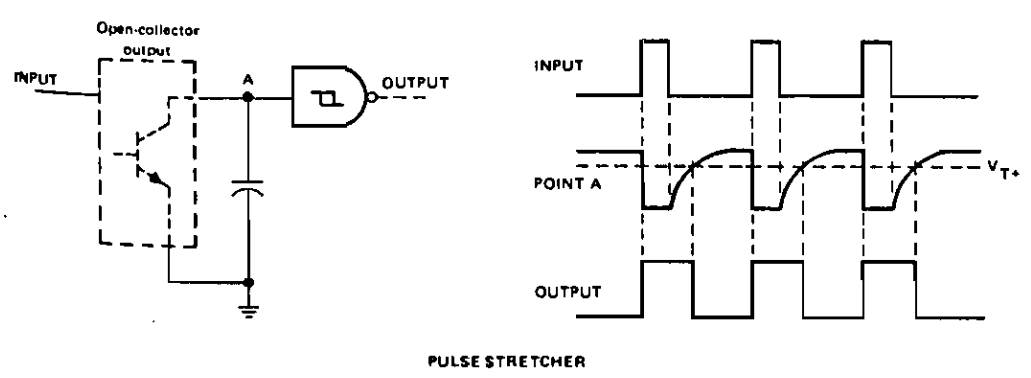
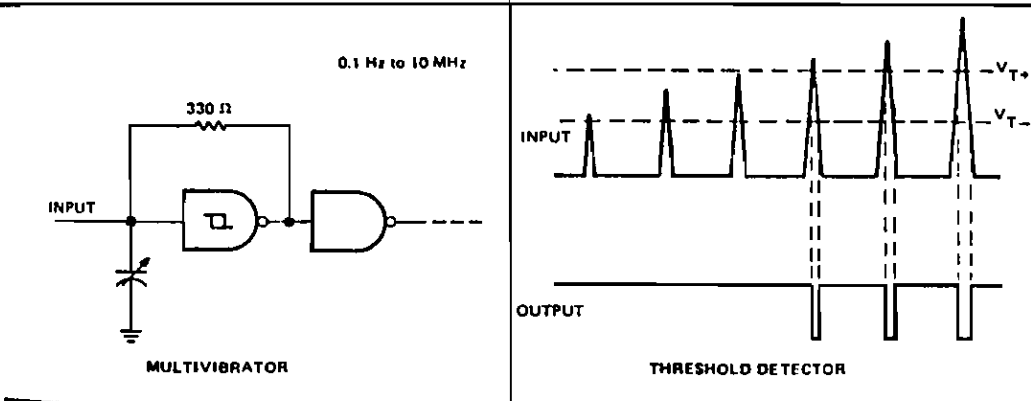
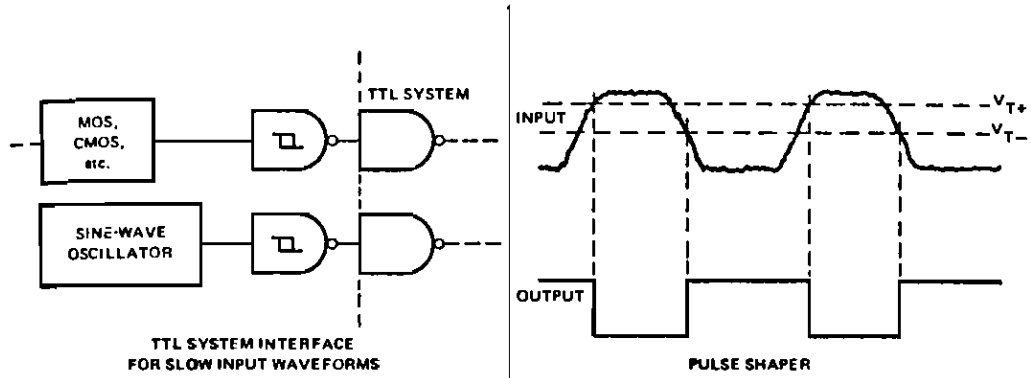
### TYPICAL CHARACTERISTICS OF 'LS13, 'LS14, AND 'LS132 CIRCUITS†



† Data for temperatures below 0°C and above 70°C and supply voltages below 4.75 V and above 5.25 are applicable for SN54LS13, SN54LS14, and SN74LS13 only.

## SCHMITT-TRIGGER POSITIVE-NAND GATES AND INVERTERS WITH TOTEM-POLE OUTPUTS

### TYPICAL APPLICATION DATA



# INTRODUCTORY

SPECIFICATIONS BASED ON EVALUATION OF LIMITED NUMBER OF DEVICES

# MAXIM

## +5V Powered RS-232 Drivers/Receivers

### General Description

The MAX230 family of line drivers/receivers is intended for all RS-232 and V28/V24 communications interfaces, and in particular, for those applications where  $\pm 12V$  is not available. The MAX230 and MAX236 are particularly useful in battery powered systems since their low power shutdown mode reduces power dissipation to less than 5 $\mu$ W. The MAX233 and MAX235 use no external components and are recommended for applications where printed circuit board space is critical.

All of the MAX230 family except the MAX231 and MAX239 need only a single +5V supply for operation. The MAX230 family RS-232 driver/receivers have on-board charge pump voltage converters which convert the +5V input power to the  $\pm 10V$  needed to generate the RS-232 output levels. The MAX231 and MAX230, designed to operate from +5V and +12V, contain a +12V to -12V charge pump voltage converter.

Since nearly all RS-232 applications need both line drivers and receivers, the MAX230 family includes both receivers and drivers in one package. Since the wide variety of RS-232 applications require differing numbers of transmitters and receivers, Maxim offers a wide selection of RS-232 driver/receiver combinations in order to minimize the package count (see table below).

Both the receivers and the line drivers (transmitters) meet all EIA RS-232C and CCITT V28 specifications.

### Features

- ◆ Operates from Single 5V Power Supply (+5V and +12V — MAX231 and MAX239)
- ◆ Meets All RS-232C and V28 Specifications
- ◆ Multiple Drivers and Receivers
- ◆ On-board DC-DC Converters
- ◆  $\pm 9V$  Output Swing with +5V Supply
- ◆ Low Power Shutdown —  $< 1\mu A$
- ◆ 3-State TTL/CMOS Receiver Outputs
- ◆  $\pm 30V$  Receiver Input Levels

### Applications

Computers  
Peripherals  
Modems  
Printers  
Instruments

MAX230-239\*

### Selection Table

Part Number	Power Supply Voltage	No. of RS-232 Drivers	No. of RS-232 Receivers	External Components	Low Power Shutdown and TTL 3-State	No. of Pins
MAX230	+5V	5	0	4 capacitors	Yes	20
MAX231	+5V and +7.5V to 13.2V	2	2	2 capacitors	No	14
MAX232	+5V	2	2	4 capacitors	No	18
MAX233	+5V	2	2	None	No	20
MAX234	+5V	4	0	4 capacitors	No	16
MAX235	+5V	5	5	None	Yes	24
MAX236	+5V	4	3	4 capacitors	Yes	24
MAX237	+5V	5	3	4 capacitors	No	24
MAX238	+5V	4	4	4 capacitors	No	24
MAX239	+5V and +7.5V to 13.2V	3	5	2 capacitors	Yes	24

\* Patent Pending

MAXIM

MAXIM is a registered trademark of Maxim Integrated Products

Maxim Integrated Products 1

LECTOR DE CODIGO DE BARRAS

## +5V Powered RS-232 Drivers/Receivers

### ABSOLUTE MAXIMUM RATINGS

$V_{CC}$ .....	+6V	Short Circuit Duration	
$V_I$ .....	( $V_{CC} - 0.3V$ ) to +15V	$T_{OUT}$ .....	continuous
$V^-$ .....	-15V	Power Dissipation	
Input Voltages		CEM-DIP .....	675mW
$T_{IN}$ .....	-0.3 to ( $V_{CC} + 0.3V$ )	(operate @ 5mW/°C above +70°C)	
$R_{IN}$ .....	$\pm 30V$	Plastic DIP .....	375mW
Output Voltages		(operate 7mW/°C above +70°C)	
$T_{OUT}$ .....	( $V^+ + 0.3V$ ) to ( $V^- - 0.3V$ )	Small Outline (SO) .....	375mW
$R_{OUT}$ .....	-0.3V to ( $V_{CC} + 0.3V$ )	(operate 7mW/°C above +70°C)	

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only and functional operation of the device at these or any other conditions above those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

### ELECTRICAL CHARACTERISTICS

( $V_{CC} = 5V \pm 10\%$ ,  $V^+ = 7.5V$  to 13.2V (MAX231 and MAX239 Only),  $T_A =$  Operating Temperature Range, Figures 3-12, unless otherwise noted.)

PARAMETER	CONDITIONS	MIN.	TYT	MAX.	UNITS
Output Voltage Swing	All Transmitter Outputs loaded with 3k $\Omega$ to Ground	$\pm 5$	$\pm 9$		V
$V_{CC}$ Power Supply Current	No load, $T_A = +25^\circ C$		6	10	mA
Power Supply Current	No load, $V^+ = 12V$ MAX231 and MAX239 only		5	10	mA
Power Supply	Figure 1, $T_A = +25^\circ C$		1	10	$\mu A$
Input Logic Threshold Low	$T_{IN}$			0.8	V
Input Logic Threshold High	$T_{IN}$	2.0			V
Logic Pullup Current	$T_{IN} = 0V$		15	200	$\mu A$
RS-232 Input Voltage Oversting Range		-30		+30	V
RS-232 Input Threshold Low	$V_{CC} = 5V$ , $T_A = +25^\circ C$	0.8	1.2		V
RS-232 Input Threshold High	$V_{CC} = 5V$ , $T_A = +25^\circ C$		1.7	2.4	V
RS-232 Input Hysteresis	$V_{CC} = 5V$	0.2	0.5	1.0	V
RS-232 Input Resistance	$T_A = +25^\circ C$	3	5	7	k $\Omega$
TTL/CMOS Output Voltage Low	$I_{OUT} = -3.2mA$			0.4	V
TTL/CMOS Output Voltage High	$I_{OUT} = +1.0mA$	3.5			V
TTL/CMOS Output Leakage Current	$EN = V_{CC}$ , $0V \leq R_{OUT} \leq V_{CC}$		0.01	1.1	$\mu A$
Output Enable Time	MAX235, MAX238, MAX239, Figure 2		400		ns
Output Disable Time	MAX235, MAX238, MAX239, Figure 2		260		ns
Propagation Delay	RS-232 to TTL		0.5		ns
Instantaneous Slew Rate	$C_L = 10pF$ , $R_L = 3-7k\Omega$ , $T_A = +25^\circ C$ (Note 1)			30	V/ $\mu s$
Transition Region Slew Rate	$R_L = 3k\Omega$ , $C_L = 250pF$ Measured from +3V to -3V or -3V to +3V		3		V/ $\mu s$
Output Resistance	$V_{CC} = V^+ = V^- = 0V$ , $V_{OUT} = \pm 2V$	300			$\Omega$
RS-232 Output Short Circuit Current			10		mA

Note 1: Sample tested

MAXIM

# +5V Powered RS-232 Drivers/Receivers

## Typical Operating Characteristics

**MAX230-239**

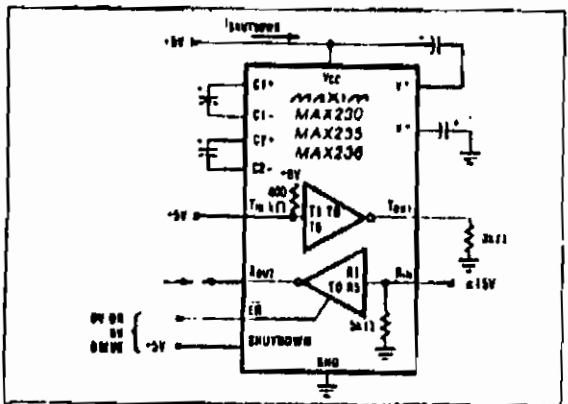
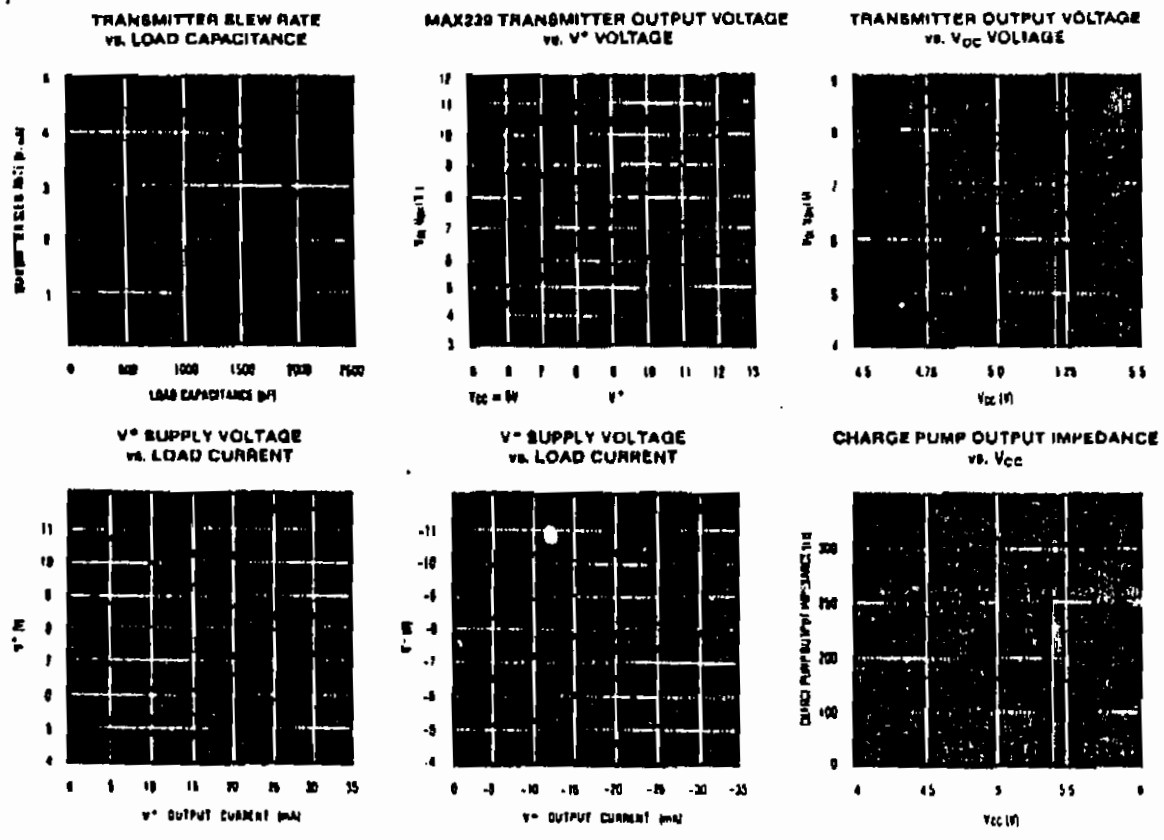


Figure 1. Shutdown Current Test Circuit

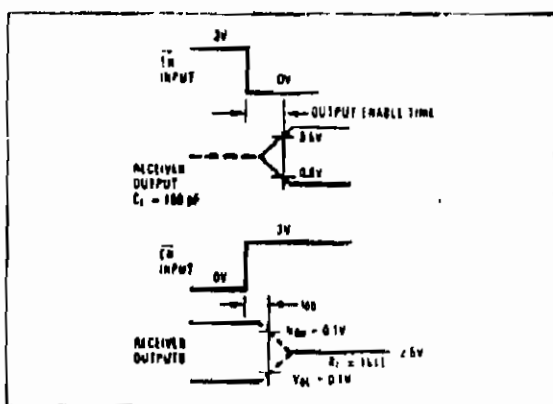


Figure 2. Receiver Output Enable and Disable Timing

**MAXIM**

## +5V Powered RS-232 Drivers/Receivers

### MAX230-239

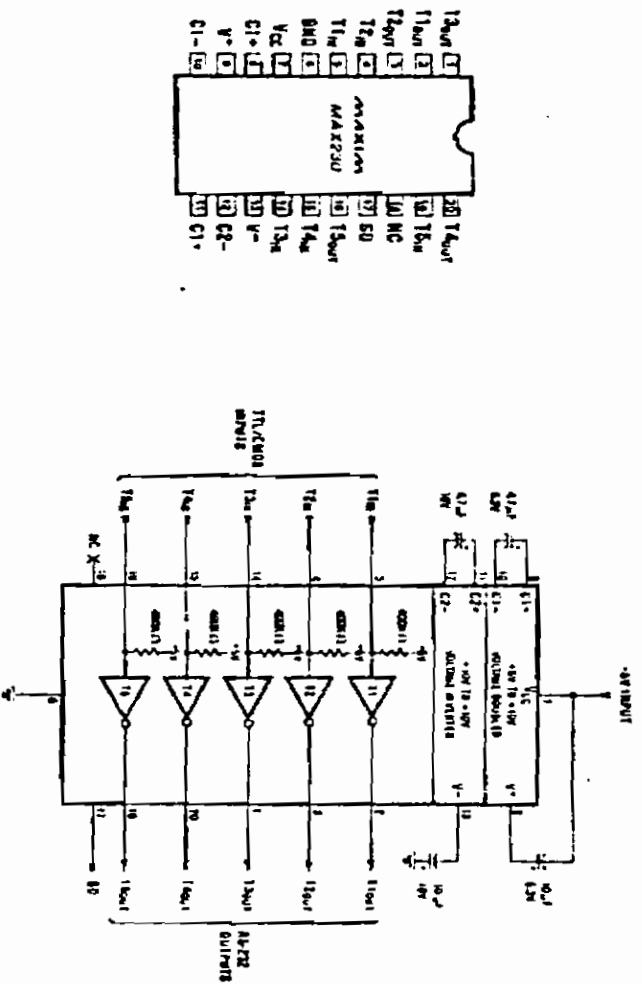


Figure 3 MAX230 Typical Operating Circuit

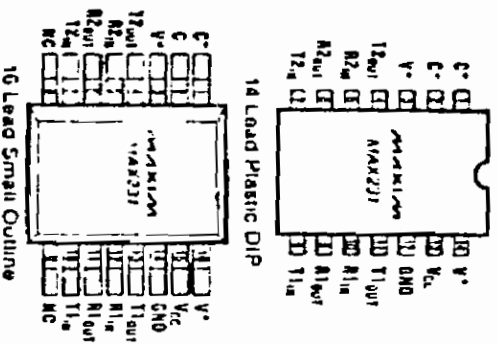
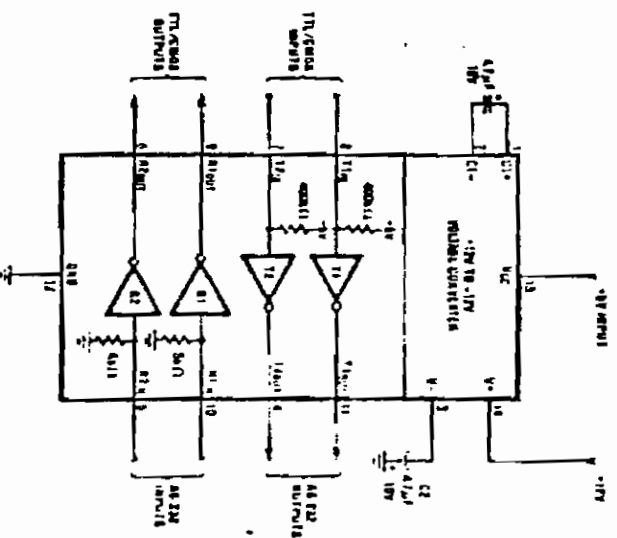


Figure 4 MAX232 Typical Operating Circuit



MAXIM





## +5V Powered RS-232 Drivers/Receivers

### MAX230-239

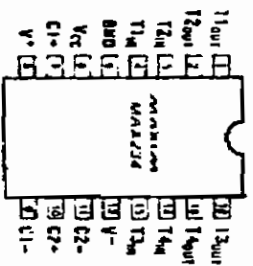
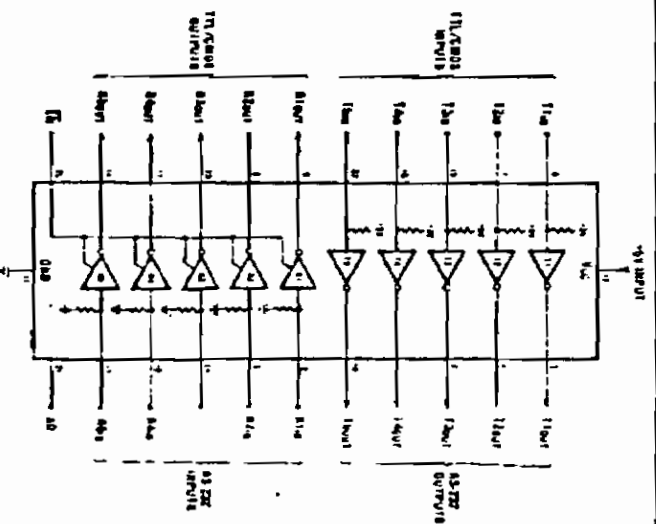
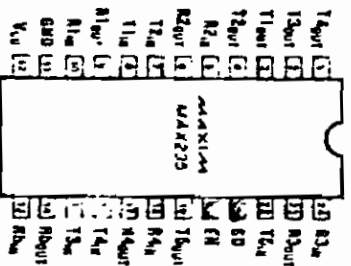
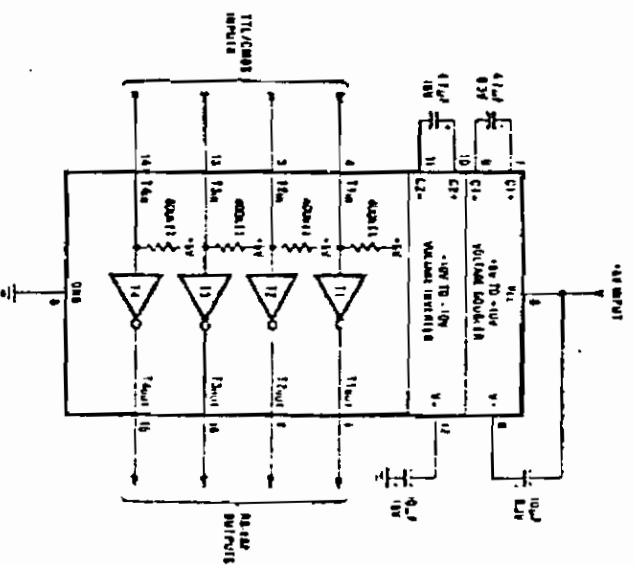


Figure 7 MAX230 Typical Operating Circuit



MAXIM Integrated Circuits

SECTION OF LOGIC OF THE MAXIM

MAXIM

## +5V Powered RS-232 Drivers/Receivers

**MAX230-239**

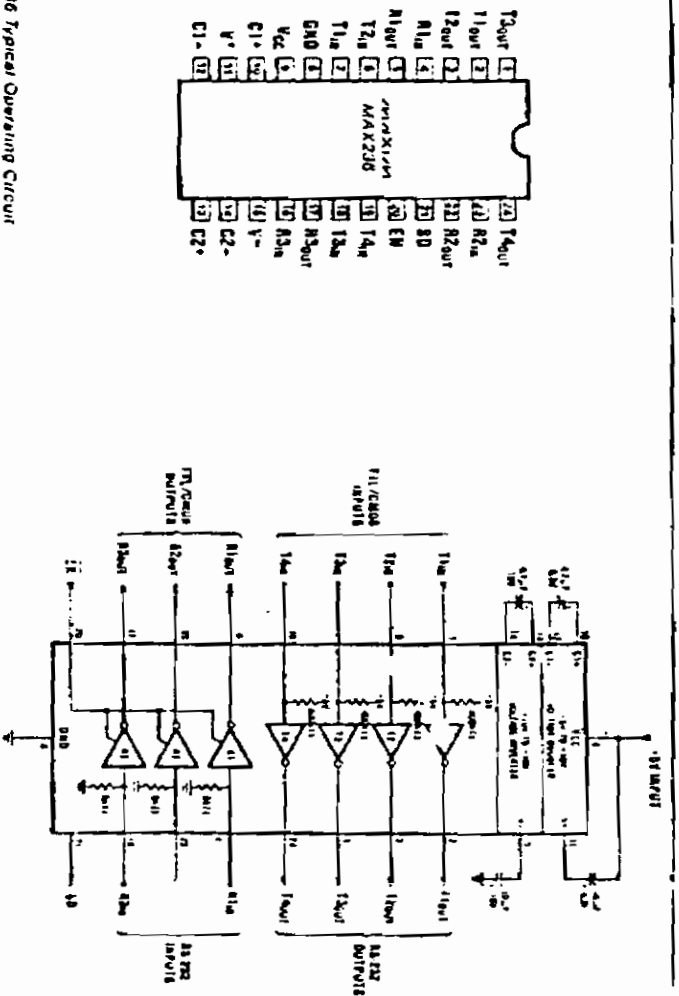


Figure 8 MAX230 Typical Operating Circuit

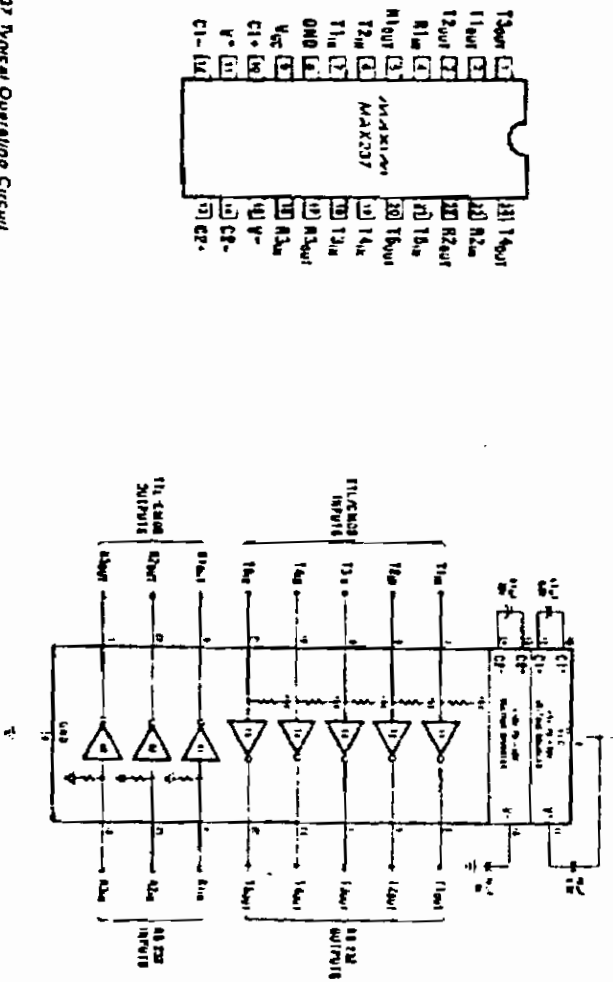


Figure 10 MAX239 Typical Operating Circuit

**MAXIM**

L E F T O N D E C O R P O R A T I O N D E B A R R I N G

## +5V Powered RS-232 Drivers/Receivers

MAX230-239

Figure 11 MAX238 Typical Operating Circuit

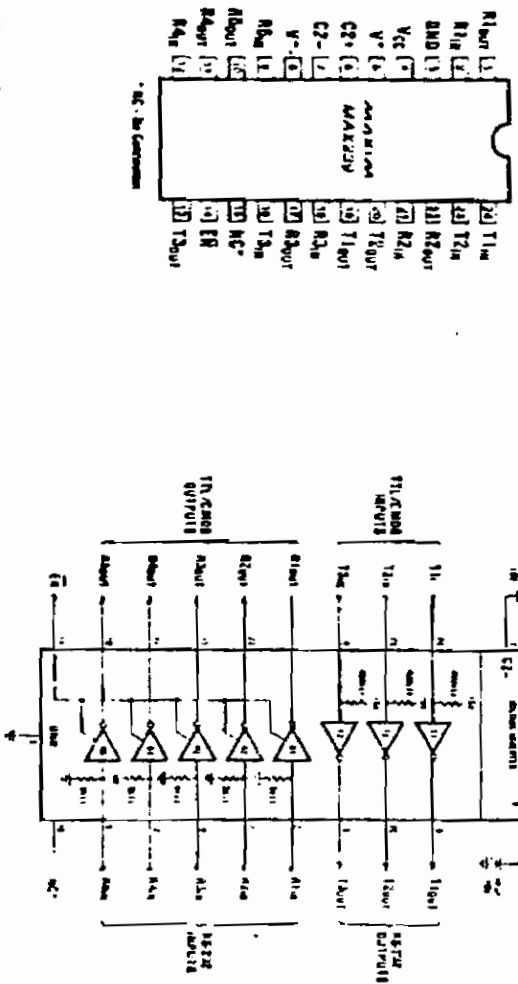
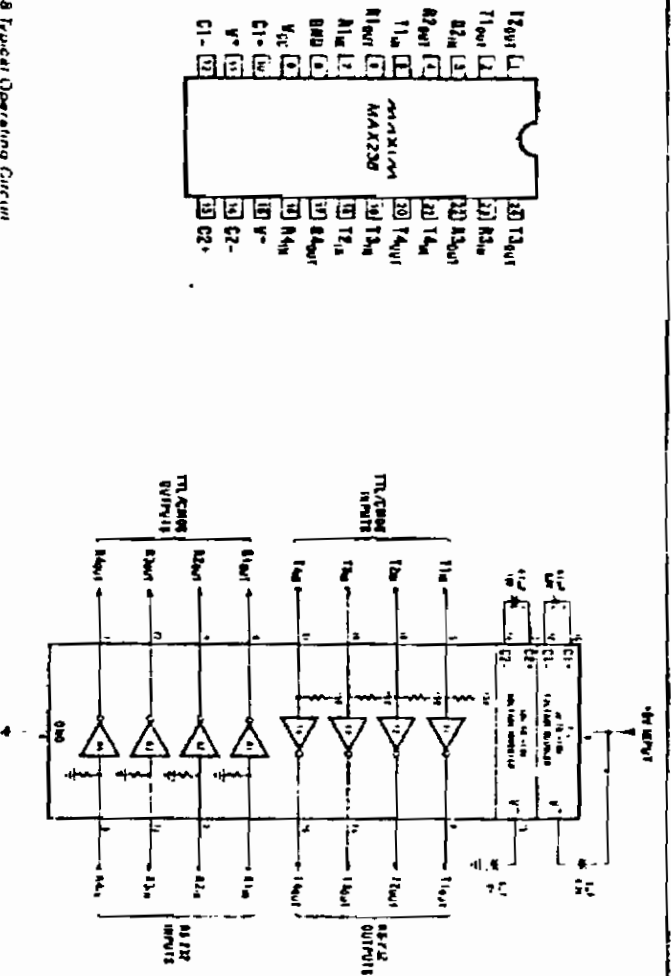


Figure 12 MAX239 Typical Operating Circuit

## +5V Powered RS-232 Drivers/Receivers

### +5V to ±10V Dual Charge Pump Voltage Converter

All but the MAX231 and MAX239 convert +5V to ±10V. This conversion is performed by two charge pump voltage converters. The first uses capacitor C1 to double the +5V to +10V, storing the +10V on the V<sup>+</sup> output filter capacitor, C3. The second charge pump voltage converter uses capacitor C2 to invert the +10V to -10V, storing the -10V on the V<sup>-</sup> output filter capacitor, C4. The equivalent circuit of the charge pump section is shown in Figure 14.

A small amount of power may be drawn from the +10V (V<sup>+</sup>) and -10V (V<sup>-</sup>) outputs to power external circuitry. The typical characteristic graphs show the typical output voltage vs. load current characteristics.

### Typical Applications

Figures 3 through 5 show typical applications. The capacitor values are non-critical. Reducing the capacitors C1 and C2 to 1μF will slightly increase the impedance of the charge pump, lowering the RS-232 driver output voltages by about 100mV. Lower values of C3 and C4 increase the ripple on the V<sup>+</sup> and V<sup>-</sup> outputs.

If the power supply input to the MAX23X device has a very fast rate-of-rise (as would occur if a PCB were to be plugged into a card cage with power already on), use the simple RC filter shown in Figure 13. This bypass network is not needed if the V<sub>CC</sub> rate-of-rise is below 1V/μs.

The MAX230 family devices are provided with an Enable pin. When the Enable pin is pulled up to V<sub>CC</sub>, the MAX230 outputs are in a low impedance state. When the Enable pin is pulled down to ground, the MAX230 outputs are in a high impedance state when it is a high impedance state.

The Shutdown pin provides control of the MAX230 MAX239. When the Shutdown pin is pulled up to V<sub>CC</sub>, the MAX230 MAX239 outputs are in a low impedance state. When the Shutdown pin is pulled down to ground, the MAX230 MAX239 outputs are in a high impedance state.

### Detailed Description

The following sections provide supplementary information for those designers with non-standard applications and for those with interest in the internal operation of the MAX230 family.

The MAX230 family devices consist of 3 sections, the transmitters, the receivers, and the charge pump DC DC voltage converter.

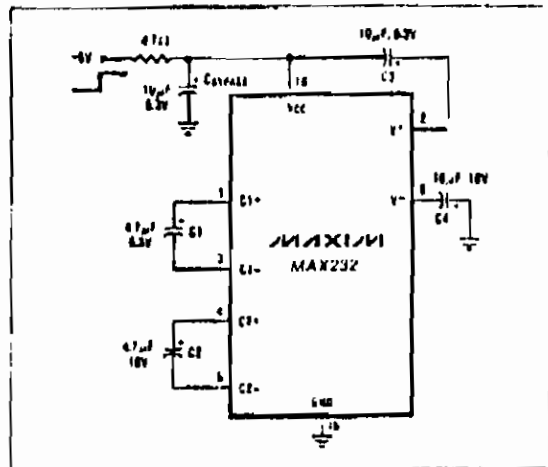


Figure 13. Protection from High  $\frac{dV}{dt}$

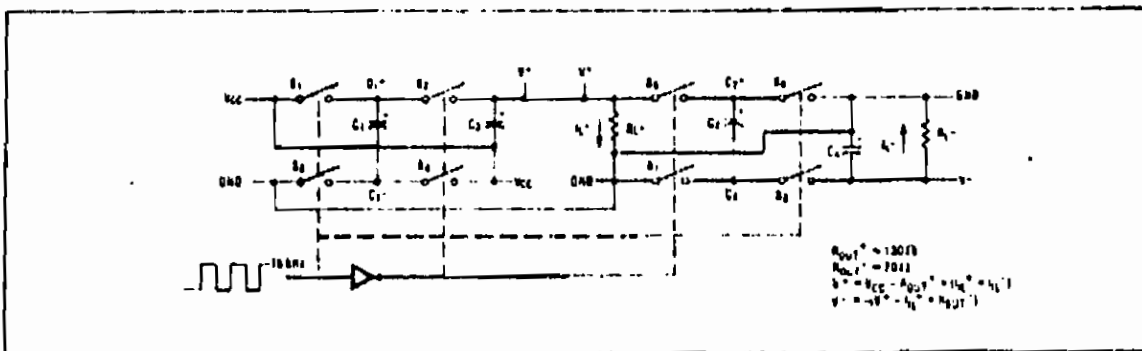


Figure 14. Charge Pump Diagram

## +5V Powered RS-232 Drivers/Receivers

For applications needing only the +5V to  $\pm 10V$  charge pump voltage converter, the MAX680 is available.

The capacitor values for C1 through C4 are non-critical. At the 15kHz typical switching frequency of the voltage converter, a 1 $\mu$ F capacitor has approximately 10 $\Omega$  impedance, and replacing the 4.7 $\mu$ F and 10 $\mu$ F capacitors shown in the typical applications with 1 $\mu$ F for C1 and C2 will increase the output impedance of the V<sup>+</sup> output by about 10 $\Omega$  and the output impedance of V<sup>-</sup> by about 20 $\Omega$ . Lowering the value of C3 and C4 increases the ripple on the V<sup>+</sup> and V<sup>-</sup> outputs. Where operation to the upper temperature limit is not required, or V<sub>CC</sub> will not go below 4.75V, C1 and C2 can be 1 $\mu$ F, and C3 and C4 can be 1 $\mu$ F per output channel (1 $\mu$ F if one transmitter is used, 5 $\mu$ F if five transmitters are used).

There are parasitic diodes which become forward biased if V<sup>+</sup> goes below V<sub>CC</sub> or V<sup>-</sup> goes above ground. When in the shutdown mode (MAX230, MAX235 and MAX236 only), V<sup>+</sup> is internally connected to V<sub>CC</sub> by a 1k $\Omega$  pulldown, and V<sup>-</sup> is internally connected to ground via a 1k $\Omega$  pullup.

The MAX233 and MAX235 contain all charge pump components, including the capacitors, and operate with NO external components.

The MAX231 and MAX239 include only the V<sup>+</sup> to V<sup>-</sup> charge pump and are intended for applications which have a +5V supply and either a +12V  $\pm 10\%$  supply or a 7.5V to 13.2V battery voltage. When operating with V<sup>+</sup> greater than 8.0V, both C2 and C4 can be 1 $\mu$ F (C1 and C3 are not used with the MAX231 and MAX239).

### Transmitter Section

The transmitters or line drivers are inverting level translators which convert the CMOS or TTL input levels to RS-232 or V.28 voltage levels. With +5V V<sub>CC</sub>, the typical output voltage swing is  $\pm 9V$  when loaded with the nominal 5k $\Omega$  input resistance of an RS-232 receiver. The output swing is guaranteed to meet the RS-232/V.28 specification of  $\pm 5V$  minimum output swing under the worst case conditions of all transmitters driving the 3k $\Omega$  minimum allowable load impedance, V<sub>CC</sub> = 4.5V, and maximum operating ambient temperature. The open circuit output voltage swing is from (V<sup>+</sup> - 0.6V) to V<sup>-</sup>.

The input thresholds are both CMOS and TTL compatible, with a logic threshold of about 25% of V<sub>CC</sub>. The inputs of unused transmitter sections can be left unconnected; an internal 400k $\Omega$  input pullup resistor to V<sup>+</sup> will pull the inputs high, forcing the unused transmitter outputs low. The input pullup resistors source about 12 $\mu$ A, and the transmitter inputs should be driven high or open circuited to minimize power supply current in the shutdown mode.

When in the low power shutdown mode, the transmitter outputs are turned off and their leakage current is less than 1 $\mu$ A with the transmitter output pulled to ground. The transmitter output leakage remains less than 1 $\mu$ A, even if the transmitter output is back-driven between 0V and (V<sub>CC</sub> + 6V). Below -0.5V the transmitter is diode clamped to ground with 1k $\Omega$  series impedance. The transmitter is also zener clamped to approximately V<sub>CC</sub> + 6V, with a series impedance of 1k $\Omega$ . As required by the RS232 and V28, the slew rate is limited to less than 30V/ $\mu$ s. This limits the maximum usable baud rate to 19,200 baud.

### Receiver Section

All but the MAX230 and MAX234 contain RS-232/V.28 receivers. These receivers convert the  $\pm 5V$  to  $\pm 15V$  RS-232 signals to 5V TTL/CMOS outputs. Since the RS-232C/V.28 specifications define a voltage level greater than +3V as a 0, the receivers are inverting. Maxim has set the guaranteed input thresholds of the receivers to 0.8V minimum and 2.4V maximum, which are significantly tighter than the -3.0V minimum and +3.0V maximum required by the RS-232 and V.28 specifications. This allows the receivers to respond both to RS-232/V.28 levels and TTL level inputs. The receivers are protected against input overvoltage up to  $\pm 30V$ .

The 0.8V guaranteed lower threshold is important to ensure that the receivers will have a logic 1 output if the receiver is not being driven because the equipment containing the line driver is turned off or disconnected, or if the connecting cable has an open circuit or short circuit. In other words, the MAX230 family receivers implements Type 1 interpretation of fault conditions (§7 of V.28, §2.5 of RS-232C). While a 0V or even a -3V receiver threshold would be acceptable for the data lines, these lower thresholds would not give proper indication on the control lines such as DTR and DSR. The MAX230 series receivers, on the other hand, have a full 0.8V noise margin for detecting the power-down or cable-disconnected states.

The receivers have a hysteresis of approximately 0.5V, with a minimum guaranteed hysteresis of 200mV. This aids in obtaining clean output transitions, even with slow rise and fall time input signals with moderate amounts of noise and ringing. The propagation delays of the receivers are 350ns for negative-going input signals, and 650ns for positive-going input signals (see Typical Characteristics graphs).

The MAX239 has a receiver 3-state control line and the MAX235 and MAX236 have both a receiver 3-state control line and a low power shutdown control. The receiver TTL/CMOS outputs are in a high impedance 3-state mode whenever the 3-state ENable line is high, and are also high impedance whenever the Shutdown control line is high.

## +5V Powered RS-232 Drivers/Receivers

### Review of EIA Standard RS-232-C and CCITT

#### Recommendations V.28 and V.24

The most common serial interface between electronic equipment is the "RS232" interface. This serial interface has been found to be particularly useful for the interface between units made by different manufacturers since the voltage levels are defined by the EIA Standard RS-232-C and CCITT Recommendation V.28. The RS-232 specification also contains signal circuit definitions and connector pin assignments, while CCITT circuit definitions are contained in a separate document, Recommendation V.24. Originally intended to interface modems to computers and terminals, these standards have many signals which are not used for computer-to-computer or computer-to-peripheral communication. V.24 contains 43 circuit definitions, while RS-232 contains 22 circuit definitions. These circuit names and pin assignments are shown in Table 1, which is an extract of Figures 3.1 and 4.1 of the RS-232C specification, and Table 1 of CCITT recommendation V.24. Most computer-to-computer or computer-to-peripheral communication circuits only use the subset shown in Table 2 of this document.

Serial interfaces can be used with a variety of transmission formats. The most popular by far is the asynchronous format, generally at one of the standard baud rates of 300, 600, 1200, etc. The maximum recommended baud rate for RS-232 and V.28 is 20,000 baud, and the fastest commonly used baud rate is 19,200 baud. Asynchronous serial links use a variety of combinations of the number of data bits, what type (if any) of parity bit, and the number of stop bits. A typical combination is 7 data bits, even parity, and 1 stop bit.

RS232/V.28 physical links are also suitable for synchronous transmission protocols. These higher level protocols often use the standard RS-232C/V.28 voltage levels. Note that one type of physical link (such as RS-232/V.28 voltage levels) can be used for a variety of higher level protocols. Table 3 summarizes the voltage levels and other requirements of V.28 and RS-232.

#### Comparison of RS-232C/V.28 with other standards

The other two most common serial interface specifications are the EIA RS423 and RS422/RS485 (CCITT recommendations V.10 and V.11). While the RS-232 or V.28/V.24 interface is the most common interface for communication between equipment made by different manufacturers, the RS423/V.10 interface and RS422/V.11 interfaces can operate at higher baud rates. In addition, the RS485 interface can be used for low cost local area networks.

The RS423 and V.10 interfaces are unbalanced or

"single-ended" interfaces which use a differential receiver. This standard is intended for data signaling rates up to 100 kbit/s (100 kilobaud). It achieves this higher baud rate through more precise requirements on the waveshape of the transmitters and through the use of differential receivers to compensate for ground potential variations between the transmitting and receiving equipment. With certain limitations, this interface is compatible with RS-232 and V.28. The limitations are:

- 1) less than 20,000 baud rate.
- 2) maximum cable lengths determined by RS-232 performance.
- 3) RS423/V.10 DTE and DCE signal return paths must be connected to the RS232/V.28 signal ground.
- 4) the RS-232 transmitter output voltages must be limited to  $\pm 12V$ , or additional protection must be provided for the RS423/V.10 receiver.
- 5) not all RS232/V.28 receivers will allow proper power-off detection of V.10 transmitter outputs.

Maxim's MAX230 and MAX232-MAX238 meet restrictions 4 and 5 over the entire range of recommended operating conditions. The MAX231 and MAX239 meet restrictions 4 and 5 provided that the  $V^+$  voltage is 12.5V or less.

The RS422, RS485, and V.10 interfaces are balanced double-current interchanges suitable for baud rates up to 10 Mbit/s. These interfaces are not compatible with RS-232 or V.28 voltage levels.

### Application Hints

#### Operation at High Baud Rates

V.28 states that "the time required for the signal to pass through the transition region during a change in state shall not exceed 1 millisecond or 3 per cent of the nominal element period on the interchange circuit, whichever is less." RS-232C allows the transition time to be 4 per cent of the duration of a signal element. At 19,200 baud, the "nominal element period" is approximately  $50\mu s$ , of which 3 per cent is  $1.5\mu s$ . Since the transition region is from  $-3V$  to  $+3V$ , this means the V.28 slew rate would ideally be faster than  $6V/1.5\mu s = 4V/\mu s$  at 19.2 kbaud and  $2V/\mu s$  at 9600 baud. The RS-232 requirement is equivalent to  $3V/\mu s$  at 19.2 kbaud,  $1.5V/\mu s$  at 9600 baud, etc. The slew rate of the MAX230 series devices is about  $3V/\mu s$  with the maximum recommended load of 2500pF. In practice, the effect of less than optimum slew rate is a distortion of the recovered data, where the 1's and 0's no longer have equal width. This distortion generally has negligible effect and the MAX230 family can be reliably used for 19.2 kbaud serial links when the cable capacitance is kept below 2500pF. With very low capacitive loading, the MAX230 and MAX234-239 may even be used at 38.4 kbaud, since the typical slew rate is  $5V/\mu s$  when loaded with 500pF in parallel

MAX230-239

## +5V Powered RS-232 Drivers/Receivers

MAX230-239

Table 1. Circuit Definitions/Pin Assignments

PIN	EIA RS-232 CIRCUIT	CCITT V.24 CIRCUIT	DESCRIPTION	TYPE	SOURCE
1	AA	101	Protective Ground	Ground	Ground
2	BA	103	Transmitted Data	Data	DTE
3	BB	104	Received Data	Data	DCE
4	CA	105	Request to Send	Control	DTE
5	CB	106	Clear to Send	Control	DCE
6	CC	107	Data Set Ready	Control	DCE
7	AB	102	Signal Ground/Return	Ground	Ground
8	CF	109	Received Line Signal Detector	Control	DCE
9	--	--	Reserved for Data Set	Testing	
10	--	--	Reserved for Data Set	Testing	
11	--	--	Unassigned		
12	SCF	122	Sec. Received Line Signal Detector	Control	DCE
13	SCB	121	Secondary CTS	Control	DCE
14	SBA	118	Sec. Transmitted Data	Data	DTE
15	DB	114	Transmission Signal Element Timing (DCE)	Timing	DCE
16	SBB	119	Sec. Received Data	Data	DCE
17	DD	115	Receiver Signal Element Timing (DCE)	Timing	DCE
18	--	--	Unassigned		
19	SCA	120	Secondary RTS	Control	DTE
20	CD	108.2	Data Terminal Ready	Control	DTE
21	CG	110	Signal Quality Det	Control	DCE
22	CE	125	Ring Indicator	Control	DCE
23	CHCI	111/112	Data Signal Rate Selector (DCE/DTE)	Control	DTE/DCE
24	DA	113	Transmit Signal Element Timing (DTE)	Timing	DTE
25	--	--	Unassigned		

Table 2. Circuits Commonly Used for RS-232C and V.24 Asynchronous Interfaces

PIN	CIRCUIT
1	Protective Ground Connect to Earth Ground
2	Transmit Data (TD) Data from DTE
3	Receive Data (RD) Data from DCE
4	Request To Send (RTS) Handshake from DTE
5	Clear to Send (CTS) Handshake from DCE
6	Data Set ready (DSR) Handshake from DCE
7	Signal Ground Reference Point for Signals
8	Received Line Signal Detector (sometimes called Carrier Detect, CDC)
11	Printer Busy Signal Handshake from Printer
20	Data Terminal Ready Handshake from DTE
22	Ring Indicator Handshake from DCE

Table 3. Summary of RS-232C and V.28 Electrical Specifications

PARAMETER	SPECIFICATION	COMMENTS
Driver Output Voltage		
0 level	-5V to +15V	With 1-7k $\Omega$ load
1 level	-5V to -15V	With 3-7k $\Omega$ load
Max. output	$\pm$ 25V Max	No Load
Receiver Input Thresholds (data and clock signals)		
0 level	-3V to +25V	
1 level	+3V to +25V	
Receiver Thresholds RTS, DSR, DTR		
On level	+3V to +25V	
Off level	Open Circuit or -3V to -25V	Detects Power Off Condition at Driver
Receiver Input Resistance	3k $\Omega$ to 7k $\Omega$	
Driver Output Resistance, power off condition	300 $\Omega$ Min	$V_{out} < \pm 2V$
Driver Slew Rate	30V/ $\mu$ s Max	3k $\Omega < R_L < 7k\Omega$ , $C_{in} < C_L < 2500pF$
Signaling Rate	Up to 20kbits/sec	
Cable Length	50/15 m Recommended Max. Length	Longer cables permissible, if $C_{load} < 2500pF$

## +5V Powered RS-232 Drivers/Receivers

### Non-Inverting Drivers and Receivers

Occasionally a non-inverting transmitter or receiver is needed instead of the inverting transmitters and receivers of the MAX230 family. Simply use one of the receivers as a TTL/CMOS inverter to get the desired operation (Figure 15). If the logic output driving the receiver input has less than 1mA of output source capability, then add the 2.2k $\Omega$  pullup resistor.

The MAX230 series receiver TTL outputs can directly drive the input of another receiver to form a non-inverting RS-232 receiver.

### Protection for Shorts to $\pm 15V$ Supplies

All MAX230 family devices transmitter outputs except the MAX231, MAX232 and MAX233 are protected against short circuits to  $\pm 15V$ , which is the maximum allowable loaded output voltage of an RS-232/V28 transmitter. The MAX231, MAX232, and MAX233 can be protected against short circuits to  $\pm 15V$  power supplies by the addition of a series 220 $\Omega$  resistor in each output. This protection is not needed to protect against short circuits to most RS-232 transmitters such as the 1488, since they have an internal short circuit current limit of 12mA.

The power dissipation of the MAX230 and MAX234-MAX239 is about 200mW with all transmitters shorted to  $\pm 15V$ .

### Isolated RS-232 Interfaces

RS-232 and V28 specifications require a common ground connection between the two units communicating via the RS-232/V28 interface. In some cases, there may be large differences in ground potential between the two units, and in other cases it may be desired to avoid ground loop currents by isolating the two grounds. In other cases, a computer or control system must be protected against accidental connection of the RS-232/V28 signal lines to 110/220VAC power lines. Figure 16 shows a circuit with this isolation. The power for the MAX233 is generated by a MAX635 DC-DC converter. When the MAX635 regulates point "A" to -5V, the isolated output at point "B" will be semi-regulated to +5V. The two optocouplers maintain isolation between the system ground and the RS-232 ground while transferring the data across the isolation barrier. While this circuit will not withstand 110VAC between the RS-232 ground and either the receiver or transmitter lines, the voltage difference between the two grounds is only limited by the optocoupler and DC-DC converter transformer breakdown ratings.

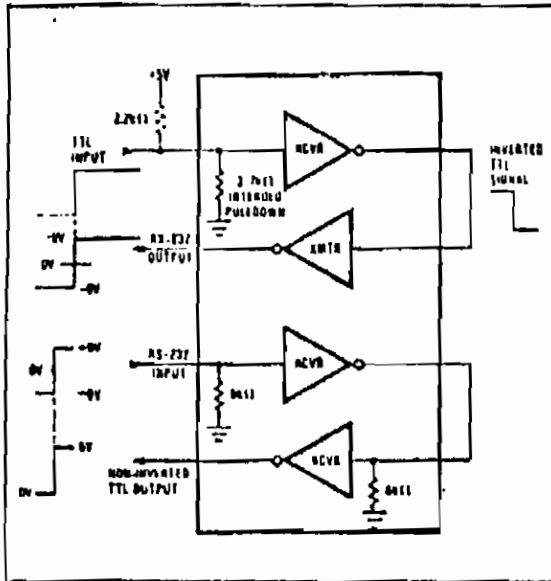


Figure 15. Non-Inverting RS-232 Transmitters and Receivers

with 5k $\Omega$ . Under no circumstance will the slow rate exceed the RS-232/V28 maximum spec of 30V/ $\mu$ s and, unlike the MC1488 driver, no external compensation capacitors are needed under any load condition.

### Driving Long Cables

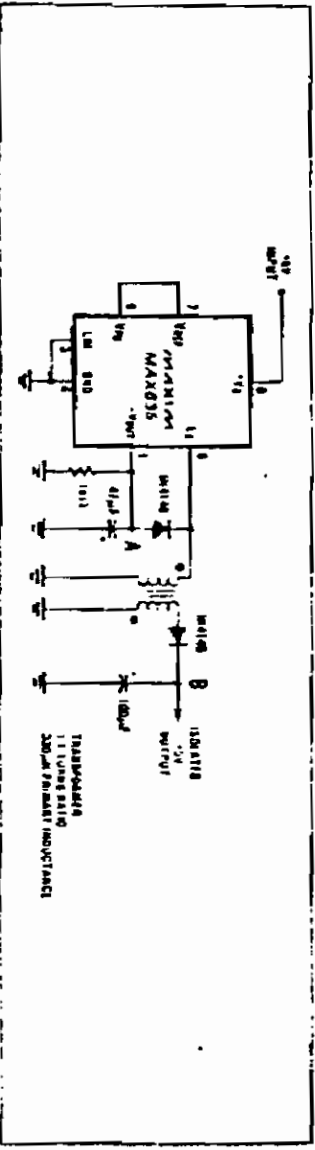
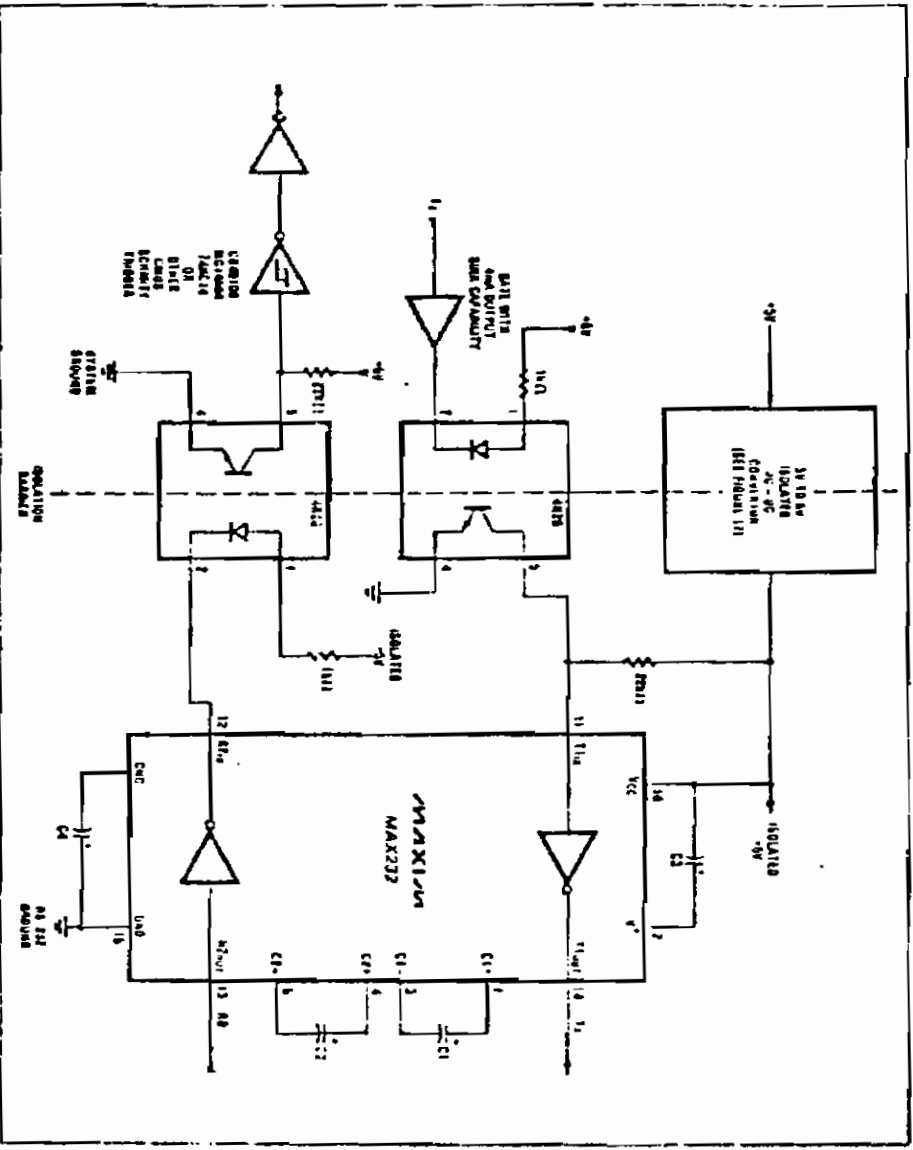
The RS-232 standard states that "The use of short cables (each less than approximately 50 feet or 15 meters) is recommended; however, longer cables are permissible, provided that the load capacitance does not exceed 2500pF."

Baud rate and cable length can be traded off: use lower baud rates for long cables, use short cables if high baud rates are desired. For both long cables and high baud rates, use H5422/V.11. The maximum cable length for a given baud rate is determined by several factors, including the capacitance per meter of cable, the slow rate of the driver under high capacitive loading, the receiver threshold and hysteresis, and the acceptable bit error rate. The MAX230 family receivers have 0.5V of hysteresis, and the transmitters are designed such that the slow rate reduction caused by capacitive loading is minimized (see Typical Characteristics).



# +5V Powered RS-232 Drivers/Receivers

## MAX230-239



## +5V Powered RS-232 Drivers/Receivers

### Ordering Information

**MAX230-239**

PART	TEMP RANGE	PACKAGE
<b>MAX230</b>		
MAX230CPP	0°C to +70°C	20 Lead Plastic DIP
MAX230CWP	0°C to +70°C	20 Lead Small Outline
MAX230C/D	0°C to +70°C	Dice
MAX230FPP	-40°C to +85°C	20 Lead Plastic DIP
MAX230FWP	-40°C to +85°C	20 Lead Small Outline
MAX230FJP	-40°C to +85°C	20 Lead CERDIP
MAX230MJP	-55°C to +125°C	20 Lead CERDIP
<b>MAX231</b>		
MAX231CPD	0°C to +70°C	14 Lead Plastic DIP
MAX231CWE	0°C to +70°C	18 Lead Small Outline
MAX231C/D	0°C to +70°C	Dice
MAX231EPD	-40°C to +85°C	14 Lead Plastic DIP
MAX231EWE	-40°C to +85°C	18 Lead Small Outline
MAX231EJD	-40°C to +85°C	14 Lead CERDIP
MAX231MJD	-55°C to +125°C	14 Lead CERDIP
<b>MAX232</b>		
MAX232CPF	0°C to +70°C	16 Lead Plastic DIP
MAX232CWE	0°C to +70°C	16 Lead Small Outline
MAX232C/D	0°C to +70°C	Dice
MAX232EPE	-40°C to +85°C	16 Lead Plastic DIP
MAX232EJE	-40°C to +85°C	16 Lead CERDIP
MAX232EWE	-40°C to +85°C	16 Lead Small Outline
MAX232MJE	-55°C to +125°C	16 Lead CERDIP
<b>MAX233</b>		
MAX233CPP	0°C to +70°C	20 Lead Plastic DIP
MAX233EPP	-40°C to +85°C	20 Lead Plastic DIP
<b>MAX234</b>		
MAX234CPE	0°C to +70°C	16 Lead Plastic DIP
MAX234CWE	0°C to +70°C	16 Lead Small Outline
MAX234C/D	0°C to +70°C	Dice
MAX234EPE	-40°C to +85°C	16 Lead Plastic DIP
MAX234EWP	-40°C to +85°C	16 Lead Small Outline
MAX234EJE	-40°C to +85°C	16 Lead CERDIP
MAX234MJE	-55°C to +125°C	16 Lead CERDIP
<b>MAX235</b>		
MAX235CPD	0°C to +70°C	24 Lead Plastic DIP*
MAX235CPG	-40°C to +85°C	24 Lead Plastic DIP*
MAX235EPD	-40°C to +85°C	24 Lead Plastic DIP*
MAX235MDG	-55°C to +125°C	24 Lead Ceramic*

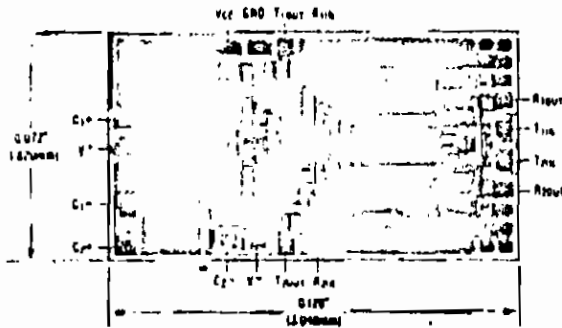
PART	TEMP RANGE	PACKAGE
<b>MAX236</b>		
MAX236CNG	0°C to +70°C	24 Lead Plastic DIP
MAX236CWG	0°C to +70°C	24 Lead Small Outline
MAX236C/D	0°C to +70°C	Dice
MAX236ENG	-40°C to +85°C	24 Lead Plastic DIP
MAX236EWG	-40°C to +85°C	24 Lead Small Outline
MAX236ERG	-40°C to +85°C	24 Lead CERDIP
MAX236MRG	-55°C to +125°C	24 Lead CERDIP
<b>MAX237</b>		
MAX237CNG	0°C to +70°C	24 Lead Plastic DIP
MAX237CWG	0°C to +70°C	24 Lead Small Outline
MAX237C/D	0°C to +70°C	Dice
MAX237ENG	-40°C to +85°C	24 Lead Plastic DIP
MAX237EWG	-40°C to +85°C	24 Lead Small Outline
MAX237ERG	-40°C to +85°C	24 Lead CERDIP
MAX237MRG	-55°C to +125°C	24 Lead CERDIP
<b>MAX238</b>		
MAX238CNG	0°C to +70°C	24 Lead Plastic DIP
MAX238CWG	0°C to +70°C	24 Lead Small Outline
MAX238C/D	0°C to +70°C	Dice
MAX238ENG	-40°C to +85°C	24 Lead Plastic DIP
MAX238EWG	-40°C to +85°C	24 Lead Small Outline
MAX238ERG	-40°C to +85°C	24 Lead CERDIP
MAX238MRG	-55°C to +125°C	24 Lead CERDIP
<b>MAX239</b>		
MAX239CNG	0°C to +70°C	24 Lead Plastic DIP
MAX239CWG	0°C to +70°C	24 Lead Small Outline
MAX239C/D	0°C to +70°C	Dice
MAX239EWG	-40°C to +85°C	24 Lead Small Outline
MAX239ERG	-40°C to +85°C	24 Lead CERDIP
MAX239MRG	-55°C to +125°C	24 Lead CERDIP

\* = 0.600" package, all other packages are 0.300" wide

# +5V Powered RS-232 Drivers/Receivers

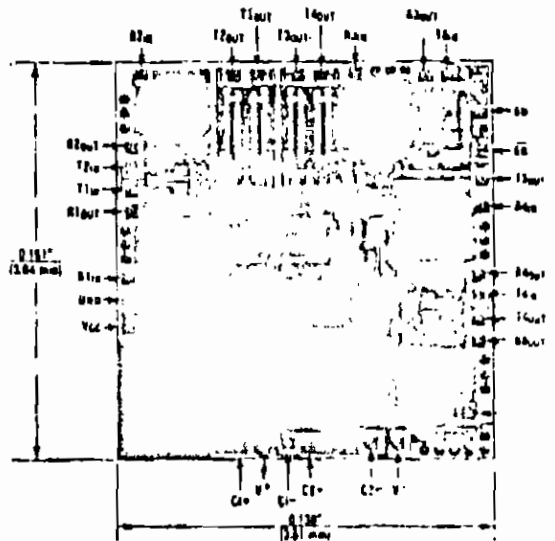
## Chip Topography

MAX230-239



MAX231 and MAX232

Note: Connect substrate to V<sup>+</sup>

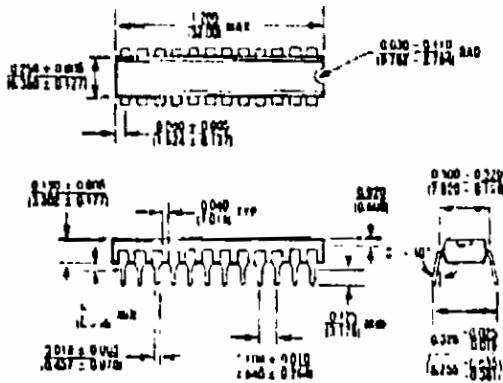


MAX230 and MAX234 - MAX238

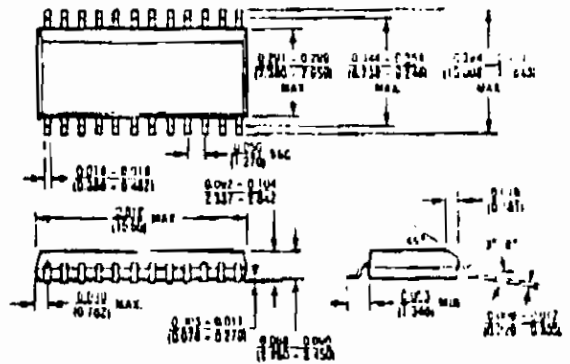
**Notes:**

1. 50 pin of MAX234, MAX237, MAX238 and MAX239 are internally connected to ground.
2. Connect substrate to V<sup>+</sup>.

## Package Information



24 Pin Plastic DIP (NO)  
 $\theta_{JA} = 120^{\circ}\text{C/W}$   
 $\theta_{JC} = 60^{\circ}\text{C/W}$



24 Lead Small Outline, Wide (WQ)  
 $\theta_{JA} = 90^{\circ}\text{C/W}$   
 $\theta_{JC} = 50^{\circ}\text{C/W}$

Maxim cannot assume responsibility for use of any circuitry other than circuitry entirely enclosed in a Maxim product. No circuit patent licenses are implied. Maxim reserves the right to change the circuitry and specifications without notice at any time.

18 Maxim Integrated Products, 810 N. Peckerin Avenue, Sunnyvale, CA 94086 (408) 737-7000

©1986 Maxim Integrated Products

Printed USA

11/86

LECTOR DE COIRO DE BARRAS

# MCS<sup>®</sup>-51 ARCHITECTURE

## 6.0 INTRODUCTION

The MCS<sup>®</sup>-51 family of 8-bit microcontrollers consists of the devices listed in Table 1, all of which are based on the MCS-51 architecture shown in Figure 6-1. The original 8051 was built in HMOS I technology. The HMOS II version, which is the device currently in production, is called the 8051AH. The term "8051," however, is still

often used to generically refer to all of the MCS-51 family members. This is the case throughout this manual, except where specifically stated otherwise. Also for brevity, the term "8052" is used to refer to both the 8052 and the 8032, unless otherwise noted.

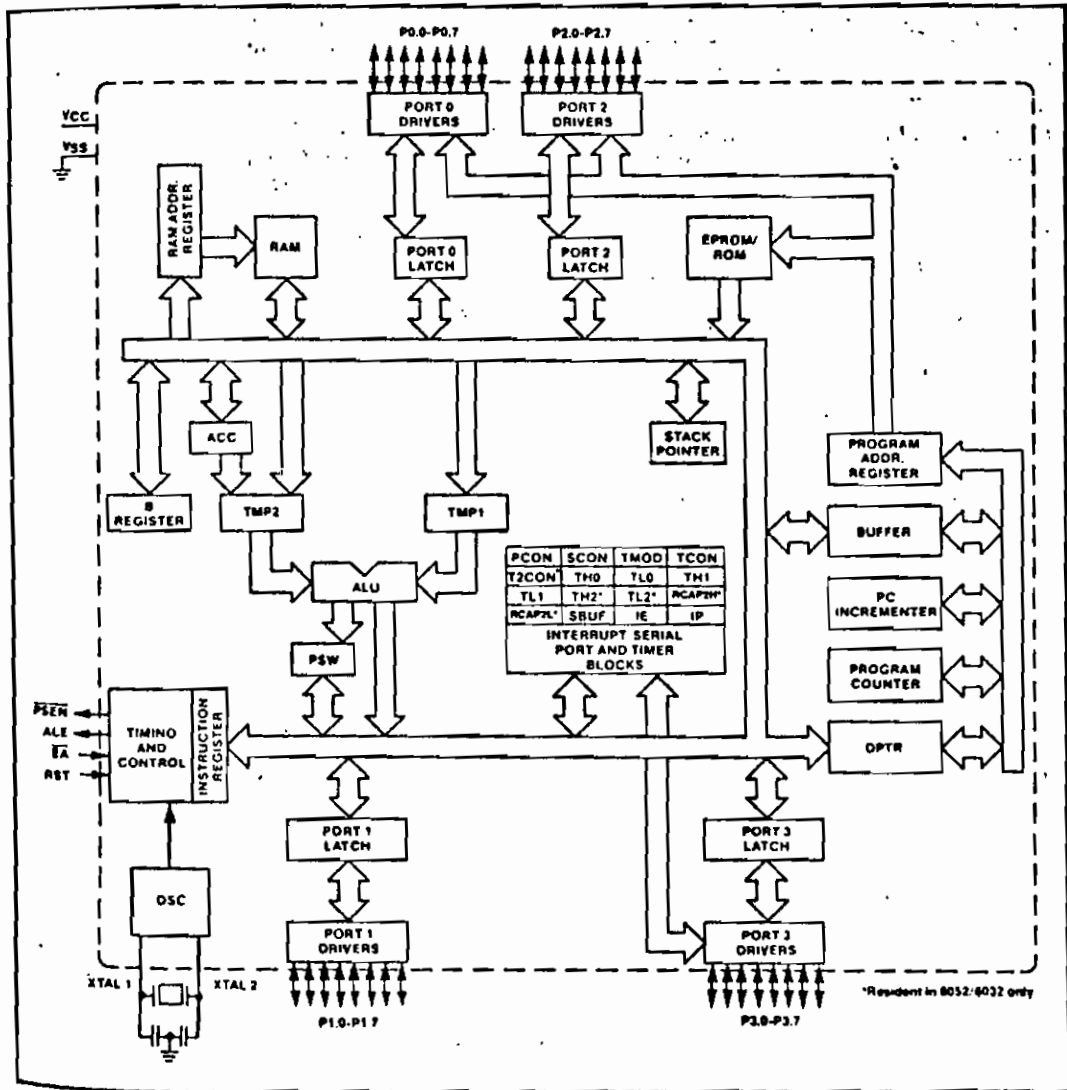


Figure 6-1. MCS-51 Architectural Block Diagram

The newest MCS-51 members, the 8032 and 8052, have more on-chip memory and an additional 16-bit timer/counter. The new timer can be used as a timer, a counter, or to generate baud rates for the serial port. As a timer/counter, it operates in either a 16-bit auto-reload mode or a 16-bit "capture" mode. This new feature is described in Section 6.6.2.

Pinouts are shown in the individual data sheets and on the inside back cover of this handbook.

Table 1. MCS<sup>®</sup>-51 Family Members

PART	TECHNOLOGY	ON-CHIP PROGRAM MEMORY	ON-CHIP DATA MEMORY
8051AH	HMOS II	4K — ROM	128
8031AH	HMOS II	NONE	128
8751H	HMOS I	4K — EPROM	128
80C51	CHMOS	4K — ROM	128
80C31	CHMOS	NONE	128
8052	HMOS II	8K — ROM	256
8032	HMOS II	NONE	256

The major MCS<sup>®</sup>-51 features are:

- 8-Bit CPU
- On-Chip oscillator and clock circuitry
- 32 I/O lines
- 64K address space for external data memory
- 64K address space for external program memory
- Two 16-bit timer/counters (three on 8032/8052)
- A five-source interrupt structure (six sources on 8032/8052) with two priority levels
- Full duplex serial port
- Boolean processor

## 6.1 MEMORY ORGANIZATION

The 8051 has separate address spaces for Program Memory and Data Memory. The Program Memory can be up to 64K bytes long. The lower 4K (8K for 8052) may reside on-chip. The Data Memory can consist of up to 64K bytes of off-chip RAM, in addition to which, it includes 128 bytes of on-chip RAM (256 bytes for the 8052), plus a number of "SFRs" (Special Function Registers) as listed below.

Symbol	Name	Address
*ACC	Accumulator	0E0H
*B	B Register	0F0H
*PSW	Program Status Word	0D0H
SP	Stack Pointer	81H
DPTR	Data Pointer (consisting of DPH and DPL)	83H 82H
*P0	Port 0	80H
*P1	Port 1	90H

Symbol	Name	Address
*P2	Port 2	0A0H
*P3	Port 3	0B0H
*IP	Interrupt Priority Control	0B8H
*IE	Interrupt Enable Control	0A8H
TMOD	Timer/Counter Mode Control	89H
+*T2CON	Timer/Counter Control	88H
TCON	Timer/Counter 2 Control	0C8H
TH0	Timer/Counter 0 (high byte)	8CH
TL0	Timer/Counter 0 (low byte)	8AH
TH1	Timer/Counter 1 (high byte)	8DH
TL1	Timer/Counter 1 (low byte)	8BH
+TH2	Timer/Counter 2 (high byte)	0CDH
+TL2	Timer/Counter 2 (low byte)	0CCH
+RCAP2H	Timer/Counter 2 Capture Register (high byte)	0CBH
+RCAP2L	Timer/Counter 2 Capture Register (low byte)	0CAH
*SCON	Serial Control	98H
SBUF	Serial Data Buff	99H
PCON	Power Control	97H

The SFRs marked with an asterisk (\*) are both bit- and byte-addressable. The SFRs marked with a plus sign (+) are present in the 8052 only. The functions of the SFRs are described as follows.

### ACCUMULATOR

ACC is the Accumulator register. The mnemonics for accumulator-specific instructions, however, refer to the accumulator simply as A.

### B REGISTER

The B register is used during multiply and divide operations. For other instructions it can be treated as another scratch pad register.

### PROGRAM STATUS WORD

The PSW register contains program status information as detailed in Figure 6-2.

### STACK POINTER

The Stack Pointer register is 8 bits wide. It is incremented before data is stored during PUSH and CALL executions. While the stack may reside anywhere in on-chip RAM.

MCS-51 ARCHITECTURE

the Stack Pointer is initialized to 07H after a reset. This causes the stack to begin at location 08H.

DATA POINTER

The Data Pointer (DPTR) consists of a high byte (DPH) and a low byte (DPL). Its intended function is to hold a 16-bit address. It may be manipulated as a 16-bit register or as two independent 8-bit registers.

PORTS 0 to 3

P0, P1, P2 and P3 are the SFR latches of Ports 0, 1, 2 and 3, respectively.

SERIAL DATA BUFFER

The Serial Data Buffer is actually two separate registers, a transmit buffer and a receive buffer register. When data is moved to SBUF, it goes to the transmit buffer where it is held for serial transmission. (Moving a byte to SBUF is what initiates the transmission.) When data is moved from SBUF, it comes from the receive buffer.

TIMER REGISTERS

Register pairs (TH0, TL0), (TH1, TL1), and (TH2, TL2) are the 16-bit counting registers for Timer/Counters 0, 1, and 2, respectively.

CAPTURE REGISTERS

The register pair (RCAP2H, RCAP2L) are the capture registers for the Timer 2 "capture mode." In this mode, in response to a transition at the 8052's T2EX pin, TH2 and TL2 are copied into RCAP2H and RCAP2L. Timer

2 also has a 16-bit auto-reload mode, and RCAP2H and RCAP2L hold the reload value for this mode. More about Timer 2's features in Section 6.6.2.

CONTROL REGISTERS

Special Function Registers IP, IE, TMOD, TCON, T2CON, SCON, and PCON contain control and status bits for the interrupt system, the timer/counters, and the serial port. They are described in later sections.

6.2 OSCILLATOR AND CLOCK CIRCUIT

XTAL1 and XTAL2 are the input and output of a single-stage on-chip inverter, which can be configured with off-chip components as a Pierce oscillator, as shown in Figure 6-3. The on-chip circuitry, and selection of off-chip components to configure the oscillator are discussed in Section

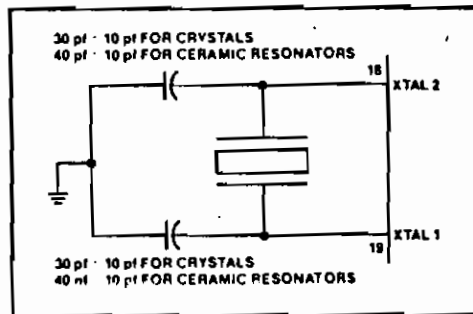


Figure 6-3. Crystal/Ceramic Resonator Oscillator

(MSB)								(LSB)	
CY	AC	FO	RS1	RS0	OV	—	—	P	
Symbol	Position	Name and Significance			Symbol	Position	Name and Significance		
CY	PSW.7	Carry flag.			OV	PSW.2	Overflow flag.		
AC	PSW.6	Auxiliary Carry flag. (For BCD operations.)			—	PSW.1	(reserved)		
FO	PSW.5	Flag 0 (Available to the user for general purposes.)			P	PSW.0	Parity flag. Set/cleared by hardware each instruction cycle to indicate an odd/even number of "one" bits in the accumulator, i.e., even parity.		
RS1	PSW.4	Register bank Select control bits 1 & 0. Set cleared by software to determine working register bank (see Note).			Note— the contents of (RS1, RS0) enable the working register banks as follows: (0,0)—Bank 0 (00H-07H) (0,1)—Bank 1 (08H-0FH) (1,0)—Bank 2 (10H-17H) (1,1)—Bank 3 (18H-1FH)				
RS0	PSW.3								

Figure 6-2. PSW: Program Status Word Register

MCS<sup>®</sup>-51 ARCHITECTURE

6.13. A more detailed discussion will be found in Application Note AP-155, "Oscillators for Microcontrollers," which is included in this manual.

The oscillator, in any case, drives the internal clock generator. The clock generator provides the internal clocking signals to the chip. The internal clocking signals are at half the oscillator frequency, and define the internal

phases, states, and machine cycles, which are described in the next section.

6.3 CPU TIMING

A machine cycle consists of 6 states (12 oscillator periods). Each state is divided into a Phase 1 half, during which the Phase 1 clock is active, and a Phase 2 half,

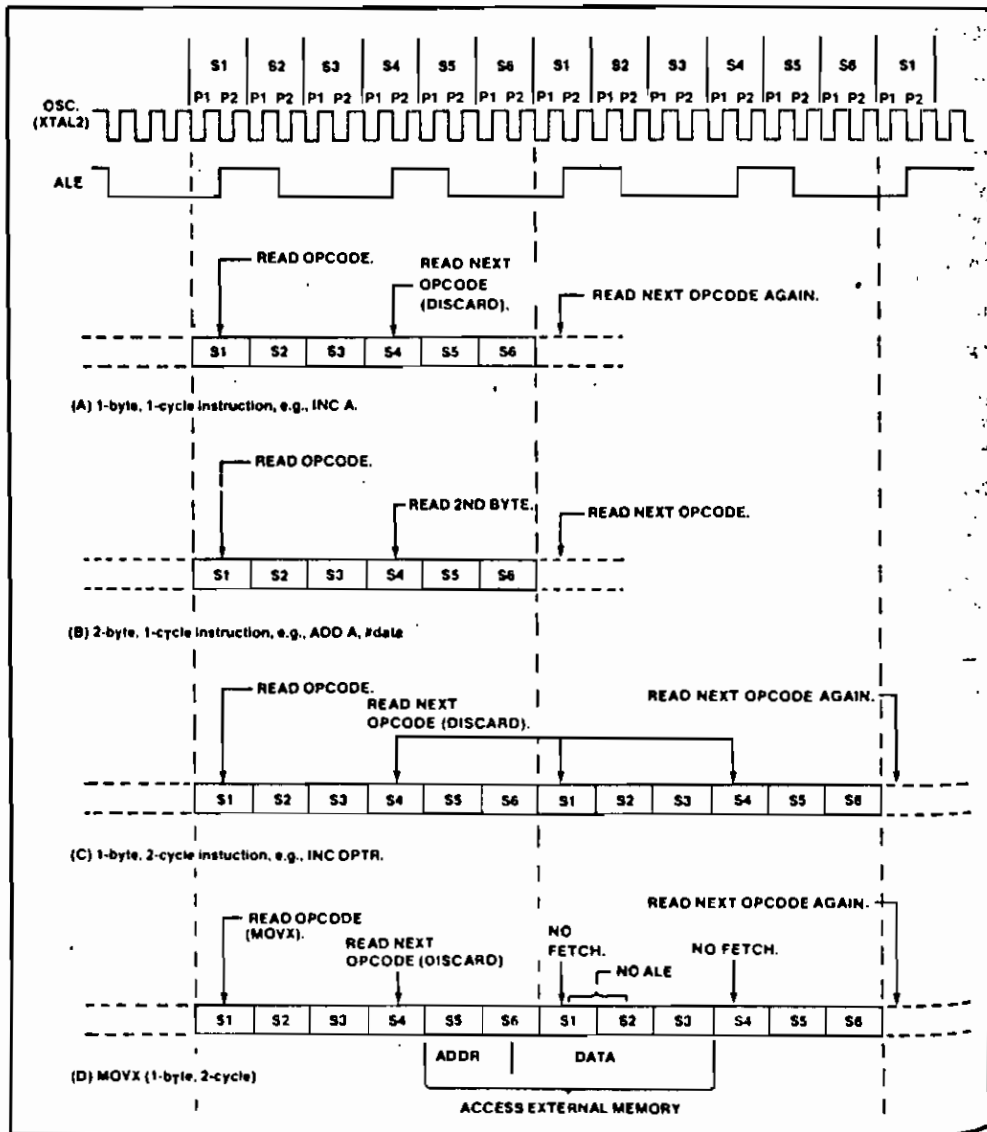


Figure 6-4. 8051 Fetch/Execute Sequences

## MANUAL DE USO

El sistema lector de código de barras requiere de un computador PC-XT con sistema operativo DOS y pórtico serial.

Para una adecuada instalación del equipo externo se utiliza un cable normalizado RS232 (terminales DB25S), con la configuración descrita en la FIGURA H.1.

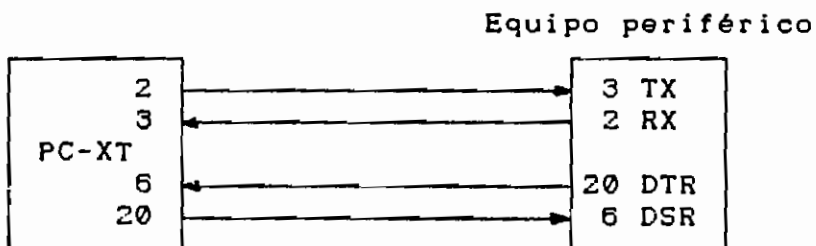


Fig H.1 INTERCONEXION DEL EQUIPO

Los nombre utilizados para los programas inician con el prefijo "BS", las siguientes tres siglas definen el programa utilizado y puede ser "IMP", "DEC" y "GEN".

Los parámetros deben ser pasados utilizando la siguiente sintaxis:



BSXXX.EXE (Opción= Parámetro\Opción= Parámetro\....)

{Opción=:            Sigla del parámetro a ser modificado y  
                      depende de cada programa.

Parámetro:            Caracter(es) válido(s) para cada opción.

BSIMP.EXE (Opción= Parámetro\Opción= Parámetro\....)

Rutina de impresión de etiquetas. Módulo independiente que requiere del archivo "TABLA.CBR" para su normal funcionamiento donde la detección de las unidades disponibles se realiza en forma automática.

Los parámetros de BSIMP.EXE son:

Unidad de trabajo.- "U="

La unidad predefinida es "A" y puede ser alterada con la opción "U=x", "x" es la identificación de alguna de las unidades disponibles.

Nombre del archivo.- "A="

Permite definir el nombre del archivo de la tabla del

Ancho de las barras.- "H="

La opción "H=x" modifica la dimensión de ancho de las barras, "x" es un número entero que multiplica el ancho mínimo del modo gráfico utilizado con: 3/240 de pulgada (0.317 mm) para ceros (0) y 12/240 de pulgada (0.635 mm) para unos (1), siendo el valor predefinido 2.

Número de copias.- "C="

Esta opción permite definir el número de copias de una misma etiqueta con "C=x", "x" es número entero.

Caracteres de la etiqueta.- "D="

"D" permite pasar a la rutina BSIMP.EXE los caracteres que serán representados en código de barras en la etiqueta. Los caracteres válidos serán aquellos que tengan su equivalencia binaria en el archivo "TABLA.CBR".

Los errores que pudiesen ocurrir durante la ejecución de BSIMP.EXE interrumpen el procedimiento y presentan en pantalla el mensaje de error correspondiente, con el propósito de realizar las acciones pertinentes para superar la condición de error; en los casos que el error no pueda

ser corregido el programa debe ser abortado con la tecla ESCAPE "ESC".

BSDEC.EXE {Opción= Parámetro\Opción= Parámetro\....}

La decodificación realizada por el módulo BSDEC.EXE requiere del archivo "TABLA.CBR" para su normal ejecución. Los caracteres recuperados pueden ser presentados en pantalla o en un archivo de texto creado por ésta rutina en la unidad de disco virtual nombrado "LECT.CBR".

La rapidez de su ejecución está determinada por el tiempo de acceso a la unidad que contiene el archivo "TABLA.CBR" por lo que es conveniente copiarlo previamente en la unidad disponible de más rápido acceso.

Los siguientes parámetros de BSDEC.EXE realizan similar función que los descritos para el programa BSIMP.EXE.

Unidad de trabajo.- "U="

Nombre del archivo.- "A="

Número de bits.- "N="

Los parámetros de uso específico de BSDEC.EXE son:

Número de bits de valor uno.- "M=x"

El parámetro "M" pasa al programa la información relativa al número de bits de valor uno "1" por caracter; definiendo así el formato del código que será decodificado. El valor de la opción "M" es transferido al circuito externo para ser utilizado como criterio de error de formato. "x" debe ser un número entero de valor menor al definido con el parámetro "N".

Dispositivo de salida.- "l=x"

Define el modo de salida de los caracteres decodificados, las opciones admisible son: "l=l" para reporte en pantalla e "l=A" para reporte en un archivo cuyo nombre es "LECT.CBR" creado en la unida de disco virtual. Este archivo podrá ser utilizado para recuperar los caracteres de una etiqueta por otros programas de aplicación específica, desarrollados aprovechando las facilidades del lector de código de barras.

La unidad de disco virtual deberá tener una capacidad de almacenamiento acorde a las necesidades de la rutina. El tamaño del archivo "TABLA.CBR" es 71936 KBytes, los mensajes

de error requieren de almenos 4 KBytes y el archivo "LECT.CBR" de 20 Bytes.

En los casos en los cuales sea imprescindible abortar el programa BSDEC.EXE será conveniente reinicializar el circuito externo con el botón de "RESET".

BSGEN.EXE {Opción= Parámetro\Opción= Parámetro\....}

El programa BSGEN.EXE requiere de parámetros ya definidos y que pueden ser modificados pasándolos como argumentos de BSGEN. Las siguientes opciones permiten alterar los parámetros en la forma descrita para BSIMP.EXE y/o BSDEC.EXE:

Unidad de trabajo.- "U="

Nombre del archivo.- "A="

Número de bits.- "N="

Número de unos.- "M="

Los programas están en capacidad de detectar errores en procedimientos del sistema operativo DOS, presentando el mensaje de error correspondiente para que la acción

pertinente sea efectuada por el usuario, con el objeto de corregir la causa del error o abortar la ejecución del programa.

A continuación se describen los errores verificados:

Error 57 " Device I/O error ". Error de Entrada/Salida se produce durante accesos a dispositivos de entrada salida.

Error 68 "device unavailable". Dispositivo no disponible ocurre el momento de acceder con funciones de Entrada/Salida a recursos no presentes o no inicializados adecuadamente.

Error 24 " Device timeout". Recurso no disponible por el momento, cuando un dispositivo de Entrada/Salida no puede ser accedido por encontrarse en un modo de operación diferente o está siendo utilizado en otra tarea.

Error 25 " Device fault ". Impresor fuera de línea. Desconectado o apagado.

Error 27 " Out of paper ". Impresor no tiene papel.

Error 53 " File no found ". Archivo no encontrado.

## BIBLIOGRAFIA

- HEWLETT-PACKARD JOURNAL  
Enero 1981. Volumen 32. Numero 1

pág 3

Handheld Scanner Maked Reading Bar Codes Easy an  
Inexpensive.  
John J. Uebbing, Donald L. Lubin, Edward G. Weaver

pág 11

Reading Bar Codes for the HP-41C Programable Calculator  
David R. Conklin, Thomas L. Revere

- BAR CODE READER  
TANDY TRS-80  
For Model 100 Owner's Manual

- MICROCONTROLLER HANDBOOK 1984  
Intel corporation.

- PC MAGAZINE  
Noviembre 12, 1985

pág 201-218

PEEKing st Your PC's Memory  
David I. Schneider

- PC MAGAZINE  
Noviembre 26, 1985

pág 187-208

PEEKing st Your PC's Memory, Part 2  
David I. Schneider

- MASTERING SERIAL COMMUNICATIONS  
Peter W. Gofton  
Sybex Inc. 1986

pág 202-217

Capitulo 15  
Communications in BASIC

- GUIA DEL PROGRAMADOR PARA EL IBM PC  
Peter Norton  
Red Editorial Iberoamericana, Andes Ltda.  
Bogotá, D.E. Colombia 1987 ( edición original 1985)

pág 58-67

Referencia de las direcciones bajas de memoria

pág 95-97  
Memoria de imagen.

- ACTION PRINTER T-750  
User's Manual  
Epson America Inc. 1988
- ACTION PRINTER APX-80  
User's Manual  
Epson America Inc. 1988
- CITTIZEN MSP-40/45  
User's Manual  
Cittizen America Corporation 1987
- MICROPROCESOR APPLICATIONS MANUAL  
MOTOROLA Semiconductor Products Inc.  
Mc Graw- Hill company 1975.

pág 5-12 - 5-40

Scanning wand for capturing data from printed symbols.

- ERROR CONTROL CODING  
Shu Lin - Costelo

pág 65