

**ESCUELA POLITECNICA NACIONAL  
FACULTAD DE INGENIERIA ELECTRICA**

**DISEÑO Y CONSTRUCCION DE UN  
SISTEMA DE CONTROL DE TIEMPOS DE  
RECORRIDO EN LOS BUSES DE  
TRANSPORTE PUBLICO**

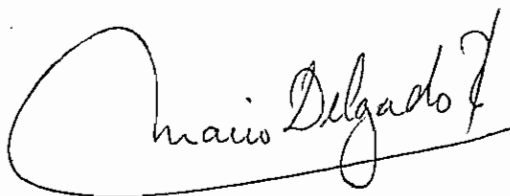
**MARIO A. DELGADO R.**

**QUITO NOVIEMBRE DE 2000**

## DECLARACION

Yo Mario Delgado Reyes, declaro que el trabajo aquí descrito es de mi autoría, que no ha sido previamente presentado para ningún grado o calificación profesional y, que he consultado las referencias bibliográficas que se incluyen en este documento.

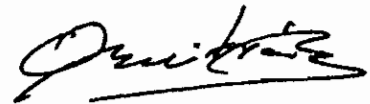
La Escuela Politécnica Nacional, puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley, Reglamento de Propiedad Intelectual y por la normativa institucional vigente.

A handwritten signature in black ink, reading "Mario Delgado Reyes". The signature is written in a cursive style with a large, looping initial "M" and a distinct "R" at the end.

Mario Delgado Reyes

## CERTIFICACION

Certifico que el presente trabajo fue desarrollado por Mario Delgado Reyes, bajo mi supervisión.



INGENIERO OSWALDO BUITRON  
DIRECTOR DE PROYECTO

## **AGRADECIMIENTO**

A todos los que me han apoyado en la elaboración de este trabajo,  
Especialmente a mi Director de Tesis el Ingeniero Oswaldo Buitrón  
Y a mi muy buen amigo Ingeniero Marcelo Guarderas.

## **DISEÑO Y CONSTRUCCION DE UN SISTEMA DE CONTROL DE TIEMPOS DE RECORRIDO EN LOS BUSES DE TRANSPORTE PUBLICO**

### **Justificación.-**

Se ha visto que el principal problema que enfrentan las cooperativas de transporte, entre ellas e internamente en cada una de las mismas, es el incumplimiento del tiempo que debe tomar cada unidad en realizar cada etapa de su recorrido por la ruta. Al incumplirse surgen peleas con choferes de otras cooperativas e internamente con los propios compañeros.

Para evitar estos problemas se están usando relojes marcadores de tarjeta de cartulina en diferentes puntos de una ruta determinada y estas marcaciones son evaluadas manualmente para el cálculo de las diferencias de tiempo y las consiguientes sanciones.

Con el presente trabajo se desarrollará un sistema similar pero con el uso de microprocesadores para lectura de tarjetas magnéticas y comunicación a un Computador Personal (PC), con lo que se logrará mayor velocidad de procesamiento de los datos y se evitarán errores voluntarios e involuntarios en el procesamiento de la información.

### **Alcance.-**

El prototipo a ser diseñado se lo realizará sobre la base del microcontrolador DS2250 del fabricante Dallas Semiconductors, el mismo que recibirá los datos por medio de un lector de tarjeta magnética y controlará una pantalla donde se mostrará la hora y otros datos requeridos en la aplicación.

Además, los datos almacenados en el prototipo, serán transferidos a un PC, donde se realizará el procesamiento, mediante programas en QBASIC y FOXPRO.

Para mayor facilidad de operación en el campo, la comunicación entre el prototipo y el computador se la realizará por medio de una interfaz serial de tipo *óptico*.

Temario.-

**CAPITULO I.- GENERALIDADES.**

- 1.1 EL TRANSPORTE PÚBLICO EN LA CIUDAD DE QUITO.
- 1.2 ANÁLISIS DE LAS ALTERNATIVAS EXISTENTES.
- 1.3 METODOLOGÍA ACTUAL.
- 1.4 SOLUCIÓN DEL PROBLEMA (CARACTERÍSTICAS).
- 1.5 CONCLUSIONES.

**CAPITULO II.- DISEÑO DE LOS CIRCUITOS.**

- 2.1 DIAGRAMA DE BLOQUES DEL PROTOTIPO.
- 2.2 DISEÑO DE CADA UNA DE LAS ETAPAS.
- 2.3 CONSIDERACIONES PARA LA CONSTRUCCIÓN Y MONTAJE DEL EQUIPO.

**CAPITULO III.- DESARROLLO DEL SOFTWARE.**

- 3.1 PROGRAMA PRINCIPAL DE CONTROL DEL EQUIPO.
- 3.2 SUBROUTINAS DEL PROGRAMA DE CONTROL.
- 3.3 PROGRAMA DE PROCESAMIENTO DEL PC.
- 3.4 SUBROUTINAS DEL PROGRAMA DE PROCESAMIENTO.

## **CAPITULO IV.- PRUEBAS, COMENTARIOS Y CONCLUSIONES.**

**4.1 PRUEBAS DE FUNCIONAMIENTO.**

**4.2 ANÁLISIS ECONÓMICO.**

**4.3 COMENTARIOS Y CONCLUSIONES.**

### **ANEXOS:**

- **BIBLIOGRAFÍA.**
- **MANUAL DE USO.**
- **LISTADOS DE LOS PROGRAMAS.**
- **HOJAS DE DATOS.**



## CAPITULO I.- GENERALIDADES.

### 1.1 EL TRANSPORTE PÚBLICO EN LA CIUDAD DE QUITO.

El problema que tienen las cooperativas y las compañías de transporte público tanto urbano como interparroquial, es la falta de cumplimiento en los tiempos de recorrido en los diferentes intervalos de las rutas asignadas. Este incumplimiento es la causa de que ciertos choferes vayan más despacio en los tramos de mayor afluencia de pasajeros en diferentes horarios, con lo que impide al bus que va atrás que tome pasajeros en forma normal; como resultado se provocan las carreras entre ellos, lo que a su vez es motivo de múltiples accidentes que causan no solo daños materiales sino también la pérdida de vidas humanas.

Al 1° de Junio de 1998, que son los datos disponibles en la Unidad de Planificación y Gestión de Transporte U.P.G.T., en el Distrito Metropolitano de Quito existen las siguientes empresas de transporte público, que ofrecen sus servicios en buses y busetas:

Cooperativas de Transporte Urbano:	27
Sociedades de Transporte Urbano:	35
Empresas Públicas de Transporte Urbano:	2
<b>Urbanas:</b>	<b>64</b>
Cooperativas Interparroquiales:	17
Sociedades Interparroquiales:	14
<b>Interparroquiales:</b>	<b>31</b>
Cooperativas Transporte Escolar:	5
Sociedades de Transporte Escolar	16
<b>Escolares:</b>	<b>21</b>
<b>Tótal</b>	<b>116</b>

Los datos obtenidos, llevan a concluir que existe una cantidad importante de compañías y unidades que brindan el servicio y que por lo tanto dado el grado del problema existente, que se adopte de manera urgente una solución por medio de un sistema de control de tiempos seguro y confiable, que sea suficientemente flexible, de tal forma que sea posible trasladarlo o reubicarlo con facilidad, que se adapte a las distintas rutas y condiciones particulares de control, en cada caso.

## **1.2 ANÁLISIS DE LAS ALTERNATIVAS EXISTENTES.**

Actualmente se están usando relojes marcadores de tarjeta de cartulina, ubicados en diferentes puntos de las rutas, los cuales se pueden identificar con siglas para que no exista la posibilidad de que una persona imprima todos los puntos del recorrido en un mismo reloj sin haber efectuado el recorrido correspondiente.

En cada tarjeta de cartulina, se registra el cálculo de los tiempos que le ha tomado a una unidad en particular el cumplir con una ruta determinada y entre los puntos definidos en la misma y en caso de existir diferencias con los tiempos establecidos, generalmente menores, se procede al cobro de una multa según una tabla acordada previamente.

Con este sistema de control se necesita una persona que controle todos los tiempos registrados comparándolos con la tabla establecida, pero el inconveniente surge cuando una ruta, por ejemplo, tenga 4 marcaciones por vuelta y 7 vueltas en el día y 40 unidades; con el ejemplo será necesario revisar y calcular los tiempos entre marcaciones y totales de 1120 impresiones. Si se toma en cuenta que para ese cálculo se requiere 30 segundos por cada uno (considerando que con la práctica lo hace muy rápido), esto significa que la persona encargada se demoraría en chequear alrededor de 9 horas ininterrumpidas y realizando cálculos. Por esta razón, en la mayoría de los casos estos cálculos se los hace por muestreo y por ser un proceso manual se producen errores voluntarios e involuntarios.

En el pasado, la Policía de Tránsito controlaba los tiempos anotando en una tarjeta la hora, pero este sistema salió de uso, ya que no se aceptaba que las empresas de transporte paguen por ese servicio, aduciendo que a la Policía ya le pagaba el estado.

### **1.3 METODOLOGÍA ACTUAL**

Actualmente se tiene en cada punto de control un equipo electromecánico o electrónico que marca, mediante una cinta entintada en una tarjeta de cartulina preimpresa, la fecha y hora en una posición que debe ser ubicada manualmente por el ayudante del chofer que es quien se baja a marcarla. En el siguiente punto debe realizar la misma operación en la siguiente posición, lo que a veces resulta en una sobremarcación al no tener la acuciosidad suficiente para la marcación y consecuentemente se producen discrepancias en la hora y pérdida del control de tiempos, con los consiguientes problemas entre el chofer y el recaudador.

Adicionalmente la ubicación física de los equipos es un inconveniente más, debido a que se deben conseguir locales en donde estén protegidos contra la intemperie y daños provocados por terceros, y principalmente que dispongan de energía eléctrica en la mayoría de casos.

Por lo anteriormente expuesto podemos indicar las siguientes ventajas del equipo electrónico propuesto sobre el sistema actual:

- Cada unidad debe usar una tarjeta diaria para cada ruta, lo que implica un costo permanente durante el año y todos los años. Con la tarjeta magnética este costo se lo hace una vez cada dos o tres años.

- La ubicación física del equipo electrónico es menos problemática, ya que el equipo puede operar conectado a la red pública de energía eléctrica o con una batería.

- Se eliminan las sobremarcaciones ya que el registro es electrónico y no hay lugar a confusiones y no depende del ayudante.

- La ventaja fundamental está en el cálculo de los tiempos y el cobro de las multas, ya que estos no dependen de la subjetividad de la persona que hace los cálculos, sino que por medio de un programa, todos los cálculos son precisamente realizados.

Por lo expuesto, las ventajas del equipo electrónico propuesto, permiten lograr una disminución de los problemas, al ser un método imparcial, no subjetivo, confiable y que permite que todos los cálculos se efectúen óptimamente.

#### **1.4 SOLUCIÓN DEL PROBLEMA (CARACTERÍSTICAS).**

Como se indicó, en la actualidad prácticamente la totalidad de las operaciones de adquisición de datos y el control son manuales, lo cual determina que no se realicen con mucha frecuencia, que la información para tomar decisiones sea muy escasa, que nunca se cuente a tiempo con la misma y que además constituye un sistema muy frágil donde se pueden introducir múltiples errores.

Para solucionar los problemas anotados, en especial de velocidad de procesamiento y evitar los errores en los cálculos, se ha pensado en un sistema que incluye un equipo electrónico digital de control que consta de los siguientes elementos:

- Terminales recolectores de datos a través de tarjeta de banda magnética, en sustitución de los equipos marcadores de tarjeta de cartulina. Al usarse tarjetas con banda magnética, se obtiene un ahorro sustancial en tarjetas de cartulina, ya que la tarjeta magnética puede durar por lo menos un año.

- Computador portátil para leer la información de todos los terminales recolectores de datos con un programa en QBASIC y posteriormente procesarlos con un programa en FOXPRO. Con esto se obtiene velocidad en el procesamiento de los datos almacenados, prácticamente la eliminación de errores y la disponibilidad oportuna de toda la información requerida.
  
- Comunicación entre equipo de control y PC con una interfaz óptica para facilidad en la recolección de información de todos los terminales y eliminación de posibles problemas por suciedad o desgaste por uso de los conectores si se usaran éstos.

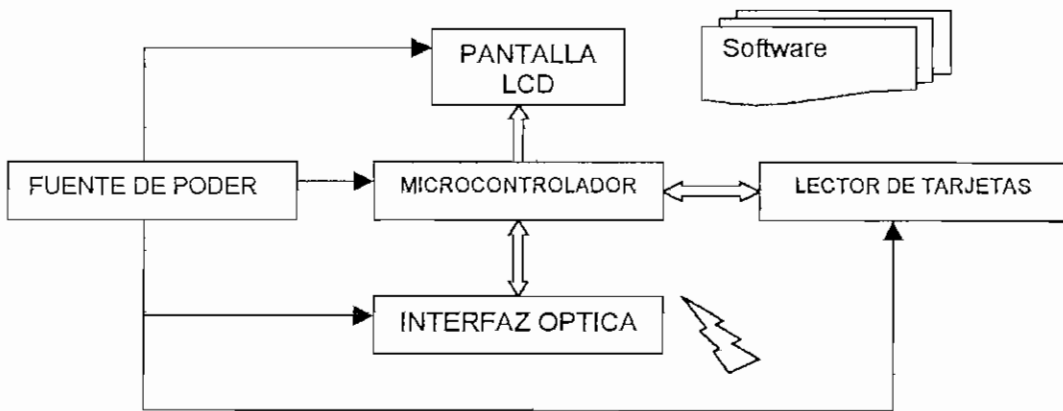
## **1.5 CONCLUSIONES.**

Sobre la base de lo expuesto se puede decir que el sistema propuesto tiene ventajas fácilmente visibles sobre la solución actualmente utilizada, tanto en la parte de velocidad de procesamiento de los datos registrados como en la confiabilidad de los resultados de los mismos y en disponer de la información en forma oportuna.

## CAPITULO II.- DISEÑO DE LOS CIRCUITOS.

### 2.1 DIAGRAMA DE BLOQUES DEL PROTOTIPO.

El equipo de control y registro, como se puede apreciar en la figura 2.1 donde se presenta su diagrama en bloques, funciona en base a un circuito microcontrolador, cuyo programa interno de control se lo ha desarrollado para cumplir con las opciones y especificaciones establecidas para el funcionamiento del equipo; además, la parte circuital está conformada por los siguientes bloques: fuente de poder, lector de tarjetas magnéticas, interfaz serial óptica y pantalla LCD.



**Figura 2.1** Diagrama en bloques del equipo de control

La concepción del equipo y su diseño se la ha realizado, en la perspectiva de conseguir un circuito lo más simple posible al fin de que su costo sea razonable y posibilite su aplicación práctica.

## 2.2 DISEÑO DE CADA UNA DE LAS ETAPAS.

La explicación relativa al diseño se la ha orientado principalmente a la interfaz serial infrarroja, en vista de que la fuente de poder, la pantalla LCD y el lector de tarjetas, que son las demás etapas que conforman el sistema, son dispositivos comerciales ya desarrollados y que se los utiliza para acoplarlos o integrarlos dentro del diseño, y en las hojas de datos y referencias se dispone de amplia información para su uso, por lo que la explicación que aquí se presenta no es en todo detalle.

### 2.2.1 PANTALLA LCD.

La pantalla de LCD se usa para mostrar principalmente la fecha y hora actual que posee el equipo, información que se graba internamente en la memoria del mismo junto con el código leído de la tarjeta magnética para la

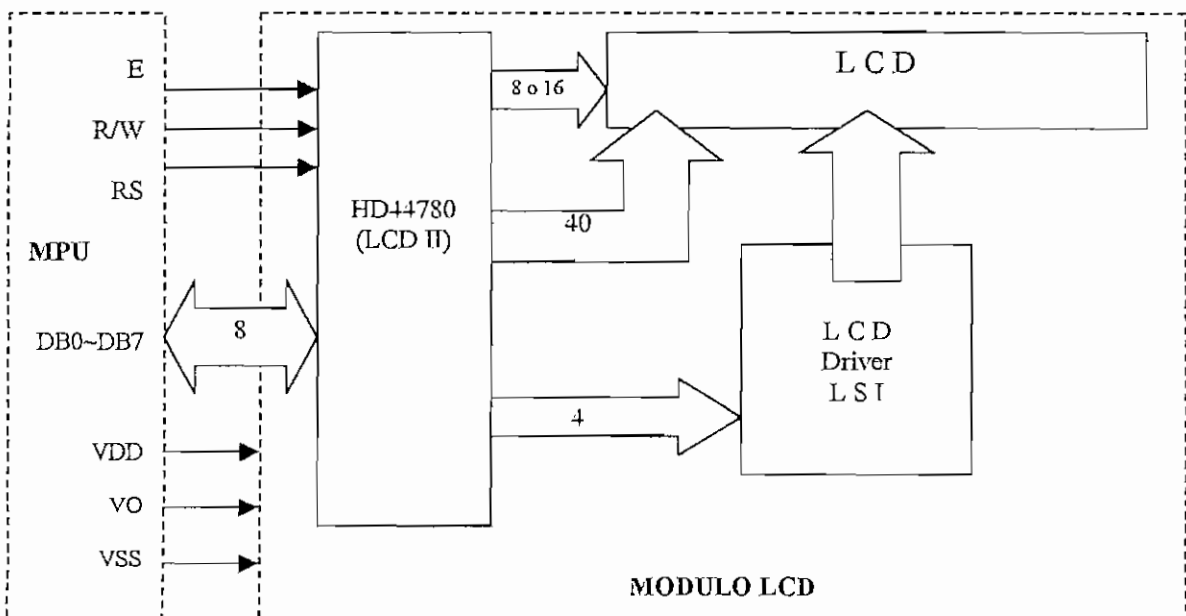


Figura 2.2 Diagrama de bloques del display Hitachi con controlador HD44780.

creación de una transacción que posteriormente se envía al computador de proceso. Adicionalmente se usa la pantalla para indicar el código de la tarjeta leída y si es válida o no. Para la aplicación, se ha establecido como suficiente una pantalla de dos líneas por 16 caracteres. El display tiene un controlador LSI HD44780 que es el que maneja directamente el display y se comunica con el microcontrolador DS2250T del equipo, como se muestra en la figura 2.2

Con el propósito de tener una fácil operación y un mejor aprovechamiento del equipo, en la pantalla se tienen los siguientes mensajes:

MENSAJE 1 (Fecha y Hora)

00/06/25 13:25

Nombre de estación

MENSAJE 2 (Número de tarjeta)

00/06/25 13:25

TARJETA CORRECTA

MENSAJE 3

00/06/25 13:25

DATOS LEIDOS

### **Inicialización por circuito de reposición interno.**

El controlador LSI HD44780 dispone del circuito que propicia la operación de inicialización de los circuitos internos. La condición de reposición se cumple cuando la fuente de polarización alcanza el valor mínimo de 4.5 VDC; en la subida al valor nominal de trabajo; 10 ms más tarde la bandera de ocupado (BF) se pone



en 1L y se mantiene en esa condición hasta que termine la inicialización de todos los circuitos internos.

Los siguientes pasos se cumplen durante la etapa de inicialización con los valores que se indican y que corresponden a los que el circuito pone por defecto, para esto se usa el registro de datos DD (Display Data Ram DB0~DB7) y los pines de control RS y RW en el I/O Buffer del display.

1) Borrar display

2) Establecimiento de función

DL = 1 : interface de datos de 8 bits (DB4)

N = 0 : 1 línea de display (DB3)

F = 0 : fuente de caracteres de matriz 5x7 (DB2)

3) Control de display ON/OFF

D = 0 : display OFF (DB2)

C = 0 : cursor OFF (DB1)

B = 0 : titileo OFF (DB0)

4) Armado del modo de entrada, donde se especifica si el cursor se incrementa (mueve a la derecha) o decrementa (mueve a la izquierda) y si se acompaña con el movimiento del display.

I/D = 1 : +1 (incremento) (DB1)

S = 0 : no desplazamiento (DB0)

Cuando el tiempo de subida del voltaje de la fuente desde 0,2V a 4,5 V entre 0,1 s a 10ms no se cumple, o cuando el ancho del pulso de nivel bajo

(power OFF) es menor a 1ms, el circuito de reset interno no operará normalmente. Por esa razón se usa la **Inicialización por Instrucciones**, para tener la seguridad de que el display se reinicializa como se quiere.

### **Inicialización por instrucciones.**

Como una alternativa para la inicialización de los circuitos de control del display y cuando no se deseen los valores que por defecto toma el display con el reset automático o en el caso de que no se haya dado de manera satisfactoria el reset automático, se puede proceder a inicializar por medio de instrucciones, tomando en cuenta que se utiliza la interface de 8 bits; como se indicó el display es de 2 líneas por 16 caracteres, se usa una matriz de carácter de 5x7 puntos; no se usa el cursor ni el titileo; el procedimiento seguido, es el que indica a continuación, en que se refieren todos los pines al I/O Buffer del display que contiene DB0~DB7, RS y R/W:

- 1) Encendido
- 2) Espera mayor a 15 ms hasta que Vcc alcance por lo menos 4.5V
- 3) No se puede chequear la bandera BF antes de esta instrucción; la bandera BF es de ocupado (Busy Flag).

Se define el funcionamiento a 8 bits poniendo en el bit 4 DB4 del registro de datos un 1L, que corresponde al bit DL

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	1(DL)	*	*	*	*

- 4) Retardo mayor a 4.1 ms

5) No se puede chequear BF antes de esta instrucción

Se repite la condición de funcionamiento a 8 bits

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	1(DL)	*	*	*	*

6) Retardo mayor a 100  $\mu$ s

7) No se puede chequear BF antes de esta instrucción

Repite el seteo de funcionamiento a 8 bits

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	1(DL)	*	*	*	*

La bandera de ocupado BF puede ser chequeada después de las instrucciones que siguen. Cuando no se chequea BF, el tiempo de espera entre instrucciones es más largo que el tiempo de ejecución de instrucción.

8) Pone el modo de trabajo a 8 bits, 2 filas y matriz 5x7, para todas las transferencias y transacciones posteriores.

En el registro de datos DD (Display Data Ram), para la configuración general de funcionamiento se debe poner: DL =1L (DB4), configurando el modo de trabajo a 8 bits, DB3 =1L para usar 2 filas de trabajo (N = 1) y DB2=0L para usar caracteres correspondientes a una matriz de 5x7 puntos (F=0). Con RS Y R/W a 0L.

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	1(DL)	1(N)	0(F)	*	*

9) Para apagar el display con los datos retenidos en la memoria RAM integrada en el circuito de display, se debe poner en el registro DD (Data Display Ram) antes referido lo siguiente: DB3=1L y todos los demás en 0L:

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	0	0	0

10) Toda la memoria RAM de datos del display se llena con el número 20H, el contador de direcciones se pone en cero y el cursor vuelve a la primera posición (totalmente a la izquierda en la primera fila, dirección=0).

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

Es importante anotar que, después de esta programación no se puede modificar el formato del display ni el número de líneas del mismo, excepto si se vuelve a programar.

11) Armado del modo de entrada, se incrementa la dirección de entrada con DB1=1L (I/D=1) y el display no se desplaza con DB0=0L (S=0) cuando se escribe un dato a la memoria RAM generadora de caracteres, DB2 debe estar en 1L y los demás registros en 0L.

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	1(I/D)	0(S)

Por otra parte, para que el circuito de control reconozca que la información del registro DD (Display Data Ram) se ha enviado como una dirección, la bandera RS que se controla externamente y está en el buffer I/O del display, debe ponerse en 1L y para que reconozca el display esa información como un dato, dicha bandera debe ponerse a 0L.

### **Contraste**

El contraste es la menor o mayor dificultad que se tiene para leer el display en base a las diferentes condiciones de iluminación, ángulo de visión y temperatura, tomando en cuenta el fondo que va a tener el display y la diferencia (contraste) con los segmentos activados.

El contraste se lo puede variar para las diferentes condiciones de iluminación y temperatura ambientales, en caso de ser usados los equipos en el interior, pero se ha llegado a un compromiso de funcionamiento y en vez de dejarlo variable, usando el potenciómetro CONT de 10K $\Omega$ , se lo ha puesto fijo en base a experimentación.

### **2.2.2 LECTOR DE TARJETAS MAGNÉTICAS.**

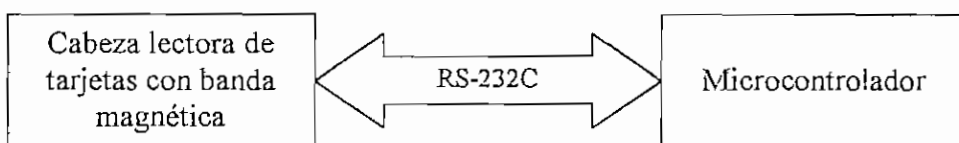
Para capturar los datos del código de identificación de las unidades de transporte hay varias alternativas dentro de las cuales se podría tener: lectores de tarjetas con banda magnética en las pistas I, II y III, lectores de tarjetas con códigos de barras, lectores de tarjetas de proximidad y sistemas de identificación de vehículos a distancia.

Se descartaron las tarjetas de banda magnética en pistas I y III por la dificultad de grabar localmente en pista III y la poca capacidad de grabación de datos en la pista I.

Se decidió no utilizar las tarjetas de código de barras por ser fácilmente duplicables con fotocopiadora y el costo de la cabeza lectora es mayor que la cabeza lectora de tarjetas de banda magnética.

En cuanto a los lectores de tarjeta de proximidad y los sistemas de identificación a distancia de vehículos si bien representan una opción más confiable, se decidió no utilizarlas por el alto costo actual de implementación, que no permitiría la aplicación del proyecto en la práctica.

En base a lo señalado, se decidió usar un lector de tarjetas de banda magnética de pista II, de los más simples que se encuentran en el mercado, por ser los más económicos, se dispone de mayor apoyo en la confección de las tarjetas y por ser de uso más extendido. Las tarjetas son codificadas solamente con números que es lo que permite la pista II.



**Figura 2.3** Comunicación entre la cabeza lectora de tarjetas con banda magnética y el microcontrolador con interface serial RS-232C.

La comunicación entre el lector y el microprocesador se realiza con una interfaz serial, la cual viene implementada en los dos elementos, ver Figura 2.3.

### 2.2.3 MICROCONTROLADOR.

Se decidió realizar el diseño en base a un microcontrolador y de las múltiples posibilidades existentes al momento, se resolvió utilizar el microcontrolador DS2250T del fabricante Dallas Semiconductor; entre otras por las siguientes razones principales:

- El conjunto de instrucciones de este microcontrolador es prácticamente el mismo que el de la familia MCS-51 de Intel, que son los circuitos de más amplia difusión en el mercado y que además es el que se estudia en la Facultad de Ingeniería Eléctrica de la E.P.N.
- Por otra parte, en la aplicación del presente trabajo es muy importante que el registro de los datos de los recorridos del transporte público tenga como referencia la fecha y hora; para lograr esto el microcontrolador seleccionado posee como parte de su circuitería un reloj en tiempo real.
- Dentro del proceso a realizar se debe almacenar la información, la misma que, frente a una pérdida del suministro de energía al circuito, debe permanecer inalterada, para su posterior descarga al computador de proceso; para este requisito el microcontrolador DS2250T es ideal, ya que posee internamente memoria NVRAM, la cual garantiza que los datos permanezcan almacenados si existe un corte de energía.
- Los espacios de memoria de programa y datos se los establece con toda libertad, de acuerdo a las necesidades.
- La programación de este microcontrolador se la realiza desde el computador por el puerto serial, sin necesidad de dispositivos especiales de programación.
- Posee un puerto serial que se lo puede programar para comunicación a diferentes velocidades. Esta característica se la usa para la comunicación con el

lector de tarjetas, interfaz óptica para la comunicación con el computador de proceso; así como también para la programación del microcontrolador.

- Por otra parte, a través de sus puertos paralelos, se puede manejar directamente la pantalla LCD de 16 caracteres y 2 líneas que se usará para los mensajes e interacción con los operadores del equipo.

#### **2.2.4 FUENTE DE PODER.**

En vista de las ventajas con que se cuenta en la actualidad y por no ser el propósito fundamental del desarrollo del prototipo, se usa una fuente de poder de las comerciales del tipo “switching” por sus características, entre otras, resisten amplias variaciones de voltaje así como de frecuencia en la red, son menos susceptibles a ruidos, son mucho más estables que las lineales, la eficiencia es mayor (mejor relación de potencia consumida a utilizada), son más compactas, han sido bien probadas y por tanto son muy confiables.

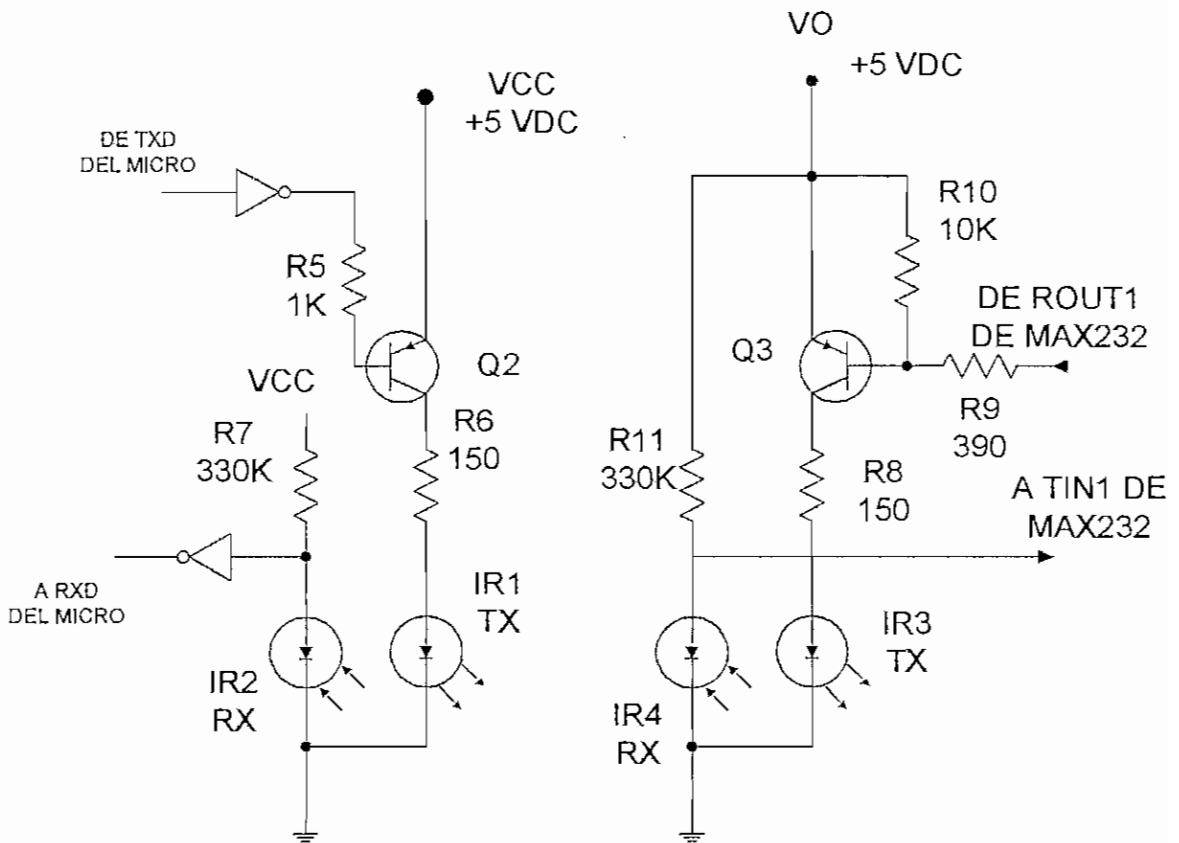
#### **2.2.5 INTERFAZ INFRARROJA CON EL PC.**

Esta interfaz se la ha implantado con el objetivo de no tener conectores que puedan deteriorarse con el uso continuo y diario de la misma, dada la aplicación del equipo.

Cabe destacar que la interfaz infrarroja no presenta ningún problema por interferencias de otras fuentes luminosas como luz natural o artificial como focos incandescentes o fluorescentes, en vista de que hay un adecuado acoplamiento entre transmisor y receptor. Adicionalmente al trabajar en el rango de 940 nm se logra una mejor eficiencia entre la energía eléctrica aplicada a la energía luminosa que se emite, un poco menor que los diodos de 850 nm pero con un menor precio.



Está implementada para una distancia máxima de 2 cm entre la ventana del recolector de datos y el dispositivo de acoplamiento con el PC; no se consideró necesario una mayor distancia para tener el menor consumo de energía posible y lograr que el circuito de acoplamiento al computador de proceso consuma lo mínimo y la duración de la batería, usada en el mismo, sea la mayor posible. Se usa una batería de 9 VDC cuyo voltaje se lo regula a los 5 VDC ( $V_o$ ) que son los necesarios para los circuitos digitales como se muestra en la figura 2.5.



**Figura 2.4 .- Interface infrarroja en la parte correspondiente al recolector de datos y al cable que va conectado al computador de proceso.**

### 2.2.5.1

### Diseño de la interfaz infrarroja.

La interfaz infrarroja está diseñada con dos pares de elementos: diodo emisor de luz infrarroja y fotorreceptor de luz infrarroja. El un par está situado a la salida de la interfaz serial del prototipo (IR1, IR2) y el otro en el cable de conexión

al puerto serial del computador de proceso (IR3, IR4), como se ilustra en la figura 2.4.

En el prototipo se maneja el diodo emisor de luz IR1 a través de la salida de transmisión de la interfaz serial del microcontrolador TXD y con un buffer para disponer de la corriente necesaria para que el diodo pueda emitir una cantidad de luz suficiente de tal forma que pueda ser recibida por el fotorreceptor IR4 que se encuentra en el cable que viene del computador de proceso.

Si se toma en cuenta que se tiene un voltaje VCC de 5 voltios, una caída de voltaje en el diodo IR1 de alrededor de 1,3 voltios y 0,6 voltios de caída en la juntura base-colector del transistor Q2, se puede determinar el valor de la resistencia R6, lo que da  $150 \Omega$  para obtener una corriente de unos 20 mA en el diodo emisor IR1, que es una corriente cuatro veces menor que la corriente máxima del diodo, con el objeto de que el consumo en el mismo no sea muy grande y ahorrar energía, especialmente en el lado del cable que usa batería. Con un razonamiento similar se calcula la resistencia R8 para la corriente en el diodo IR3 del acople que va en el cable de conexión al PC. Adicionalmente la disipación de potencia máxima que se tiene en los dos diodos es de  $20 \text{ mA} \times 1,6 \text{ voltios} = 32 \text{ mW}$  que está por debajo de la máxima disipación de 100 mW especificada por el fabricante.

En la entrada serial del prototipo se tiene el fotorreceptor IR2 que recibe la señal del fotoemisor IR3 que se encuentra en la parte móvil del prototipo en el cable que se conecta al PC, y esta señal se la pasa a través de una compuerta schmitt trigger, implementada con un integrado 74HC14 cuya entrada es del tipo schmitt trigger, para poder obtener una onda cuadrada sin ruido en su salida, que podría provocar un mal funcionamiento, toda vez que la señal va a la entrada de recepción de la interfaz serial RXD del microcontrolador.

Para los fotodiodos IR2 e IR4, que son los receptores, se tiene una corriente en oscuridad de 3 nA y si se hace que el voltaje que cae en la resistencia R7 sea pequeño menor que 1 voltio, se tiene la salida hacia el pin

RXD del microcontrolador (tomada del ánodo del fotodiodo IR2) en 1L, esta resistencia R7 es alrededor de 330 K $\Omega$ . El momento que el IR2 recibe luz, la corriente sube hasta unos 20 mA, con lo cual la caída de voltaje en el mismo es 0,3 voltios lo cual provoca que la salida hacia el pin RXD del microcontrolador (ánodo del fotodiodo IR2) sea 0L.

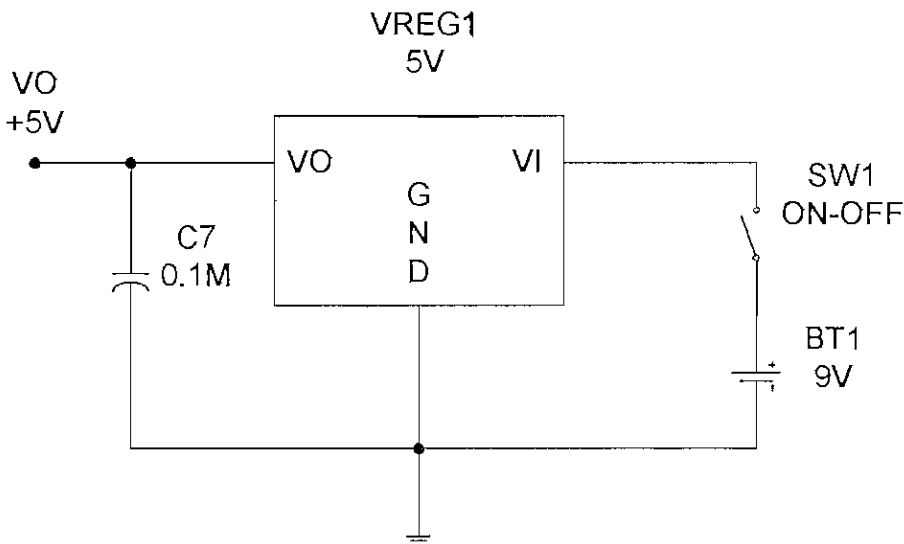


Figura 2.5 .- Alimentación usada en la parte de la interface infrarroja que se conecta al computador de proceso. Se usa una batería de 9 voltios y un regulador a +5V.

Del mismo modo se tiene el par óptico en el cable de conexión al puerto serial del PC, pero en este caso se cuenta con una batería de 9 VDC, con el propósito de poder polarizar y conseguir un funcionamiento adecuado de estos dispositivos como se ilustra en la figura 2.5.

En la figura 2.5 se muestra una batería de 9VDC, que es la que alimenta la interface serial que se conecta al computador de proceso. Se tiene un interruptor para no descargar la batería cuando la interface no está en uso. Se usa adicionalmente un regulador para mantener los +5VDC necesarios para polarizar el integrado MAX232 que es el que maneja los voltajes necesarios entre el computador de proceso y el par infrarrojo de la interface.

## 2.2.5.2

## Diseño de los circuitos de acoplamiento.

Debido a que el microcontrolador posee un solo puerto para comunicación serial, se usan circuitos lógicos con compuertas NAND e INVERSORES para poder discriminar si la comunicación serial se la va a realizar con el computador de proceso o con el lector de tarjetas magnéticas, se implementa la circuitería según la siguiente tabla:

Salida T del lector	Salida pin10 de compuerta U2C, entrada a RXD de micro
0	Datos del PC
1	Datos de lector de tarjetas

En base a la tabla anterior, se implementa el circuito que se muestra en la figura 2.6, en la que se usan cuatro compuertas **NOR** y se tienen como entradas el pin de control P1.0 del microcontrolador, la señal de datos proveniente del lector de tarjetas y la señal de datos provenientes del PC de proceso; y como salida los datos que se reciben en el pin RXD del microcontrolador.

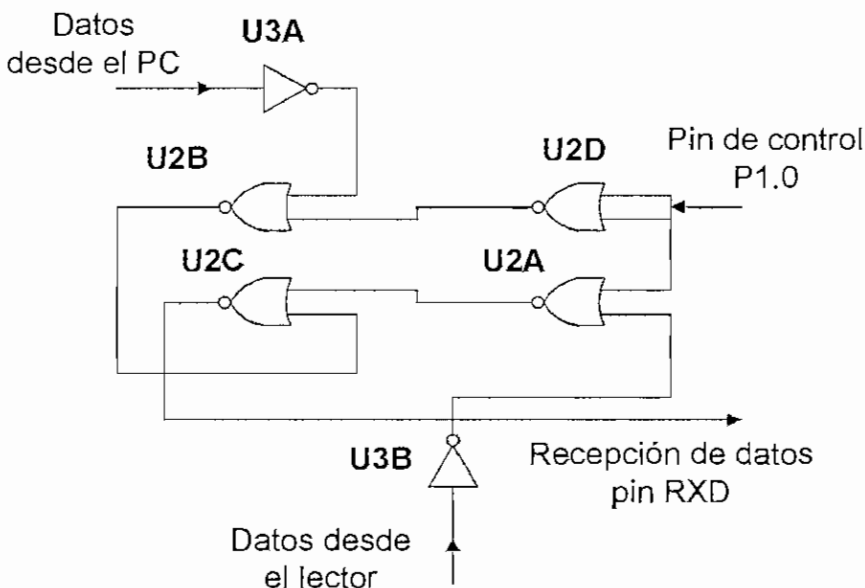


Figura 2.6 .- Disposición de las compuertas NOR para discriminar el origen de los datos a ser procesados.

Se diseñó el programa del microcontrolador para tener siempre, como predeterminada, la comunicación serial, a través de la interfaz infrarroja, con el computador de proceso y cambiar, si existe una interrupción, a comunicación serial del microcontrolador con el lector de tarjetas.

De acuerdo con lo señalado, el computador de proceso va a comunicarse con el microcontrolador, debido a que se necesita leer o escribir datos del o al equipo, por lo que el microcontrolador va a trabajar como se explica en los siguientes incisos y haciendo referencia a la Figura 2.6.

Como se indicó anteriormente se tiene como predeterminada la comunicación serial del microcontrolador con el computador de proceso, para esto se mantiene el pin **P1.0** en 1L. Este 1L aplicado a la compuerta **NOR U2D**, que actúa como inversor, produce a su salida un 0L, el cual a su vez habilita la compuerta **NOR U2B** para que permita el paso de los datos que llegan de la interfaz infrarroja a través del inversor **U3A**. Adicionalmente el 1L del pin P1.0 aplicado a la compuerta **NOR U2A** bloquea el paso de información que podría provenir del lector de tarjetas a través del inversor **U3B** y fuerza que a la salida de **NOR U2A** se tenga un 0L que habilita a la compuerta **NOR U2C** para permitir el paso de datos desde el computador de proceso que pasaron por **NOR U2B** y puedan llegar al pin **RXD** del microcontrolador. De acuerdo con la lógica planteada y el circuito diseñado, queda establecida la comunicación con el computador de proceso y deshabilitada la comunicación con el lector de tarjetas.

Si suponemos ahora que el lector de tarjetas es el que va a comunicarse con el microcontrolador, debido a que se ha introducido una tarjeta, el lector de tarjetas envía una señal de "tarjeta presente" (salida T del lector de tarjetas con banda magnética) que provoca una señal de interrupción en el pin **INT1** del microcontrolador, que está programado para que con esa interrupción el pin **P1.0** se ponga en 0L. Este 0L habilita la compuerta **NOR U2A** para que pasen los datos del código de la tarjeta que vienen de la salida L del lector de tarjetas con banda magnética a través del inversor **U3B**. La salida 0L del pin **P1.0** también va a la compuerta **NOR U2D**, usada como inversor, que da como resultado un 1L y

esto inhabilita el paso de datos por la compuerta **NOR U2B** que vienen de la interface infrarroja por la compuerta inversora **U3A** y la salida de **NOR U2B** se fuerza a 0L que a su vez habilita la compuerta **NOR U2C** para que continúe el paso de datos que llegan desde el lector de tarjetas a través del inversor **U3B** y la compuerta **NOR U2A** hacia el pin **RXD** de la comunicación serial del microcontrolador. Como puede establecerse, en esta condición de trabajo del circuito, se permite la comunicación con el lector de tarjetas y se inhabilita la comunicación con el computador de proceso.

### **2.3 CONSIDERACIONES PARA LA CONSTRUCCIÓN Y MONTAJE DEL EQUIPO.**

Para la construcción y montaje del equipo, se han tomado en consideración las distintas variables que van a influir tanto en su instalación, operación, así como aspectos muy determinantes para que el equipo tenga una amplia aceptación y pueda ser utilizado en la práctica; esas condiciones son las siguientes:

1. Ambiente de operación.- Este equipo debe operar en ambientes muy contaminados por polvo y el humo del escape de los vehículos, ya que su instalación va a ser en lugares muy cercanos a calles y carreteras, por lo que se ha considerado que debe estar en una caja bien cerrada para evitar el ingreso de estos contaminantes externos que podrían alterar su funcionamiento.

2. Calidad de energía.- La energía de alimentación corresponde a la red pública de la ciudad que es de 110 voltios, 60 Hz., pero debe considerarse la posibilidad de cortes de la misma, por lo que se ha previsto que el equipo pueda mantener sus programas y datos durante esos eventos, y como se señaló, es la razón para haber escogido un microprocesador que dispone de respaldo de baterías de litio como energía de emergencia.

3. Los usuarios.- Como las personas que van a operar los equipos, en su mayoría, no son idóneos, se ha diseñado el equipo, pensando en que tiene que haber la menor intervención posible y que a su vez cumpla el propósito de su aplicación; esto es, en el caso del registro de las unidades de transporte solo tiene que pasar la tarjeta por el lector y en el caso de la descarga de la información y su procesamiento en un PC, si se tendrá una persona suficiente entrenada.

4. El costo.- A la hora de conseguir que efectivamente el equipo sea aceptado y utilizado, un factor determinante es su costo, ya que va a tener que competir con los sistemas de control que se utilizan en la actualidad, cuyo costo es relativamente bajo, aunque menos eficientes; por lo señalado el diseño fue pensado de tal manera que se utilice el menor número de componentes, tan solo los indispensables para una operación que cumpla con lo requerido y por lo tanto que su costo sea el menor posible.

5. El montaje.- Para el montaje debe considerarse que pueden existir dos posibilidades: montaje fijo o movable (transportable), lo cual está directamente relacionado con la forma de trabajo que para el control realizan las distintas cooperativas de transporte; en todo caso, en cualquiera de las dos posibilidades, se debe tomar en cuenta la seguridad de la instalación, principalmente, en cuanto a que no pueda ser removido fácilmente para evitar robos.

6. La caja.- La caja o chasis debe tener una inclinación, gracias a la cual tanto leer la pantalla como el pasar la tarjeta magnética sea fácil y no forzado; por otro lado el acabado es importante a fin de que no sufra un deterioro por estar expuesto a un medio altamente contaminado y es la razón de haber decidido que el chasis sea metálico, con acabado de pintura electrostática.

### **CAPITULO III.- DESARROLLO DEL SOFTWARE.**

La parte circuital se completa con los programas requeridos para hacer aprovechable la misma; es así como se dispone para la comunicación entre el microcontrolador y el PC, de un programa desarrollado en lenguaje QuickBasic, el cual es el encargado de enviar las instrucciones al microcontrolador para conseguir igualar al reloj, leer transacciones, etc., y cuyo objetivo principal es crear un archivo en ASCII con la información que ha sido descargada desde el equipo de registro.

La información recuperada y que se encuentra en ASCII, se la procesa con un programa desarrollado en FoxPro, para el cálculo y la generación de reportes desde el computador de gestión.

Por otra parte, existe el programa de control del equipo en el lenguaje assembler que es el que el microprocesador DS2250T reconoce y procesa los datos por el recibidos.

Por lo tanto los programas del sistema están conformados por dos partes fundamentales a saber: el programa de control de funcionamiento del equipo de registro que es el que supervisa todas las acciones que ejecuta el microcontrolador y los circuitos adicionales y el programa de gestión de los datos almacenados que se lo ejecuta en el computador.

En cuanto al programa de control para el equipo de registro, en los anexos se presenta el listado del mismo y en los siguientes subcapítulos una breve explicación a través de diagramas de flujo:



### **3.1 DISEÑO DEL PROGRAMA DEL MICROCONTROLADOR.**

Para una adecuada comprensión de los diagramas de flujo de los programas mencionados, en la figura 3.1 se presenta la simbología con la que se ha desarrollado dichos diagramas.

## Convención de símbolos

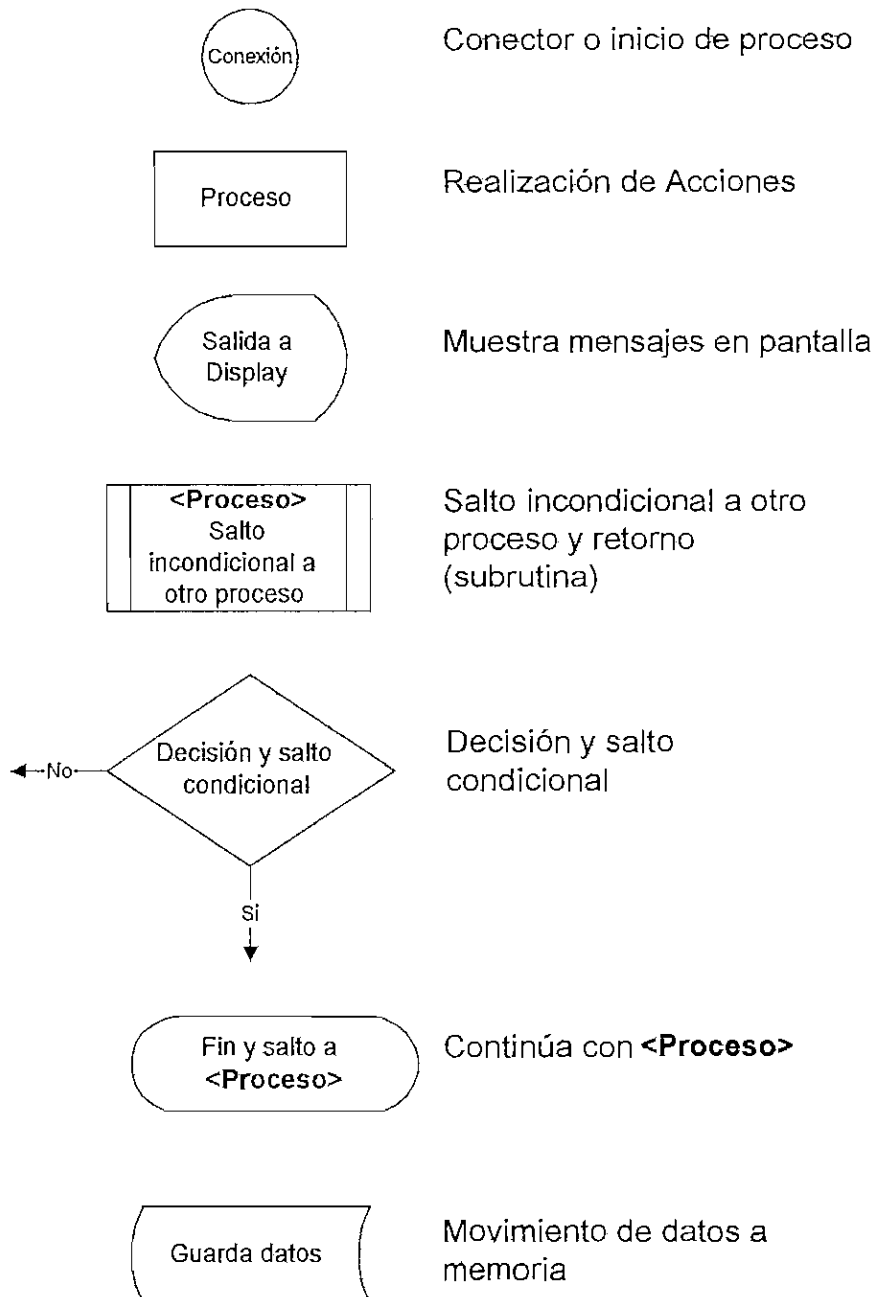


Figura 3.1.- Significado de los Símbolos usados en los diagramas de flujo.

## INICIO DEL PROGRAMA AL ENCENDER EL EQUIPO

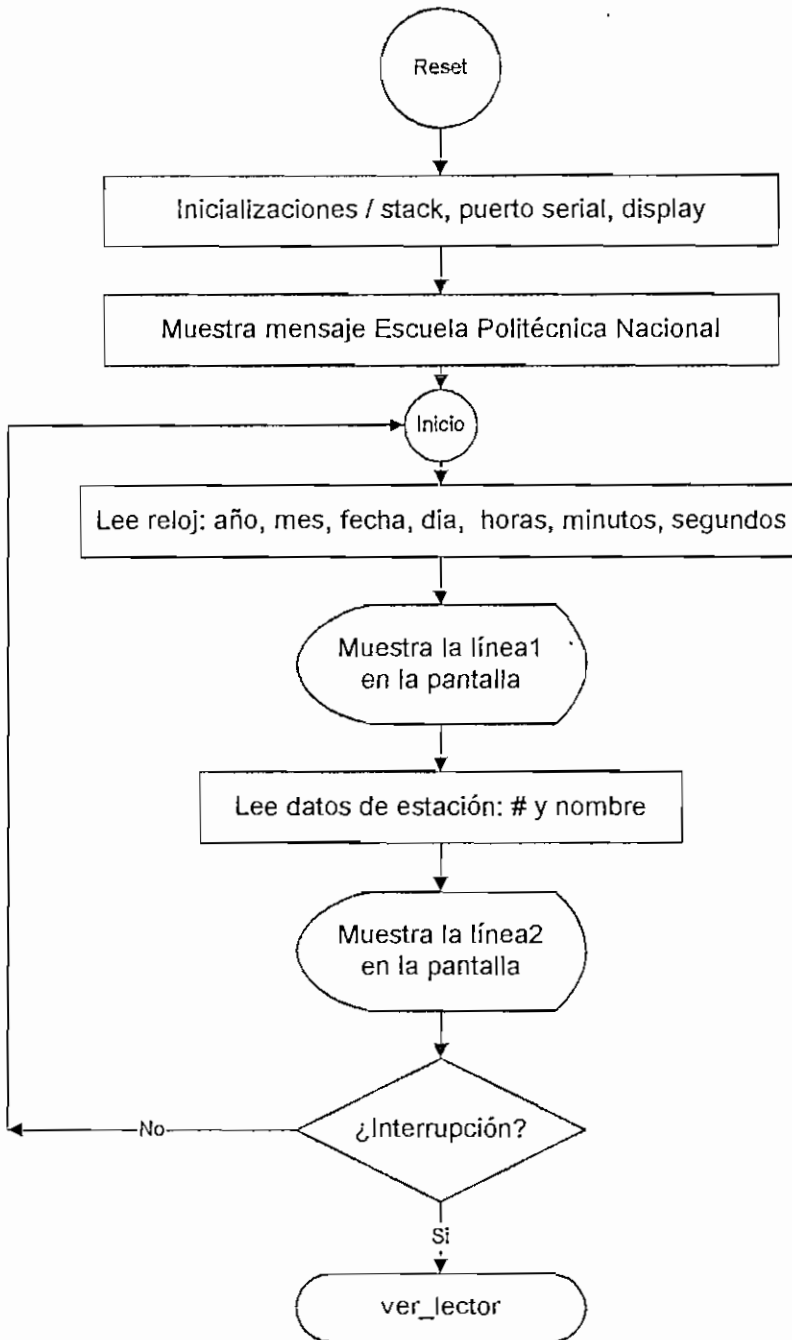


Figura 3.2.- Programa principal

En la Figura 3.2 se presenta el diagrama de flujo del programa principal, con el cual se inicializan los registros internos y se define el modo de funcionamiento general del circuito, a continuación se muestran los mensajes de inicio y espera hasta que exista una interrupción y va a rutina **ver\_lector** o **serial**.

### Rutina ver\_lector

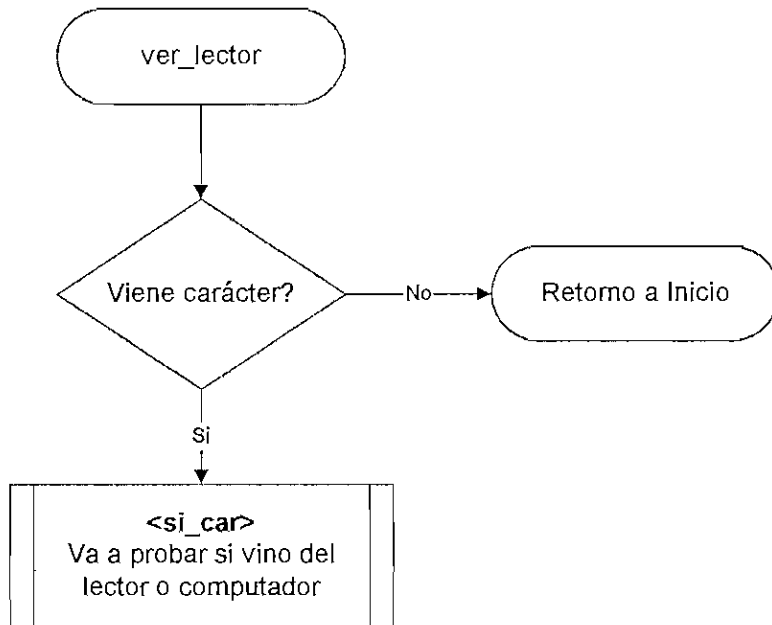


Figura 3.3.- Subrutina `ver_lector`

En la Figura 3.3 se puede observar el diagrama de flujo de la subrutina que chequea que realmente venga un carácter de la lectora de tarjetas y no del PC. Si viene un carácter, va a rutina `si_car`, caso contrario retorna al programa principal en Inicio.

Rutina <si\_car> verifica de donde vino  
la interrupción, PC o lector

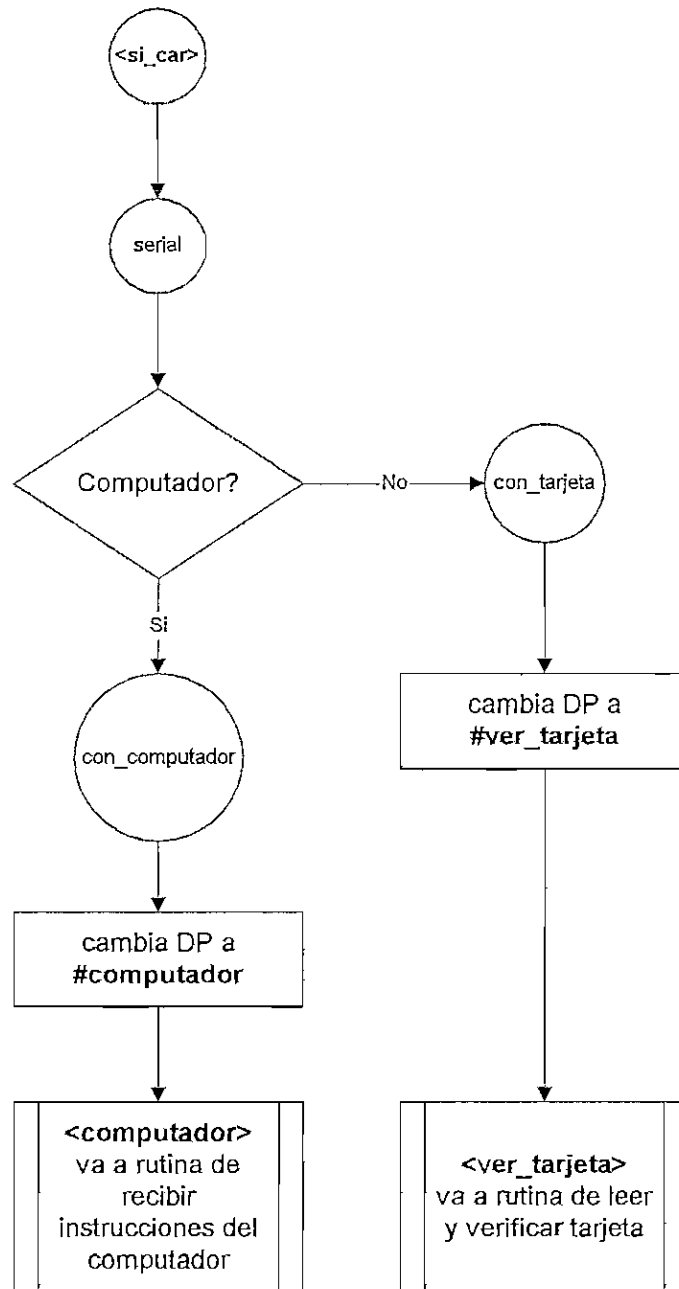


Figura 3.4.- Rutina <si\_car>

La Figura 3.4 muestra la rutina que chequea si el carácter vino desde la lectora o el computador y de acuerdo a eso, dirige el programa hacia la rutina **ver\_tarjeta** o **computador**.

### Rutina <ver\_tarjeta> procesa los caracteres leídos por el lector

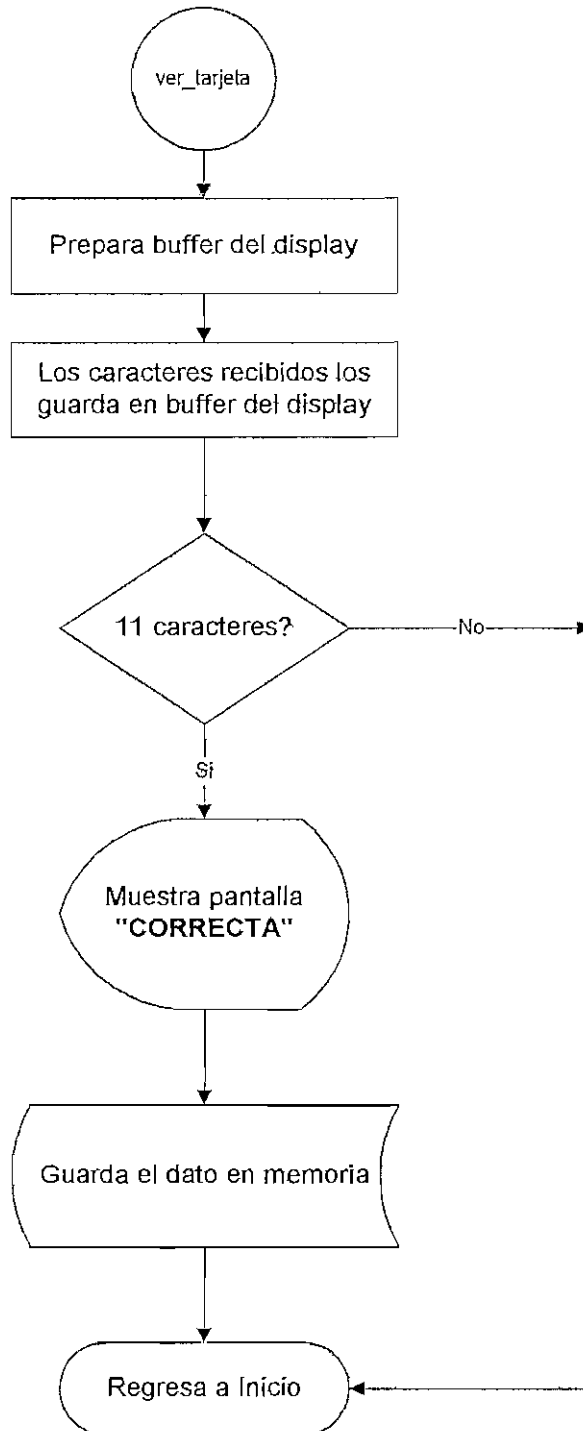


Figura 3.5 .- Rutina <ver\_tarjeta>

En la Figura 3.5 se presenta el diagrama de la rutina que toma los datos del lector y se verifica que sean 11 dígitos y se da los mensajes que correspondan. Si son válidos, se los guarda. Al final se regresa al programa principal Inicio.

**Rutina <computador> recibe los  
comandos del computador**

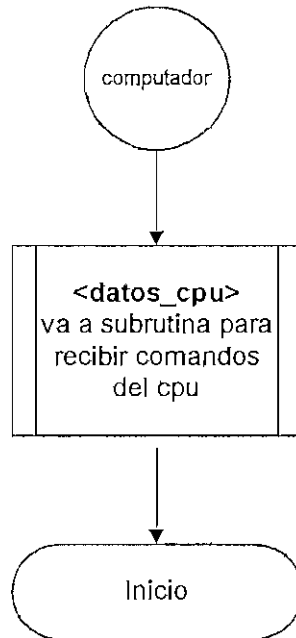


Figura 3.6 .- Rutina <computador>

En la Figura 3.6 se presenta el diagrama de flujo correspondiente a la rutina <computador> que dirige el proceso a la subrutina <datos\_cpu> para recibir datos e instrucciones desde el computador. Una vez realizado esto, regresa al programa principal **Inicio**.

## Subrutina &lt;datos\_cpu&gt;

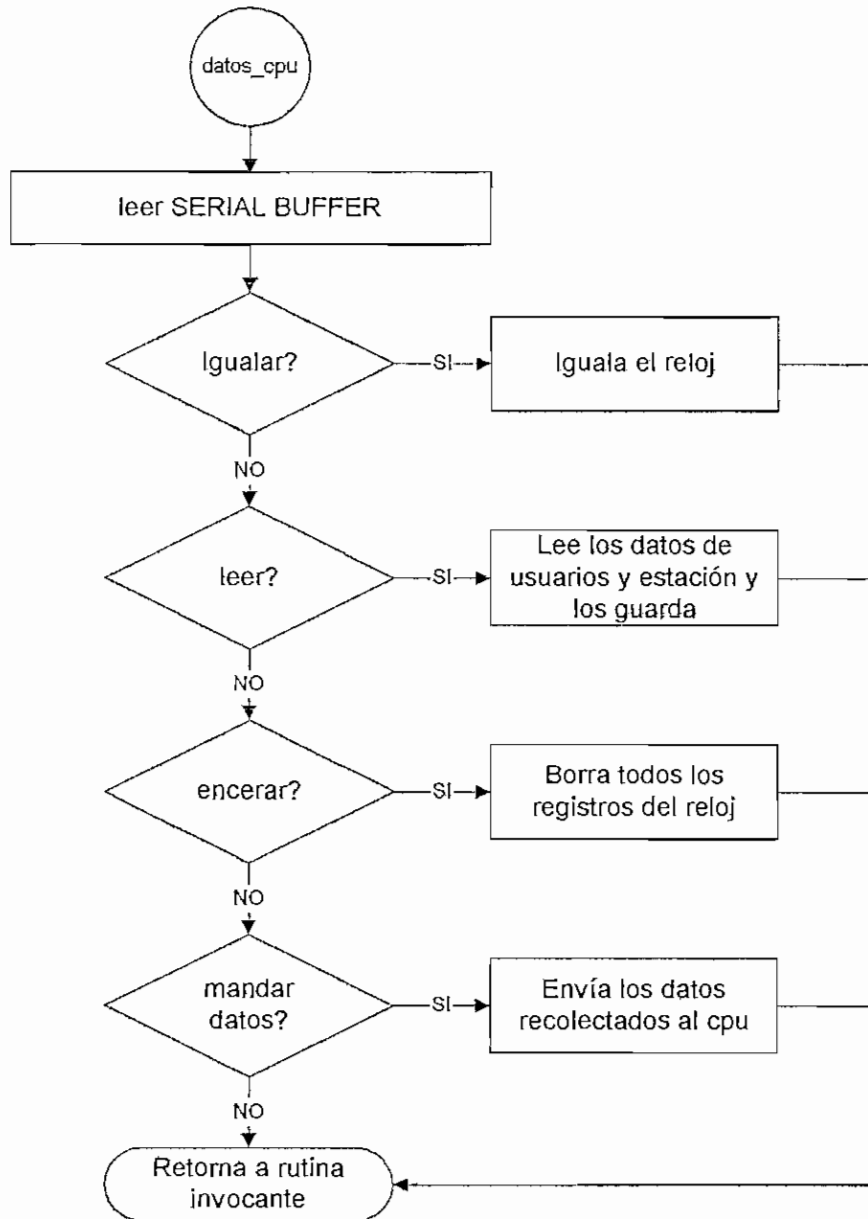


Figura 3.7 .- Subrutina &lt;datos\_cpu&gt;

En la Figura 3.7 se presente el diagrama de flujo de la subrutina <datos\_cpu> que recibe los datos del buffer serial con instrucciones del PC y decide que acción tomar, luego de lo cual retorna al punto de donde fue invocada esta subrutina.



**Subrutina <init\_display> inicializa el display  
según procedimiento del fabricante**

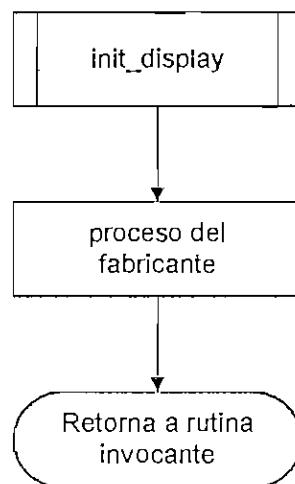
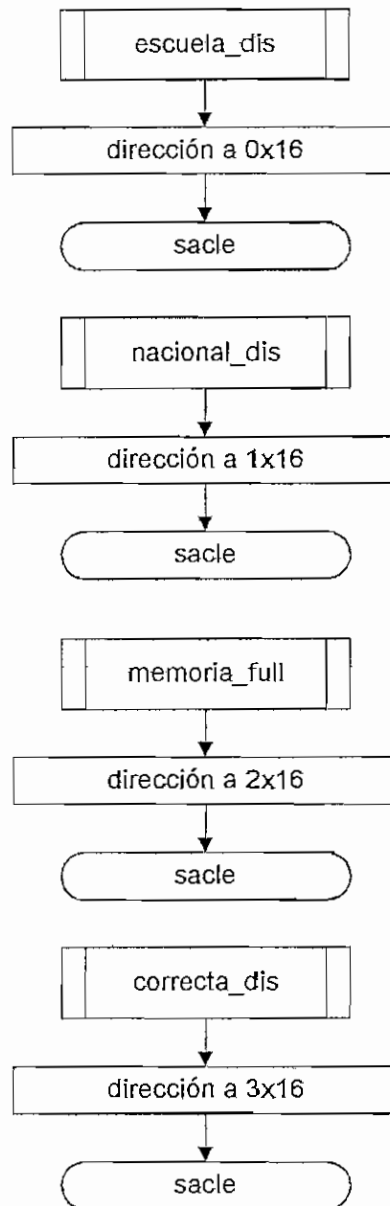


Figura 3.8 .- Subrutina de inicialización del display

En la Figura 3.8 se ilustra el diagrama de flujo de la subrutina que inicializa el display y lo predetermina para su funcionamiento en 2 líneas, matriz de 5x7 y que no se muestre el cursor, según el proceso recomendado por el fabricante del mismo.

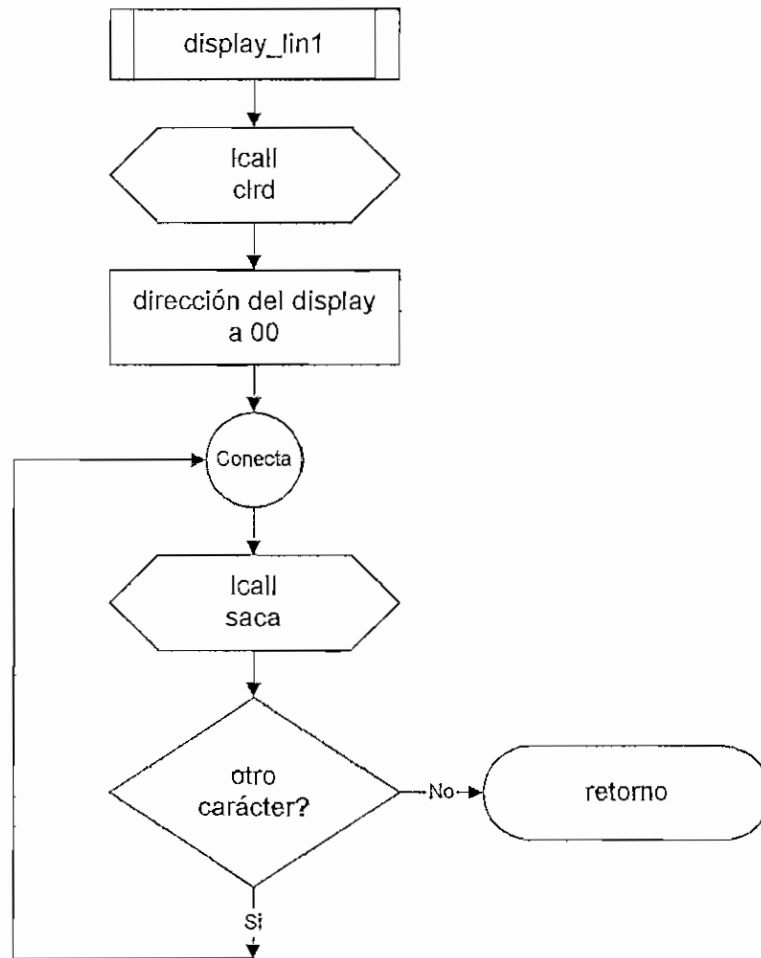
### Subrutinas de letreros en el display



**Figura 3.9 .- Subrutinas de indicación de letreros en el display.**

En la Figura 3.9 se ilustra el diagrama de flujo de las subrutinas que muestran los letreros de Escuela Politécnica Nacional, Correcta, Memoria Llena, cambiando las direcciones donde están guardados estos datos como constantes. Cada vez se llama a la rutina de <sacle> saca letrero.

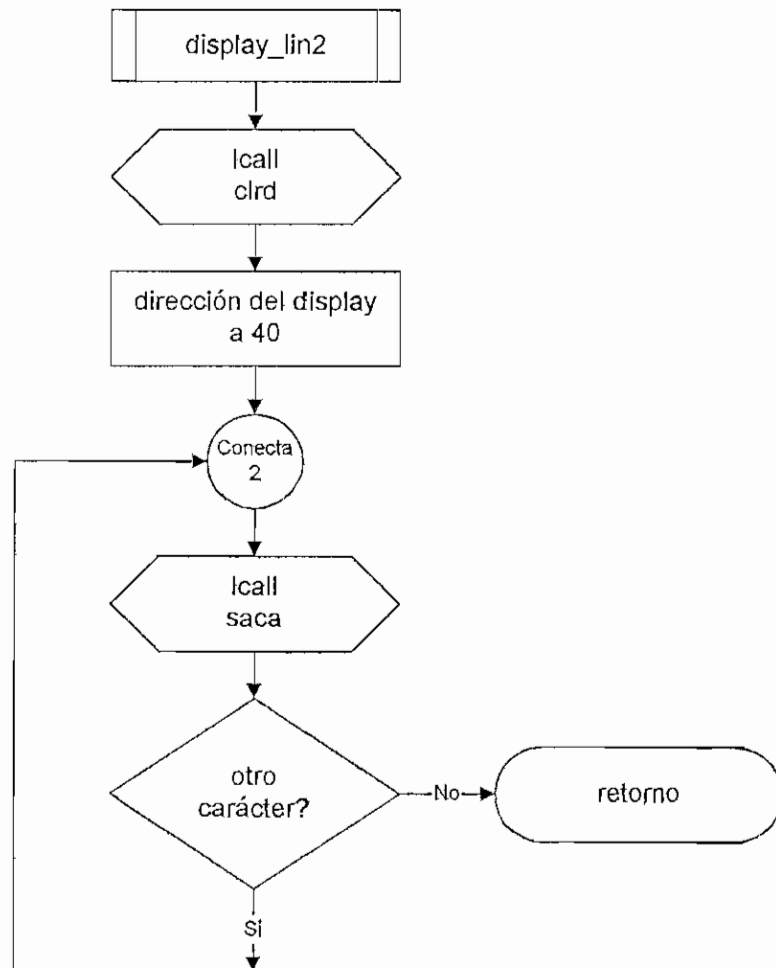
**Subrutina de muestra de primera línea  
y proceso de carácter en carácter.**



**Figura 3.10 .- Subrutina de línea 1 y muestra de carácter en carácter.**

En la Figura 3.10 se ilustra el diagrama de la subrutina con la que se muestra la primera línea y saca el letrero de carácter en carácter, usando llamadas a las subrutinas de <clrd> (clear display) y <saca> (muestra carácter)

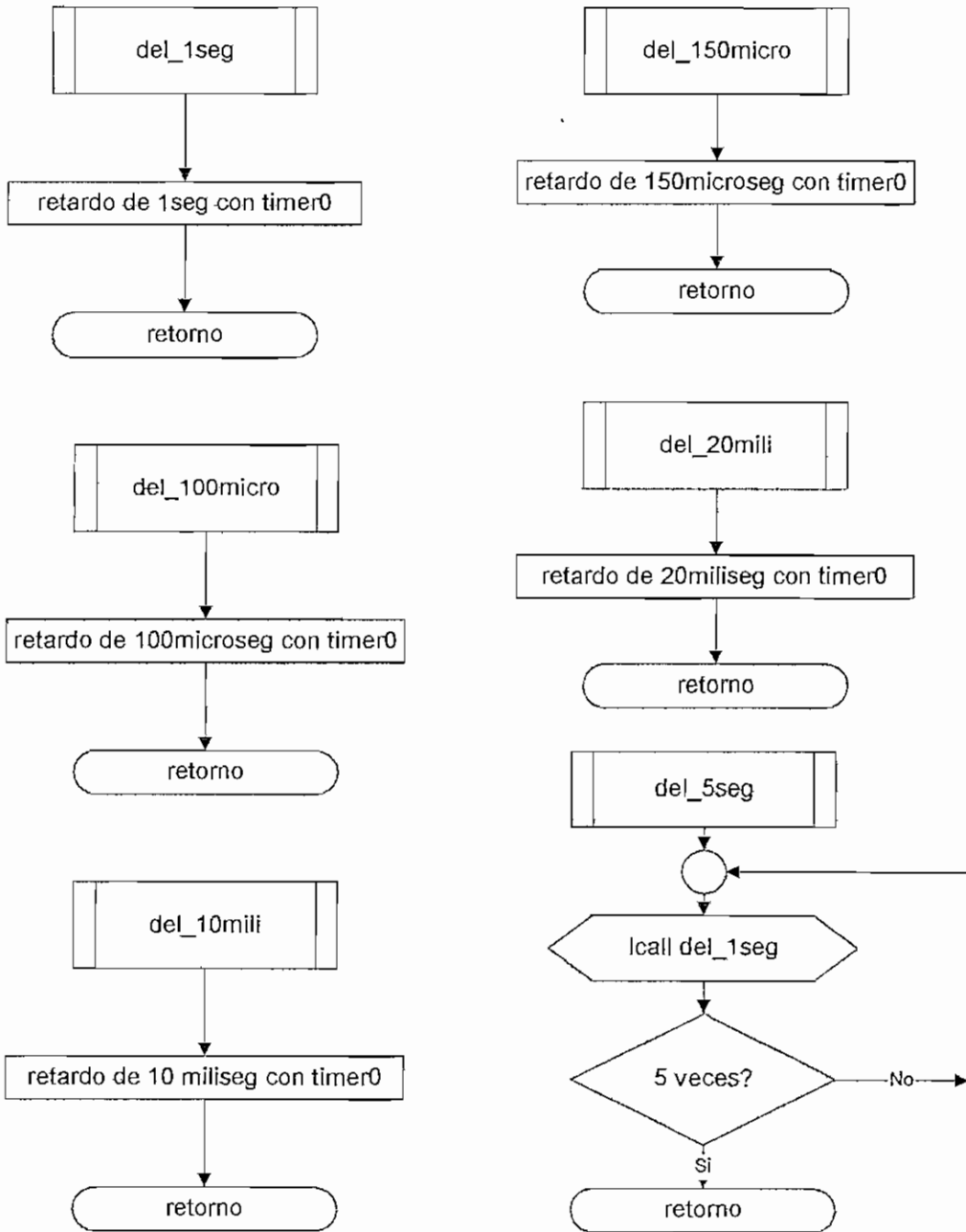
**Subrutina de muestra de segunda línea  
y proceso de carácter en carácter.**



**Figura 3.11 .- Subrutina de línea 2 y muestra de carácter en carácter.**

En la Figura 3.11 se ilustra el diagrama de la subrutina con la que se muestra la segunda línea y saca el letrero de carácter en carácter, usando llamadas a las subrutinas de <clrd> (clear display) y <saca> (muestra carácter), de manera similar a la primera línea.

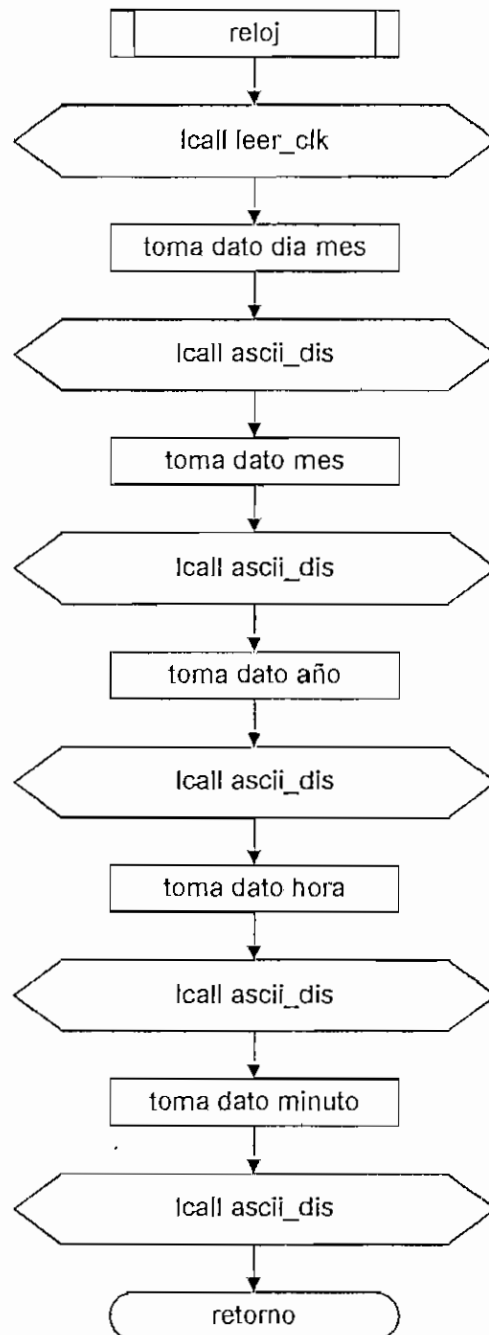
### Subrutinas de retardo



**Figura 3.12 .-** En esta figura se ilustran las subrutinas de los diferentes retardos que se usan en el programa.

En la Figura 3.12 podemos ver las diferentes subrutinas usadas para los retardos necesarios, especialmente en la inicialización del display.

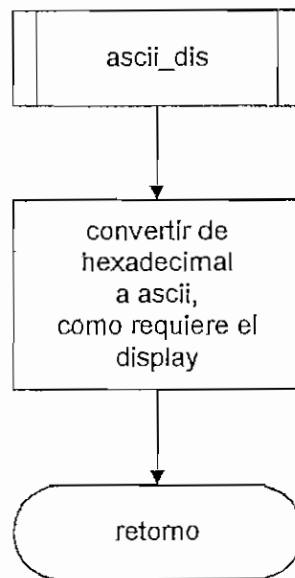
### Subrutina de lectura del reloj interno



**Figura 3.13 .-** En esta figura se ilustra como se lee el reloj interno y se va mostrando en la pantalla la fecha y la hora.

En la Figura 3.13 se ilustra el diagrama de la subrutina con la que se lee el reloj interno del microprocesador DS2250T en tiempo real y luego se muestra en la pantalla la fecha y la hora. Posteriormente se regresa a la rutina invocante.

### Subrutina de conversión de hexadecimal a ascii



**Figura 3.14 .-Subrutina de conversión de hexadecimal a ascii**

En la Figura 3.14 solamente se da la idea de la subrutina que se usa para la conversión de los datos que están en sistema hexadecimal a ascii, (lo cual se hace con un algoritmo conocido), que es el tipo de datos que maneja el display.

### Subrutina de lectura de nombre y número de estación, provenientes del PC

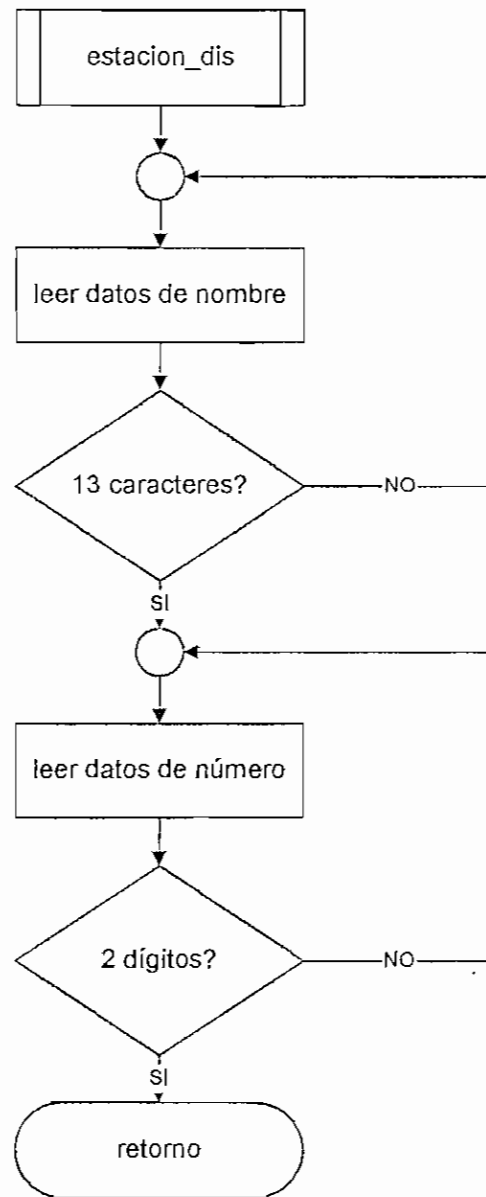


Figura 3.15 .- Diagrama de recepción de nombre y número de estación desde el PC

En la Figura 3.15 se ve la forma en que se reciben los datos de 13 caracteres para el nombre de la estación y dos dígitos para el código de la misma, enviados desde el PC de proceso.



### 3.2 DISEÑO DEL PROGRAMA DE GESTIÓN EN EL COMPUTADOR DE PROCESO.

Por no ser objeto del presente trabajo se describe brevemente el Programa de Gestión en el computador de Proceso y el mismo está disponible en su versión ejecutable.

El programa de gestión en el computador de proceso está desarrollado en FoxPro y tiene los siguientes componentes o módulos:

- Manejo de datos de buses
- Manejo de tipo de buses
- Manejo de datos de rutas
- Manejo de datos de choferes
- Cálculo de transacciones (adelantos o retrasos)
- Reportes
- Mantenimiento del programa

Todos los módulos de manejo de datos tienen la posibilidad de: ingreso de nuevos datos, eliminación, modificación e impresión de los datos correspondientes.

En el módulo de Cálculo de Transacciones se tiene la posibilidad de justificar un adelanto o atraso de la unidad, antes de realizar el reporte de excepciones en el módulo de reportes.

En el módulo de Mantenimiento (Acceso) se puede mantener las claves de ingreso al programa y el reindexamiento de las bases de datos y el mantenimiento de la bitácora.

## **CAPITULO IV.- PRUEBAS, COMENTARIOS Y CONCLUSIONES.**

### **4.1 PRUEBAS DE FUNCIONAMIENTO.**

Una vez construido el prototipo en una caja metálica terminada con pintura electrostática para una gran resistencia a la corrosión y al maltrato, se realizaron las siguientes consideraciones prácticas:

- Parte de la interface óptica que se conecta al computador, los fotodiodos exclusivamente, se los montó en las tapas de un conector DB9 que adicionalmente tiene pegado un imán que es el que da sujeción al conector DB9 sobre la caja del prototipo, cuando se necesite la comunicación entre el prototipo y el PC.

- El resto de la interface se la montó sobre una caja que contiene la batería de alimentación y un interruptor de ON/OFF.

- La parte superior de la caja del prototipo tiene una pequeña inclinación para que pueda el aparato ser usado tanto sobre escritorio como sujeto a una pared y no se tenga inconveniente en la lectura de la pantalla y sus mensajes.

Se realizaron pruebas de comunicación entre el prototipo y el PC de proceso, con los siguientes resultados:

- Comunicación desde PC de proceso a microcontrolador para la carga inicial del programa de assembler: correcta.

- Comunicación para igualación de reloj: correcta.

- Comunicación para reprogramación de Nombre de Estación y Número de Estación: correcta.
  
- Lectura de tarjetas magnéticas por parte del prototipo con la aceptación de los códigos válidos: correcta.
  
- Comunicación desde el prototipo al PC de proceso para el envío de los datos recolectados: correcta. Se adjunta, en los anexos, una impresión del archivo de texto que se crea en el PC al recibir los datos.
  
- Comunicación para el borrado de datos almacenados en el prototipo ya leídos: correcta.
  
- Pruebas de las tablas creadas con FoxPro de: bus, tipo de bus, ruta, chofer con las opciones en cada una de: ingreso, consulta y eliminación, modificación e impresión: correctas.
  
- Pruebas de cálculo de los atrasos o adelantos de las unidades en base a las rutas que seguirá cada una: correctas.
  
- Impresión de reportes diario y de período de los datos procesados por el PC: correcta. Se incluye en los anexos una impresión de cada uno de los reportes.

## **4.2 ANÁLISIS ECONÓMICO.**

El prototipo construido tiene un costo aproximado de US\$350,00 que incluye todos los elementos electrónicos, la placa de circuito impreso, la caja metálica, la fuente de poder, el cable de interface con el PC con su pequeña caja y el soporte de los fotodiodos. No se ha incluido el costo que tiene el software desarrollado tanto para el microcontrolador como para la comunicación entre el

prototipo y el PC y tampoco el del software de aplicación para la generación de los reportes.

Asumiendo que el costo de software esté incluido en un precio de venta de US\$500,00 hemos realizado una comparación de costos con el sistema tradicional de tarjetas de cartulina, en el cual cada equipo está alrededor de US\$450,00.

Se debe tener en cuenta que el sistema con tarjetas magnéticas tiene un costo inicial alto (costo unitario US\$3,50) en la confección de las mismas, pero estas durarán alrededor de un año. Por el contrario las tarjetas de cartulina se las usa una por día (costo US\$0,06 cada una).

Adicionalmente con el sistema tradicional se necesita reemplazar las cintas entintadas que tienen una duración de alrededor de 30.000 marcaciones bien legibles (costo US\$25,00). Por otro lado, en el sistema magnético es necesario reemplazar la cabeza lectora cada 300.000 lecturas (costo US\$60).

En el sistema de tarjetas de cartulina normalmente se tiene el costo del control de las mismas por parte de una persona diariamente (una hora) que se ha proyectado a un valor de alrededor de US\$15,00 mensuales. En el sistema de tarjetas magnéticas toma el procesamiento menos de un minuto. El tiempo que hay que considerar es el de la lectura de los datos de cada uno de los sitios en que estaría instalado un recolector de datos.

Se ha hecho un cuadro comparativo de los costos anualizados de implementación de los dos sistemas y de los costos de operación de los mismos en base a 20 buses por ruta, 4 estaciones por ruta y cinco vueltas por día, el que se muestra en la tabla 4.1.

**Tabla 4.1.- Comparación de costos anuales de los dos sistemas.**

	Sistema de Tarjetas cartulina		Sistema de Tarjetas magnéticas	
<b>Equipo estación</b>	1,800.00	4 equipos	2,000.00	4 equipos
<b>Tarjetas</b>	438.00	20 diarias	70.00	1 anual
<b>Cinta</b>	100.00	1 trimestral	0.00	ninguna
<b>Cabeza lectora</b>	0.00	ninguna	30.00	1 cada 2 años
<b>Proceso</b>	180.00	US\$15 mensual	3.00	US\$0.25 mensual
<b>Mantenimiento</b>	80.00	Trimestral	10.00	Anual
<b>Total</b>	2,598.00		2,113.00	

### 4.3 COMENTARIOS Y CONCLUSIONES.

De la tabla 4.1 podemos deducir que al año existirá un ahorro de US\$500,00 con el sistema de tarjetas magnéticas, aparte de las ventajas discutidas inicialmente.

Con lo que se ha logrado en el desarrollo del presente trabajo vemos que se ha justificado plenamente el mismo, ya que se ha comprobado que el sistema de tarjetas magnéticas es más económico, más eficaz, más confiable, menos manipulable, más veloz, versátil y susceptible de ser mejorado con el avance de la tecnología electrónica y su constante reducción en costos.

## ANEXOS.

### BIBLIOGRAFÍA.-

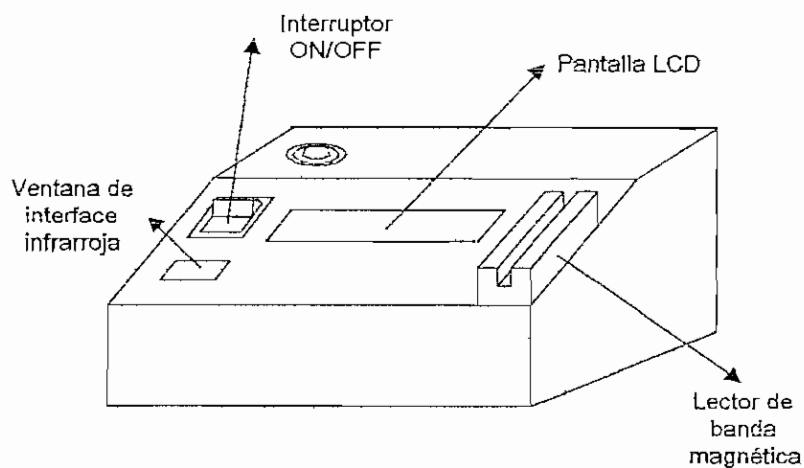
1. – ARACIL, R. Y JIMENEZ, A.; **Sistemas discretos de control**; Departamento de publicaciones de la ETSII; Madrid; 1981
2. – BAGÚ, Sergio; **Tiempo, realidad social y conocimiento**; Siglo Veintiuno Editores S.A.; 1ª Ed.; Buenos Aires; 1970
3. – BUBAN, Peter y OTROS; **Electricidad y Electrónica, aplicaciones prácticas**; McGraw Hill, México; 1990
4. – DALLAS SEMICONDUCTOR; **DS5000 Soft Microcontroller User's Guide**; Dallas Semiconductor Corp.; Dallas; 1989
5. – DALLAS SEMICONDUCTOR; **Product Data Book**; Dallas Semiconductor Corp.; Dallas; 1989
6. – GONZALEZ, Fabio y MORENO, César; **Depurador y kit de desarrollo para el microcontrolador DS5000T**; Escuela Politécnica Nacional; Quito; 1991
7. – GONZALEZ, José Adolfo; **Introducción a los microcontroladores**; McGraw Hill; Madrid; 1992
8. – LEIBSON, Steve; **The handbook of microcomputer interfacing**; Tab Books, Inc.; California; 1985
9. – MAXIM; **Integrated circuits data book**; Maxim Integrated Products, Inc.; California; 1989
10. – OLLERO BATURONE, Aníbal; **Control por computadora**; Marcombo S.A.; Barcelona; 1991
11. – SKROKOV, M.R.; **Mini and microcomputer control in industrial processes**; Van Nostrand Reinhold; New York; 1980
12. – TOLEDO, Francisco; **Temporizador programable controlado por un microcomputador**; Escuela Politécnica Nacional; Quito; 1995
13. – ZELENOVSKY, Ricardo; **El PC para ingenieros electrónicos**; Escuela Politécnica del Ejército; Quito; 1994

## MANUAL DE USO.-

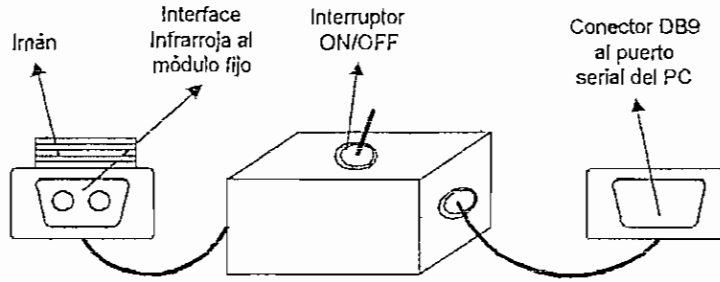
**1.- Instalación.-** La instalación del prototipo se la realiza sobre escritorio o mesa, en el modelo de producción se lo podrá hacer también fijo sobre una pared.

Como todo equipo electrónico se debe escoger un sitio preferentemente donde no esté sujeto ni a temperatura ni humedad extremas y en lo posible cuidarlo de excesivo polvo.

**2.- Nombre de cada una de las partes.-** En la figura anexo 1, se indican las partes que tiene el prototipo en su módulo fijo y en la figura anexo 2 las del cable de conexión al PC.



**Figura anexo 1.- Partes que conforman el módulo fijo del prototipo.**



**Figura anexo 2.- Partes que conforman el módulo móvil del prototipo que es el que se conecta al módulo fijo por un lado y al PC por el otro .**

**3.- Funcionamiento.-** Los dos módulos: fijo y móvil son independientes. El módulo fijo funciona todo el tiempo y el móvil se lo usa solamente en el momento de comunicar al módulo fijo con el PC.

Inicialmente se prende el módulo fijo, y mostrará en la pantalla el letrero "Escuela Politécnica Nacional" , después de un momento se mostrará la fecha, la hora, el nombre de la estación y el número de la misma con lo que está listo para funcionar.

Se pasan las tarjetas magnéticas con identificación del bus y con cada pasada, se mostrará en el display si la misma tiene una codificación válida o no. Este procedimiento anterior es el modo normal de trabajo durante todo el día.

Al final del día se conecta el módulo fijo al PC a través del módulo movible y se leen los registros almacenados durante el día. Posteriormente se encera el módulo fijo. El anterior es el procedimiento para todos los equipos instalados en los diferentes puntos de las rutas. Para la comunicación del PC de proceso con el terminal fijo, se usa un programa hecho en QBASIC cuyo listado se incluye con el de ASSEMBLER en forma separada. Las pantallas que muestra este programa de QBASIC se las ve a continuación:



09-02-2000	PROGRAMA PARA CONTROL DE BUSES	19:14:26
------------	--------------------------------	----------

<b>MENU DE OPCIONES</b>	
1.-	Igualar el reloj
2.-	Borrar registros
3.-	Leer registros
4.-	Cargar estacion
9.-	Fin de operaci3n
Escoja su opci3n : ==>	

09-02-2000	PROGRAMA PARA CONTROL DE BUSES	19:16:04
------------	--------------------------------	----------

<b>LECTURA DE DATOS DEL REGISTRADOR</b>	
Para leer todos los datos <T>	
Para salir presione <ENT>	
Escoja su opci3n :	

09-03-2000	PROGRAMA PARA CONTROL DE BUSES	22:44:15
------------	--------------------------------	----------

<b>MENU DE OPCIONES</b>	
1.-	Igualar el reloj
2.-	Borrar registros
3.-	Leer registros
4.-	Cargar estacion

Nombre de estacion	:? La Marina*****
Numero de la estacion	:? 53

Si quiere CARGAR digite <S> s	
-------------------------------	--

Los datos leídos de los diferentes equipos se los reúne y son procesados para el cobro de las sanciones respectivas si las hubiere.

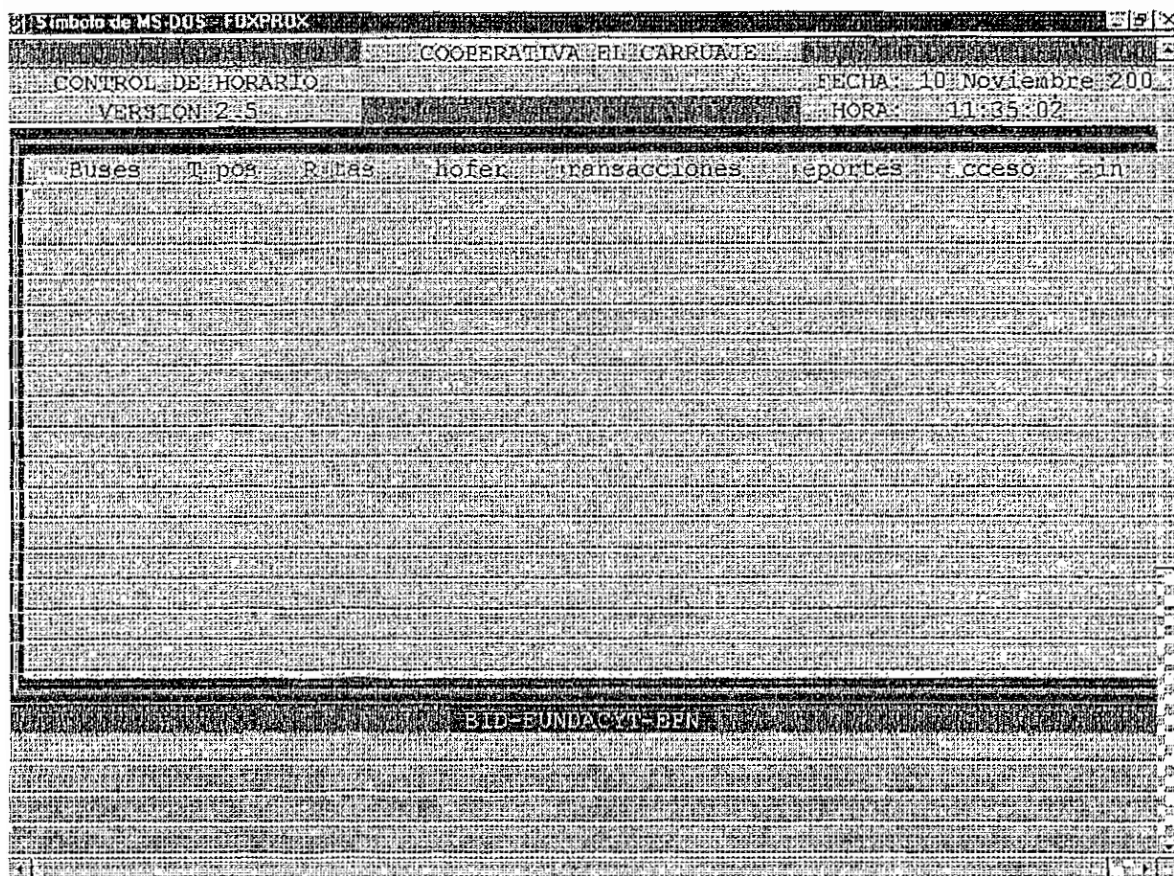
De la misma forma en que se conecta el PC con el prototipo para leer los datos se debe conectar para: igualar el reloj, cambiar el número o el nombre de la estación o reprogramar el microcontrolador.

En las siguientes páginas se presenta un resumen del funcionamiento del programa de proceso realizado en FOXPRO, el cual mantiene las tablas de Buses, Tipos de Bus, Rutas y Puntos, Choferes, Justificaciones de Atrasos/Adelantos, Proceso de datos, Reportes, Acceso al Sistema y Fin o Salida del programa.

## PANTALLAS DEL PROGRAMA DE PROCESO REALIZADO EN FOXPRO

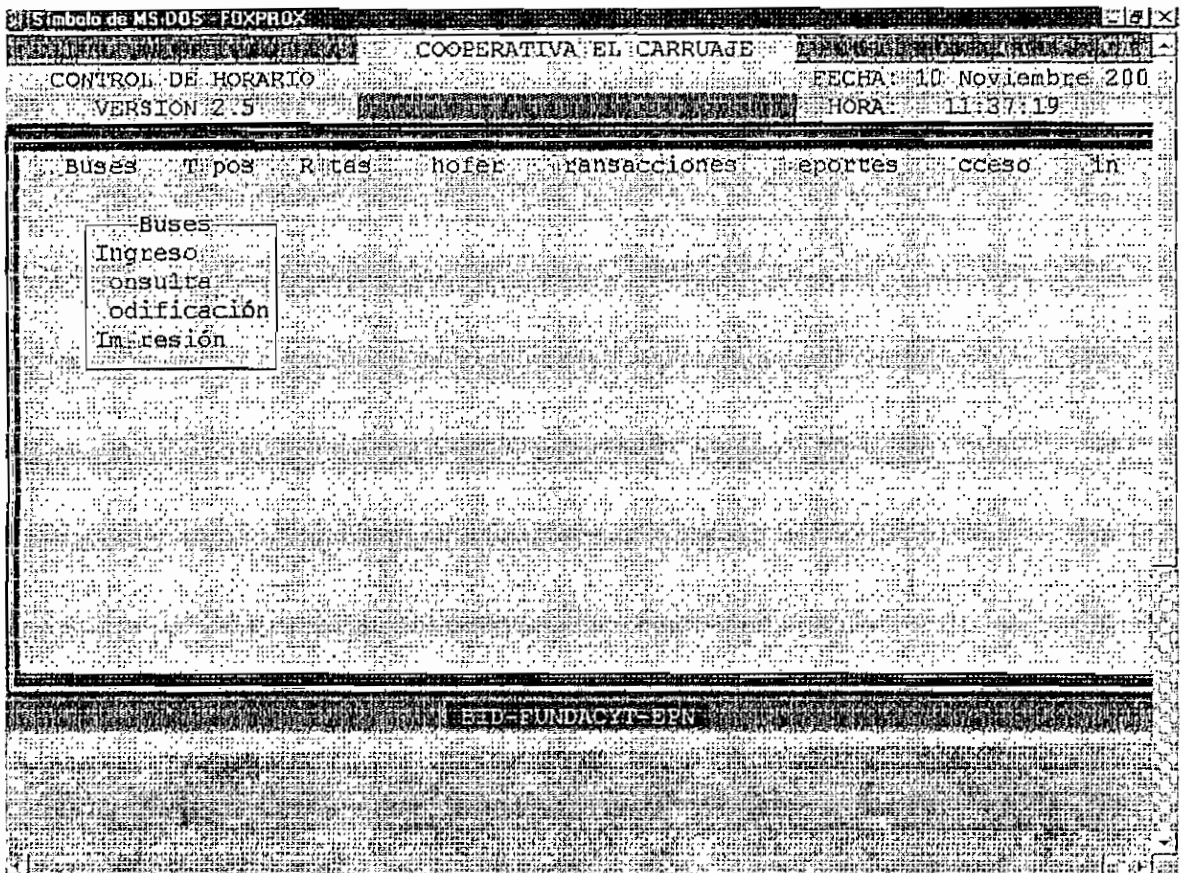
A continuación se presentan las pantallas del programa de proceso del archivo **datos.txt**, que es el que se recobra con el programa realizado en Qbasic desde los terminales.

### PANTALLA PRINCIPAL



En esta pantalla se muestra el menú de inicio en donde se tienen las diferentes posibilidades de manejar las tablas de: buses, tipos, rutas con sus puntos, choferes y las transacciones, reportes, acceso y salida (Fin). Desde aquí podemos escoger el proceso que deseamos realizar.

## PANTALLA DE MANEJO DE LA TABLA DE BUSES



En Esta pantalla tenemos las opciones para la tabla de buses, en donde podemos ingresar nuevos buses, consultar los datos de los mismos, modificarlos o imprimirlos.

Todas las tablas adicionales de Tipos y Choferes se las maneja de la misma forma con similares opciones. La tabla que tiene un manejo un poco diferente es la de Rutas con sus Puntos individuales.

## PANTALLA DE MANTENIMIENTO DE BUSES

```

Simbolo de MS-DOS: FOXPROX
COOPERATIVA EL CARRUAJE
CONTROL DE HORARIO
FECHA: 10 Noviembre 200
VERSION 2.5
HORA: 11:39:47

MANTENIMIENTO DE BUSES

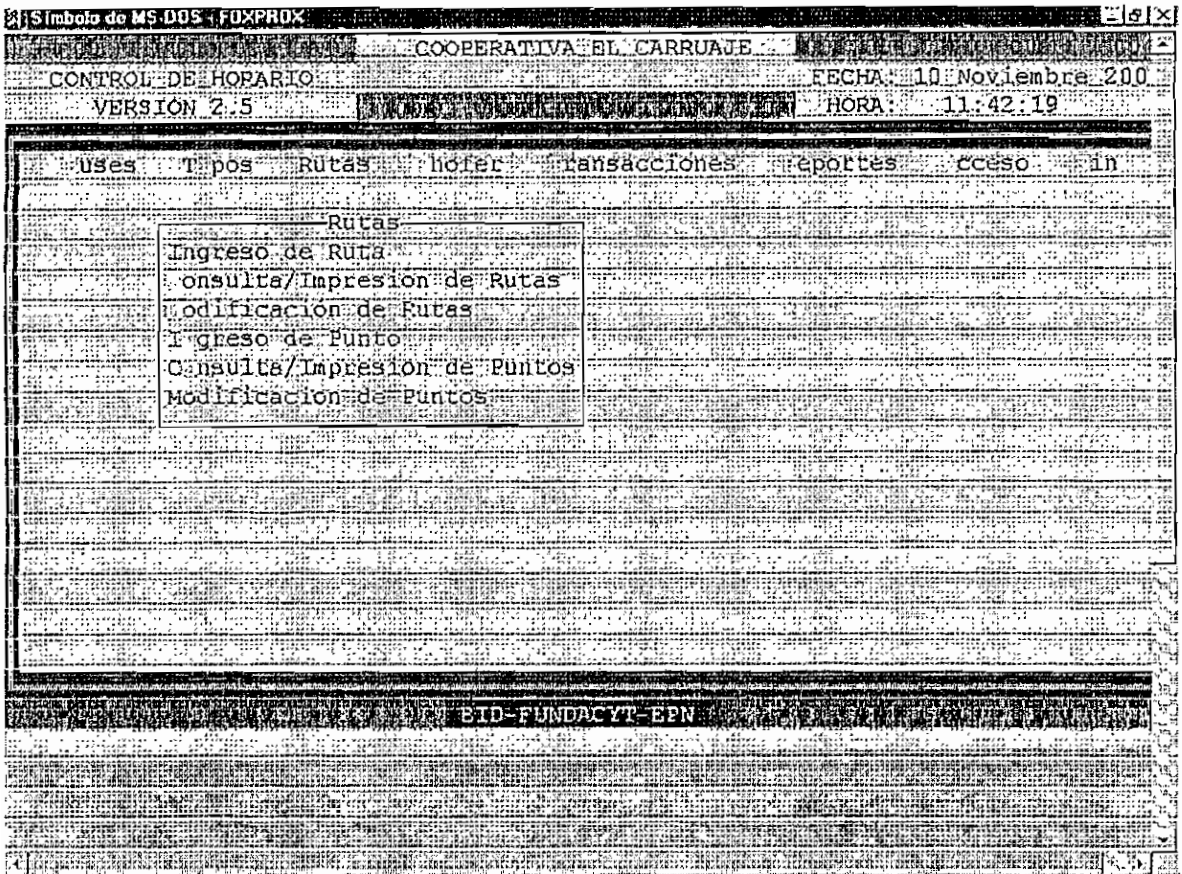
<F1> Elegir Chofer <F2> Elegir Ruta
<F3> Elegir Tipo <ESC> si no desea modificar

Codigo      : 1708935000
Placa       : E-23012
Chofer      : 464 MANZANO PEDRO
Ruta       : 01 COLON-CAMAL
Tipo       : 18 SELECTIVO URBANO
  
```

Se escogió esta opción, ya que es una muestra del mantenimiento de la tabla de buses y es similar a la de ingreso de nuevas unidades. Como se ve es fácil de usar.

La pantalla de consulta muestra todas las unidades y es en ella desde donde se las puede eliminar.

## PANTALLA DE TABLAS DE RUTAS Y PUNTOS



Con esta opción se realiza el control y mantenimiento de las diferentes Rutas y dentro de cada un de ellas de sus Puntos o Estaciones.

El manejo de estas tablas es similar a las otras, excepto que cada Ruta tiene como referencia su propia tabla de Puntos.

## PANTALLA DE MANTENIMIENTO DE RUTAS Y PUNTOS

Simbolo de MS-DOS - FOXPROX

COOPERATIVA EL CARRUAJE

CONTROL DE HORARIO FECHA: 10 Noviembre 200

VERSION 2.5 HORA: 11:44:15

**MODIFICACION DE PUNTOS**

<Enter> Escoger <F2> Ruta por Nombre <F3> Ruta por Código <ESC> salir

<PgUp> Pant. Arriba <PgDn> Pant. Abajo <TAB> <↑> <↓> Moverse entre campos

Código de Ruta: 01

Nombre de Ruta: COBON-CAMAJ

Ingrese Código de Punto: 03

Ingrese Descripción de Punto: 6 DE DICIEMBRE

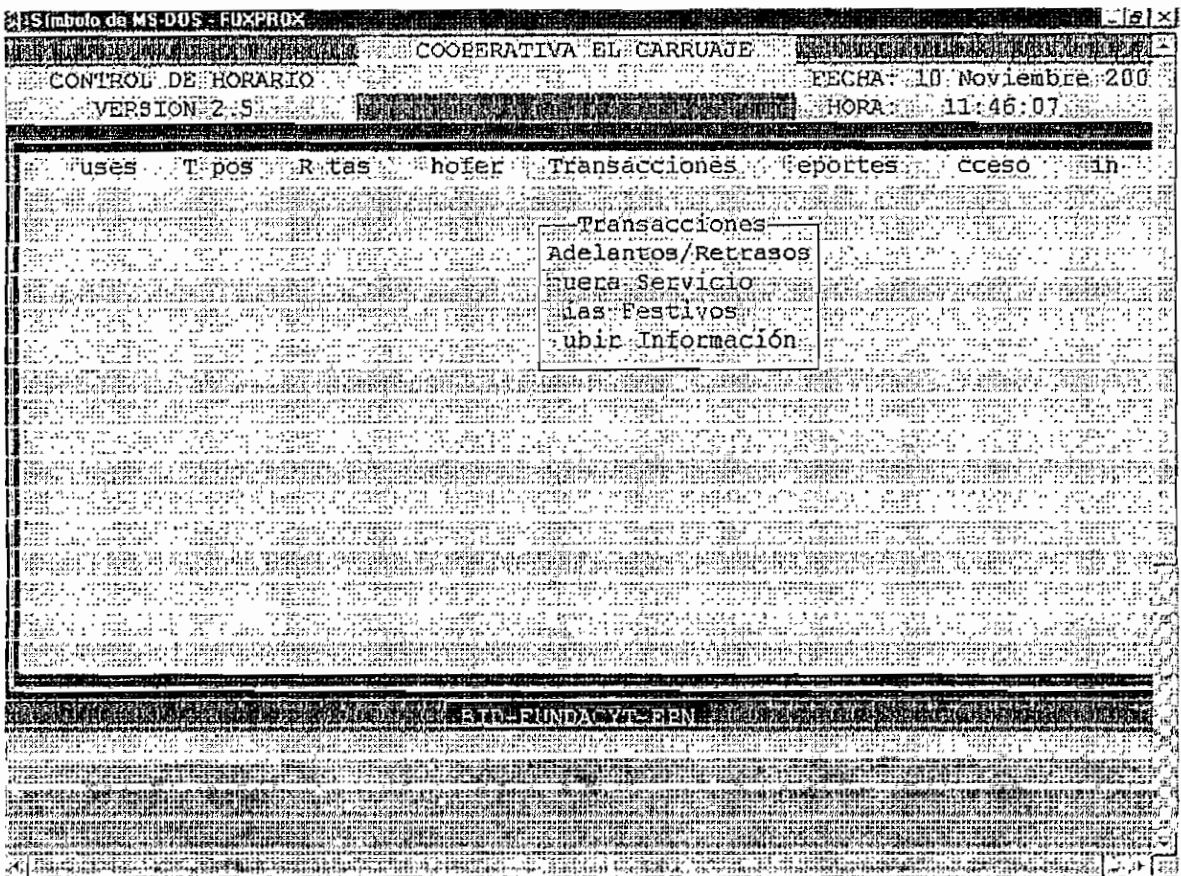
Ingrese Tipo de Punto: INTER

Ingrese Tiempo al Punto: 1:00

Como se ve en esta opción se pueden modificar los datos de cada ruta y de cada punto. El formato es similar al de ingreso de nuevos datos.

Lo que se debe tener en cuenta es que existen puntos INTER (intermedios) y puntos INICIO (donde empieza la ruta).

## PANTALLA DE TRANSACCIONES



En esta pantalla tenemos las diferentes opciones de transacciones:

- **Atrasos/adelantos**, que es donde se pueden justificar los mismos, a más de observarlos.
- **Fuera de servicio**, para unidades que están dañadas o no van a trabajar en uno o varios días.
- **Dias festivos**, para ingresar los días en que no tienen obligación de trabajar y necesidad de control horario.
- **Subir información**, con esta opción se procesa la información recabada desde los terminales en el archivo datos.txt.



## PANTALLA DE INGRESO A JUSTIFICACIONES DE ATRASOS/ ADELANTOS

Símbolo de MS-DOS FOXPROX
[a]x

---

**COOPERATIVA EL CARRUAJE**

CONTROL DE HORARIO FECHA: 10 Noviembre 200  
 VERSIÓN 2.5 HORA: 11:48:05

---

**Adelantos/Retrasos por Justificar**  
 Desde: 2000/10/12 Hasta: 2000/11/10

<F1> Justificar    <F2> Buscar Nombre    <F3> Buscar Código    <ESC> Salir  
 <PgUp> Pant. Arriba    <PgDn> Pant. Abajo    <TAB> <↑> <↓> Moverse entre campos

---

Fecha	Hora	Código Bus	Nombre Chofer	Transacción	Excedido
2000/10/12	9:23	0914185913	RODRIGUEZ ALFONS	RETRASO	0.01
2000/10/12	10:24	0914185913	RODRIGUEZ ALFONS	RETRASO	0.01
2000/10/12	11:21	0914185913	RODRIGUEZ ALFONS	ADELANTO	0.03
2000/10/12					
2000/10/12	7:13	1170926146	PEREZ FEDERICO	RETRASO	0.01
2000/10/12	8:12	1170926146	PEREZ FEDERICO	ADELANTO	0.01

En esta pantalla se escoge el bus que va a ser justificado en su adelanto o atraso en la marcación de la hora establecida. En la pantalla siguiente se tiene el justificativo individual.

## PANTALLA DE JUSTIFICACION INDIVIDUAL

Simbolo de MS-DOS - FOXPROX

COOPERATIVA EL CARRUAJE

CONTROL DE HORARIO      FECHA: 10 Noviembre 200

VERSION 2.5      HORA: 11:50:13

Desde: 2000/10/12    Hasta: 2000/11/10

<F1> Justificar    <F2> Buscar Nombre    <F3> Buscar Código    <ESC> Salir

<PgUp> P      e campos

<ENTER> Grabar Modificación    <ESC> Salir sin Grabar

Fecha de Transacción aaaa/mm/dd: 2000/10/12

Fec	Nombre del Chofer :	Perez FEDERICO	do
200	Código del Bus :	1170926146	03
200	Tipo de Transacción :	RETRASO	01
200	Hora Marcada :	7:13	01
200	Tiempo acordado :	1.00	01
200	Tiempo excedido :	0 Horas y 2 Minutos	03
200	Justificativo :		04

Aquí se tiene el ingreso de la justificación de la excepción, atraso o adelanto, y la causa del mismo. Al hacer esto, se corrige la hora de marcación con la preprogramada.

## PANTALLA DE REPORTE DIARIO

Símbolo de MS-DOS - FOXPROX  
 COOPERATIVA EL CARRUAJE  
 CONTROL DE HORARIO  
 FECHA: 10 Noviembre 200  
 VERSIÓN 2.5  
 HORA: 11:52:01

**Reporte Diario de Buses con Excepciones**  
 FECHA: 2000/10/12 DIA: Jueves HORA: 11:51:42  
 <F1> Buscar por Nombre <ESC> Salir <TAB> <↑> <↓> Entre campos  
 <F2> Buscar por Código <PgUp> Pant. Arriba <PgDn> Pant. Abajo  
 Código Ruta: 01 Nombre Ruta: COLON-CAMAL

Código Bus	Punto	Nombre del Chofar	Retraso	Adelanto	Falta
0914185913	03	RODRIGUEZ ALFONSO	0:02	0:00	
0914185913	05	RODRIGUEZ ALFONSO	0:02	0:00	
0914185913	06	RODRIGUEZ ALFONSO	0:00	0:05	
0914185913					
1170926146	02	PEREZ FEDERICO	0:02	0:00	
1170926146	03	PEREZ FEDERICO	0:00	0:02	

En esta pantalla se exhiben las excepciones que han ocurrido en el día y ruta escogidos, para todos los buses de dicha Ruta. Esta presentación puede ser impresa para tener un documento en blanco y negro.

## PANTALLA DE REPORTE MENSUAL

```

Simbolo de MS-DOS - FOXPROX
COOPERATIVA EL CARRUAJE
CONTROL DE HORARIO FECHA: 27 Octubre 2000
VERSION 2.5 HORA: 12:39:28

Reporte de Llegado Mensual de Buses
FECHA: 2000/10/27 HORA: 12:39:08
Codigo : 1170926146 Chofer : PEREZ FEDERICO
ATRASOS: 0.10 ADELANTOS: 0.07
FALTAS: 0.00 <+> <-> Moverse entre dias <ESC> Otro BUS

Fecha      Codigo Bus Punto Nombre del Chofer      Retras Adela
2000/10/12
2000/10/12 1170926146 03 PEREZ FEDERICO      0.00 0.01
2000/10/12 1170926146 04 PEREZ FEDERICO      0.00 0.01
2000/10/12 1170926146 05 PEREZ FEDERICO      0.03 0.00
2000/10/12 1170926146 06 PEREZ FEDERICO      0.00 0.04
2000/10/12 1170926146 07 PEREZ FEDERICO      0.06 0.00

<ESC> Salir <S> Siguiente Bus <A> Anterior Bus : S

```

En esta pantalla se presenta el reporte del mes escogido, para la unidad seleccionada, presentándose un resumen en la parte superior.

En la pantalla principal se tienen las opciones de **Acceso** y de **Fin**, la de Fin es lógicamente para salir del programa. En Acceso se tienen las opciones para ingresar nuevas claves, modificarlas, mantener la Bitácora del sistema y también la opción de reindexar las tablas en caso de que los índices se dañen, especialmente en el momento de un corte de energía.

**4.- Mantenimiento.-** Normalmente el único mantenimiento necesario es el de la limpieza de la cabeza lectora con una tarjeta especial que tiene una pequeña tela o en su defecto con un paño con un poco de alcohol industrial.

**5.- Solución de problemas.-** En caso de que surja algún problema, lo recomendable es apagar el equipo durante unos cinco segundos y volverlo a encender para reiniciarlo.

**ESTRUCTURA DEL ARCHIVO "DATOS.TXT" QUE ENVIA  
EL PROTOTIPO AL PC DE PROCESO.**

<b>23</b>	<b>,2000</b>	<b>,05</b>	<b>,12</b>	<b>,10</b>	<b>,36</b>	<b>,091418591310</b>
ESTACION	AÑO	MES	DIA	HORA	MINUTOS	CODIGO BUS

```

23,2000,05,06,17,36,170939582410
23,2000,05,06,17,36,170893500010
23,2000,05,07,08,27,117092614630
23,2000,05,07,08,27,170974641410
23,2000,05,07,08,27,091418591310
23,2000,05,07,08,27,170974641410
23,2000,05,07,08,27,171257278110
23,2000,05,07,08,27,117092614630
23,2000,05,07,08,42,170974641410
23,2000,05,07,12,52,091418591310
23,2000,05,08,09,24,171257278110
23,2000,05,08,09,24,117092614630
23,2000,05,08,09,24,170974641410
23,2000,05,08,09,24,091418591310
23,2000,05,08,12,34,171257278110
23,2000,05,08,12,34,117092614630
23,2000,05,08,12,34,170974641410
23,2000,05,08,12,34,091418591310
23,2000,05,08,12,35,091418591310
23,2000,05,12,10,33,170974641410
23,2000,05,12,10,33,170974641410
23,2000,05,12,10,34,091418591310
23,2000,05,12,10,34,117092614630
23,2000,05,12,10,35,117092614630
23,2000,05,12,10,35,117092614630
23,2000,05,12,10,35,171257278110
23,2000,05,12,10,35,170974641410
23,2000,05,12,10,36,170974641410
23,2000,05,12,10,36,171257278110
23,2000,05,12,10,36,117092614630

```

## EJEMPLO DEL REPORTE DIARIO

Fecha 2000/10/12		REPORTE DIARIO DE BUSES		
		Hora: 19:41:21		
Codigo Bus	Nombre Chofer	Punto	Retraso	Adelanto Falta
0914185913	RODRIGUEZ ALFONSO	03	0.02	0.00
0914185913	RODRIGUEZ ALFONSO	05	0.02	0.00
0914185913	RODRIGUEZ ALFONSO	06	0.00	0.05
0914185913	RODRIGUEZ ALFONSO	07	0.05	0.00
1170926146	PEREZ FEDERICO	02	0.02	0.00
1170926146	PEREZ FEDERICO	03	0.00	0.02
1170926146	PEREZ FEDERICO	04	0.00	0.02
1170926146	PEREZ FEDERICO	05	0.05	0.00
1170926146	PEREZ FEDERICO	06	0.00	0.07
1170926146	PEREZ FEDERICO	07	0.10	0.00
1708935000	MANZANO PEDRO	02	0.00	0.02
1708935000	MANZANO PEDRO	03	0.03	0.00
1708935000	MANZANO PEDRO	04	0.00	0.03
1708935000	MANZANO PEDRO	05	0.02	0.00
1708935000	MANZANO PEDRO	07	0.02	0.00
1709395824	JACOME LUIS	02	0.00	0.03
1709395824	JACOME LUIS	03	0.03	0.00
1709395824	JACOME LUIS	04	1.00	0.00
1709395824	JACOME LUIS	05	0.00	1.00
1709395824	JACOME LUIS	06	0.00	0.98
1709395824	JACOME LUIS	07	1.00	0.00
1709746414	NARVAEZ HECTOR	02	0.02	0.00
1709746414	NARVAEZ HECTOR	03	0.02	0.00
1709746414	NARVAEZ HECTOR	04	0.00	0.02
1709746414	NARVAEZ HECTOR	05	0.02	0.00
1709746414	NARVAEZ HECTOR	07	0.02	0.00

**EJEMPLO DEL REPORTE DETALLADO MENSUAL**

## REPORTE DETALLADO DE BUSES

Cód.Bus : 0914185913 Nombre Chofer: RODRIGUEZ ALFONSO

ATRASOS: 0.05 ADELANTOS: 0.03  
FALTAS: 0.00

Fecha	Cod.Bus	Nombre Chofer	Punto Retraso	Adelanto	Falta
2000/10/12	0914185913	RODRIGUEZ ALFONSO	03	0.01	0.00
2000/10/12	0914185913	RODRIGUEZ ALFONSO	05	0.01	0.00
2000/10/12	0914185913	RODRIGUEZ ALFONSO	06	0.00	0.03
2000/10/12	0914185913	RODRIGUEZ ALFONSO	07	0.03	0.00



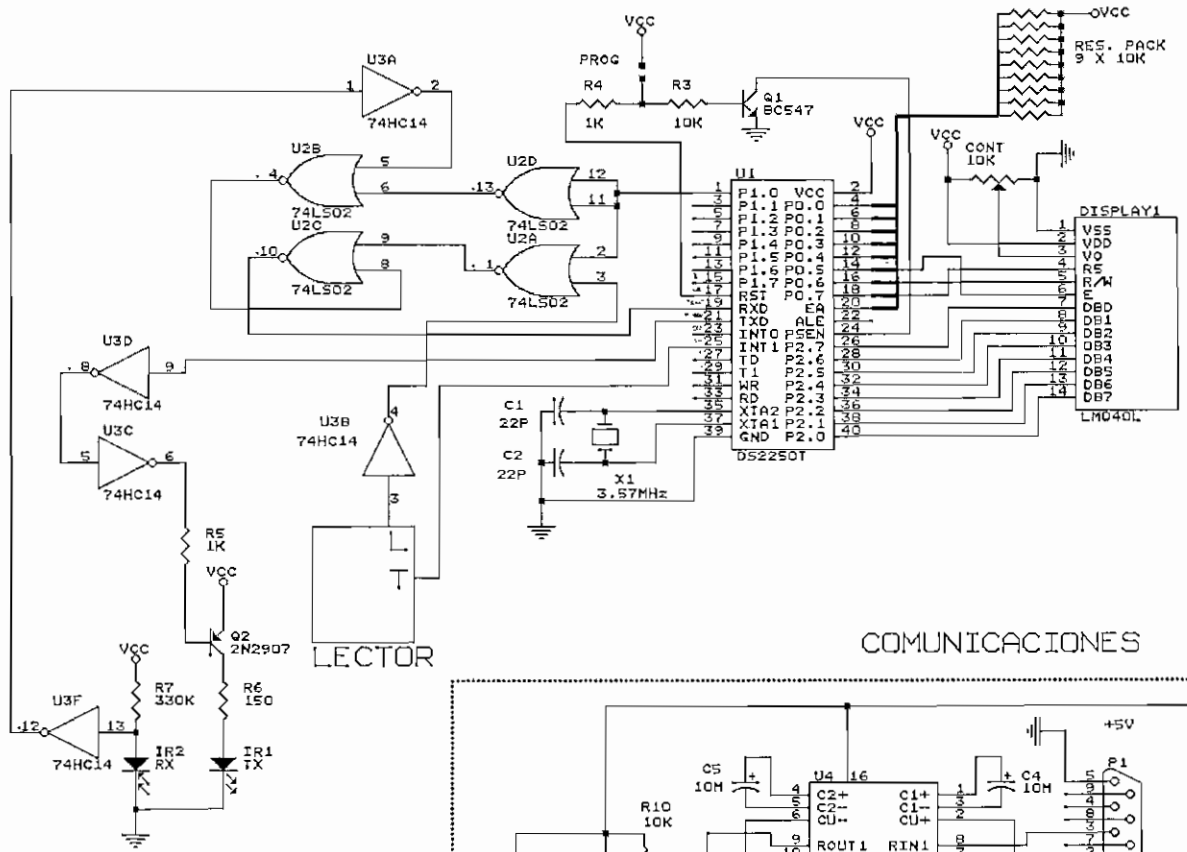
## REPORTE DETALLADO DE BUSES

Cód.Bus : 1170926146 Nombre Chofer: PEREZ FEDERICO

ATRASOS: 0.10 ADELANTOS: 0.07

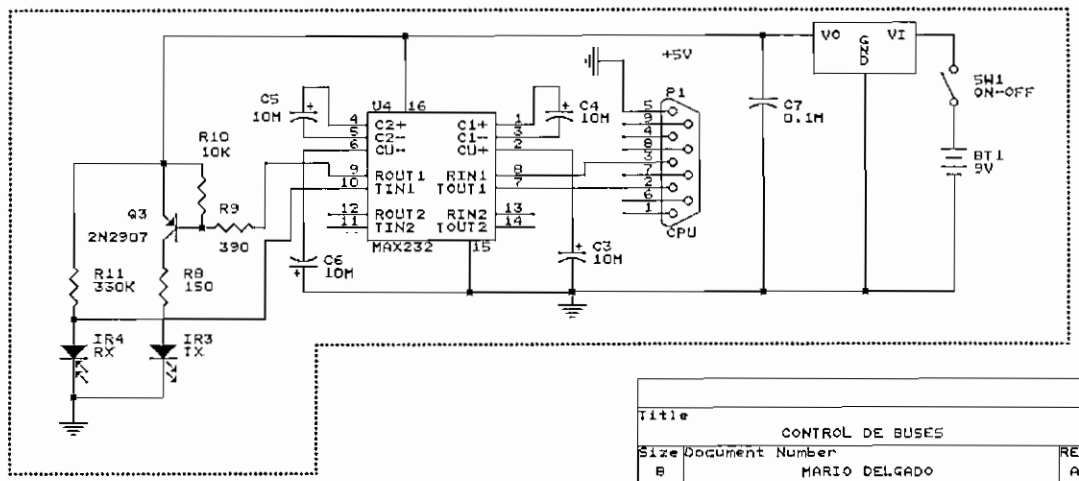
FALTAS: 0.00

Fecha	Cod.Bus	Nombre Chofer	Punto	Retraso	Adelanto	Falta
2000/10/12	1170926146	PEREZ FEDERICO	02	0.01	0.00	
2000/10/12	1170926146	PEREZ FEDERICO	03	0.00	0.01	
2000/10/12	1170926146	PEREZ FEDERICO	04	0.00	0.01	
2000/10/12	1170926146	PEREZ FEDERICO	05	0.03	0.00	
2000/10/12	1170926146	PEREZ FEDERICO	06	0.00	0.04	
2000/10/12	1170926146	PEREZ FEDERICO	07	0.06	0.00	



LECTOR

COMUNICACIONES



Title		CONTROL DE BUSES
Size	Document Number	REV
B	MARIO DELGADO	A
Date:	October 30, 2000	Sheet of

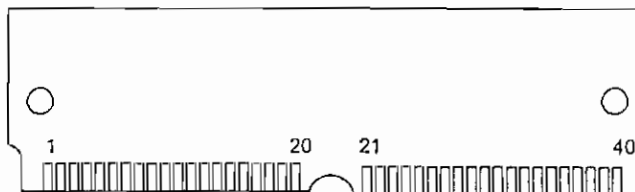
### FEATURES

- 8-bit 8051 compatible microcontroller adapts to task-at-hand:
  - 8K, 32K, or 64K bytes of nonvolatile RAM for program and/or data memory storage
  - Initial downloading of software in end system via on-chip serial port
  - Capable of modifying its own program and/or data memory in end use
- High-reliability operation:
  - Maintains all nonvolatile resources for 10 years in the absence of  $V_{CC}$
  - Power-fail reset
  - Early warning power-fail interrupt
  - Watchdog timer
- Software Security Feature:
  - Executes encrypted software to prevent unauthorized disclosure
- On-chip, full-duplex serial I/O ports
- Two on-chip timer/event counters
- 32 parallel I/O lines
- Compatible with industry standard 8051 instruction set
- Permanently Powered real time clock

### DESCRIPTION

The DS2250(T) Soft Microcontroller Module is a fully 8051 compatible 8-bit CMOS microcontroller that offers "softness" in all aspects of its application. This is accomplished through the comprehensive use of nonvolatile technology to preserve all information in the absence of system  $V_{CC}$ . The internal program/data memory space is implemented using 8K, 32K, or 64K bytes of nonvolatile CMOS SRAM. Furthermore, inter-

### PIN ASSIGNMENT



40-PIN SIMM

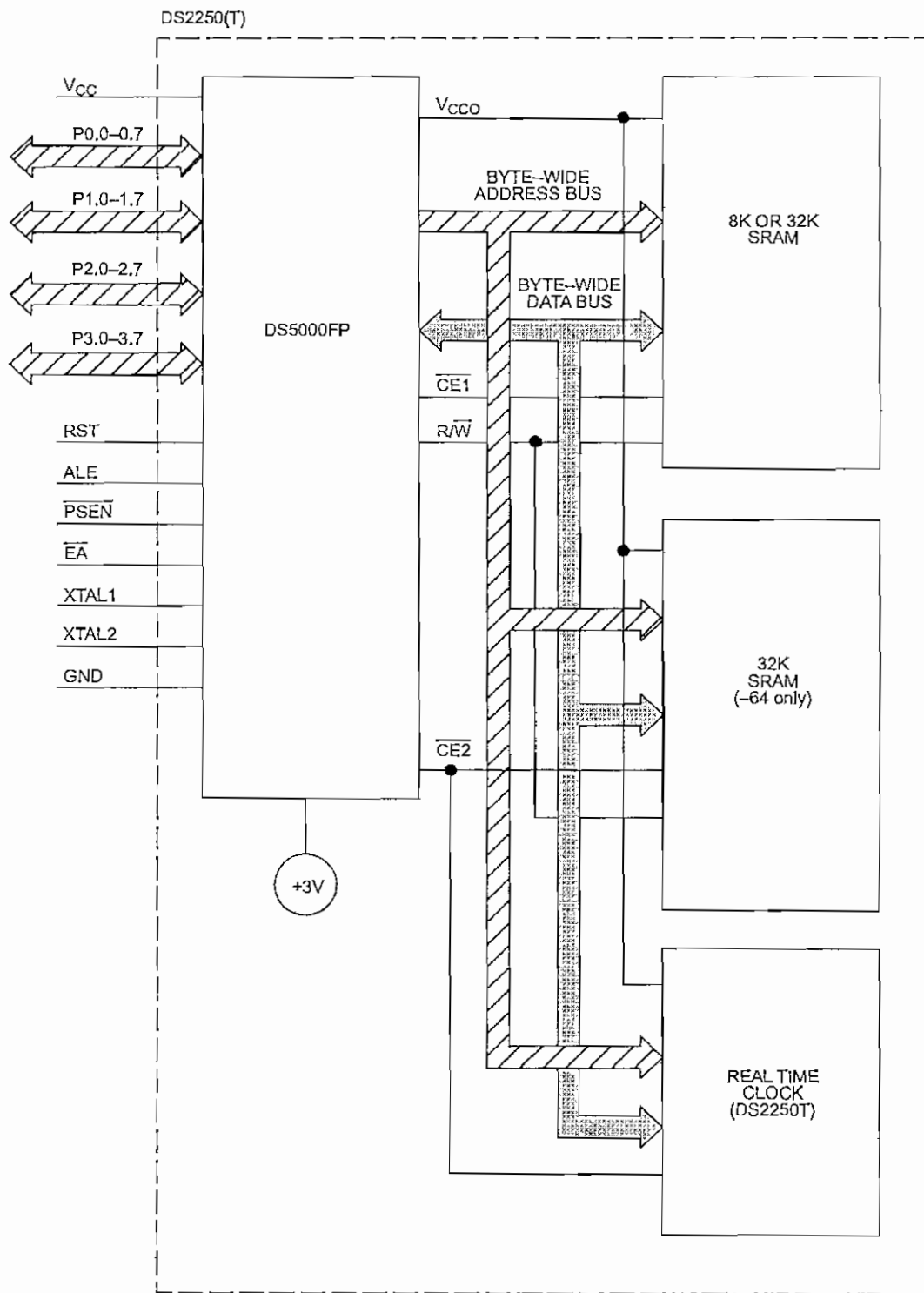
nal data registers and key configuration registers are also nonvolatile. An optional real time clock gives permanently powered timekeeping. The clock keeps time to a hundredth of a second using an on-board crystal. All nonvolatile memory and resources are maintained for over 10 years at room temperature in the absence of power.

**ORDERING INFORMATION**

<b>PART NUMBER</b>	<b>RAM SIZE</b>	<b>MAX CRYSTAL SPEED</b>	<b>TIMEKEEPING?</b>
DS2250-8-16	8K bytes	16 MHz	No
DS2250-32-16	32K bytes	16 MHz	No
DS2250-64-16	64K bytes	16 MHz	No
DS2250T-8-16	8K bytes	16 MHz	Yes
DS2250T-32-16	32K bytes	16 MHz	Yes
DS2250T-64-16	64K bytes	16 MHz	Yes

Operating information is contained in the User's Guide section of the Secure Microcontroller Data Book. This data sheet provides ordering information, pinout, and electrical specifications.

DS2250(T) BLOCK DIAGRAM Figure 1



## PIN DESCRIPTION

PIN	DESCRIPTION
1, 3, 5, 7, 9, 11, 13, 15	<b>P1.0 – P1.7.</b> General purpose I/O Port 1
17	<b>RST</b> -- Active high reset input. A logic 1 applied to this pin will activate a reset state. This pin is pulled down internally so this pin can be left unconnected if not used. An RC power-on reset circuit is not needed and is not recommended.
19	<b>P3.0 RXD.</b> General purpose I/O port pin 3.0. Also serves as the receive signal for the on board UART. This pin should not be connected directly to a PC COM port.
21	<b>P3.1 TXD.</b> General purpose I/O port pin 3.1. Also serves as the transmit signal for the on board UART. This pin should not be connected directly to a PC COM port.
23	<b>P3.2 <math>\overline{\text{INT0}}</math>.</b> General purpose I/O port pin 3.2. Also serves as the active low External Interrupt 0.
25	<b>P3.3 <math>\overline{\text{INT1}}</math>.</b> General purpose I/O port pin 3.3. Also serves as the active low External Interrupt 1.
27	<b>P3.4 T0.</b> General purpose I/O port pin 3.4. Also serves as the Timer 0 input.
29	<b>P3.5 T1.</b> General purpose I/O port pin 3.5. Also serves as the Timer 1 input.
31	<b>P3.6 <math>\overline{\text{WR}}</math>.</b> General purpose I/O port pin. Also serves as the write strobe for Expanded bus operation.
33	<b>P3.7 <math>\overline{\text{RD}}</math>.</b> General purpose I/O port pin. Also serves as the read strobe for Expanded bus operation.
35, 37	<b>XTAL2, XTAL1.</b> Used to connect an external crystal to the internal oscillator. XTAL1 is the input to an inverting amplifier and XTAL2 is the output.
39	<b>GND</b> -- Logic ground.
26, 28, 30, 32, 34, 36, 38, 40	<b>P2.7–P2.0.</b> General purpose I/O Port 2. Also serves as the MSB of the Expanded Address bus.
24	<b><math>\overline{\text{PSEN}}</math></b> -- Program Store Enable. This active low signal is used to enable an external program memory when using the Expanded bus. It is normally an output and should be unconnected if not used. $\overline{\text{PSEN}}$ also is used to invoke the Bootstrap Loader. At this time, $\overline{\text{PSEN}}$ will be pulled down externally. This should only be done once the DS2250(T) is already in a reset state. The device that pulls down should be open drain since it must not interfere with $\overline{\text{PSEN}}$ under normal operation.
22	<b>ALE</b> -- Address Latch Enable. Used to de-multiplex the multiplexed Expanded Address/Data bus on Port 0. This pin is normally connected to the clock input on a '373 type transparent latch. When using a parallel programmer, this pin also assumes the $\overline{\text{PROG}}$ function for programming pulses.
20	<b><math>\overline{\text{EA}}</math></b> -- External Access. This pin forces the DS2250(T) to behave like an 8031. No internal memory (or clock) will be available when this pin is at a logic low. Since this pin is pulled down internally, it should be connected to +5V to use NV RAM. In a parallel programmer, this pin also serves as $V_{PP}$ for super voltage pulses.

PIN	DESCRIPTION
4, 6, 8, 10, 12, 14, 16, 18	<b>P0.0–P0.7.</b> General purpose I/O Port 0. This port is open–drain and can not drive a logic 1. It requires external pull–ups. Port 0 is also the multiplexed Expanded Address/Data bus. When used in this mode, it does not require pull–ups.
2	$V_{CC} + - 5$ volts.

### INSTRUCTION SET

The DS2250(T) executes an instruction set which is object code compatible with the industry standard 8051 microcontroller. As a result, software development packages which have been written for the 8051 are compatible with the DS2250(T), including cross–assemblers, high–level language compilers, and debugging tools. Note that the DS2250(T) is functionally identical to the DS5000(T) except for package and the 64K memory option.

A complete description for the DS2250(T) instruction set is available in the User’s Guide section of the Secure Microcontroller Data Book.

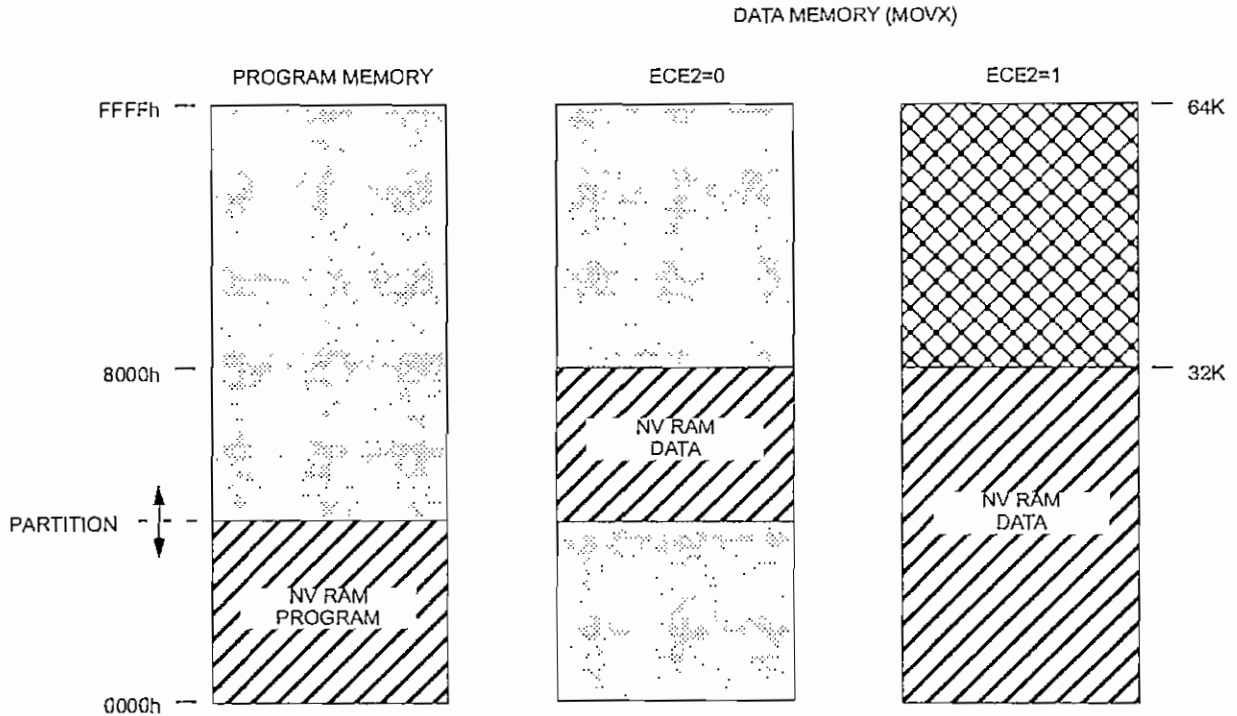
### MEMORY ORGANIZATION

Figure 2 illustrates the address spaces which are accessed by the DS2250(T). As illustrated in the figure,

separate address spaces exist for program and data memory. Since the basic addressing capability of the machine is 16 bits, a maximum of 64K bytes of program memory and 64K bytes of data memory can be accessed by the DS2250(T) CPU. The 8K or 32K byte RAM area inside of the DS2250(T) can be used to contain both program and data memory. A second 32K RAM is available for data only.

The Real Time Clock (RTC) in the DS2250(T) is reached in the memory map by setting a SFR bit. The MCON.2 bit (ECE2) is used to select an alternate data memory map. While ECE2=1, all MOVXs will be routed to this alternate memory map. The real time clock is a serial device that resides in this area. A full description of the RTC access and example software is given in the User’s Guide section of the Secure Microcontroller Data Book.

DS2250(T) MEMORY MAP Figure 2



## LEGEND:



= NVRAM MEMORY



= EXPANDED BUS (PORTS 0 AND 2)



= NOT AVAILABLE

**PROGRAM LOADING**

The Program Load Modes allow initialization of the NV RAM Program/Data Memory. This initialization may be performed in one of two ways:

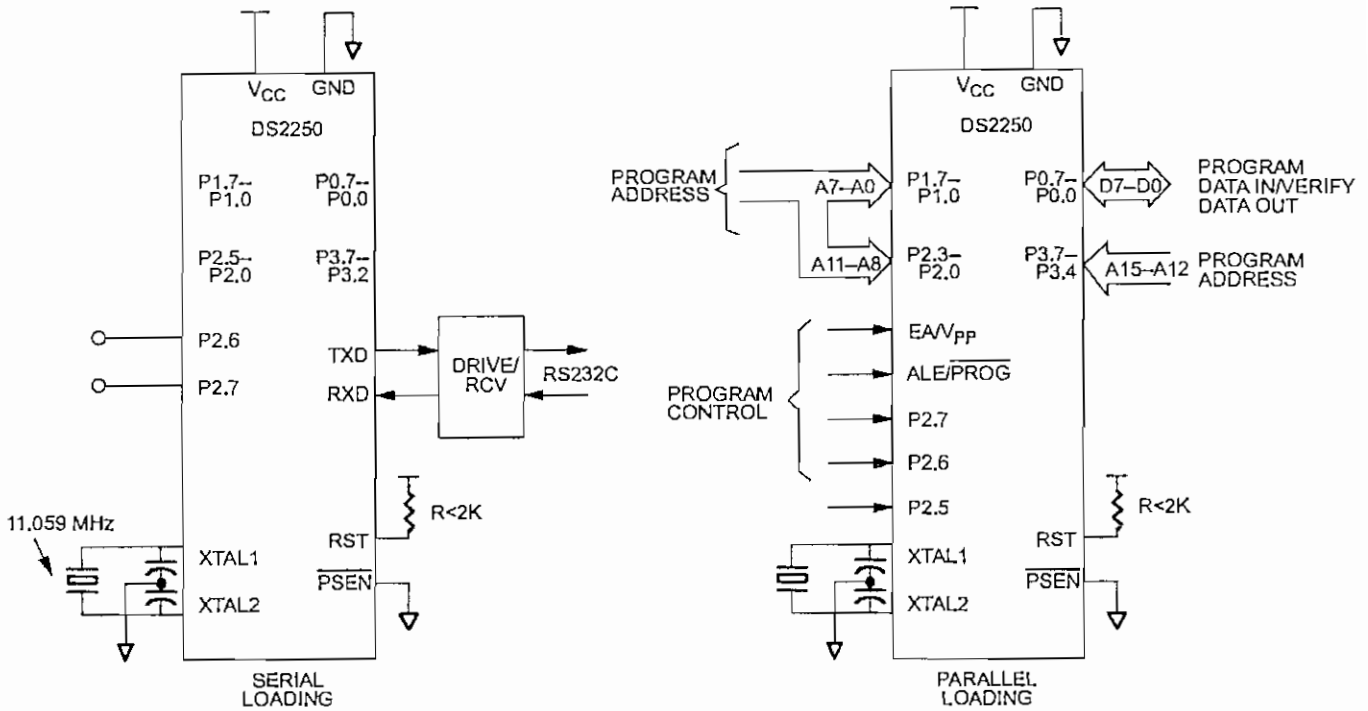
1. Serial Program Loading which is capable of performing Bootstrap Loading of the DS2250(T). This feature allows the loading of the application program to be delayed until the DS2250(T) is installed in the end system.
2. Parallel Program Load cycles which perform the initial loading from parallel address/data information presented on the I/O port pins. This mode is timing-set compatible with the 87C51H microcontroller programming mode.

The DS2250(T) is placed in its Program Load configuration by simultaneously applying a logic 1 to the RST pin and forcing the  $\overline{\text{PSEN}}$  line to a logic 0 level. Immediately following this action, the DS2250(T) will look for a parallel Program Load pulse, or a serial ASCII carriage return (0DH) character received at 9600, 2400, 1200, or 300 bps over the serial port.

The hardware configurations used to select these modes of operation are illustrated in Figure 3.



## PROGRAM LOADING CONFIGURATIONS Figure 3



### SERIAL BOOTSTRAP LOADER

The Serial Program Load Mode is the easiest, fastest, most reliable, and most complete method of initially loading application software into the DS2250(T) nonvolatile RAM. Communication can be performed over a standard asynchronous serial communications port. A typical application would use a simple RS232C serial interface to program the DS2250(T) as a final production procedure. The hardware configuration which is required for the Serial Program Load mode is illustrated in Figure 3. Port pins 2.7 and 2.6 must be either open or pulled high to avoid placing the device in a parallel load cycle. Although an 11.0592 MHz crystal is shown in Figure 3, a variety of crystal frequencies and loader baud rates are supported, shown in Table 2. The serial loader is designed to operate across a three-wire interface from a standard UART. The receive, transmit, and ground wires are all that are necessary to establish communication with the DS2250(T).

The Serial Bootstrap Loader implements an easy-to-use command line interface which allows an application program in an Intel hex representation to be loaded into and read back from the device. Intel hex is the typical format which existing 8051 cross-assemblers output.

The serial loader responds to single character commands which are summarized below:

COMMAND	FUNCTION
C	Return CRC-16 checksum of embedded RAM
D	Dump Intel Hex File
F	Fill embedded RAM block with constant
K	Load 40-bit Encryption Key
L	Load Intel Hex File
R	Read MCON register
T	Trace (Echo) incoming Intel Hex data
U	Clear Security Lock
V	Verify Embedded RAM with incoming Intel Hex
W	Write MCON register
Z	Set Security Lock
P	Put a value to a port
G	Get a value from a port

Table 1 summarizes the selection of the available Parallel Program Load cycles. The timing associated with these cycles is illustrated in the electrical specs.

PARALLEL PROGRAM LOAD CYCLES Table 1

MODE	RST	$\overline{\text{PSEN}}$	$\overline{\text{PROG}}$	$\overline{\text{EA}}$	P2.7	P2.6	P2.5
Program	1	0	0	$V_{PP}$	1	0	X
Security Set	1	0	0	$V_{PP}$	1	1	X
Verify	1	X	X	1	0	0	X
Prog Expanded	1	0	0	$V_{PP}$	0	1	0
Verify Expanded	1	0	1	1	0	1	0
Prog MCON or Key registers	1	0	0	$V_{PP}$	0	1	1
Verify MCON registers	1	0	1	1	0	1	1

The Parallel Program Cycle is used to load a byte of data into a register or memory location within the DS2250(T). The Verify Cycle is used to read this byte back for comparison with the originally loaded value to verify proper loading. The Security Set Cycle may be used to enable and the Software Security feature. One may also enter bytes for the MCON register or for the five encryption registers using the Program MCON cycle. When using this cycle, the absolute register address must be presented at Ports 1 and 2 as in the normal program cycle (Port 2 should be 00H). The MCON contents can likewise be verified using the Verify MCON cycle.

When the DS2250(T) first detects a Parallel Program Strobe pulse or a Security Set Strobe pulse while in the Program Load Mode following a Power On Reset, the internal hardware of the device is initialized so that an existing 4K byte program can be programmed into a DS2250(T) with little or no modification. This initialization automatically sets the Range Address for 8K bytes and maps the lowest 4K byte bank of Embedded RAM

as program memory. The next 4K bytes of Embedded RAM are mapped as Data Memory.

In order to program more than 4K bytes of program code, the Program/Verify Expanded cycles can be used. Up to 32K bytes of program code can be entered and verified. Note that the expanded 32K byte Program/Verify cycles take much longer than the normal 4K byte Program/Verify cycles.

A typical parallel loading session would follow this procedure. First, set the contents of the MCON register with the correct range and partition only if using expanded programming cycles. Next, the encryption registers can be loaded to enable encryption of the program/data memory (not required). Then, program the DS2250(T) using either normal or expanded program cycles and check the memory contents using Verify cycles. The last operation would be to turn on the security lock feature by either a Security Set cycle or by explicitly writing to the MCON register and setting MCON.0 to a 1.

**SERIAL LOADER BAUD RATES FOR DIFFERENT CRYSTAL FREQUENCIES Table 2**

CRYSTAL FREQ (MHz)	BAUD RATE					
	300	1200	2400	9600	19200	57600
14.7456		Y	Y	Y	Y	
11.0592	Y	Y	Y	Y	Y	Y
9.21600	Y	Y	Y	Y		
7.37280	Y	Y	Y	Y		
5.52960	Y	Y	Y	Y		
1.84320	Y	Y	Y	Y		

**ADDITIONAL INFORMATION**

A complete description for all operational aspects of the DS2250(T) is provided in the User's Guide section of the Secure Microcontroller Data Book.

**DEVELOPMENT SUPPORT**

Dallas Semiconductor offers a kit package for developing and testing user code. The DS5000TK Evaluation

Kit allows the user to download Intel hex formatted code directly to the DS2250(T) from a PC-XT/AT or compatible computer. The kit consists of a DS5000T-32-12, an interface pod, demo software, and an RS232 connector that attaches to the COM1 or COM2 serial port of a PC. The kit can be used with a DS2250(T). A mechanical adaptor, the DS9075-40V, allows a DS2250(T) to be used in the DS5000TK. See the Secure Microcontroller User's Guide for further details.

**ABSOLUTE MAXIMUM RATINGS\***

Voltage on Any Pin Relative to Ground  
 Operating Temperature  
 Storage Temperature  
 Soldering Temperature

-0.3V to +7.0V  
 0°C to 70°C  
 -40°C to +70°C  
 260°C for 10 seconds

\* This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

**DC CHARACTERISTICS**(t<sub>A</sub> = 0°C to 70°C; V<sub>CC</sub> = 5V ± 5%)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Input Low Voltage	V <sub>IL</sub>	-0.3		+0.8	V	1
Input High Voltage	V <sub>IH1</sub>	2.0		V <sub>CC</sub> +0.3	V	1
Input High Voltage RST, XTAL1	V <sub>IH2</sub>	3.5		V <sub>CC</sub> +0.3	V	1
Output Low Voltage @ I <sub>OL</sub> =1.6 mA (Ports 1, 2, 3)	V <sub>OL1</sub>		0.15	0.45	V	
Output Low Voltage @ I <sub>OL</sub> =3.2 mA (Ports 0, ALE, PSEN)	V <sub>OL2</sub>		0.15	0.45	V	1
Output High Voltage @ I <sub>OH</sub> =-80 μA (Ports 1, 2, 3)	V <sub>OH1</sub>	2.4	4.8		V	1
Output High Voltage @ I <sub>OH</sub> =-400 μA (Ports 0, ALE, PSEN)	V <sub>OH2</sub>	2.4	4.8		V	1
Input Low Current V <sub>IN</sub> = 0.45V (Ports 1, 2, 3)	I <sub>IL</sub>			-50	μA	
Transition Current; 1 to 0 V <sub>IN</sub> = 2.0V (Ports 1, 2, 3)	I <sub>TL</sub>			-500	μA	
Input Leakage Current 0.45 < V <sub>IN</sub> < V <sub>CC</sub> (Port 0)	I <sub>L</sub>			±10	μA	
RST, EA Pulldown Resistor	R <sub>RE</sub>	40		125	KΩ	
Stop Mode Current	I <sub>SM</sub>			80	μA	4
Power Fail Warning Voltage	V <sub>PFW</sub>	4.15	4.6	4.75	V	1
Minimum Operating Voltage	V <sub>CCmin</sub>	4.05	4.5	4.65	V	1
Programming Supply Voltage (Parallel Program Mode)	V <sub>PP</sub>	12.5		13	V	1
Program Supply Current	I <sub>PP</sub>		15	20	mA	
Operating Current DS2250-8K DS2250-32K @ 12 MHz DS2250(T)-64-16 @ 16 MHz	I <sub>CC</sub>			43 48 54	mA	2
Idle Mode Current @ 8 MHz	I <sub>CC</sub>			6.2	mA	3

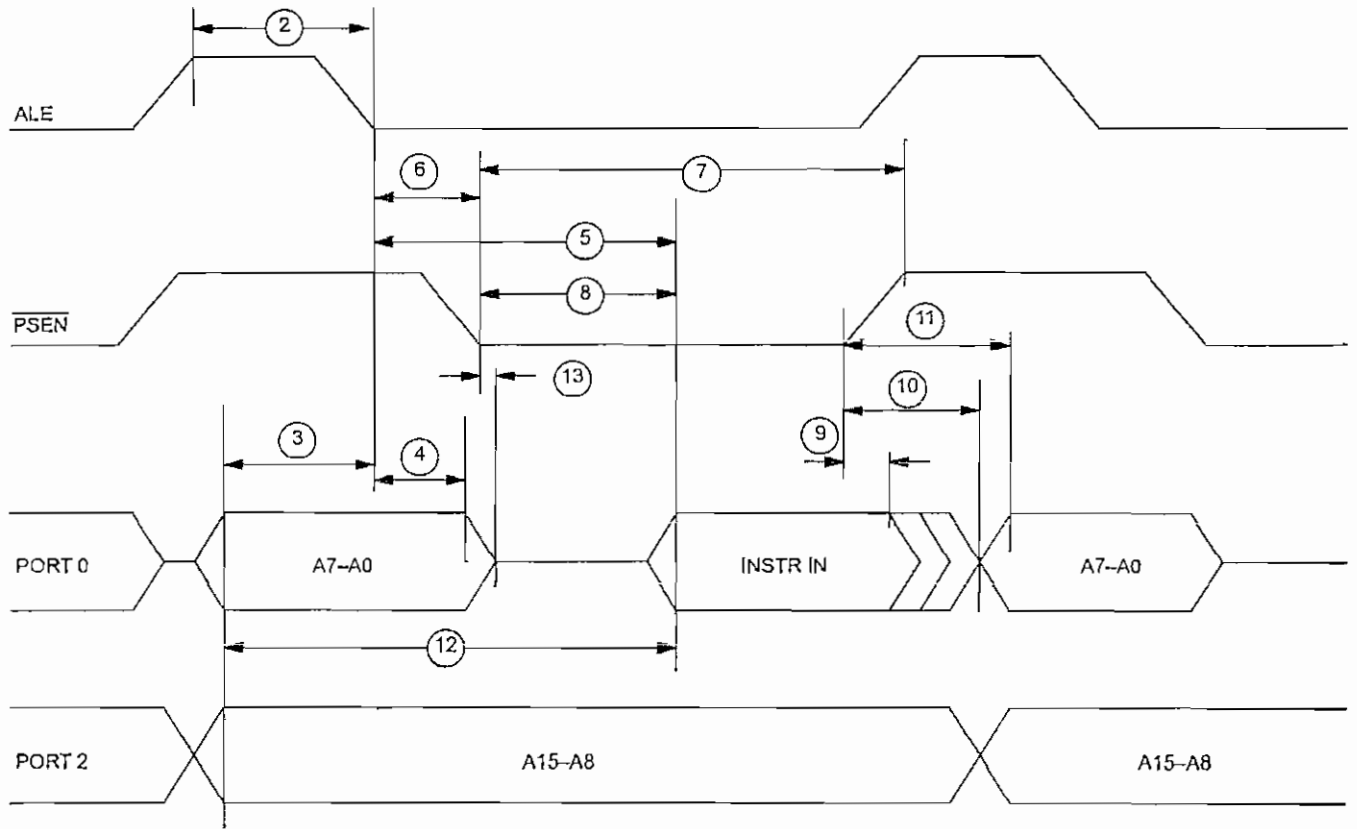
## AC CHARACTERISTICS

### EXPANDED BUS MODE TIMING SPECIFICATIONS

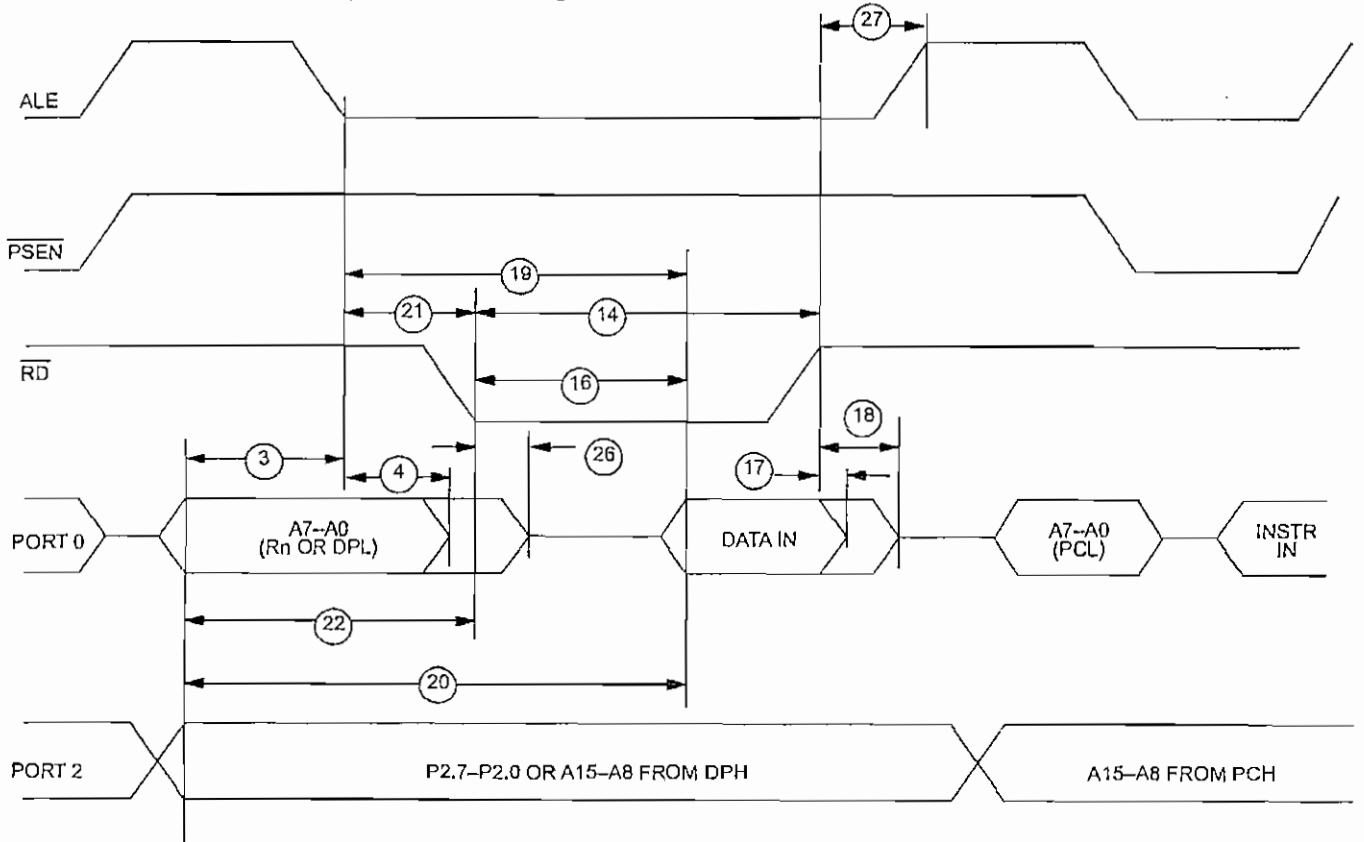
 $(t_A = 0^\circ\text{C to } 70^\circ\text{C}; V_{CC} = 5\text{V} \pm 5\%)$ 

#	PARAMETER	SYMBOL	MIN	MAX	UNITS
1	Oscillator Frequency	$1/t_{CLK}$	1.0	16 (-16)	MHz
2	ALE Pulse Width	$t_{ALPW}$	$2t_{CLK} - 40$		ns
3	Address Valid to ALE Low	$t_{AVALL}$	$t_{CLK} - 40$		ns
4	Address Hold After ALE Low	$t_{AVAAV}$	$t_{CLK} - 35$		ns
5	ALE Low to Valid Instr. In @12 MHz @16 MHz	$t_{ALLVI}$		$4t_{CLK} - 150$ $4t_{CLK} - 90$	ns
6	ALE Low to PSEN Low	$t_{ALLPSL}$	$t_{CLK} - 25$		ns
7	PSEN Pulse Width	$t_{PSPW}$	$3t_{CLK} - 35$		ns
8	PSEN Low to Valid Instr. In @12 MHz @16 MHz	$t_{PSLVI}$		$3t_{CLK} - 150$ $3t_{CLK} - 90$	ns ns
9	Input Instr. Hold after PSEN Going High	$t_{PSIV}$	0		ns
10	Input Instr. Float after PSEN Going High	$t_{PSIX}$		$t_{CLK} - 20$	ns
11	Address Hold after PSEN Going High	$t_{PSAV}$	$t_{CLK} - 8$		ns
12	Address Valid to Valid Instr. In @12 MHz @16 MHz	$t_{AVVI}$		$5t_{CLK} - 150$ $5t_{CLK} - 90$	ns ns
13	PSEN Low to Address Float	$t_{PSLAZ}$	0		ns
14	RD Pulse Width	$t_{RDPW}$	$6t_{CLK} - 100$		ns
15	WR Pulse Width	$t_{WRPW}$	$6t_{CLK} - 100$		ns
16	RD Low to Valid Data In @12 MHz @16 MHz	$t_{RDLDV}$		$5t_{CLK} - 165$ $5t_{CLK} - 105$	ns ns
17	Data Hold after RD High	$t_{RDHDV}$	0		ns
18	Data Float after RD High	$t_{RDHDZ}$		$2t_{CLK} - 70$	ns
19	ALE Low to Valid Data In @12 MHz @16 MHz	$t_{ALLVD}$		$8t_{CLK} - 150$ $8t_{CLK} - 90$	ns ns
20	Valid Addr. to Valid Data In @12 MHz @16 MHz	$t_{AVDV}$		$9t_{CLK} - 165$ $9t_{CLK} - 105$	ns ns
21	ALE Low to RD or WR Low	$t_{ALLRDL}$	$3t_{CLK} - 50$	$3t_{CLK} + 50$	ns
22	Address Valid to RD or WR Low	$t_{AVRDL}$	$4t_{CLK} - 130$		ns
23	Data Valid to WR Going Low	$t_{DVWRL}$	$t_{CLK} - 60$		ns
24	Data Valid to WR High @12 MHz @16 MHz	$t_{DVWRH}$	$7t_{CLK} - 150$ $7t_{CLK} - 90$		ns ns
25	Data Valid after WR High	$t_{WRHDV}$	$t_{CLK} - 50$		ns
26	RD Low to Address Float	$t_{RDLAZ}$		0	ns
27	RD or WR High to ALE High	$t_{RDHALH}$	$t_{CLK} - 40$	$t_{CLK} + 50$	ns

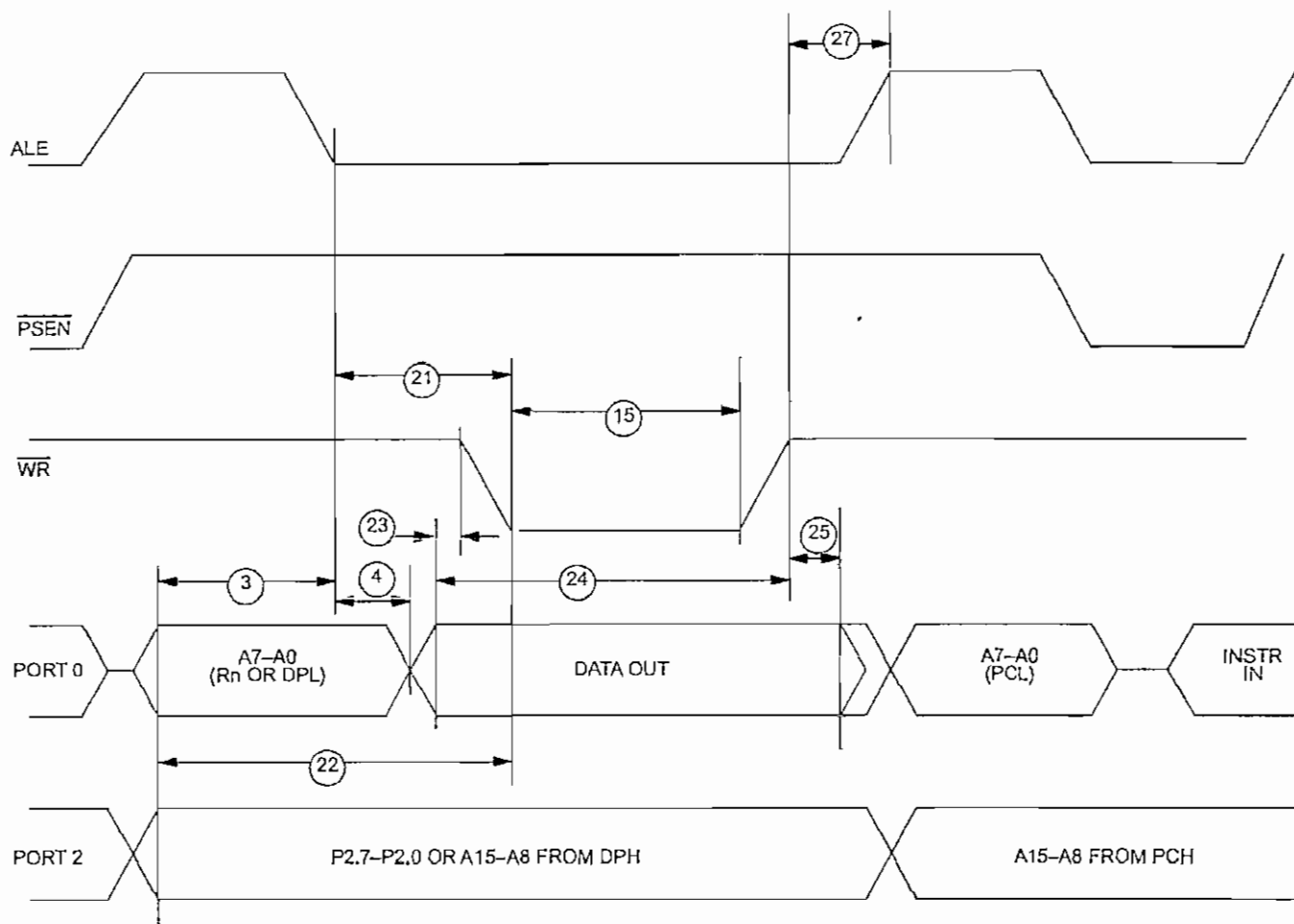
EXPANDED PROGRAM MEMORY READ CYCLE



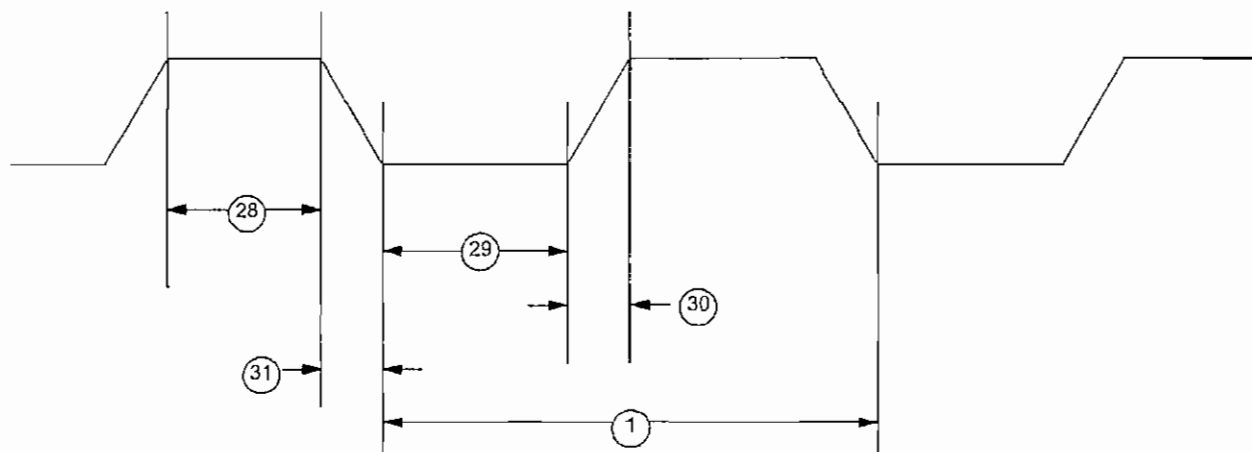
EXPANDED DATA MEMORY READ CYCLE



## EXPANDED DATA MEMORY WRITE CYCLE



## EXTERNAL CLOCK TIMING



## AC CHARACTERISTICS (cont'd)

## EXTERNAL CLOCK DRIVE

 $(t_A = 0^\circ\text{C to } 70^\circ\text{C}; V_{CC} = 5V \pm 5\%)$ 

#	PARAMETER	SYMBOL	MIN	MAX	UNITS
28	External Clock High Time	$t_{CLKHPW}$	@12 MHz	20	ns
			@16 MHz	15	
29	External Clock Low Time	$t_{CLKLPW}$	@12 MHz	20	ns
			@16 MHz	15	
30	External Clock Rise Time	$t_{CLKR}$	@12 MHz	20	ns
			@16 MHz	15	
31	External Clock Fall Time	$t_{CLKF}$	@12 MHz	20	ns
			@16 MHz	15	

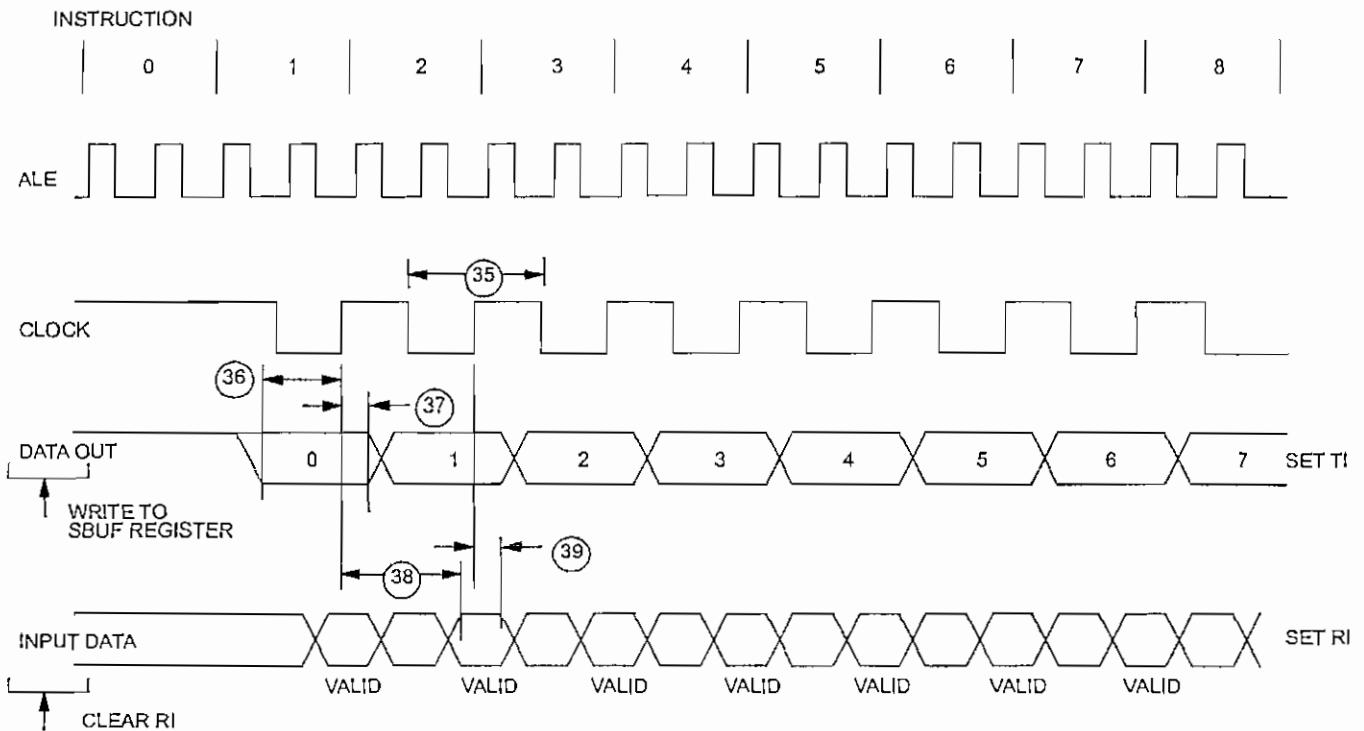
## AC CHARACTERISTICS (cont'd)

## SERIAL PORT TIMING – MODE 0

 $(t_A = 0^\circ\text{C to } 70^\circ\text{C}; V_{CC} = 5V \pm 5\%)$ 

#	PARAMETER	SYMBOL	MIN	MAX	UNITS
35	Serial Port Cycle Time	$t_{SPCLK}$	$12t_{CLK}$		$\mu\text{s}$
36	Output Data Setup to Rising Clock Edge	$t_{DOCH}$	$10t_{CLK} - 133$		ns
37	Output Data Hold after Rising Clock Edge	$t_{CHDO}$	$2t_{CLK} - 117$		ns
38	Clock Rising Edge to Input Data Valid	$t_{CHDV}$		$10t_{CLK} - 133$	ns
39	Input Data Hold after Rising Clock Edge	$t_{CHDIV}$	0		ns

## SERIAL PORT TIMING – MODE 0





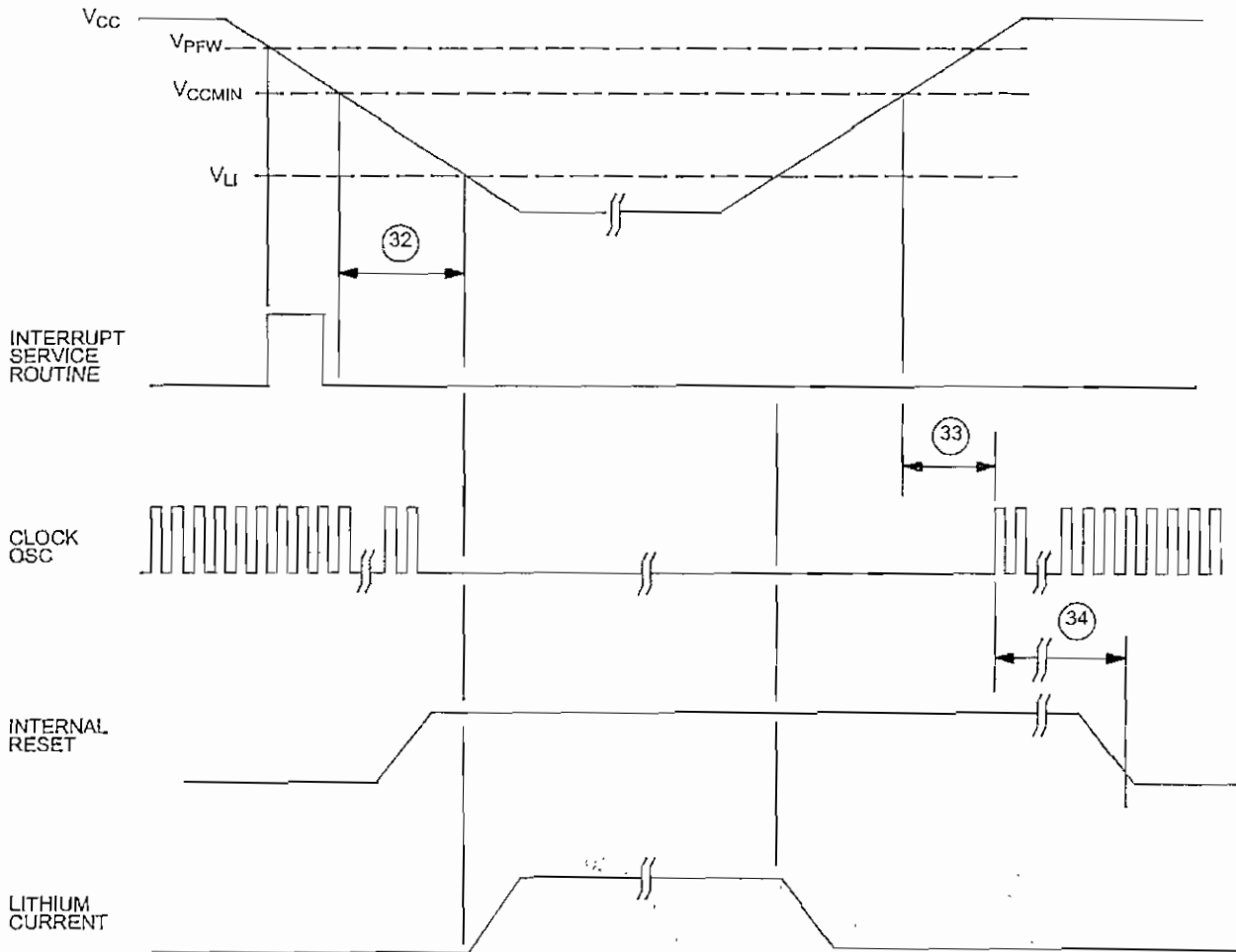
# AC CHARACTERISTICS (cont'd)

## POWER CYCLING TIMING

( $t_A = 0^\circ\text{C}$  to  $70^\circ\text{C}$ ;  $V_{CC} = 5\text{V} \pm 5\%$ )

#	PARAMETER	SYMBOL	MIN	MAX	UNITS
32	Slew Rate from $V_{CCmin}$ to 3.3V	$t_F$	40		$\mu\text{s}$
33	Crystal Start-up Time	$t_{CSU}$		(note 5)	
34	Power-On Reset Delay	$t_{POR}$		21504	$t_{CLK}$

## POWER CYCLE TIMING

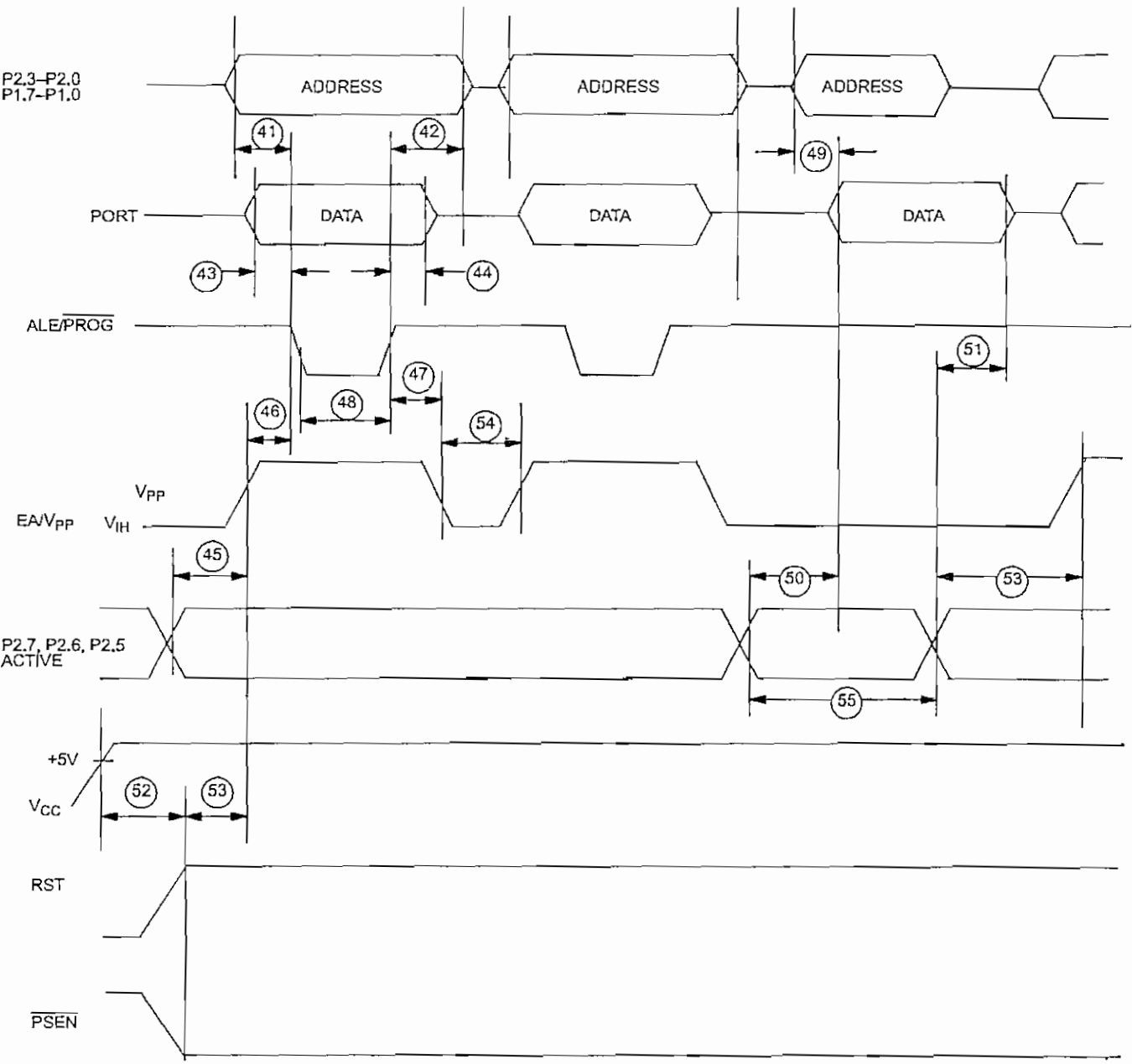


**AC CHARACTERISTICS (cont'd)**  
**PARALLEL PROGRAM LOAD TIMING**
 $(t_A = 0^\circ\text{C to } 70^\circ\text{C}; V_{CC} = 5\text{V} \pm 5\%)$ 

#	PARAMETER	SYMBOL	MIN	MAX	UNITS
40	Oscillator Frequency	$1/t_{CLK}$	1.0	12.0	MHz
41	Address Setup to $\overline{PROG}$ Low	$t_{AVPRL}$	0		
42	Address Hold after $\overline{PROG}$ High	$t_{PRHAV}$	0		
43	Data Setup to $\overline{PROG}$ Low	$t_{DVPRL}$	0		
44	Data Hold after $\overline{PROG}$ High	$t_{PRHDV}$	0		
45	P2.7, 2.6, 2.5 Setup to $V_{PP}$	$t_{P27HVP}$	0		
46	$V_{PP}$ Setup to $\overline{PROG}$ Low	$t_{VPHPRL}$	0		
47	$V_{PP}$ Hold after $\overline{PROG}$ Low	$t_{PRHVPL}$	0		
48	$\overline{PROG}$ Width Low	$t_{PRW}$	2400		$t_{CLK}$
49	Data Output from Address Valid	$t_{AVDV}$		48 1800*	$t_{CLK}$
50	Data Output from P2.7 Low	$t_{DVP27L}$		48 1800*	$t_{CLK}$
51	Data Float after P2.7 High	$t_{P27HDZ}$	0	48 1800*	$t_{CLK}$
52	Delay to Reset/ $\overline{PSEN}$ Active after Power On	$t_{PORPV}$	21504		$t_{CLK}$
53	Reset/ $\overline{PSEN}$ Active (or Verify Inactive) to $V_{PP}$ High	$t_{RAVPH}$	1200		$t_{CLK}$
54	$V_{PP}$ Inactive (Between Program Cycles)	$t_{VPPPC}$	1200		$t_{CLK}$
55	Verify Active Time	$t_{VFT}$	48 2400*		$t_{CLK}$

\* Second set of numbers refers to expanded memory programming up to 32K bytes.

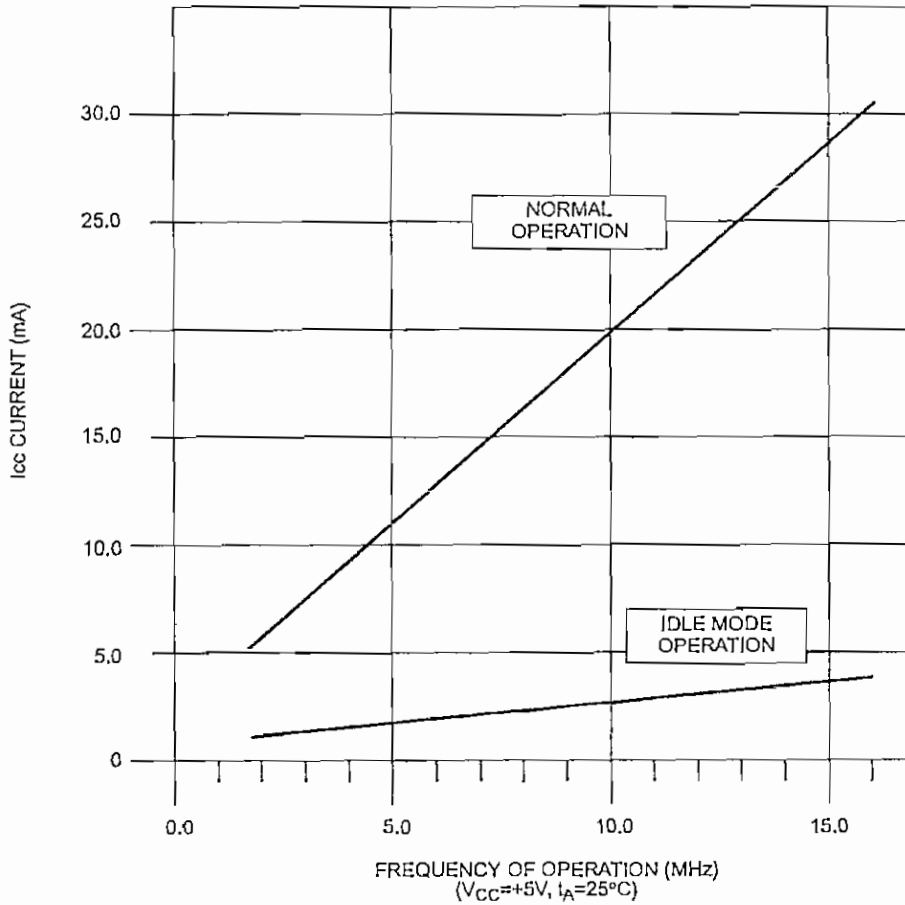
# PARALLEL PROGRAM LOAD TIMING



## CAPACITANCE

(test frequency = 1 MHz;  $t_A = 25^\circ\text{C}$ )

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Output Capacitance	$C_O$			10	pF	
Input Capacitance	$C_I$			10	pF	

DS2250(T) TYPICAL  $I_{CC}$  VS. FREQUENCY

Normal operation is measured using:

- 1) External crystals on XTAL1 and 2
- 2) All port pins disconnected
- 3) RST=0 volts and EA= $V_{CC}$
- 4) Part performing endless loop writing to internal memory

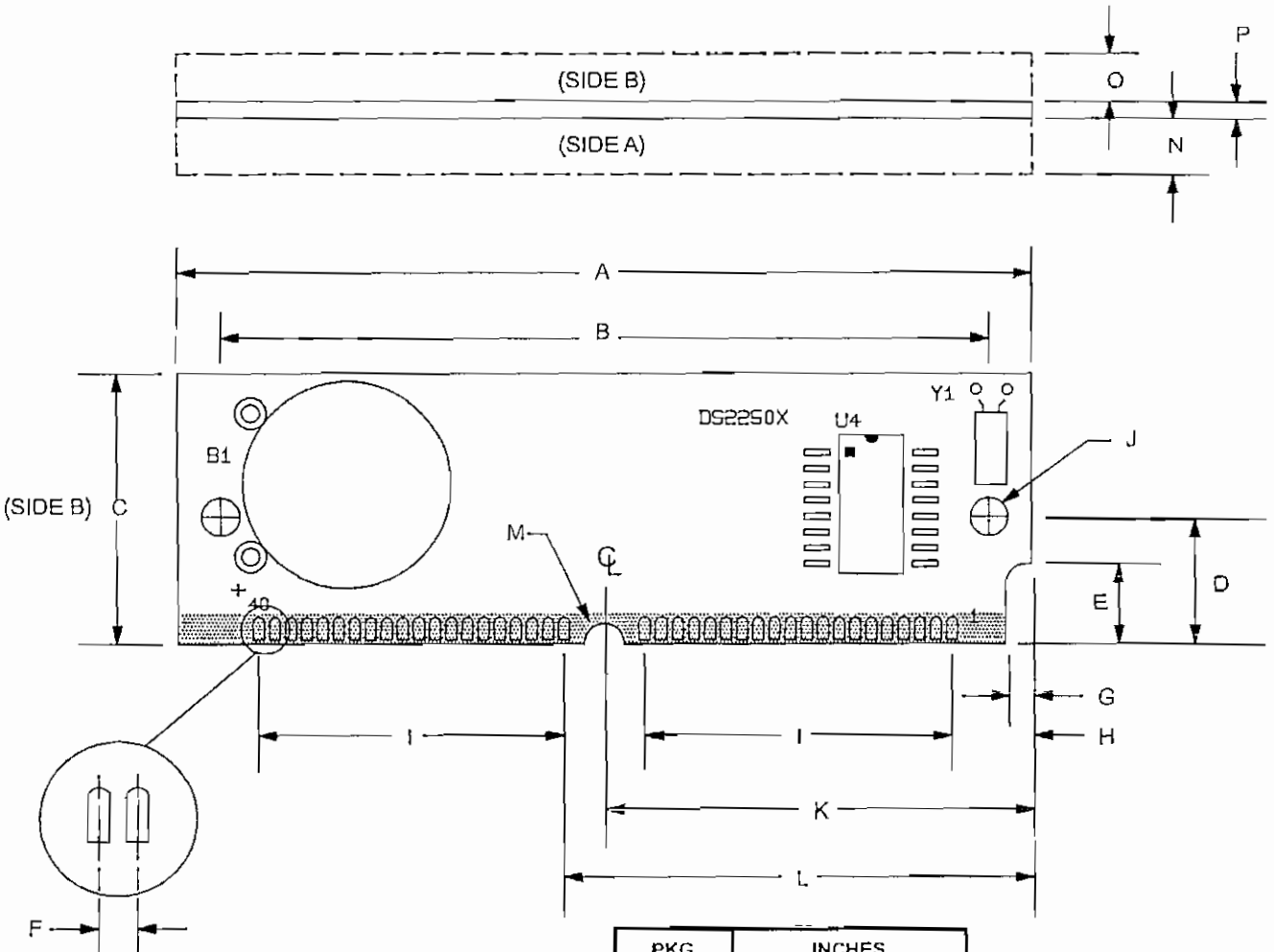
Idle mode operation is measured using:

- 1) External clock source at XTAL1; XTAL2 floating
- 2) All port pins disconnected
- 3) RST=0 volts and EA= $V_{CC}$
- 4) Part set in IDLE mode by software

#### NOTES:

1. All voltages are referenced to ground.
2. Maximum operating  $I_{CC}$  is measured with all output pins disconnected; XTAL1 driven with  $t_{CLKR}$ ,  $t_{CLKF}=10$  ns,  $V_{IL}=0.5$ V; XTAL2 disconnected;  $\overline{EA}=RST=PORT0=V_{CC}$ .
3. Idle mode  $I_{CC}$  is measured with all output pins disconnected; XTAL1 driven at 8 MHz with  $t_{CLKR}$ ,  $t_{CLKF}=10$  ns,  $V_{IL}=0.5$ V; XTAL2 disconnected;  $\overline{EA}=PORT0=V_{CC}$ , RST =  $V_{SS}$ .
4. Stop mode  $I_{CC}$  is measured with all output pins disconnected;  $\overline{EA}=PORT0=V_{CC}$ ; XTAL2 not connected; RST =  $V_{SS}$ .
5. Crystal start-up time is the time required to get the mass of the crystal into vibrational motion from the time that power is first applied to the circuit until the first clock pulse is produced by the on-chip oscillator. The user should check with the crystal vendor for the worst case spec on this time.

PACKAGE DRAWING



PKG DIM	INCHES	
	MIN	MAX
A	2.645	2.655
B	2.379	2.389
C	0.845	0.855
D	0.395	0.405
E	0.245	0.255
F	0.050 BSC	
G	0.075	0.085
H	0.245	0.255
I	0.950 BSC	
J	0.120	0.130
K	1.320	1.330
L	1.445	1.455
M	0.057	0.067
N	-	0.160
O	-	0.195
P	0.047	0.054

## DATA SHEET REVISION SUMMARY

The following represent the key differences between 12/13/95 and 08/16/96 version of the DS2250(T) data sheet. Please review this summary carefully.

1. Correct Figure 3 to show RST active high.
2. Add minimum value to PCB thickness.

# LM074L

- 16 character × 2 lines
- Controller LSI HD44780 is built-in (see section 6).
- +5V single power supply
- Large numeral font set (see page 186).

## MECHANICAL DATA (Nominal dimensions)

Module size	84W x 44H x 12T (max.) mm
Effective display area	61W x 15.8H mm
Character size (5 × 7 dots)	2.96W x 4.86H mm
Character pitch	3.55 mm
Dot size	0.56W x 0.66H mm
Weight	about 35 g

ABSOLUTE MAXIMUM RATINGS	min.	max.
Power supply for logic ( $V_{DD}-V_{SS}$ )	0	7.0 V
Power supply for LCD drive ( $V_{DD}-V_O$ )	0	13.5 V
Input voltage ( $V_i$ )	$V_{SS}$	$V_{DD}$ V
Operating temperature ( $T_a$ )	0	50°C
Storage temperature ( $T_{stg}$ )	-20	70°C

## ELECTRICAL CHARACTERISTICS

$T_a = 25^\circ\text{C}$ , $V_{DD} = 5.0 \text{ V} \pm 0.25 \text{ V}$	
Input "high" voltage ( $V_{iH}$ )	2.2 V min.
Input "low" voltage ( $V_{iL}$ )	0.6 V max.
Output "high" voltage ( $V_{OH}$ ) ( $I_{OH} = 0.2 \text{ mA}$ )	2.4 V min.
Output "low" voltage ( $V_{OL}$ ) ( $I_{OL} = 1.2 \text{ mA}$ )	0.4 V max.
Power supply current ( $I_{DD}$ ) ( $V_{DD} = 5.0 \text{ V}$ )	1.0 mA typ. 3.0 mA max.
Power supply for LCD drive (Recommended) ( $V_{DD}-V_O$ )	Duty = 1/16
$T_a = 0^\circ\text{C}$	4.6 V typ.
$T_a = 25^\circ\text{C}$	4.4 V typ.
$T_a = 50^\circ\text{C}$	4.2 V typ.

OPTICAL DATA . . . . . See page 15.

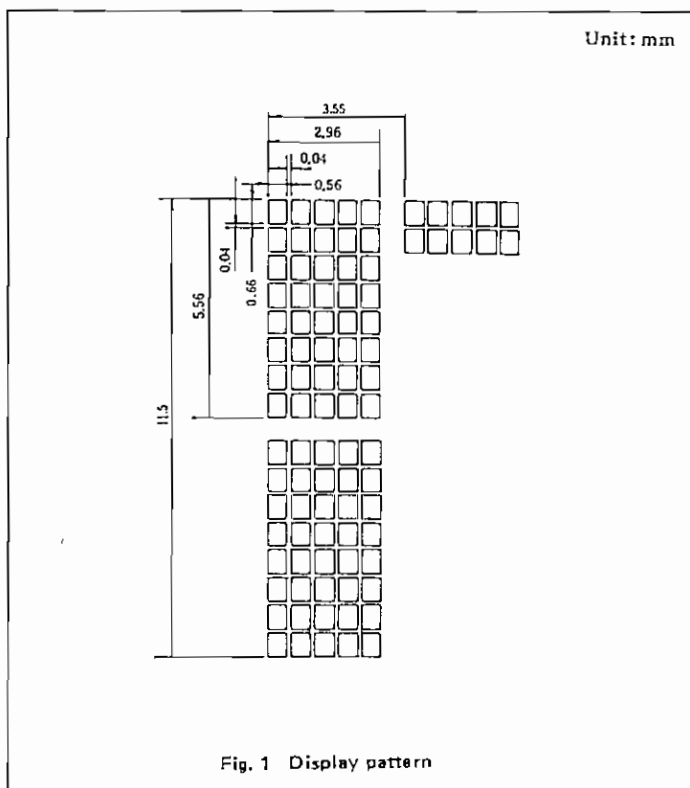
## INTERNAL PIN CONNECTION

Pin No.	Symbol	Level	Function
1	$V_{SS}$	--	0V
2	$V_{DD}$	-	+5V
3	$V_O$	-	-
4	RS	H/L	L: instruction code input H: Data input
5	R/W	H/L	H: Data read (LCD module → MPU) L: Data write (LCD module ← MPU)
6	E	H, H→L	Enable signal
7	DB0	H/L	Data bus line Note (1), (2)
8	DB1	H/L	
9	DB2	H/L	
10	DB3	H/L	
11	DB4	H/L	
12	DB5	H/L	
13	DB6	H/L	
14	DB7	H/L	

### Notes:

In the HD44780, the data can be sent in either 4-bit 2-operation or 8-bit 1-operation so that it can interface to both 4 and 8 bit MPU's.

- (1) When interface data is 4 bits long, data is transferred using only 4 buses of  $DB_4 \sim DB_7$  and  $DB_0 \sim DB_3$  are not used. Data transfer between the HD44780 and the MPU completes when 4-bit data is transferred twice. Data of the higher order 4 bits (contents of  $DB_4 \sim DB_7$  when interface data is 8 bits long) is transferred first and then lower order 4 bits (contents of  $DB_0 \sim DB_3$  when interface data is 8 bits long).
- (2) When interface data is 8 bits long, data is transferred using 8 data buses of  $DB_0 \sim DB_7$ .



Unit: mm

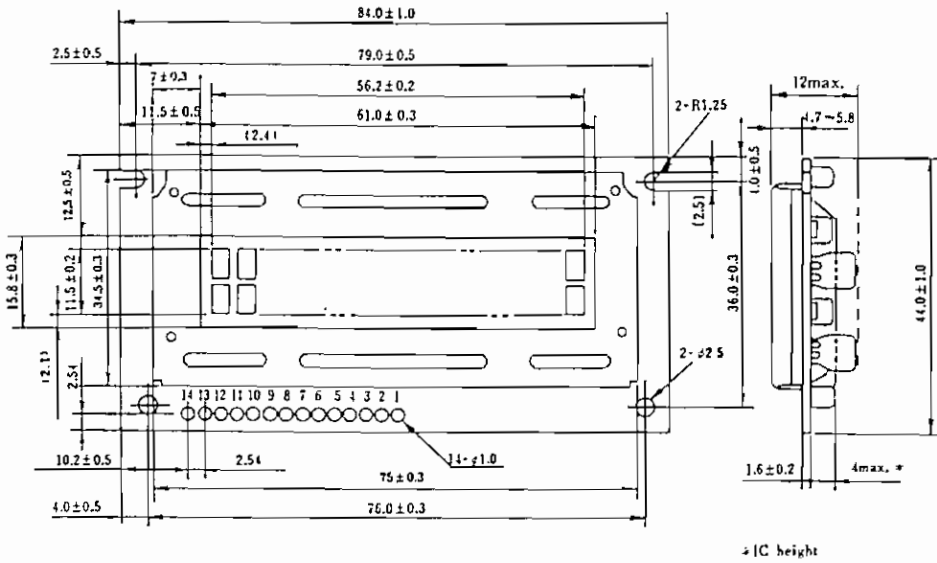


Fig. 2 External dimensions

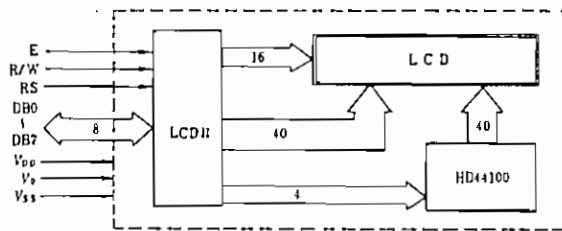


Fig. 3 Block diagram

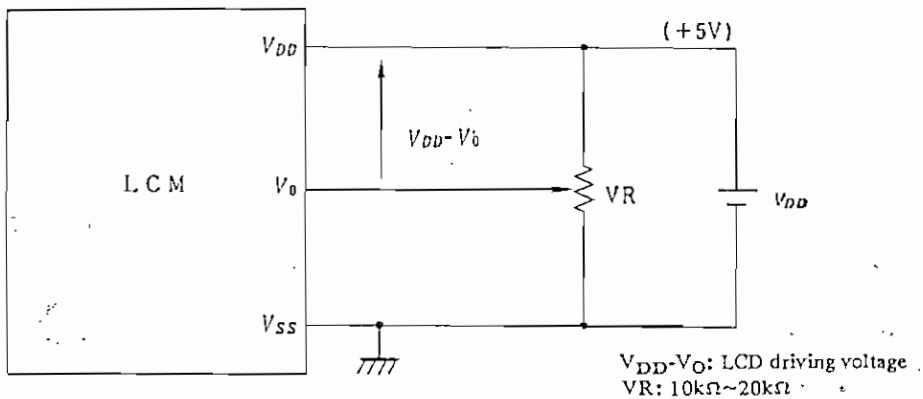


Fig. 4 Power supply



## TIMING CHARACTERISTICS

Item	Symbol	Test condition	Min.	Typ.	Max.	Unit
Enable cycle time	$t_{cyc}$	Fig. 5, Fig. 6	1.0	—	—	$\mu s$
Enable pulse width	$P_{WEH}$	Fig. 5, Fig. 6	450	—	—	ns
Enable rise/fall time	$t_{Er}, t_{Ef}$	Fig. 5, Fig. 6	—	—	25	ns
RS, R/W set up time	$t_{AS}$	Fig. 5, Fig. 6	140	—	—	ns
Data delay time	$t_{DDR}$	Fig. 6	—	—	320	ns
Data set up time	$t_{DSW}$	Fig. 5	195	—	—	ns
Hold time	$t_H$	Fig. 5, Fig. 6	20	—	—	ns

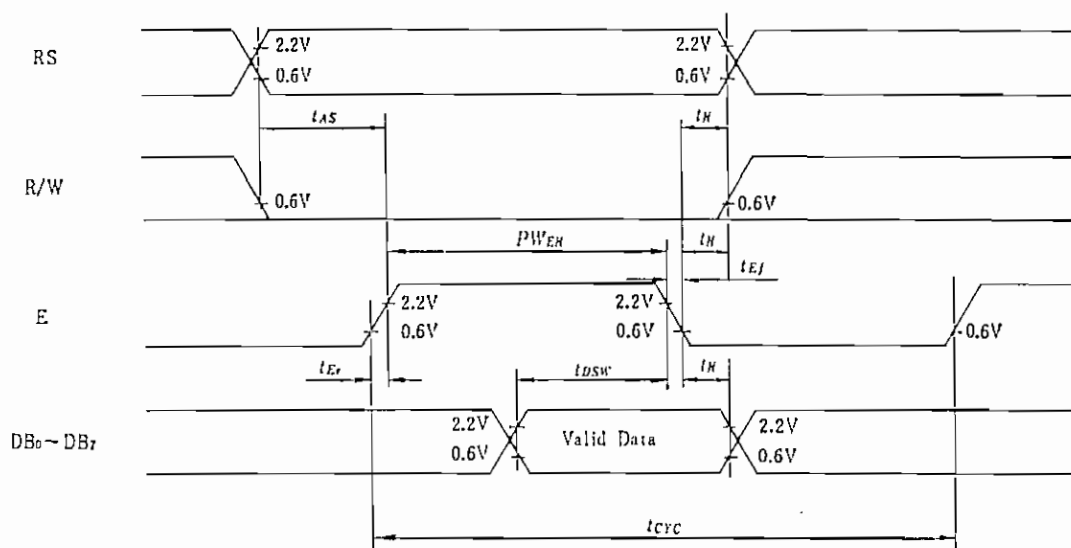


Fig. 5 Interface timing (data write)

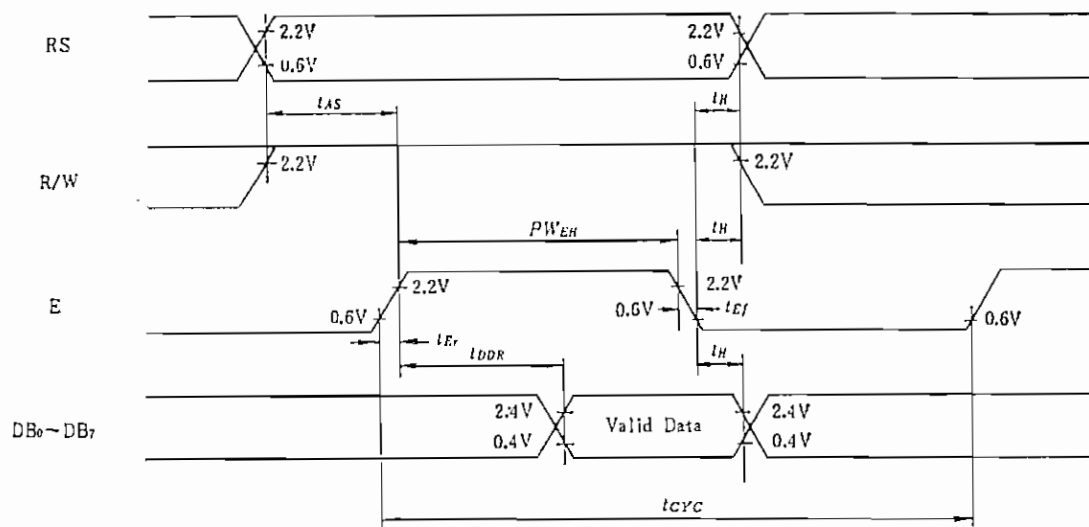


Fig. 6 Interface timing (data read)

Higher Lower 4bit 4bit	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
XXXX0000	CG RAM (1)												
XXXX0001	(2)												
XXXX0010	(3)												
XXXX0011	(4)												
XXXX0100	(5)												
XXXX0101	(6)												
XXXX0110	(7)												
XXXX0111	(8)												
XXXX1000	(1)												
XXXX1001	(2)												
XXXX1010	(3)												
XXXX1011	(4)												
XXXX1100	(5)												
XXXX1101	(6)												
XXXX1110	(7)												
XXXX1111	(8)												

Fig. 7 Character dot pattern

# HOW TO USE HITACHI'S BUILT-IN CONTROLLER DRIVER

## LCD-II (HD44780) DOT MATRIX LCD MODULE

### ■ INTRODUCTION

The LCD-II (HD44780) is a dot matrix liquid crystal display controller & driver LSI that displays alphanumeric, kana characters and symbols. It drives dot matrix liquid crystal display under 4-bit or 8-bit microcomputer or microprocessor control. All the functions required for dot matrix liquid crystal display drive are internally provided on one chip. The user can complete dot matrix liquid crystal display systems with less number of chips by using the LCD-II (HD44780). If a driver LSI HD44100H is externally connected to the HD44780, up to 80 characters can be displayed. The LCD-II is produced in the CMOS process. Therefore, the combination of the LCD-II with a CMOS microcomputer or microprocessor can accomplish a portable battery-drive device with lower power dissipation.

### ■ FEATURES

- Capable of interfacing to 4-bit or 8-bit MPU.
- Display data RAM . . . . . 80 x 8 bits  
(80 characters, max.)
- Character generator ROM . . . .  
Character font 5 x 7 dots: 160 characters  
Character font 5 x 10 dots: 32 characters
- Both display data and character generator RAMs can be read from the MPU.
- Wide range of instruction functions  
Display clear, Cursor home, Display ON/OFF, Cursor ON/OFF, Display character blink, Cursor shift, Display shift
- Internal automatic reset circuit at power ON. (Internal reset circuit)

### 1. Applicable type

- (1) 1 line series  
LM054 • H2570 • LM015 • LM568AF • LM020L • LM070L • LM038 • LM027 • H2571 • LM058
- (2) 2 line series  
LM052L • LM016L • LM032L • LM060L • LM017L • LM018L • LM075L • LM074L • LM068L • LM061L
- (3) 4 lines series  
LM041L • LM044L
- (4) Compact version  
LM104L • LM105L • LM107L

### 2. Connecting MPU with LCM

#### 2.1 Driver circuit block diagram

Figure 1 shows the driver circuit block diagram of LCM with built-in controller LSI. Controller LSI HD44780 (LCD-II) is built-in this LCM. Also extended LCD driver LSI is built in the LCM that displays more than 16 digits.

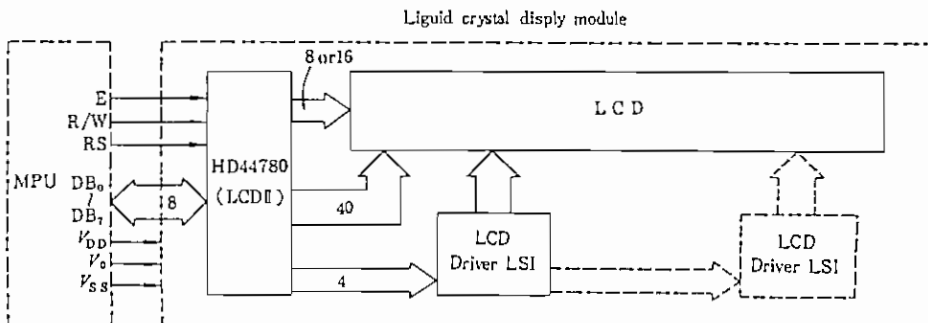


Fig. 1 Driver circuit block diagram

Interfacing to MPU

the HD44780, data can be sent in either 4-bit 2-operation 8-bit 1-operation so it can interface to both 4 and 8 bit U's.

When interface data is 4-bits long, data is transferred using only 4 buses: DB<sub>4</sub> ~ DB<sub>7</sub>. DB<sub>0</sub> ~ DB<sub>3</sub> are not used. Data transfer between the HD44780 and the MPU completes when 4-bit data is transferred twice. Data of the higher order 4 bits (contents of DB<sub>4</sub> ~ DB<sub>7</sub> when interface data is 8 bits long) is transferred first, then the

lower order 4 bits (content of DB<sub>0</sub> ~ DB<sub>3</sub> when interface data is 8 bits long) is transferred. Check the busy flag after 4-bit data has been transferred twice (one instruction). A 4-bit 2-operation will then transfer the busy flag and address counter data.

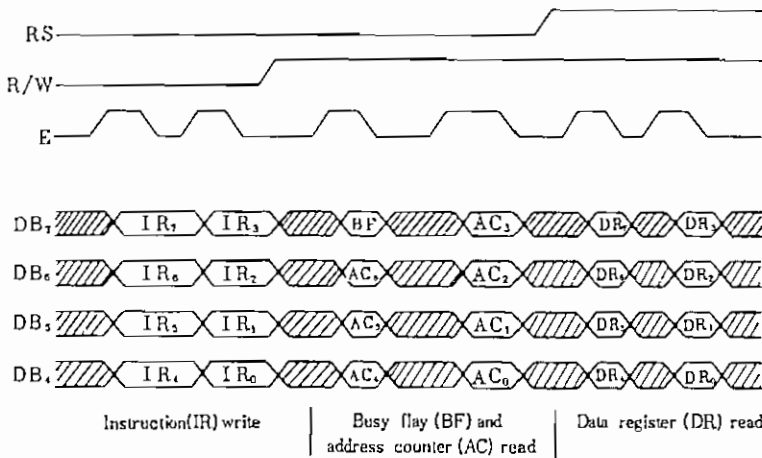


Fig. 2 4-bit data transfer example

When interface data is 8 bit long, data is transferred using the 8 data buses of DB<sub>0</sub> ~ DB<sub>7</sub>.

Interface to MPU  
Interface to 8-bit MPU

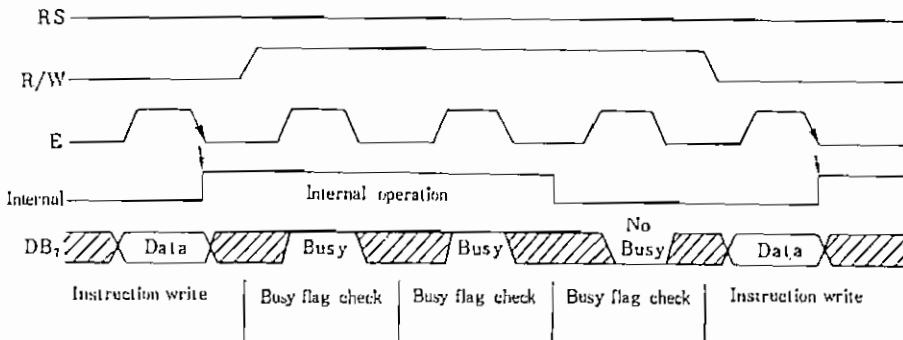


Fig. 3 Example of busy flag check timing sequence

When connecting to 8-bit MPU through PIA  
Fig. 4 is an example of using a PIA or I/O port (for single chip microcomputer) as an interface device. Input and output of the device is TTL compatible. In the example, PB<sub>0</sub> to PB<sub>7</sub> are connected to the data

buses DB<sub>0</sub> to DB<sub>7</sub> and PA<sub>0</sub> to PA<sub>2</sub> are connected to E, R/W and RS respectively. Pay attention to the timing relation between E and other signals when reading or writing data and using PIA as an interface.

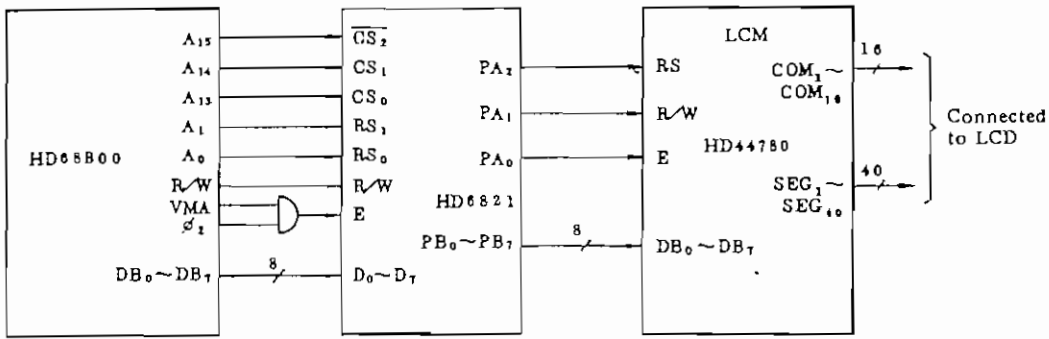
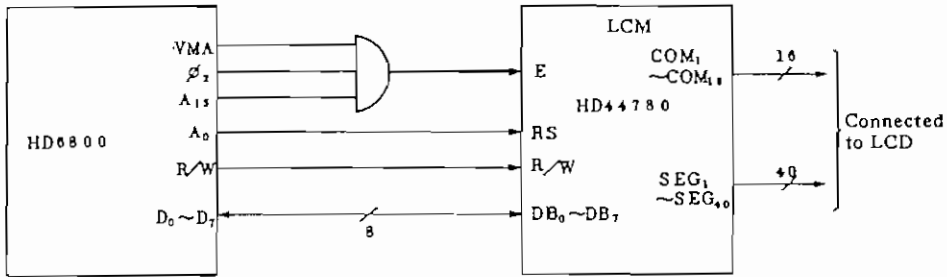
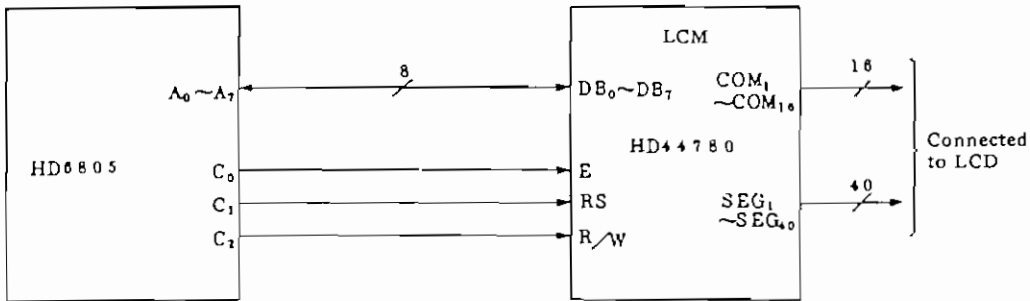


Fig. 4 Example of interface to HD68800 using PIA (HD6821)

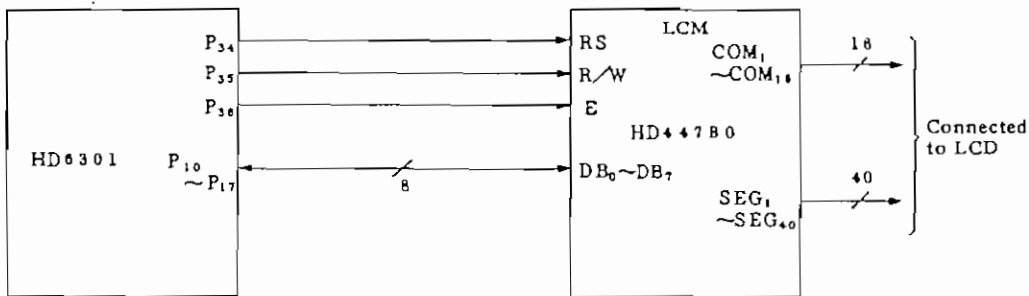
② Connecting directly to the 8 bit MPU bus line



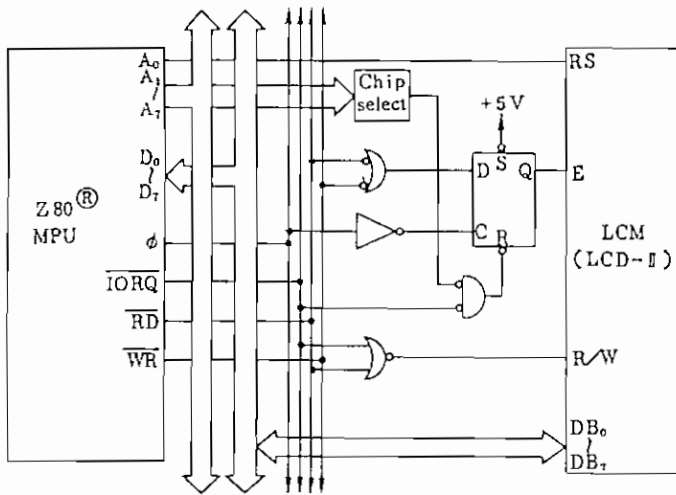
③ Example of interfacing to the HD6805



④ Example of interfacing to the HD6301



Example of interfacing to Z80 MPU



Note: 280 is the trademark of ZILOG, U.S.A.

Above circuit is an example of connection with Z80 MPU and HD44780A00 as an I/O equipment. It can be used as a part of memories by using MREQ signal.

- A0 signal can be used for RS signal.
- A0 = 0: Instruction register is selected.
- A0 = 1: Data register is selected.

(c) In order to check busy flag, transfer the data of DB<sub>0</sub> ~ DB<sub>7</sub> to A register (accumulator) by executing In/Out instruction. After that, busy flag can be easily checked by examining DB<sub>7</sub>.

Example of interfacing to 80 CPU family

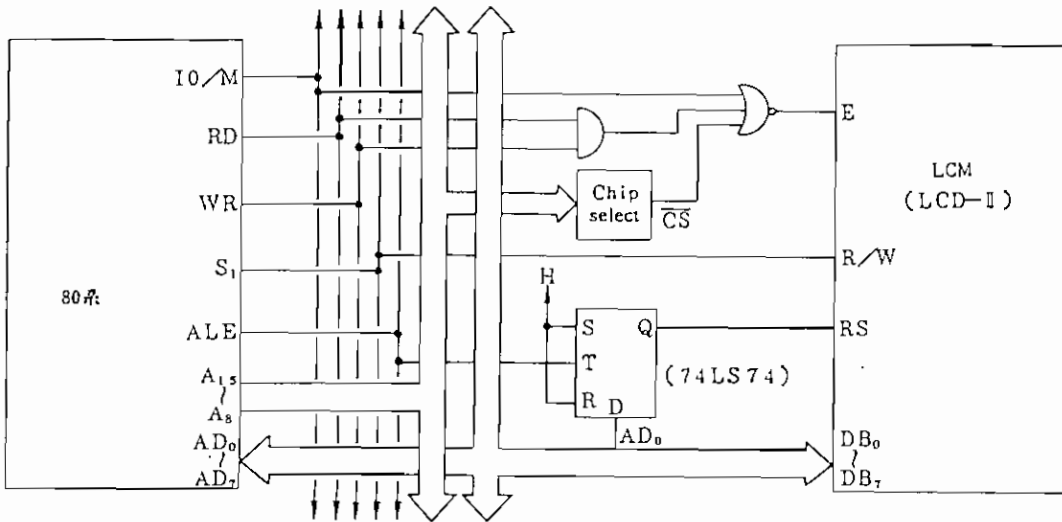


Fig. 5 Example of connection with LCM being used as a part of memories on the determined address.

Fig. 5 is an example of connection with LCD module used as a part of memories on the determined address.

It generates RS signal (Register Select signal) by latching the content of AD<sub>0</sub> at the rising edge of ALE signal. By using this method, you can obtain RS signal from the content among 8 bit addresses generated at the clock of the machine cycle. In case of using LCD module as an

I/O equipment, chip select signal is necessarily activated when IO/M signal is "High" level.

Furthermore, by using A8 for RS signal, the interface is easily realized.

By both methods, busy flag can be checked by storing status data into A register (Accumulator) and examining the bit 7 by software.

(2) Interface to 4-bit MPU

The HD44780 can be connected to a 4-bit MPU through the 4-bit MPU I/O port. If the I/O port has enough bits, data can be transferred in 8-bit lengths, but if the bits are insufficient, the transfer is made in two operations of 4 bits each (with designation of interface data length for 4 bits). In the latter case, the timing sequence becomes

somewhat complex. (See Fig. 6)

Fig. 7 shows an example of interface to the HMCS43C. Note that 2 cycles are needed for the busy flag check as well as the data transfer. 4-bit operation is selected by program.

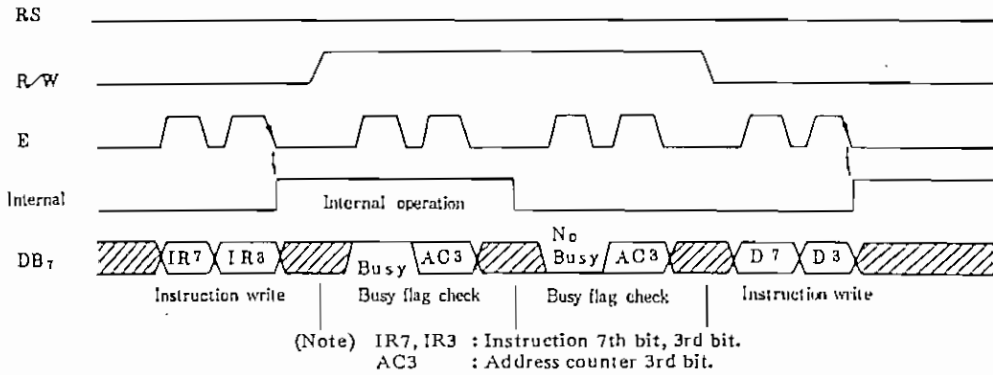


Fig. 6 An example of 4 bit data transfer timing sequence

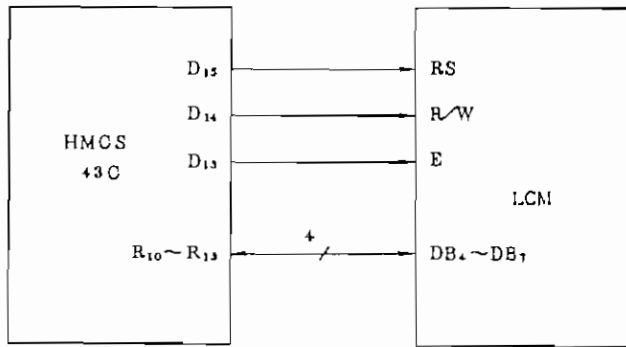


Fig. 7 Example of interface to the HMCS43C

3. Precautions on constituting hardwares

3.1 Chip select

HD44780 has no CS (chip select) terminals. Therefore, when this LSI is connected directly to Data Bus line not through PIA and so on, add the circuit that inhibits the output of Enable signal at the address which is not assigned for HD44780.

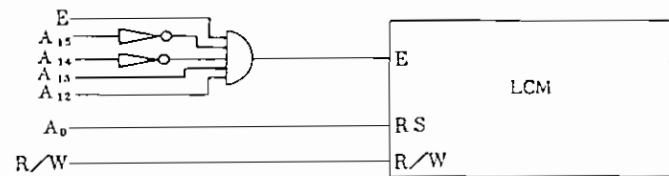


Fig. 8 Example of addresses  $(3000)_{16} \sim (3FFF)_{16}$  being assigned for HD44780

3.2 Ability of driving bus line

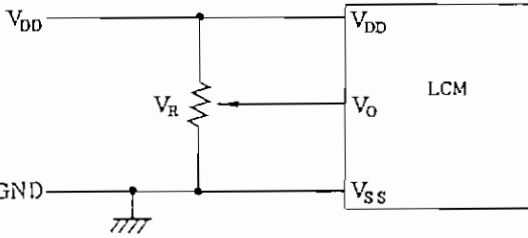
$DB_0$  to  $DB_7$  can drive one TTL or capacitance of 130 pF. The data bus terminals have three-state constructions and remain in high impedance state while Enable signal being low level.

Since the data bus has pull up MOS, it outputs high level voltage during the data bus being opened.

3.3 Power supply voltage for liquid crystal display drive

At Interface of liquid crystal display module, there are three power supply terminals,  $V_{DD}$ , GND, and  $V_0$ . LCD module is driven by the voltage that is equal to  $V_{DD} - V_0$ , when supplying power for liquid crystal display drive to  $V_0$  terminal. Since suitable voltage of power supply for LCD shifts according to temperature change adjust supplying power to LCD by referring to Fig. 9 or Fig. 10.

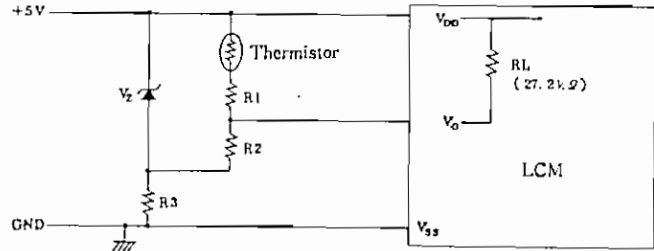
Example of variable driving voltage by a variable resistance (VR)  
 The driving voltage can be changed by VR to compensate the influence of surrounding temperature.



Recommended VR value = 10kΩ ~ 20kΩ

Fig. 9 Variable driving voltage circuit

(2) Example of a thermal compensator circuit  
 When setting the voltage, refer to Table-1



Thermistor { RT=15kΩ (Ta=25°C)  
 B=4300

Fig. 10 Example of a thermal compensator circuit

1

Duty	Recommended driving voltage		Typical circuit parameter			
	Ta (°C)	VDD - VO (V)	Vz (V)	R1 (kΩ)	R2 (kΩ)	R3 (kΩ)
1 / 8	0	4.0	4.5	2.2	2.8	1.0
	25	3.7				
	50	3.3				
1 / 11	0	4.3	4.5	2.2	3.2	0.3
	25	3.9				
	50	3.3				
1 / 16	0	4.6	5.0	0.1	1.3	0.1
	25	4.4				
	50	4.2				

Initialization

Initializing by internal reset circuit

HD44780 automatically initializes (resets) when power is on using the internal reset circuit. The following operations are executed in initialization. The busy flag (BF) is in busy state until initialization ends. (BF = 1) The busy flag returns to 0 10 ms after VCC rises to 4.5 V.

Display clear

Function set . . . . . DL = 1 : 8 bit long interface data  
 N = 0 : 1-line display  
 F = 0 : 5 x 7 dot character font

Display ON/OFF

Control . . . . . D = 0 : Display OFF  
 C = 0 : Cursor OFF  
 B = 0 : Blink OFF

Shift mode set . . . . I/D = 1 : +1 (increment)  
 S = 0 : No shift

(5) Write DD RAM

When the rise time of power supply (0.2 → 4.5) is out of the range 0.1 ms ~ 10 ms, or when the low level width of power OFF (less than 0.2 V) is less than 1 ms, the internal reset circuit will not operate normally.

In this case, initialization will not be performed normally. Initialize by MPU according to "4.2 initializing by instruction" at the head of program.

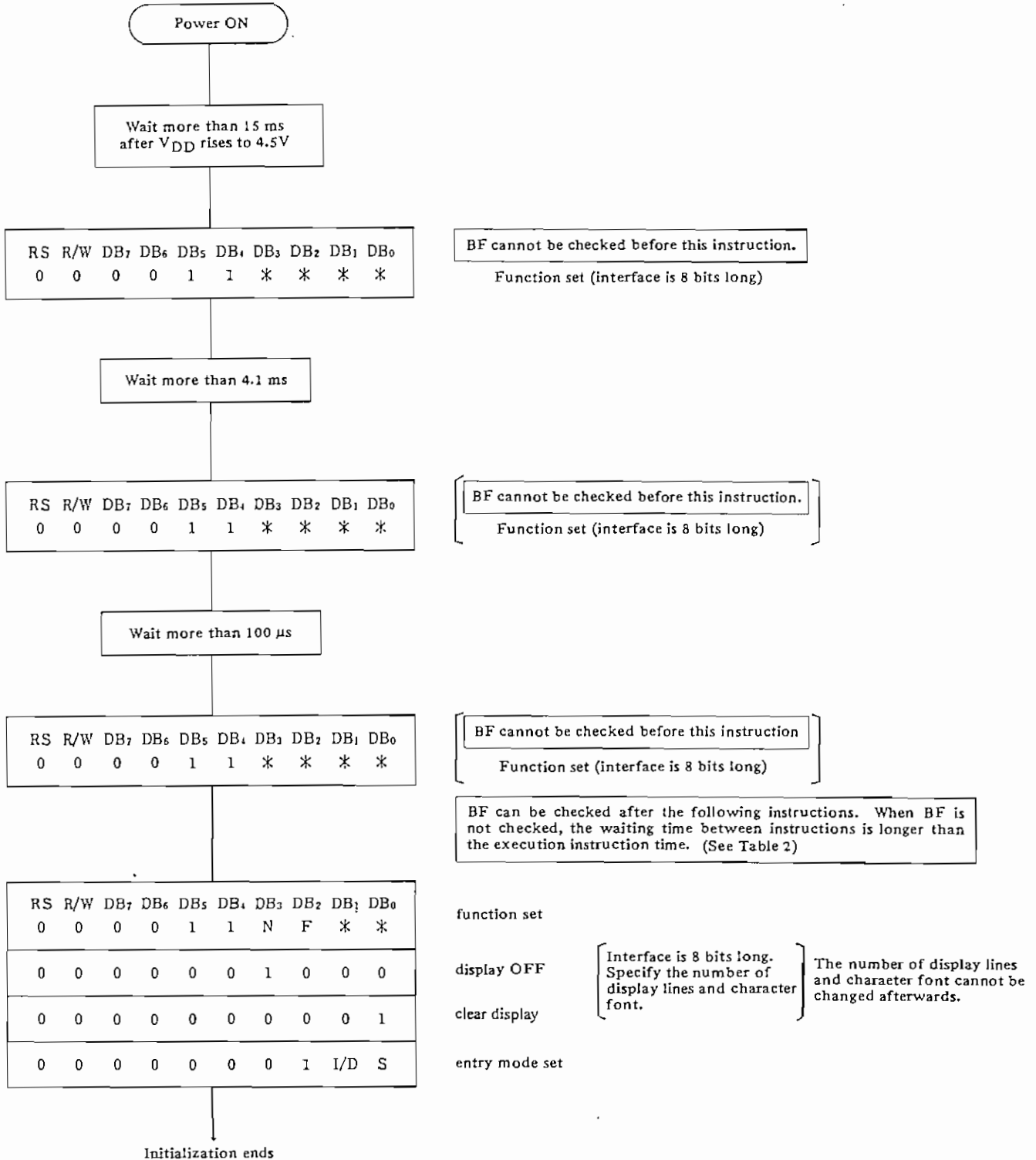


4.2 Initializing by instruction

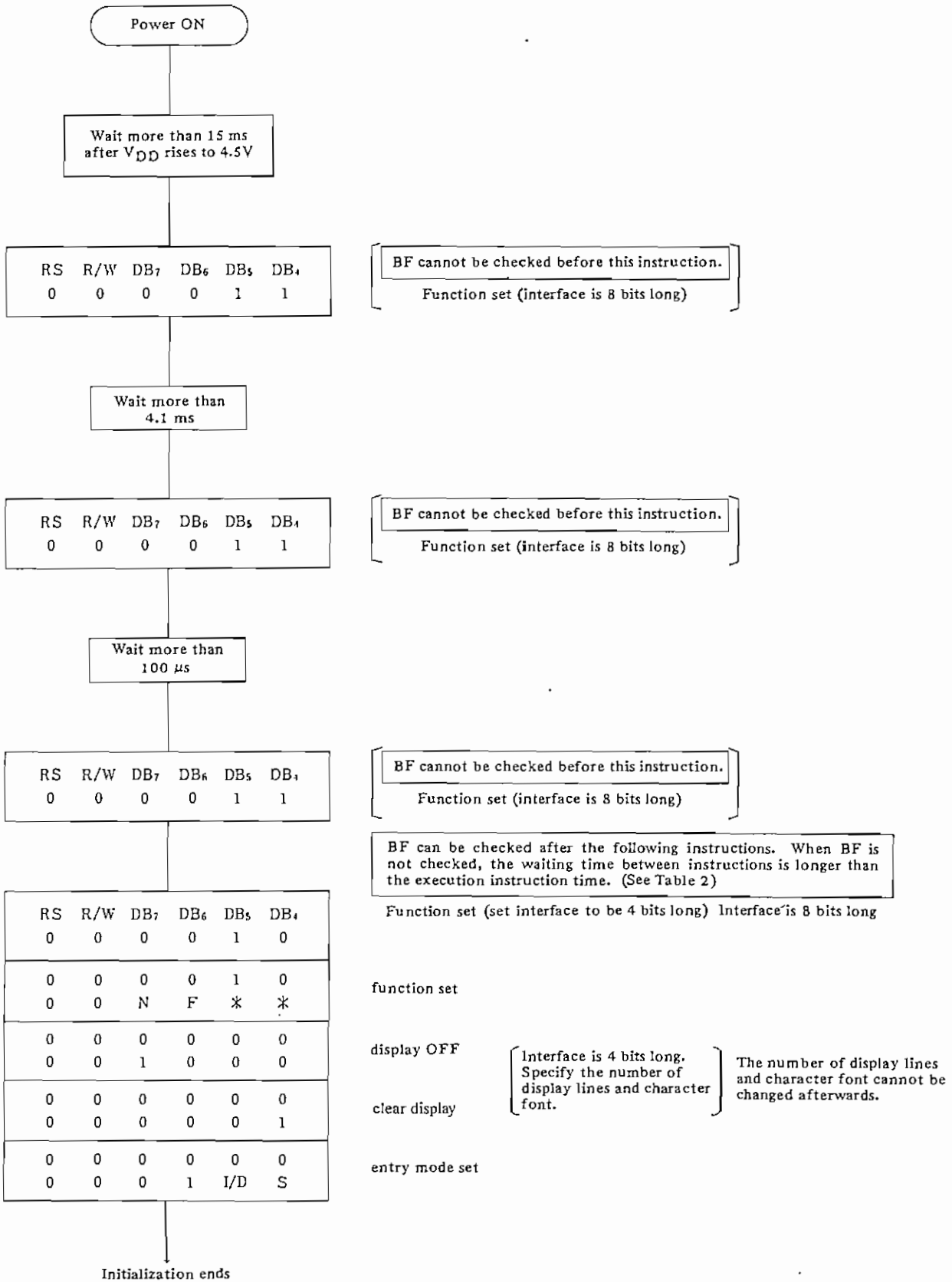
If the power supply conditions for correctly operating the internal reset circuit are not met, initialization by instruction is required.

Use the following procedure for initialization.

(1) When interface is 8 bits long;



When interface is 4 bits long



## 5. Instruction

### 5.1 Outline

Only two HD44780 registers, the Instruction Register (IR) and the Data Register (DR) can be directly controlled by the MPU. Prior to internal operation start, control information is temporarily stored in these registers, to allow interface from HD44780 internal operation to various types of MPUs which operate in different speeds or to allow interface to peripheral control ICs. HD44780 internal operation is determined by signals sent from the MPU. These signals include register selection signals (RS), read/write signals (R/W) and data bus signals (DB<sub>0</sub> ~ DB<sub>7</sub>), and are called instructions, here. Table 2 shows the instructions and their execution time. Details are explained in subsequent sections.

Instructions are of 4 types, those that,

- (1) Designate HD44780 functions such as display format, data length, etc.
- (2) Give internal RAM addresses.
- (3) Perform data transfer with internal RAM
- (4) Others

In normal use, category (3) instructions are used most frequently. However, automatic incrementing by +1 (or decrementing by -1) of HD44780 internal RAM addresses after each data write lessens the MPU program load. The display shift is especially able to perform concurrently with display data write, enabling the user to develop systems in minimum time with maximum programming efficiency. For an explanation of the shift function in its relation to display, see 5.3. When an instruction is executing during internal operation, no instruction other than the busy flag/address read instruction will be executed.

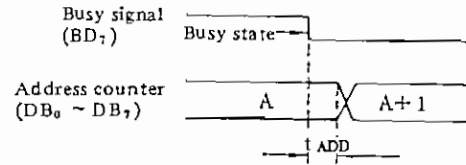
Because the busy flag is set to "1" while an instruction is being executed, check to make sure it is on "1" before sending an instruction from the MPU.

#### Note 1

Make sure the HD44780 is not in the busy state (BF = 0) before sending the instruction from the MPU to the HD44780. If the instruction is sent without checking the busy flag, the time between first and next instructions is much longer than the instruction time. See Table 2 for a list of each instruction execution time.

#### Note 2

After executing instruction of writing data to CG/DD RAM or reading data from CG/DD RAM, RAM address counter is automatically incremented by 1 (or decremented by 1). In this case, this shift is executed after Busy Flag is set to "Low".  $t_{ADD}$  is stipulated the time from the fall edge of busy flag to the end of address counter's renewal.



$t_{ADD}$  depends on the operating frequency

$$t_{ADD} = \frac{1.5}{f_{CP} \text{ or } f_{osc}} \text{ (s)}$$

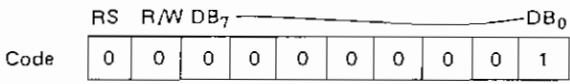
Table 2 Instructions

Instruction	Code										Description	Execution time (when fosc is 250 kHz) Note 1	Execution time (when fosc is 160 kHz) Note 2
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0			
Clear display	0	0	0	0	0	0	0	0	0	1	Clears all display and returns the cursor to the home position (Address 0).	82 μs ~ 1.64 ms	120 μs ~ 4.9 ms
Return home	0	0	0	0	0	0	0	0	1	*	Returns the cursor to the home position (Address 0). Also returns the display being shifted to the original position. DD RAM contents remain unchanged.	40 μs ~ 1.6 ms	120 μs ~ 4.8 ms
Cursor move mode	0	0	0	0	0	0	0	1	I/D	S	Sets the cursor move direction and specifies or not to shift the display. These operations are performed during data write and read.	40 μs	120 μs
Cursor ON/OFF control	0	0	0	0	0	0	1	D	C	B	Sets ON/OFF of all display (D), cursor ON/OFF (C), and blink of cursor position character (B).	40 μs	120 μs
Cursor and display shift	0	0	0	0	0	1	S/C	R/L	*	*	Moves the cursor and shifts the display without changing DD RAM contents	40 μs	120 μs
Interface set	0	0	0	0	1	DL	N	F	*	*	Sets interface data length (DL) number of display lines (L) and character font (F).	40 μs	120 μs
CG RAM address	0	0	0	1	ACG					Sets the CG RAM address. CG RAM data is sent and received after this setting.		40 μs	120 μs
DD RAM address	0	0	1	ADD					Sets the DD RAM address. DD RAM data is sent and received after this setting.		40 μs	120 μs	
Read busy flag & address	0	1	BF	AC					Reads Busy flag (BF) indicating internal operation is being performed and reads address counter contents.		1 μs	1 μs	
Write data to CG or DD RAM	1	0	Write Data					Writes data into DD RAM or CG RAM.		40 μs	120 μs		
Read data from CG or DD RAM	1	1	Read Data					Reads data from DD RAM or CG RAM.		40 μs	120 μs		
I/D = 1: Increment (+1) I/D = 0: Decrement (--1) S = 1: Accompanies display shift. S/C = 1: Display shift S/C = 0: Cursor move R/L = 1: Shift to the right. R/L = 0: Shift to the left. DL = 1: 8 bits DL = 0: 4 bits N = 1: 2 lines N = 0: 1 line F = 1: 5 x 10 dots F = 0: 5 x 7 dots BF = 1: Internally operating BF = 0: Can accept instruction											DD RAM: Display data RAM CG RAM: Character generator RAM ACG: CG RAM address ADD: DD RAM address Corresponds to cursor address. AC: Address counter used for both of DD and CG RAM address.	Execution time changes when frequency changes. (Example) When fosc is 270 kHz: $40 \mu s \times \frac{250}{270} = 37 \mu s$	

effect  
 1. Applied to models driven by 1/8 duty or 1/11 duty.  
 2. Applied to models driven by 1/16 duty.

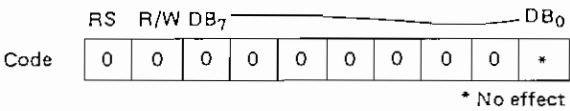
5.2 Description of details

(1) Clear display



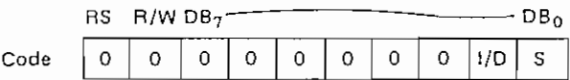
Writes space code "20" (hexadecimal) (character pattern for character code "20" must be blank pattern) into all DD RAM addresses. Sets DD RAM address 0 in address counter. Returns display to its original status if it was shifted. In other words, the display disappears and the cursor or blink go to the left edge of the display (the first line if 2 lines are displayed). Set I/D = 1 (Increment Mode) of Entry Mode. S of Entry Mode doesn't change.

(2) Return home



Sets the DD RAM address 0 in address counter. Returns display to its original status if it was shifted. DD RAM contents do not change. The cursor or blink go to the left edge of the display (the first line if 2 lines are displayed).

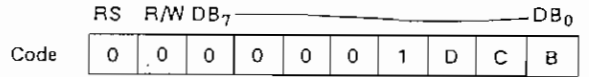
(3) Entry mode set



I/D: Increments (I/D = 1) or decrements (I/D = 0) the DD RAM address by 1 when a character code is written into or read from the DD RAM. The cursor or blink moves to the right when incremented by 1 and to the left when decremented by 1. The same applies to writing and reading of CG RAM.

S: Shifts the entire display either to the right or to the left when S is 1; to the left when I/D = 1 and to the right when I/D = 0. Thus it looks as if the cursor stands still and the display moves. The display does not shift when reading from the DD RAM when writing into or reading out from the CG RAM does it shift when S = 0.

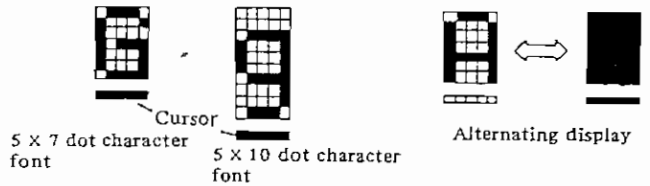
(4) Display ON/OFF control



D: The display is ON when D = 1 and OFF when D = 0. When off due to D = 0, display data remains in the DD RAM. It can be displayed immediately by setting D = 1.

C: The cursor displays when C = 1 and does not display when C = 0. Even if the cursor disappears, the function of I/D, etc. does not change during display data write. The cursor is displayed using 5 dots in the 8th line when the 5 x 7 dot character font is selected and 5 dots in the 11th line when the 5 x 10 dot character font is selected.

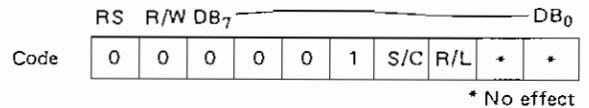
B: The character indicated by the cursor blinks when B = 1. The blink is displayed by switching between all blank dots and display characters at 409.6 ms interval when  $f_{CP}$  or  $f_{osc} = 250$  kHz. The cursor and the blink can be set to display simultaneously. (The blink frequency changes according to the reciprocal of  $f_{CP}$  or  $f_{osc}$ :  $409.6 \times \frac{250}{270} = 379.2$  ms when  $f_{CP} = 270$  kHz.)



(a) Cursor Display Example

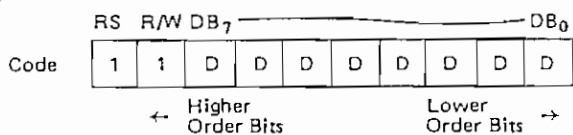
(b) Blink Display Example

(5) Cursor or display shift



Shifts cursor position or display to the right or left without writing or reading display data. This function is used to correct or search for the display. In a 2-line display, the cursor moves to the 2nd line when it passes the 40th digit of the 1st line. Notice that the 1st and 2nd line displays will shift at the same time. When the displayed data is shifted repeatedly each line only moves horizontally. The 2nd line display does not shift into the 1st line position.

(11) Read data from CG or DD RAM



Reads binary 8 bit data DDDDDDDD from the CG or DD RAM. The previous designation determines whether the CG or DD RAM is to be read. Before entering the read instruction, you must execute either the CG RAM or DD RAM address set instruction. If you don't, the first read data will be invalidated. When serially executing the "read" instruction, the next address data is normally read from the second read. The "address set" instruction need not be executed just before the "read" instruction when shifting the cursor by cursor shift instruction (when reading out DD RAM). The cursor shift instruction operation is the same as that of the DD RAM's address set instruction.

After a read, the entry mode automatically increases or decreases the address by 1. However, display shift is not executed no matter what the entry mode is.

(Note) The address counter (AC) is automatically incremented or decremented by 1 after "write" instructions to either CG RAM or DD RAM. RAM data selected by the AC cannot than be read out even if "read" instructions are executed. The conditions for correct data read out are: execute either the address set instruction or cursor shift instruction (only with DD RAM), just before reading out execute the "read" instruction from the second time the "read" instruction is serial.

5.3 Instruction and display correspondence

(1) 8-bit operation, 8-digit x 1-line display (using internal reset)

Following table shows an example of 8-bit x 1-line display in 8-bit operation.

The HD44780 functions must be set by Function Set prior to display. Since the display data RAM can store data for 80 characters, as explained before, the RAM can be used for displays like the lightening board when combined with display shift operation.

Since the display shift operation changes display position only and DD RAM contents remain unchanged, display data entered first can be output when the return home operation is performed.

HD44780

operation, 8-digit 1-line display example (using internal reset)

Instruction	Display	Operation
Power supply ON (HD44780 is initialized by the internal reset circuit)	<input type="text"/>	Initialized. No display appears.
Function Set RS R/W DB <sub>7</sub> <span style="margin-left: 150px;">DB<sub>0</sub></span> 0 0 0 0 1 1 0 0 * *	<input type="text"/>	Sets to 8-bit operation and selects 1-line display lines and character font. (Number of display lines and character fonts cannot be changed hereafter.)
Display ON/OFF Control 0 0 0 0 0 0 1 1 1 0	<input type="text"/>	Turns on display and cursor. Entire display is in space mode because of initialization.
Entry Mode Set 0 0 0 0 0 0 0 1 1 0	<input type="text"/>	Sets mode to increment the address by one and to shift the cursor to the right at the time of write to the DD/CG RAM. Display is not shifted.
Write Data to CG RAM/DD RAM 1 0 0 1 0 0 1 0 0 0	<input type="text" value="H _"/>	Write "H". The DD RAM has already been selected by initialization when the power is turned on. The cursor is incremented by one and shifted to the right.
Write Data to CG RAM/DD RAM 1 0 0 1 0 0 1 0 0 1	<input type="text" value="H I _"/>	Writes "I".
⋮	⋮	
Write Data to CG RAM/DD RAM 1 0 0 1 0 0 1 0 0 1	<input type="text" value="H I T A C H I _"/>	Writes "I".
Entry Mode Set 0 0 0 0 0 0 0 1 1 1	<input type="text" value="-H I T A C H I _"/>	Sets mode for display shift at the time of write.
Write Data to CG RAM/DD RAM 1 0 0 0 1 0 0 0 0 0	<input type="text" value="I T A C H I _"/>	Writes "Space".
Write Data to CG RAM/DD RAM 1 0 0 1 0 0 1 1 0 1	<input type="text" value="T A C H I M _"/>	Writes "M".
⋮	⋮	
Write Data to CG RAM/DD RAM 1 0 0 1 0 0 1 1 1 1	<input type="text" value="M I C R O K O _"/>	Writes "O".
Cursor or Display Shift 0 0 0 0 0 1 0 0 * *	<input type="text" value="M I C R O K O"/>	Shifts only the cursor position to the left.
Cursor or Display Shift 0 0 0 0 0 1 0 0 * *	<input type="text" value="M I C R O K O"/>	Shifts only the cursor position to the left.
Write Data to CG RAM/DD RAM 1 0 0 1 0 0 0 0 1 1	<input type="text" value="I C R O C O"/>	Writes "C" (correction). The display moves to the left.
Cursor or Display Shift* 0 0 0 0 0 1 1 1 * *	<input type="text" value="M I C R O C O"/>	Shifts the display and cursor position to the right.
Cursor or Display Shift 0 0 0 0 0 1 0 1 * *	<input type="text" value="M I C R O C O"/>	Shifts display and cursor position to the right.
Write Data to CG RAM/DD RAM 1 0 0 1 0 0 1 1 0 1	<input type="text" value="I C R O C O M _"/>	Writes "M".
⋮	⋮	
Return Home 0 0 0 0 0 0 0 0 1 0	<input type="text" value="H I T A C H I"/>	Returns both display and cursor to the original position (Address 0).

(2) 4-bit operation, 8-digit x 1-line display (using internal reset)

The program must set functions prior to 4-bit operation. The following table shows an example. When power is turned on, 8-bit operation is automatically selected and the first write is performed as an 8-bit operation. Since

nothing is connected to DB<sub>0</sub> ~ DB<sub>3</sub>, a rewrite is then required. However, since one operation is completed in two access of 4-bit operation, a rewrite is needed as a function (see the following table).

Thus, DB<sub>4</sub> ~ DB<sub>7</sub> of the function set is written twice.

4 bit operation, 8-digit 1-line display (using internal reset)

No.	Instruction	Display	Operation
1	Power supply ON (HD44780 is initialized by the internal reset circuit)	<input type="text"/>	Initialized. No display appears.
2	Function Set RS R/W DB <sub>7</sub> DB <sub>6</sub> DB <sub>5</sub> DB <sub>4</sub> 0 0 0 0 1 0	<input type="text"/>	Sets to 4-bit operation. In this case, operation is handled as 8 bits by initialization, and only this instruction completes with one write.
3	Function Set 0 0 0 0 1 0 0 0 0 0 * *	<input type="text"/>	Sets 4-bit operation and selects 1-line display and 5 x 7 dot character font. 4-bit operation starts from this point on and resetting is needed. (Number of display lines and character fonts cannot be changed hereafter.)
4	Display ON/OFF Control 0 0 0 0 0 0 0 0 1 1 1 0	<input type="text" value="-"/>	Turns on display and cursor. Entire display is in space mode because of initialization.
5	Entry Mode Set 0 0 0 0 0 0 0 0 0 1 1 0	<input type="text" value="-"/>	Sets mode to increment the address by one and to shift the cursor to the right, at the time of write, to the DD/CG RAM. Display is not shifted.
6	Write Data to CG RAM/DD RAM 1 0 0 1 0 0 1 0 1 0 0 0	<input type="text" value="H_"/>	Writes "H". The cursor is incremented by one and shifts to the right.

Hereafter, control is the same as 8-bit operation.



8-bit operation, 8-digit x 2-line display

For 2-line display, the cursor automatically moves from the first to the second line after the 40th digit of the 1st line has been written. Thus, if there are only 8 characters on the first line, the DD RAM address must again be set after the 8th character is completed. (See the following table) Note that the first and second lines of the display

shift are performed. In the example, the display shift is performed when the cursor is on the second line. However, if shift operation is performed when the cursor is on the first line, both the first and second lines move together. When you repeat the shift, the display of the second display will only move within each line many times.

8-bit operation, 8-digit x 2-line display example (using internal reset)

Instruction	Display	Operation
Power supply ON (HD44780 is initialized by the internal reset circuit)		Initialized. No display appears.
Function Set RS R/W DB <sub>7</sub> ————— DB <sub>0</sub> 0 0 0 0 1 1 1 0 * *		Sets to 8-bit operation and selects 2-line display and 5 x 7 dot character font.
Display ON/OFF Control 0 0 0 0 0 0 1 1 1 0		Turns on display and cursor. All display is in space mode because of initialization.
Entry Mode Set 0 0 0 0 0 0 0 1 1 0		Sets mode to increment the address by one and to shift the cursor to the right, at the time of write, to the DD/CG RAM. Display is not shifted.
Write Data to CG RAM/DD RAM 1 0 0 1 0 0 1 0 0 0		Write "H". The DD RAM has already been selected by initialization when the power is turned on. The cursor is incremented by one and shifted to the right.
⋮	⋮	
Write Data to CG RAM/DD RAM 1 0 0 1 0 0 1 0 0 1		Writes "I".
Set DD RAM Address 0 0 1 1 0 0 0 0 0 0		Sets RAM address so that the cursor is positioned at the head of the 2nd line.
Write Data to CG RAM/DD RAM 1 0 0 1 0 0 1 1 0 1		Writes "M".
⋮	⋮	
Write Data to CG RAM/DD RAM 1 0 0 1 0 0 1 1 1 1		Writes "O".
Entry Mode Set 0 0 0 0 0 0 0 1 1 1		Sets mode for display shift at the time of write.
Write Data to CG RAM/DD RAM 1 0 0 1 0 0 1 1 0 1		Writes "M". Display is shifted to the right. The first and second lines' shift are operated at the same time.
⋮	⋮	
Return Home 0 0 0 0 0 0 0 0 1 0		Returns both display and cursor to the original position (Address 0).

6. Precaution on programming

(1) Instruction of function set

Perform the function at the head of program that accesses HD44780 before executing all instructions, and not change the data of the Instruction Register in the program. The data of function register can be changed by the program as follows;

- a. • Changing of DL (Data Length)
  - Perform the instruction appointed in 4.2 (2), when DL is changed from 8-bit length to 4-bit length mode.
  - Perform the instruction appointed in 4.2 (1), when DL is changed from 4-bit length to 8-bit length mode.
- b. • Changing of N (Column Number)
  - Perform the instruction of function set after executing instruction of display clear or display off.

In this case, sequence of AC and DD RAM must be changed. Thus, rewrite the address set register after that.

- c. • Changing of F (Font)
  - There is no problem in this case, but for dual-line display, the font mode of 5 x 11 cannot be selected (this mode is forbidden by hardware).

When N or F is changed, power supply voltage for LCD must be changed. If not changed, crosstalk will appear, or contrast will be poor.

(2) Busy flag check

HD44780 is produced in the CMOS process, therefore internal executing time is long. Standard time is 40 μs ~ 1.6 ms. (This varies by instruction)

When the high speed MPU controls it, check the busy flag before performing instruction or reading data.

While internal operation is active, Enable signal is not accepted. (Enable signal at reading status register for checking busy flag is accepted) Busy flag signal is output through DB<sub>7</sub>, as shown in Table 3, when RS = "0", R/W = "1", and Enable = "1".

(3) Input of unidentified instruction code

Undefined instruction code of HD44780 is only as follows;

RS	R/W	DB <sub>7</sub> ~ DB <sub>0</sub>
0	0	0 ~

(Others are included to defined instruction)

When the undefined instruction code is loaded to HD44780, it accepts the code, but does not change the internal states (RAM and other status of Flags). Busy state, however continues for maximum 40 μs by the acceptance of the code.

Table 3 The relation between the operation and the combination of RS, R/W

RS	RW	E	OPERATION
0	0		Write instruction code
0	1		Read busy flag and address counter
1	0		Write data
1	1		Read data

When performing data and instruction code by 4 bit, transfer RS, R/W every time.

How to check trouble  
 Use the flowchart below to check errors.

Error analysis flowchart

